

Modeling of Air Traffic Control System Using Formal Methods



T04431

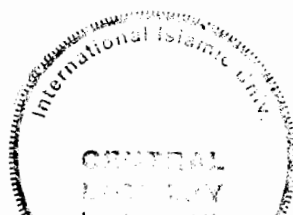
Developed by

Maryam Jamal
(103-FAS/MSSE/F05)

Supervised by

Dr. Nazir Ahmad Zafar

DEPARTMENT OF COMPUTER SCIENCE
FACULTY OF BASIC AND APPLIED SCIENCES
INTERNATIONAL ISLAMIC UNIVERSITY, ISLAMABAD
2007



16-7-2010

~~SECRET~~



Accession No JH-4431

MS
005.3
MAM.

- 1- Air traffic control - Automation
- 2- Computer Software.

md
16-7-2010

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

*In the Name of Allah The Most Beneficent
The Most Merciful*

**Department of Computer Science
International Islamic University Islamabad**

Final Approval

Dated: 13th November 2007

It is certified that we have read the project report submitted by Miss Maryam Jamal (103-FAS/MSSE/F05) and it is our judgment that this project is of sufficient standard to warrant its acceptance by the International Islamic University, Islamabad for the MS Degree in Software Engineering.

Committee


External Examiner

Dr. Aamer Nadeem
Assistant Professor,
Department of Computer Science,
Mohammad Ali Jinnah University,
Islamabad




Internal Examiner

Adnan Ashraf
Lecturer,
Department of Computer Science,
International Islamic University,
Islamabad



Supervisor

Dr. Nazir Ahmad Zafar
Principal Scientist,
Department of Computer and Information Sciences,
Pakistan Institute of Engineering & Applied Sciences,
Nilore, Islamabad



Dedicated
to
my Ever loving and Encouraging
Parents

The Dissertation is submitted to
Department of Computer Science,
International Islamic University, Islamabad
As a partial fulfillment of the requirement
For the award of the degree of
MS in Software Engineering.

DECLARATION

I hereby declare that this project report, neither as a whole nor as a part thereof has been copied out from any source. It is further declared that I have developed the project and its report while working individually, and completed the report entirely on the basis of my personal efforts made under the sincere guidance of my Project Supervisor. If any part of this report is proved to be copied out or found to be reported, I shall stand by the consequences. No portion of the work presented in this report has been submitted in support of any application for any other degree or qualification of this or any other University or Institute of learning.

Maryam Jamal

(103-FAS/MSSE/F05)

ACKNOWLEDGEMENTS

All praise to Almighty ALLAH, the most Merciful and Compassionate, who enabled me to complete this project. I would like to acknowledge the support of my Parents and family members, whose prayers are a source of strength and determination for me.

I express gratefulness to my kind supervisor Dr. *Nazir Ahmad Zafar* who trusted my talent and gave me an opportunity to work on this project under his supervision. Throughout the project he kept my morale high by his suggestions and appreciation. Working on an entirely new field of Formal Methods was a challenge that turned out to be an excellent medium of learning for me because of the strong commitment and dedication of my project supervisor. Without his precious guidance and help I would have never been able to develop this Project on time.

It is my pleasure to thank Dr. *Naveed Ikram* who has been so much cooperative and helpful throughout the project. His worthy comments and guidance made it possible to improve the quality of project. I am also very grateful to Dr. *Amir Nadeem* for thoroughly reviewing the thesis and giving valuable comments. Moreover, I would like to acknowledge the help of internal examiners; Sir *Adnan Ashraf* and Sir *Zohaib Zafar* for their precious guidance about every aspect of the research.

Finally, I would like to express thanks to all my Teachers for polishing my skills and helping me seek the knowledge.

Maryam Jamal

(103-FAS/MSSE/F05)

PROJECT IN BRIEF

Project Title: Modeling of Air Traffic Control System Using Formal Methods

Undertaken By: Maryam Jamal (103-FAS/MSSE/F05)

Supervised By: Dr. Nazir Ahmad Zafar
Principal Scientist
Department of Computer and Information Sciences,
Pakistan Institute of Engineering & Applied Sciences,
Nilore, Islamabad

Tools Used: Z Notation
Z/EVES Tool-set
Microsoft Office 2003
Microsoft Visio 2003
Adobe Acrobat Reader & Writer

Operating System: Windows 2000 Professional

System Used: Pentium III Machine

Date Started: September, 2006

Date Completed: July, 2007

ABSTRACT

Formal Methods are an emerging methodology that makes it possible to prove and analyze certain properties of the system so that errors and inconsistencies are identified during early stages of the development process. In this project, Formal Methods in terms of Z notation are applied for the specification of safety-critical system of Air Traffic Control (ATC). Automation of ATC system is considered to be one of the most challenging domains because it is a complex, highly distributed and safety critical system. Firstly, ATC system model in real world is analyzed. For connectivity of different zones of airspace, the real world ATC system is transformed into a directed graph, which is then used to formalize the components of formal ATC model, i.e., static Topology, Network State, Aircraft and Controller. The whole Formal ATC model is then presented as encapsulation of formal models of its components. The phases of flight of an aircraft are defined as state operations on formal ATC model. Finally, the Formal ATC system model is checked and analyzed with Z/EVES tool-set.

ABBREVIATIONS

ATC	Air Traffic Control System
CCF	Central Control Function
CDIS	CCF Display Information System
CICS	Customer Information Control System
CSP	Communicating Sequential Processes
DFD	Data Flow Diagram
EVES	Euclid Verification and Evaluation Ssystem
FM	Formal Methods
ICAO	International Civil Aviation Organization
IFR	Instrument Flight Rule
RSML	Requirements State Machine Language
SpecTRM-RL	Specifications Tools and Requirements Methodology – Requirements Language
TCAS	Traffic Collision Avoidance System
UML	Unified Modeling Language
V&V	Verification and Validation
VDM	Vienna Development Method
VFR	Visual Flight Rule

Table of Contents

<i>Chapter No.</i>	<i>Contents</i>	<i>Page No.</i>
1.	Introduction	1
	1.1 Background	1
	1.2 Air Traffic Control System	2
	1.3 The Project	3
	1.4 Motivation	4
	1.5 Literature Survey	6
	1.6 Conclusion	9
	1.7 Problem Statement	10
	1.8 Proposed Model	10
	1.8.1 Network Components	12
	1.8.2 State Operations	13
2.	Formal Methods	14
	2.1 An Overview	14
	2.2 Classification of Formal Methods	16
	2.3 Notations and Tools	17
	2.4 Successful Industrial Applications	19
3.	Abstract Model	22
	3.1 Air Traffic Control System	22
	3.2 Network Components	23
	3.2.1 Controller	24
	3.2.2 Aircraft	25
	3.2.3 Airport	26
	3.2.4 Airspace	26
	3.2.5 Airway	27
	3.3 Flight Profile	28
	3.3.1 Takeoff	28
	3.3.2 Departure	29
	3.3.3 Enroute	29
	3.3.4 Approach	30
	3.3.5 Landing	30

<i>Chapter No.</i>	<i>Contents</i>	<i>Page No.</i>
4.	Formal ATC Model	31
	4.1 The Z Notation	31
	4.2 ATC Model In Graph Theory	33
	4.3 Network Components	34
	4.3.1 Static Topology	34
	4.3.2 Network State	36
	4.3.3 Aircraft	37
	4.3.4 Controller	38
	4.3.5 ATC System	40
	4.4 State Operations	42
	4.4.1 Takeoff an Aircraft	42
	4.4.2 Departure of An Aircraft	43
	4.4.3 Handover an Aircraft	44
	4.4.4 Descent of an Aircraft	45
	4.4.5 Landing of an Aircraft	46
5.	Analyzing ATC Model	47
	5.1 Z/EVES Tool-Set	47
	5.2 Model Exploration in Z/EVES	48
	5.2.1 Syntax and Type Checking	49
	5.2.2 Domain Checking	49
	5.2.3 Reduction	49
	5.2.4 Prove By Reduce	49
	5.3 Exploration of Formal ATC Model	50
6.	Conclusion	52
	6.1 Concluding Remarks	52
	6.2 Future Directions	53
	References	55

List of Figures

<i>Figure No.</i>	<i>Figures</i>	<i>Page No.</i>
1.1	Formal Model of ATC System	12
3.1	An Overlapping of Protected Zones of Two Aircrafts	25
3.2	Classification of Airspace	27
3.3	Flight Profile of an Aircraft	28
4.1	ATC Model in Graph Theory	33
4.2	ATC Model with Computer-based Controllers	38
4.3	Components of Formal ATC System Model	41
5.1	Snapshot of Z/EVES Tool-set	48

List of Tables

<i>Table No.</i>	<i>Table</i>	<i>Page No.</i>
5.1	Results of Exploration of Formal Model of ATC System	50
5.2	Comparison with existing models	51

Chapter # 1

Introduction

1. INTRODUCTION

Firstly, the background of project is introduced, followed by an introduction to the Air Traffic Control system. Then the project is described after which the motivation to adopt the project has been discussed. A thorough literature survey is then presented. Then the problem statement of our work is extracted from drawbacks and shortcomings of the existing models. Finally the proposed model is described briefly.

1.1 BACKGROUND

In the current age of computers, there is remarkable trend of system automation. Due to large-sized domains and ever increasing demands of end-users, makes the automation tremendously complex. Despite of increased complexity every automated system has an inherent need of high quality and reliability. Therefore, every automated system must conform to the requirements of end users and fulfills the assigned task speedily and correctly.

The systems like nuclear power plant, avionics, railway, chemical synthesis, weapon manufacturing etc belong to domain of safety critical systems [28]. The safety critical system is a system in which any error has very critical impacts. Firstly, the precious human lives are at risk. Even a minor error can lead to fatal injuries or can lead to death. Secondly, unlike other systems safety critical systems requires high budget. Any defect in the system is very expensive to fix which causes severe economic penalties.

The automation of safety critical systems is considered to be a challenge [10]. Since the paradigm has shifted from the job of a programmer to engineer. Making a working piece of software is not sufficient. Every system especially safety critical system needs to have optimal resource (memory, CPU) utilization giving error-free correct results in units of time. Moreover, a system must fulfill the requirements of end users while coping with increasing complexities of the domain.

1.2 AIR TRAFFIC CONTROL SYSTEM

The Air Traffic Control (ATC) is one of the most extensive systems of the world. It is a system in which continuous services are provided to the aircrafts by the ground-based controllers [16, 58]. The core of ATC system is an Air Traffic controller. A team of controllers, headed by a controller, works at each ground station like local, tower, center control stations. There is wide variety of aircrafts that ATC system needs to supervise ranging from small private aircrafts to large commercial airliners. ATC system provides a wide range of services to the aircrafts like navigational, weather updates, flight profile information, emergency relief etc.

Behind every service offered by ATC system, there are two major goals [17]. First and foremost goal is to minimize or avoid aircraft collision. The term collision not only includes the mid-air collision of aircrafts but also the collision between aircraft and ground obstructions located in the way of its flight. Second major goal is to maximize the effectiveness of ATC system with little or no delays. The effectiveness of the system corresponds to greater number of aircrafts those can fly safely in the airspace without compromising the safety of all other aircrafts occupying airspace and airports. Similarly, as the number of aircrafts increases there must not be unwanted delays. The achievement of these goals can yield increased customer's confidence that can lead to the profit of whole airline industry.

ATC system is a challenging domain to work on because it is a complex, highly distributed and safety critical system [34]. The domain of ATC system is very large and complicated offering wide range of functionalities and services to the aircrafts. Therefore, the complexity of the domain of ATC system is remarkably high that is hard to be handled effectively. The task of safe journey of an aircraft is distributed among various controllers who collaborate and direct an aircraft within their area of control. This distribution of control makes ATC system a highly distributed system. Moreover, the ATC system belongs to the domain of safety critical system. Any error can not only damage the aircraft but also the lives of on-board passengers and crew are at risk.

Therefore, the automation of ATC requires a sophisticated technique to cope with all challenges imposed on the domain with an aim to fulfill the goals set for the ATC system.

1.3 THE PROJECT

Automation of ATC system is considered to be one of the most challenging domains because it is a complex, highly distributed and safety critical system. The field of safety critical applications like nuclear power plant, avionics, railway, chemical synthesis, weapon manufacturing etc is an important field because even a minor error not only causes monetary losses but also precious human lives are at risk. Secondly, any defect in the system is very expensive to fix which causes severe economic penalties.

Therefore the software of every safety critical system has a basic need of ultra-high dependability. "Dependability is the property of a computing system, which allows reliance to be justifiably placed on the services it delivers. Dependability is an overall property, which has other measures such as

- **Safety** is a measure of the continuous delivery of services free from occurrences of catastrophic failures.
- **Reliability** is a measure of continuous delivery of proper service (where service is delivered according to specified conditions) or equivalently of the time to failure.
- **Availability** is a measure of the delivery of proper service with respect to the alternation of proper and improper services." [29]

All of the three measures contribute to the dependability of the system. But it is difficult to focus on all these aspects simultaneously in this short research project. It has been found that there is a basic need and public concern of ATC system to be working properly without the occurrences of faults. Therefore, safety is like a tenet of an ATC system, so in our work we will be focusing the aspects of safety.

International Civil Aviation Organization (ICAO) [21] is an agency of United Nations that defines principles and techniques for every aspects of air navigation to ensure safety and effectiveness of an ATC system. ICAO was created in 1944 and its headquarter is in Montreal, Canada. The ICAO Council develops standards, regulations and recommended practices concerning every aspect of air navigation and international civil aviations. There are 189 contracting states of ICAO which follow its standards in order to have maximum safety level. The objectives of ICAO are two folded, firstly, to bridge the gaps between contracting states and to establish consistency of rules to be implemented uniformly throughout the globe. The safety related standards defined by ICAO [20] stresses

- To have clearly defined performance standards, e.g. separation criteria. In all situations there must be safe separation between all aircrafts utilizing ATC system's services.

Moreover some of the factors [20] those should be considered, in conducting a safety assessment, are as follows

- "Types of aircraft and their performance characteristics, including their navigation capabilities and performance;
- Traffic density and distribution;
- Airspace complexity, ATC route structure and the classification of airspace;
- Aerodrome layout, including runway, taxiway configurations and preferences." [20]

We will be using the above-mentioned clauses of ICAO as criteria to evaluate some of the existing models of ATC system.

1.4 MOTIVATION

Motivation for embarking on this line of research is due to a couple of reasons. Firstly, modeling of a complex, highly distributed and safety critical system of ATC is a challenge in the field of automation. Secondly, the use of Formal Methods (FM) in systems development especially in safety critical systems is growing rapidly. FM are being increasingly mentioned in some safety-related standards as a possible method of improving dependability [28].

There is enough research emerging for standardization of usage of FM for safety critical systems but also on the standardization of semantics of formal specification languages. The European Space Agency has mandated the use of FM in the sector of space and the US Radio Technical Commission for Aeronautics has included FM contents for Aviation in its standard DO-178B. Therefore, the use of FM in systems development especially in safety critical systems is growing exponentially.

FM is a promising field in research academia [40, 46]. After realizing the benefits of FM in industry and academia, Formal Methods Europe [18] is stimulating the use of, and research on, FM for software development. Despite of many myths and misconceptions [2] about FM, one of the good examples of using FM is Central Control Function (CCF) Display Information System (CDIS). "CDIS is a distributed real-time system running on a network of dual computers communication over token ring Local Area Network. During CDIS specifications Vienna Development Method (VDM) was used as core specification technique; prototyping was used for user interface definition; and concurrency aspects were catered using Communicating Sequential Processes (CSP). For CDIS design VDM was used along with Milner's Calculus of Communicating Systems. As a result the delivered software had a defect rate of about 0.75 faults per thousand lines of code, a figure two to ten times better than that for published projects and comparable software in air traffic control applications that did not use FM" [1].

Another promising example of FM in the field of avionics is the formal requirements specification for the Traffic Collision Avoidance System (TCAS) II, required on all commercial aircraft flying in US airspace. "The Requirements State Machine Language (RSML) was used, which is based on State charts with more focus on the readability of a formal specification. The completeness and consistency of TCAS II specifications has been checked using its automated proof support. The RSML specifications can also be refined to generate code. Therefore, the TCAS II project demonstrated the practicality of writing formal specifications for a complex process-control system and the feasibility of building a formal model of a system that can be read and reviewed by application experts without special training" [14].

Based on the above experiences, it was realized that FM is an emerging and future technology. Currently, in Pakistan, FM are not very popular and it will be really useful to introduce this state-of-the-art technology in Pakistan for enhancing the increasing trend of software engineering. This research has been carried out in this direction with this motivation. The challenging domain of ATC accelerated us to investigate if the FM can be applied successfully in this area. There are three main objectives to be achieved in this research: (i) formal approach in systems development, (ii) integration of formal and informal approaches and (iii) Proposing an abstract model ensuring correctness of formal specification of the ATC system.

1.5 LITERATURE SURVEY

ATC system has been an important research field. Many researchers have been contributing in this area using various approaches. The work presented by Bass et.al [34] is a case study of ATC system exploiting Layered Architecture. Fields et.al [47] modeled the complex ATC system as distributed representations those are primarily visible and external to human actors (like flight strips, strip board etc) those can be used as a collaborative system and a source of conflict detection. Johnson et.al [53] focused the models of Unified Modeling Language (UML) for developing ATC simulator and its Graphical User Interface. A simple multi-aircraft, hybrid and stochastic model of Air Traffic Management for conflict detection and resolution, from the point of view of an Air Traffic Controller is presented by Glover et.al [62]. The simulation model is presented by Baxter et.al [50], which is structured into ground, terminal and en-route phases of flight.

In the field of FM, ATC system is not an unexplored area. McCluskey et.al [56] gives formal specification for conflict prediction over Shanwick Oceanic airspace. Butler et.al [51] introduced a new approach to verifying safety of conflict detection algorithms. Carreno et.al [59] focused safety verification of small Aircraft Transportation system using concept of operations, which support separation, orderly arrivals and increased throughput of airports lacking radar coverage and tower.

Whittle et.al [27] presented an algorithm that translate scenarios of a system's behavior into state machines has been applied to the weather control logic subsystem of Center TRACON Automation System, which is under development at NASA Ames Research Center. Bontemps et.al [63] demonstrated the use of Live Sequence Charts for specifying part of ATC system CTAS dealing with weather updates. Degani et.al [3] gave a formal perspective to the analysis of pilot interaction with automated flight control systems. Zimmerman et.al [35] demonstrated the use of Specifications Tools and Requirements Methodology – Requirements Language (SpecTRM-RL). The vertical flight control system is modeled for a high-tech aircraft like the MD-11 which operates as part of Flight Management System and provides targets and controls necessary to maintain a predetermined vertical flight profile and provide guidance, control and annunciation functions [35].

To summarize the above discussion, some models focused the conflict prediction and resolution in an ATC system, some highlighted the issues of weather information updating, some focused on communication between pilot and air traffic controller some proved the various aspects in an ATC system using V&V and some gave the single isolated aspects of an ATC system. Since our objective is to give the formal specification of the whole ATC system, the following models of an ATC system were quite relevant to our work. It is important to note that we have evaluated these models for their aspects of safety. The limitations in their work have been identified those can lead to any possible potential catastrophic effect.

1.5.1 Integrating the Operator into Formal Models in the Air-Traffic Control Domain by David Leadbetter et.al [12]

Leadbetter et.al [12] focused the detection and reduction of errors caused by a human operator in an ATC system. A simplified abstract model of an ATC system is presented as a brief sub component using Z notation. A sector of airspace is modeled in terms of airport, waypoints and routes. Just two operations of aircraft's telemetry updates are given. Movement of aircraft in two dimensions is considered and vertical separation of aircrafts is not considered.

Based on the evaluation criteria, following issues are identified

- The vertical separation of aircrafts is not considered.
- The speed and position of an aircraft are just specified.
- Since a sector of airspace is modeled the issues of traffic density and distribution are not considered.
- Since a route is defined as a mapping of waypoints, no check is defined on a waypoint connected with itself.
- The airport is modeled abstractly as a power set of waypoints.
- The presented ATC model is not validated.

1.5.2 A Tutorial Introduction to Formal Methods by Peter A. Lindsay [45]

To demonstrate the strength of a Model based formal language, an example of a simple hypothetical ATC system using Sum Language, dialect of Z notation, and Cogito Methodology is presented by Lindsay [45]. It also gives the idea of distributed architecture of ATC system abstractly. Based on the evaluation criteria, following issues are identified

- The safe separation of aircrafts is not considered.
- The aircraft is modeled abstractly and its performance characteristics are not modeled.
- The airspace is modeled abstractly. No idea about route structure and connectivity of airspace is presented.
- The airport and its configurations are also left undefined.

1.5.3 Software Systems Engineering From Domain Analysis via Requirements Capture to Software Architecture by Dines Bjorner [13]

Bjorner [13] demonstrated application of RAISE method for domain analysis, requirements capture and software architecture. Domain analysis is done by abstract formal modeling of airspace, timetable and air traffic. Based on domain analysis requirements capture is done for two subsystems i.e ATC system and scheduling and rescheduling.

Based on the evaluation criteria, following issues are identified

- The issue of no collision is defined abstractly but safe separation is ignored.
- The aircraft is modeled abstractly and its performance characteristics are not modeled.
- Since the model is very abstract. Although the major components are modeled but they are very abstract with no details of invariants and safety properties to be defined on them.

1.5.4 Proof in VDM: A Practitioner's Guide by J. C. Bicarregui et.al [22]

Moreover to determine the strength of FM in detecting errors in requirement specifications and validation of explicit requirements, a case study of ATC system at a very abstract level using VDM is presented by Bicarregui et.al [22]. Based on the evaluation criteria, following issues are identified

- The safe separation of aircrafts is not considered.
- The aircraft is modeled abstractly and its performance characteristics are not modeled.
- The airspace is modeled abstractly. No idea about route structure and connectivity of airspace is presented.
- The airport and its configurations are also left undefined.

1.6 CONCLUSION

The models described earlier are at abstract level of representations. Certain loopholes in the existing models have been identified that can badly affect safety of an ATC system. None of them covers the real world requirements of an ATC system. Unless the model is based on realistic requirements it cannot be refined to a system that can be implemented in real world fulfilling the claims of more safety and dependability. Therefore, a model reflecting real world model and fulfilling safety criteria is required.

1.7 PROBLEM STATEMENT

Safe separation between aircrafts is an utmost demand of an ATC system according to criteria of ICAO. Loss of separation is an act of protecting an aircraft from entering proximity of another aircraft that may lead to a possible conflict. Therefore, a safe ATC system must address issues of safe separation between aircrafts effectively.

Similarly none of the model solely covers all of the defined assessment criteria. The structure of airspace, routes, aircrafts and airports are either skipped or they are defined at abstract level of representation. Similarly, connectivity of different zones of airspace is vital for safe journey of an aircraft. Therefore, a model fulfilling the demanded criteria of ICAO is required.

1.8 PROPOSED MODEL

Our aim is to give a dependable system focusing more on the level of safety. The anticipated solution focus real world ATC system and removes the problems identified in existing models. The anticipated solution designed for the problem is as following.

Our work is based on the standards set by ICAO. Our model will fulfill all the assessment criteria defined by ICAO those have been set for evaluation of existing ATC system models. ICAO standards are followed because our formal ATC model should be standard and uniformly applicable through out the globe and it should reflect real world ATC system.

Since we are at initial stages of development of an ATC system, it is difficult to model it completely along with its full complexities. Therefore, like the existing models we will be proposing an abstract model but it will be that much refined to model relevant standards set by ICAO.

Connectivity issues are resolved by applying graph theory [54]. Graph theory is considered to be a compliant means in solving various connectivity problems of real world. The real world ATC system is transformed into a directed graph. The small three-dimensional segments of airspace called zones represents the set of nodes and the airway segments connecting them represents the set of edges. The directed graph is exploited because direction of connections is worthy.

The safe separation between aircrafts is addressed by the concept of protected zone. Protected zone is an area of defined dimensions around an aircraft in which no other aircraft is allowed to intrude. If the protected zones of two aircrafts intersect it may lead to a possible collision or catastrophic damage to aircrafts. Therefore, our model will be focussing protected zones of aircrafts to ensure safety.

Since ATC system is one of the most complex domains. It is difficult to model it completely in this project. Therefore, there are certain features left for future work. Firstly, the ATC system is a highly interactive system interacting with various other third-party components but interfacing to humans and other components is outside the domain of our ATC system Model. Similarly, the communication between pilot and air traffic controller is also a tenet of ATC system but it is outside the scope of our ATC system Model. For safety in ATC system, Controller has also the job of Conflict Resolution but it is in itself a whole research area so we are not focusing on Conflict Resolution.

It is important to mention that the work done by Zafar et.al [42, 43] has been the starting point of our work but they presents modeling of railway interlocking system as an undirected network using VDM. The formal ATC Model is shown in figure 1.1.

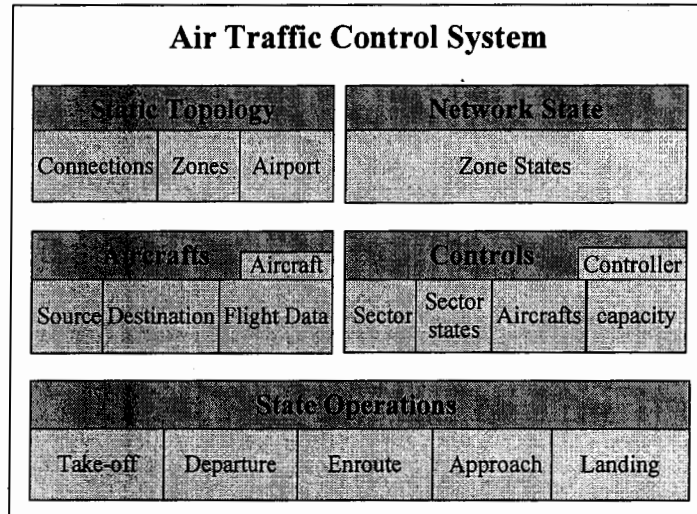


Figure 1.1
Formal Model of ATC System

1.8.1 NETWORK COMPONENTS

The major components of ATC Model are static Topology, Network State, Aircraft and Controller. The whole Formal ATC Model is then presented as encapsulation of formal models of its components. Therefore the modeling of ATC system in terms of these four components enhances the model. The invariants defined in the models helps to capture the real world safety properties those must be met to have a safe and reliable ATC system.

1.8.1.1 STATIC TOPOLOGY

The Static Topology is a fixed physical layouts of components related together to perform intended task. Therefore, Zone, Airport and Connections model the static Topology of an ATC System Model.

1.8.1.2 NETWORK STATE

The Network State of ATC system Model represents the dynamicity of aircrafts flying within the zones. It is assumed that there can be exactly one aircraft in a zone at a time.

1.8.1.3 AIRCRAFT

Modeling of aircrafts involves the maintenance of their important flight data like speed, heading and altitude.

1.8.1.4 CONTROLLER

Modeling of controllers, in our model controllers will be computer-based systems those monitor and track the aircrafts within their assigned airspace zones.

1.8.2 STATE OPERATIONS

The phases of flight of an aircraft are defined as state operations on formal ATC Model. The safety properties necessary to have a safe journey of an aircraft are defined. Freudenrich [7] described the flight profile of an aircraft, which has following five distinct phases.

- **Takeoff** - The pilot powers up the aircraft and speeds down the runway.
- **Departure** - The plane lifts off the ground and climbs to a cruising altitude.
- **En route** - The aircraft travels through one or more center of airspaces.
- **Approach** - The pilot aligns the aircraft with the designated landing runway.
- **Landing** - The aircraft lands on the designated runway.

Chapter # 2

Formal Methods

2. FORMAL METHODS

Formal Methods (FM) is an emerging technology. Before defining a concrete and rigorous model using FM it is therefore utmost necessary to have a brief introduction of FM. Therefore this chapter initially gives an overview of FM, and then the classification of FM and some commonly used Tools and Notations of FM are presented. Finally the chapter is concluded with successful industrial experiences using FM.

2.1 AN OVERVIEW

“Formal Methods (FM) refers to mathematically rigorous techniques and tools for the specification, design and verification of software and hardware systems” [48]. FM uses mathematical notations for writing specifications of the system to be developed. These mathematical notations are particularly derived from the area of set theory, discrete mathematics or graph theory. Thus formal specifications are mathematical expressions with well-defined syntax and semantics [32]. Using the technique of mathematical refinements FM can be used at every stage of software development process. Once formal specifications are written they can be refined into actually implemented system by the process of stepwise mathematical refinements. The Validation and Verification (V&V) technique [11] offered by FM is applied at each phase of the development process, which ensures the correctness, and consistency of the system giving high confidence in system to be developed.

The traditional approaches uses natural language or graphical notations to write system specifications. The multi-meaning vast vocabulary of natural language makes the specifications highly ambiguous [52]. Unlike traditional approaches formal specifications uses mathematical notations those have same interpretation throughout the globe [31]. Further the use of mathematics in writing specifications helps to have a deeper insight of system to be developed and provides an excellent medium for modeling of complex systems.

One of the major limitations of traditional approaches is that they lack the ability to prove the specifications for the presence of errors. The errors and inconsistencies are hidden behind textual or graphical requirements specifications [15], which penetrate to later phases of development process and those are identified only during Implementation and Testing phases. Errors identified during Implementation are not only difficult but also costly to fix [41]. On the other hand, the mathematical nature of specifications enables to carry out proves. The worth of conducting proves is that it explores the entire state space of the system to be developed. Therefore, only FM makes it possible to prove and analyze certain properties of the system during early stages of the development process so that errors in the requirement specifications can be identified and removed. Studies have suggested that FM have proven to be vital in improving the clearness and precision of requirements specification, and helps in identification of essential and subtle errors [55].

Another area in which FM has a promising impact is the Validation and Verification (V&V). Through the process of Validation it can be explored that whether the system to be developed conforms to its requirements. Whereas Verification attempts to establish whether the product of the particular phase of software development process meets the requirements established during previous phase. The traditional approaches cater the aspects of V&V by tedious error prone processes of audits, inspections, reviews, testing etc. However in the field of FM it involves writing various challenging theorems to check the properties of the system to be developed. Then mathematically rigorous proves of these theorems are conducted. When these theorems are proven successful it means system is correct and consistent else it needs revision.

Therefore, FM is an emerging and future technology with its focus to develop high quality and reliable systems [6, 33]. The claims of high quality and reliability is justified from the fact that FM are used in organizations attaining level 3 and above of Software Engineering Institute's process maturity framework [41]. FM are being successfully applied for development of hardware and software systems. For instance, Hardware engineers use FM, such as VHSIC Hardware Description Language (VHDL) descriptions, to model integrated circuits before fabricating them [41]. Similarly in the field of software development FM are applied to develop almost all range of systems ranging from small database applications to complex safety critical systems.

2.2 CLASSIFICATION OF FORMAL METHODS

FM can be classified in a number of ways. Following are some major classifications.

2.2.1 SOPHISTICATION-BASED FORMAL METHODS

A formal analysis technique may be classified as either light-weight or heavy-duty [55]. The user of a heavy-duty technique must be mathematically mature, a person having advanced mathematical training and theorem proving skills. The most common heavy-weight techniques are mechanical theorem provers, such as PVS and ACL2. On the other hand the user of a light-weight technique [5] does not require advanced mathematical training and theorem proving skills. "The studies demonstrate that a pragmatic, lightweight application of FM can offer a cost-effective way of improving the quality of software specifications" [55]. Examples of this lightweight approach to FM include the Alloy object modeling notation, Denney's synthesis of some aspects of the Z notation with use case driven development, and the CSK VDM Tools.

2.2.2 SEMANTIC-BASED CLASSIFICATION

Most commonly accepted classification of FM is based on semantic foundation of formal specification languages they use [23]. The two main semantic approaches are distinguished as Model Oriented and Property Oriented FM [23]. Model Oriented FM provide constructs which enables to specify a model of system's behavior in terms of a mathematical model using abstract data structures as sets, sequences, Cartesian products, maps and functions. CSP, Petri nets, statecharts, Z, VDM, RAISE, B-methods are some commonly used model-oriented FM. Whereas, Property Oriented FM languages enable the expression of necessary minimal constraints on the system's behavior, without prescribing an internal structure or a model of system. They give the designer some freedom in the further development of the system. Larch, ANNA and EHDM are some property-oriented FMs.

2.2.3 APPLICATION-BASED CLASSIFICATION

Since languages are usually focused on certain types of systems, examining the application domain makes another classification [23].

- Sequential Formalism, like VDM, Z and Larch, focuses sequential systems.
- Concurrent Formalism, like LOTOS and MTCCS, focuses on modeling processes and their interaction.
- Real-time formalism, like timed and hybrid automata, ET-LOTOS and MTCCS, allows modeling of events in time.
- Hybrid formalism, like hybrid automata and hybrid ASTRAL, are capable of modeling besides the discrete software components, the continuous physical real world environment.

2.2.4 GRAPHICAL-BASED FORMAL METHODS

FM can include graphical languages [60]. SDL are the most well-known graphical technique for specification of a system. Data Flow Diagram (DFD) and Unified Modeling Language (UML) can be considered a semi-formal method.

Petri nets provide another well-known graphical technique, which are a fully formal. Finally, finite state machines are often presented in tabular form. This does not decrease the formalism in the use of finite state machines. So the definition of FM provided earlier is quite encompassing.

2.3 NOTATIONS AND TOOLS

FM are in different stages of development, in a wide spectrum from formal languages with no tool support, to internationally standardized languages with tool support and industrial users [41]. Following are some major FM in use.

2.3.1 VIENNA DEVELOPMENT METHOD (VDM)

VDM [4] is a model oriented specification and design method. It is used for specifying the behavior of abstract data types and sequential programs. The behavior is specified in terms of preconditions and postconditions on pair of states. A precondition is a predicate that must be true in order for the result of an operation to be defined. A postcondition is a predicate that is true after the operation has completed. VDM includes both; a method, stepwise refinement, and a notation, META-IV.

2.3.2 Z NOTATION

The Z notation [24] is another model oriented approach, which is based on set theory and first order predicate logic. It is also used for specifying the behavior of abstract data types and sequential programs. A Z specification describes a state space for the system and a set of operations that may be performed on that state space. The state space corresponds to the variables that determine the program's state. The operations are defined as relations on pairs of states from the state space that are conceptually similar to VDM's preconditions and postconditions. Z allows a complex specification to be divided into smaller, more manageable and understandable, parts using schemas. A schema groups variable declarations with a list of predicates that constrain possible values of those variables. These parts can then be combined to produce the overall description of the system.

2.3.3 LARCH

Larch [25] is a property-oriented method for specification of sequential programs and abstract data types. A Larch specification consists of an axiomatic component for specifying state dependent behavior and an algebraic component for specifying state independent properties. Giving preconditions, postconditions for operations along with a list of objects whose values may be modified by the operation, specifies state dependent behavior. Providing declarations for operators specifies state independent behavior.

2.3.4 STATECHARTS

StateCharts [8] are used to specify state transitions in reactive systems. Reactive systems, in contrast to transformational systems, cannot be described in terms of simple functions that map inputs to outputs. The response that a reactive system provides to an input event, for example, depends on the current state of the system. The current state, in turn, is a function of the inputs that have already been received. Safety critical systems are considered to be reactive systems.

2.4 SUCCESSFUL INDUSTRIAL APPLICATIONS

The industry is full of major successes of FM. One of the best descriptions is portrayed by Clarke et.al [14]. Therefore the following text has been taken from [14].

2.4.1. CUSTOMER INFORMATION CONTROL SYSTEM (CICS)

Oxford University and IBM Hursley Laboratories collaborated in the 1980s on using Z to formalize part of IBM's CICS, an online transaction processing system with thousands of installations worldwide. Measurements taken by IBM throughout the development process indicated an overall improvement in the quality of the product, a reduction in the number of errors discovered, and earlier detection of errors found in the process. IBM also estimated a 9% reduction in the total development cost of the new release. The success of this work is well known and resulted in the Queen's Award for Technological Achievement. It inspired many others to follow suit.

2.4.2. THE CCF DISPLAY INFORMATION SYSTEM (CDIS)

In 1992 Praxis delivered to the UK Civil Aviation Authority the Central Control Function (CCF) Display Information System, a part of the new air traffic management system for London's airspace. CDIS is a distributed fault-tolerant system implemented on nearly 100 computers linked in a dual local-area network. Praxis used FM as an integral part of the development process and in conjunction with other software engineering, project management, and quality assurance techniques.

At the system specification stage, an abstract VDM model was developed in conjunction with concrete user interface definitions, semi-informal definitions of the concurrent behavior, and definitions of external interfaces. During design, the abstract VDM model was refined into more concrete module specifications. At a lower level, the software for the dual LAN was specified and developed formally using CCS. As a result the perceived and measured quality of the software was much higher. The delivered software had a defect rate of about 0.75 faults per thousand lines of code, a figure two to ten times better than that for published projects and comparable software in ATC applications that did not use FM.

2.4.3. LOCKHEED C130J

Praxis has worked with Lockheed on analyzing the code for the avionic software of the Lockheed C130J being supplied to the US Air Force and the RAF. The software is coded in the SPARK-annotated subset of Ada. Specifications are written in the Software Productivity Consortium's CORE notation, which is based on Parnas's tabular specifications. Many would expect that the use of SPARK would add to the cost of the software, while improving its quality. The added quality, however, decreased the overall cost of software development because of the huge savings in testing. The use of SPARK annotations to specify the behavior of the modules led to software that is close to being "correct by construction," and hence passes its tests instead of requiring expensive rework.

2.4.4. TRAFFIC COLLISION AVOIDANCE SYSTEM (TCAS)

In the early 1990s, the Safety Critical Systems Research Group at the University of California, Irvine (now at the University of Washington) produced a formal requirements specification for the Traffic Collision Avoidance System (TCAS) II, required on all commercial aircraft flying in US airspace. They used the Requirements State Machine Language (RSML), which is based on Statecharts with changes made to overcome difficulties found during the specification process.

After a group of industry and university representatives produced a first draft of the TCAS II specification, a private company on behalf of the Federal Aviation Administration took over the specification effort; official TCAS II documentation still uses RSML. Both the private company and the original university researchers have produced automated tools for RSML including simulators, test case generators and other test tools, and safety analysis tools. The TCAS II specification has been automatically checked for mathematical completeness and consistency and provably correct code can now be automatically generated from RSML specifications. The TCAS II project demonstrated

- The practicality of writing a formal requirements specification for a complex process-control system
- The feasibility of building a formal model of a system that can be read and reviewed by application experts without special training.

Chapter # 3

Abstract Model

3. ABSTRACT MODEL

In order to model the real ATC system it is foremost important to have the basic knowledge of domain. Therefore, in this chapter the ATC system has been introduced. ATC system can be considered as a network of various components performing their designated tasks. Therefore, the network components those are milestone of an ATC system, are introduced. Finally the flight profile of an aircraft is described which is essential aspect of ATC system.

3.1 AIR TRAFFIC CONTROL SYSTEM

International Civil Aviation Organization (ICAO) [21] is an agency of United Nations that defines principles and techniques for every aspects of air navigation to ensure safety and effectiveness of an ATC system. ICAO was created in 1944 and its headquarter is in Montreal, Canada. The ICAO Council develops standards, regulations and recommended practices concerning every aspect of air navigation and international civil aviations. There are 189 contracting states of ICAO which follow its standards in order to have maximum safety level. The objectives of ICAO are two folded, firstly, to bridge the gaps between contracting states and to establish consistency of rules to be implemented uniformly throughout the globe.

ATC is one of the most extensive systems of the world [16, 58]. It is a system in which continuous services are provided to the aircrafts by the ground-based controllers. The core of ATC system is an air traffic controller. A team of controllers, headed by a controller, works at each ground station like local, tower, center control stations. There is wide variety of aircrafts that ATC system needs to supervise ranging from small private aircrafts to large commercial airliners. ATC system provides a wide range of services to the aircrafts like navigational, weather updates, flight profile information, emergency relief services etc.

Behind every service offered by ATC system, there are two major goals [17]. First and foremost goal is to minimize or avoid aircraft collision. The term collision not only includes the mid-air collision of aircrafts but also the collision between aircraft and ground obstructions located in the way of its flight. Second major goal is to maximize the effectiveness of ATC system with little or no delays. The effectiveness of the system corresponds to greater number of aircrafts those can fly safely in the airspace without compromising the safety of all other aircrafts occupying airspace and airports. Similarly, as the number of aircrafts increases there must not be unwanted delays. The achievement of these goals can yield increased customer's confidence which lead to the profit of airline industry.

ATC system is a challenging domain to work on because it is a complex, highly distributed and safety critical system [34]. The domain of ATC system is very large and complicated offering wide range of functionalities and services to the aircrafts. Therefore, the complexity of the domain of ATC system is remarkably high that is hard to be handled effectively. The task of safe journey of an aircraft is distributed among various controllers who collaborate and direct an aircraft within their area of control. This distribution of control makes ATC system a highly distributed system. Moreover, the ATC system belongs to the domain of safety critical system. Any error can not only damage the aircraft but also the lives of on-board passengers and crew are at risk.

3.2 NETWORK COMPONENTS

The real world ATC system is a complex domain and it uses additional components like navigational aids, communication devices, radar and other tracking devices etc. Due to complexity of the domain it is difficult to model the whole ATC system in this short research project. Therefore only the major components of ATC system are modeled in our work [16, 58]. Following is a brief description of each component modeled by our ATC system.

3.2.1 CONTROLLER

The safe journey of an aircraft from take-off till landing is the responsibility of air traffic controller. Air traffic controller is qualified and skillful person who monitor and track aircrafts within their area of control. Each controller is assigned a unique identifier and certain dimension of airspace as its area of control. There is wide variety of aircrafts that controller needs to supervise ranging from small private aircrafts to large commercial airliners. It is important to note that a controller can service a number of aircrafts depending upon its capacity. The more efficient ATC system is the more aircrafts a controller will be able to control.

ATC system provides a wide range of services to the aircrafts like navigational, weather updates, flight profile information, emergency relief etc. Air traffic controller may use various devices and equipments like radar, radios, computer etc for effective supervision of aircrafts. Air traffic controller may communicate and coordinate with pilots and can issue relevant instructions and advisories about navigation, flight profile and emergency situation.

To avoid collision or any other conflicting situation the air traffic controller has complete information about flight profile of all aircrafts flying within its area of control. An air traffic controller callaborates with other controllers to hand-off an aircraft which has safely flown within its controlled airspace. After successful hand-off the aircraft communicates with new airspace controller and its contact with previous airspace controller comes to an end.

A team of controllers, headed by a controller, works at each ground station like local, tower, center control stations. Since we are giving an automated solution of an ATC system the controller in our model will be computer-based system monitoring and controlling aircrafts within their area of control. The purpose is not to eliminate human controllers but to assist them to do their tasks efficiently and easily.

3.2.2 AIRCRAFT

The essential component of an ATC system is an aircraft. Aircrafts have large diversity in terms of their size and budget. The small home-made gliders, helicopters, commercial regularly scheduled airbus, private aeroplane, mission-oriented military aviations and special purpose general avionics all belong to the category of aircrafts. Depending on the type, aircrafts have different speed and altitude limitation. Factors like weight, wind and weather also affects performance of the aircraft. Each aircraft utilizing ATC services is assigned a unique identification mark called as callsign. Through the callsigns aircraft communicates with air traffic controller to avail necessary services.

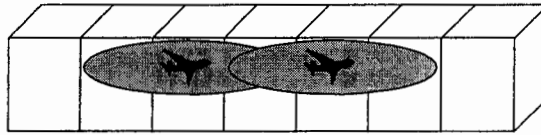


Figure 3.1

An Overlapping of Protected Zones of Two Aircrafts

The utmost goal of an ATC system is the safety of aircrafts and its onboard passengers. Safety is ensured when there is no conflicting situation and there is safe separation between aircrafts utilizing ATC services. The collision or a possible loss of separation may occur if the standards of safe separation are violated. One of the most promising concept in regard to safe separation of aircraft is the protected zone. Our work is focussed on protected zone to ensure safety.

“The protected zone is an area of defined dimension closer to the aircraft” [49]. Each aircraft utilizing ATC services has a protected zone around itself. An aircraft can not intrude protected zone of another aircraft. The conflict or loss of separation occurs when a protected zone of two or more aircrafts are overlapped. Figure 3.1 shows this scnerio. The protected zones of two aircrafts are overlapping indicating a conflict or loss of separation.

ICAO has determined the following separation standards

- “Above 29,000 feet, when aircraft are cruising at high speeds in the en route airspace, the standard is five miles of horizontal radar separation or 2,000 feet of vertical separation.
- Below 29,000 feet in the en route airspace, the vertical separation is reduced to 1,000 feet while the horizontal radar separation remains at five miles.
- When aircraft are moving at much slower speeds as they depart or approach an airport, the standard is three miles of horizontal radar separation or 1,000 feet of vertical separation.
- In certain oceanic airspace, vertical separation has been reduced to 1,000 feet at altitudes above 29,000 feet”. [7]

3.2.3 AIRPORT

An airport is considered to be an area that can offer the services of take-off and landing to the aircrafts. The airport ranges from just a small strip in an open area to modern heavily equipped airports of major cities. Some airports are for public use whereas some are reserved for military or special purpose usage.

The essential feature of any airport is a runway. Runway is a strip of ground serving the purpose of take-off and landing for aircrafts. Each runway has an assigned number and a defined length. Length of runway is vital for safe landing of aircrafts having different weights under different weather conditions.

3.2.4 AIRSPACE

The airspace around the globe is divided into large three-dimensional segments known as Centers. Each center is further divided into smaller sectors. ICAO broadly classifies the airspace as controlled and uncontrolled. The different classification of airspace [16, 58] ranging from Class A, Class B, Class C, Class D, Class E and Class G airspace are shown in figure 3.2.

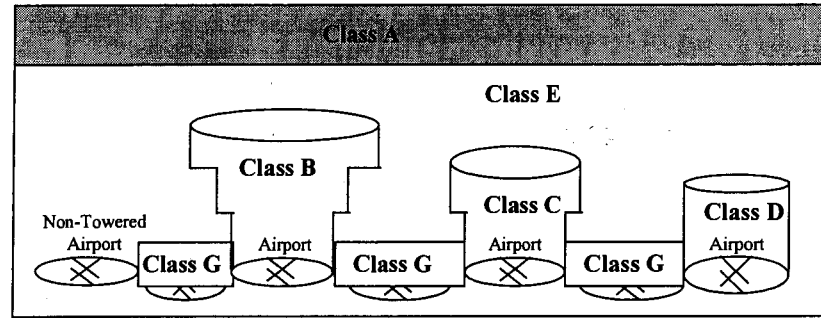


Figure 3.2
Classification of Airspace

A controlled airspace is an area in which active services are provided to aircrafts utilizing ATC system. The continuous supervision of controllers is required because the traffic load is very high in these areas like a metropolitan area, a busy airport etc. Since we are interested in a safe and efficient system to be built we will be focussing on an airspace providing maximum supervision to the aircrafts. Therefore, in our model every airspace segment is assumed to be controlled in order to have high level of safety.

3.2.5 AIRWAY

The concept of airways is prevailing in the domain of ATC system in order to have safe and orderly flow of air traffic. An airway is a designated area of airspace through which aircrafts are directed to fly. Airways start at 1,200 feet above ground level and extend upward to an altitude of 18,000 feet mean sea level. Airways are normally 8 nautical miles wide.

An airway is referred to as a corridor in the sky and most frequently termed as highway in the sky [44]. The system of airways is more closely compared to the networks of roads. Just like the vast network of roads spreads across the globe. Similarly the extensive network of airways is defined around the globe. Moreover just like roads some airways allow bi-directional whereas some allows unidirectional flow of traffic. Regardless of direction the basic goal is to allow safe and swift flow of aircrafts ranging from small private aircrafts to large commercial airliners.

3.3 FLIGHT PROFILE

When an aircraft flies from source to destination it undergoes certain phases, these phases are called Flight Profile of an aircraft. It is important to note that aircraft needs the supervision of air traffic controller so that it can complete each phase successfully. The aircraft must go through each phase safely before starting the new phase of its flight profile in order to have a safe journey. The flight profile of an aircraft [7] has five distinct phases namely; take-off, departure, en-route, approach and landing as shown in figure 3.3. A detailed description of each phase is narrated in this section.

3.3.1 TAKEOFF

The pilot of the aircraft initially files a flight plan. Along with other contents the important feature of a flight plan is the route to be followed for the journey. The air traffic controller at the airport reviews the weather and flight plan using all available devices in order to avoid any potential conflicting situation. After the successful review process the flight progress strip is generated which is a promising means to track the aircraft.

Then the controller searches a vacant runway, which is not being assigned to another aircraft. Moreover the controller also reviews the availability of airspace above the airport in order to avoid any conflict. If the review is successful the aircraft is assigned the vacant runway. The aircraft is now being cleared to take-off and it is then handed off to departure controller.

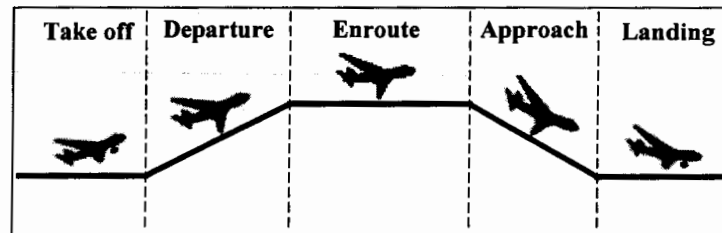


Figure 3.3
Flight Profile of an Aircraft

3.3.2 DEPARTURE

It is the phase in which aircraft ascends towards the en-route portion of the flight as shown in figure 3.4. The departure controller has the airspace of 50 miles radius above the airport as its area of control. The controller uses various devices like radar to ensure safety and avoid conflicting situations. After the successful departure the aircraft maintains the en-route cruising altitude.

3.3.3 ENROUTE

After flying 30 miles from the source airport, the aircraft ends its contact with departure controller and communicates with center controller. Enroute phase of the flight consists of the successions of handoffs between center controllers. Depending on the route the aircraft is handed off many times through out its flight. The center controllers are responsible for safety of all aircrafts flying within their area of control.

The controllers at center communicate with pilots and with other center controllers in order to ensure safe separation of aircrafts. Depending on situation center controller can issue instructions like weather updates or change in speed, heading or altitude. In some critical situations like bad weather or heavy traffic load etc the center controller has an authority to change route of the flight. Similarly the pilot can request the change in route, heading or altitude. In both cases center controller has the responsibility to ensure safety of all aircrafts flying within its area of control.

When an aircraft safely reaches end of the boundary of a center it is then handed-off to next center. There are two types of hand-offs. To ensure maximum level of safety, in most of the hand-offs details of the aircraft are transferred to the next center controller. On the other hand some handoffs are silent that means the details of aircrafts are not transferred to receiving center controllers if the aircrafts are flying in an agreed manner. After each successful hand off the aircraft ends its contact with previous controller and establishes its contact with next controller. This series of hand offs continue to proceed until aircraft reaches 150 miles before destination airport.

TH-4431

3.3.4 APPROACH

It is the phase in which aircraft descends towards the destination airport. In order to avoid any conflicting situation the controller may direct the aircraft to place themselves in a holding pattern. Holding patterns are defined routes in the airspace above airport, which are used when airport is already occupied and cannot handle further arrivals.

When airport is available for arrival the controller directs the aircrafts to adjust their speed, heading and altitude and align themselves to the designated runway. This process continues until aircraft reaches 10 miles away from destination runway. The aircraft now end its contact with previous airspace controller and communicates with local controller of the airport.

3.3.5 LANDING

The local controller monitors the clearance of assigned runway. It is ensured that any other aircraft is not utilizing the assigned runway. The controller also monitors the weather updates and issues relevant instructions. When the controller found it safe the clearance for landing is granted to the aircraft.

When an aircraft lands successfully it is then directed to the taxiway and then to the terminal gate. During this process there is a possibility of interferences with active runway. Therefore the controller also has a job to avoid any conflicting situation that can threat the safety of aircraft and onboard passengers. After successful landing the aircraft ends its communication with the local controller.

Chapter # 4

Formal ATC Model

4. FORMAL ATC MODEL

Firstly, this chapter introduces the Z notation, which is used to formalize our ATC system model. Then the chapter focuses the transformation of real world ATC system into a directed graph, which is then used to formalize the major components of formal ATC Model i.e static Topology, Network State, Aircraft and Controller. The whole formal ATC Model is then presented as encapsulation of formal models of its components. Finally, the phases of flight of an aircraft are defined as state operations on formal ATC Model. The safety properties are defined for each phase of flight so that safety of aircraft is ensured from takeoff till landing.

4.1 THE Z NOTATION

The Z notation is based upon set theory and mathematical logic. The set theory includes standard set operators, set comprehensions, Cartesian products, and power sets [24]. On the other hand the logic of Z notation is formulated using first order predicate calculus. The combination of logic and set theory makes Z notation one of the most easy and widely used FM.

The use of Z notation allows to organize the system in smaller chunks known as Schemas. In Z notation schema is used to define state of the system and the operations, which alter state of the system. The properties and constraints on a mathematical object are collected together in a schema. Therefore, schema is a pattern of declaration of the state of system. Similarly, schema also defines the way in which state of the system can be modified. While declaring state operations preconditions and post conditions for execution of a particular modification are defined. Thus usage of schema language offers a simple, easy to use and uniform way for defining a complete system.

A promising aspect of Z notation is the mathematical refinement. “In FM, program refinement is the stepwise verifiable transformation of an abstract (high-level) formal specification into a concrete (low-level) executable program” [61]. Once formal specifications in Z notation are written they can be refined into actually implemented system by the process of stepwise mathematical refinements.

There are more than 90 techniques of FM [30] but the following reasons compel us to choose Z notations to model our ATC system.

- The basic goal is the correct and error-free specifications. In Z notation every object is assigned a unique type which provides a useful link to programming practice, this notion of types means that an algorithm can be written to check the type of every object in a specification; several type-checking tools exist to support the practical use of Z [26]. The major reason to write the specifications in Z notation is to check, prove and analyze it using a powerful tool such as Z/EVES. So that the specifications have no potential safety critical errors.
- The rich mathematical notations offered by Z make it possible to reason rigorously and effectively about the behavior of specified system.
- Since we are at initial stages of requirements specification of the project, the specifications at this stage are not large or hard to be managed. Therefore, structuring in terms of objects is not an issue.
- Since we are modeling a safety critical system. Despite the advent of other counterparts, Z notation is still favorite candidate for safety critical or life critical systems. It is evident from numerous success stories of FM using Z notation [14].

Like any other tool Z notation has certain limitations as well. Z notation is not anticipated for defining non-functional properties, such as usability, performance, size, and reliability of a system. Secondly, it is not considered to be an ideal medium for declaring temporal and concurrent behavior of a system. Thirdly, Z is also not well suited for defining user interfaces of any system. However, these limitations can be overcome. There are many other FM suitable for representation of above-mentioned aspects. Therefore, Z can be used in combination with other FM to overcome its limitations.

4.2 ATC MODEL IN GRAPH THEORY

In real world, airspace around the globe is divided into large three-dimensional segments known as Centers. Connectivity between airspace segments is vital for safe journey of an aircraft. It helps to swiftly decide the route of journey and is useful in determining the position of aircraft. In real world it can be observed that the intricate network of airways extends within the airspace around the globe. Therefore, it can be inferred that an airway connects different airspace segments. In formal ATC Model the relationship between controlled airspace and airways is exploited because connectivity of airspace segments is one aspect that can strengthen safety of ATC system.

The work of Zafar et.al [42, 43] has been the starting point of our work but it presents the modeling of railway interlocking system as an undirected network using VDM. Just like the work of [42] our formal ATC Model focuses Graph Theory [57], which is considered to be a compliant means in solving problems of connectivity. Unlike [42] the ATC system model in real world is transformed into a directed graph [36, 39]. The Directed Graph is exploited because the direction of connections is worthy because some airways allow bi-directional whereas some allows unidirectional flow of traffic.

In our model the controlled airspace is further divided into smaller airspace segments termed as Zones. If there is an airway segment connecting two zones, it means the two zones are connected and aircrafts can fly directly between them. Therefore, set of all zones in the airspace represents nodes of the graph and the set of airway segments connecting them as arcs. The direction of arcs indicates the direction of flow of traffic. Figure 4.1 shows ATC Model in graph theory representing six zones z1, z2, z3, z4, z5 and z6. Ten airway segments represent the interconnections between the zones.

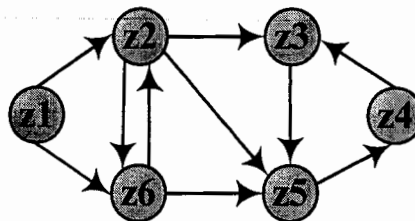


Figure 4.1
ATC Model in Graph Theory

The directed graph, in figure 4.1, shows the arc between z1 and z2 is unidirectional i.e aircrafts can fly from z1 to z2 and not from z2 to z1. This connection can be represented as ordered pair (z1,z2) whereas the connection between z2 and z6 is bi-directional i.e aircrafts can fly from z2 to z6 and from z6 to z2. This connection of zones is represented by ordered pairs (z2,z6) and (z6,z2). Therefore, graph of figure 4.1, is defined using Connections relation as shown below.

$$\text{Connections} = \{(z1, z2), (z1, z6), (z2, z3), (z2, z5), (z2, z6), (z3, z5), (z4, z3), \\ (z5, z4), (z6, z2), (z6, z5)\}$$

4.3 NETWORK COMPONENTS

The real world ATC system is transformed into a directed graph, which is then used to formalize the components of formal ATC Model i.e static Topology, Network State, Aircraft and Controller. The whole formal ATC Model is then presented as encapsulation of formal models of its components. The subsequent sections describe the basic components of ATC system in detail.

4.3.1 STATIC TOPOLOGY

The Static Topology is a fixed physical layouts of components related together to perform intended task. The topology remains unchanged and represented by fixed data structures. Therefore, Zone, Airport and Connections are used to model the static Topology of an ATC system.

4.3.1.1 ZONE

Each Zone is defined as a small three-dimensional entity of Airspace. Since an abstract model is presented, we are not concerned with its shape and geometry. Thus, it is modeled by variable Zone as a collection of abstract type POINT.

[Point]

Zone == P Point

4.3.1.2 CONNECTION

The connectivity between two zones is modeled as relation `Connection`. It not only represents connectivity but also indicates the direction of flow of air traffic. It is important to note that a zone is not connected with itself.

$$\text{Connection} == \{ z1, z2: \text{Zone} \mid z1 \neq z2 \cdot (z1, z2) \}$$

4.3.1.3 STATIC MODEL

The connectivity of all zones forming airspace around the globe and airports enclosed in the respective zones are represented in schema `StaticTopology`. It actually represents the static model of the ATC system.

[Runway]

<i>StaticTopology</i>
<i>connections</i> : P <i>Connection</i> <i>Airport</i> : <i>Runway</i> \mapsto <i>Zone</i>
$\forall z1: \text{Zone} \cdot \exists z2: \text{Zone} \cdot (z1, z2) \in \text{connections} \vee (z2, z1) \in \text{connections}$ $\forall z: \text{Zone} \mid z \in \text{ran } \text{Airport}$ $\cdot \exists z1: \text{Zone} \cdot z \mapsto z1 \in \text{connections} \wedge z1 \mapsto z \in \text{connections}$

State Variables:

The state variables defined in schema are

- The state variable `connections` is modeled as a set of connection in network.
- Although modern airports have many advanced devices for providing efficient services to aircrafts. But the most promising component, common to all airports, is a runway, represented by abstract data type `Runway`. Therefore, the variable `Airport` is modeled as a total injective function of `Runway` and `Zone`. It means each `Runway` is assigned to exactly one `Zone` and no two `Zones` can have the same `Runway`.

Invariants:

1. All zones must be connected with other zones either uni-directionally or bi-directionally. It means, if an aircraft can move directly from one zone to another it may or may not be possible for it to move in the opposite direction.
2. Every airport must have incoming and outgoing airways.

4.3.2 NETWORK STATE

The Network State of ATC system Model, defined in the schema named `DynamicTopology`, represents the dynamicity of aircrafts flying within the zones. It is assumed that there can be exactly one aircraft in a zone at a time. Each aircraft is assigned a unique identification mark represented by abstract data type `AircraftId`. The state of each zone is indicated in variable `State`. If an aircraft resides in a zone, the state of zone is marked as `OCCUPIED` else the state is `CLEAR`.

[*AircraftId*]

State ::= *CLEAR* | *OCCUPIED*

DynamicTopology

zoneStates: *Zone* → *AircraftId*

∀*z*: *Zone*

• ∀*a1, a2*: *AircraftId* • *z* → *a1* ∈ *zoneStates* ∧ *z* → *a2* ∈ *zoneStates* ⇒ *a1* = *a2*

State Variables:

The state variables defined in the schema are

- A partial function of `Zone` and `AircraftId` is declared using the state variable `zoneStates`. It means that, one zone is occupied by zero or one aircraft at a time. Similarly, an aircraft, represented by `AircraftId`, belongs to zero or more `Zone` at a time.

Invariants:

1. Each aircraft belongs to exactly one airspace zone at a time and no two zones can have a same aircraft at any particular time.

4.3.3 AIRCRAFT

The schema *Aircraft* describes flight data of Aircraft utilizing ATC services. Each Aircraft is assigned a unique identification mark called *AircraftId*, the variable *Aircrafts* represents a total injective function of *AircraftId* and *Aircraft*. It means, each *AircraftId* is assigned to exactly one *Aircraft* and no two *Aircrafts* have the same *AircraftId*, at a time.

$Aircrafts == AircraftId \rightarrow Aircraft$

$AircraftType ::= CIVIL \mid GENERAL \mid MILITARY$

<i>Aircraft</i>
<i>source</i> : Zone <i>destination</i> : Zone <i>type</i> : AircraftType <i>protectedZone</i> : P Connection <i>currentSpeed</i> : N <i>currentAltitude</i> : N <i>heading</i> : N <i>speedLimit</i> : N <i>altitudeLimit</i> : N
<i>source</i> ≠ <i>destination</i> <i>protectedZone</i> ≠ ∅ $\forall z1, z2: Zone \mid z1 \mapsto z2 \in protectedZone \cdot destination \neq z1$ <i>currentSpeed</i> < <i>speedLimit</i> <i>currentAltitude</i> < <i>altitudeLimit</i> <i>heading</i> < 360

State Variables:

The state variables defined in schema are

- A Zone from which aircraft has flown is represented by variable *source*.
- A Zone to which aircraft is destined to land is defined by variable *destination*.
- The aircraft type is represented by variable *type*, which can be Civil, General or Military Aviation.
- The protected zone of an aircraft is modeled as the power set of Zone.

- The speed, altitude, heading, speed limit, and altitude limit of an Aircraft are represented as natural numbers in variables `currentSpeed`, `currentAltitude`, `heading`, `speedLimit`, `altitudeLimit` respectively.

Invariants:

1. An Aircraft cannot have the same zone as its source and destination.
2. Every aircraft must have a protected zone around itself.
3. Protected zone of the aircraft must end at destination zone.
4. The current speed should not exceed Aircraft's speed limitation.
5. The current altitude should not exceed altitude limit of an Aircraft.
6. The heading of an aircraft should not be greater than 360 degrees.

4.3.4 CONTROLLER

The safe journey of any aircraft, from takeoff till landing, requires the services of ground-based controllers. Each controlled airspace is being monitored and controlled by a team of controllers. The controllers in our model will be computer-based systems those monitor and track the aircrafts within their assigned airspace zones. Figure 4.2 shows ATC Model with cubes representing three dimensional airspace segments termed as zones and the aircrafts within them, being controlled by computer-based controllers.

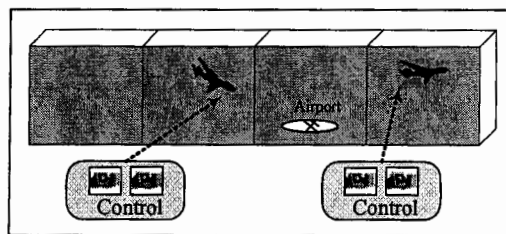


Figure 4.2

ATC Model with Computer-based Controllers

A controller is represented by schema `Controller`. Each controller monitors and directs an aircraft within the airspace zones assigned to it. It is mandated that each zone can have one aircraft at a time.

There is a unique identification mark, represented by abstract data type `ControlId`, assigned to each controller. The variable `Controls` represents a total injective function of `ControlId` and `Controller`. It means, there is exactly one `ControlId` assigned to each controller and no two controllers have the same `ControlId`.

[*ControlId*]

Controls == *ControlId* \rightarrow *Controller*

Controller

sector: P *Zone*

states: *Zone* \rightarrow *State*

aircrafts: *Aircrafts*

capacity: N

dom *states* = *sector*

aircrafts \in F (*AircraftId* \times (*altitudeLimit*: Z; *currentAltitude*: Z;
currentSpeed: Z; *destination*: P *Point*; *heading*: Z;
source: P *Point*; *speedLimit*: Z))

aircrafts < *capacity*

State Variables:

There are four variables declared in the schema

- The collection of zones is termed as a Sector. The set of `Zone` under the command of controller is represented in variable `Sector`.
- The variable `States` is defined as a total function of `Zone` and `State` that represent the state of each zone of Sector. It means each zone has a single state value either OCCUPIED or CLEAR.
- The number of aircrafts within a Sectors is represented by variable `aircrafts`.
- The variable `capacity` defined as a natural number represents the maximum number of aircrafts that controller can control.

Invariants:

1. All the zones having a state value must belong to the controlled sectors.
2. There must be a finite number of aircrafts flying within the Sector.
3. The number of aircrafts flying within the Sector must be less than or equal to capacity limit assigned to the controller.

4.3.5 ATC SYSTEM

The formal models of Static Topology, Network State, Aircraft and Controller, presented in earlier sections, are encapsulated to model the ATC system as shown in figure 4.3. Formal ATC model is defined in the schema named *ATCSys*tem.

*ATCSys*tem

\exists *StaticTopology*

\exists *DynamicTopology*

aircrafts: *Aircrafts*

controls: *Controls*

\forall *ald*: *AircraftId* | *ald* \in dom *aircrafts*

- \exists *aircraft*: *Aircraft* | *aircrafts* *ald* = *aircraft*
 - *aircraft* . *source* \in ran *Airport*
 - \wedge *aircraft* . *destination* \in ran *Airport*

\forall *ald*: *AircraftId* | *ald* \in dom *aircrafts*

- \exists *cid*: *ControlId* | *cid* \in dom *controls*
 - \exists *control*: *Controller* | *controls* *cid* = *control*
 - dom *control* . *aircrafts* \subseteq dom *aircrafts*

\forall *ald*: *AircraftId* | *ald* \in dom *aircrafts*

- \exists *z*: *Zone* | (*z*, *ald*) \in *zoneStates*
- \exists *a*: *Aircraft* | *aircrafts* *ald* = *a*
 - \forall *z1*: *Zone*
 - (*z1*, *z*) \in *a* . *protectedZone* \wedge (*z*, *z1*) \in *a* . *protectedZone*

\forall *z1*, *z2*: *Zone*

- \exists *a1*, *a2*: *Aircraft*
 - ((*z1*, *z2*) \in *a1* . *protectedZone* \vee (*z2*, *z1*) \in *a1* . *protectedZone*)
 - \wedge ((*z1*, *z2*) \in *a2* . *protectedZone* \vee (*z2*, *z1*) \in *a2* . *protectedZone*)
 - \Rightarrow *a1* = *a2*

\forall *z*: *Zone* • \forall *c1*, *c2*: *Controller* • *z* \in *c1* . *sector* \wedge *z* \in *c2* . *sector* \Rightarrow *c1* = *c2*

\forall *aid*: *AircraftId*

- \forall *c1*, *c2*: *Controller*
 - *aid* \in dom *c1* . *aircrafts* \wedge *aid* \in dom *c2* . *aircrafts* \Rightarrow *c1* = *c2*

\forall *z*: *Zone* | *z* \in dom *zoneStates*

- \exists *c*: *Controller* | *z* \in *c* . *sector* \wedge *z* \in dom *c* . *states*
 - *z* \mapsto *OCCUPIED* \in *c* . *states*

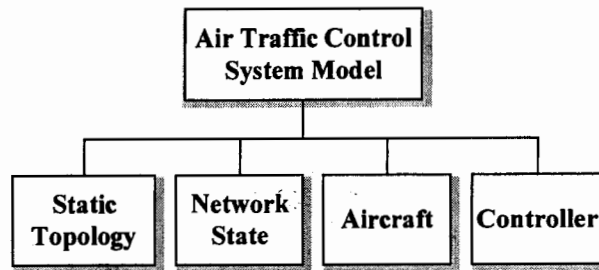


Figure 4.3

Components of Formal ATC System Model**State Variables:**

The declaration part of schema includes

- The schema `StaticTopology` is included with all its declarations and constraints but with no state change privileges granted on this schema.
- Similarly, the schema `DynamicTopology` is included with all its declarations and constraints but with no state change privileges granted on it.
- The set of aircrafts flying within airspace is represented by variable `aircrafts`.
- The set of controllers controlling all the Sectors of airspace is represented by variable `controls`.

Invariants:

1. The source and destination of all aircrafts must be airport.
2. All aircrafts in the system must be under the control of a controller.
3. For every aircraft occupying a particular zone, their protected zone consists of all zones connected with it through incoming or outgoing airway segments.
4. There must not be any overlapping of protected zones of any two aircrafts.
5. Every zone must be under the control of one controller at a time.
6. Every aircraft must be under the control of one controller at a time.
7. For all zones, occupied by aircrafts, must have state value `OCCUPIED`.

4.4 STATE OPERATIONS

The state operations are defined for the formal ATC Model, defined in previous section. An air traffic controller (computer-based system), which must collaborate with other controllers and with pilots, manages each airspace Sector. Through out the flight, controller of Sector maintains important flight data of an aircraft. The phases of flight are formalized as state operations [37, 38]. During the flight an aircraft has five distinct phases as shown in figure 3.3. The safety properties are defined for each phase of flight so that safety of aircraft is ensured from takeoff till landing.

4.4.1 TAKEOFF AN AIRCRAFT

The first phase of the flight of any aircraft is Takeoff. This operation is formalized in a schema named `TakeOff`. It takes an Aircraft as input that is ready for takeoff, after clearance of all ground activities. The aircraft must be on the airport that is defined as its source of flight. The source airport must be connected with other zones i.e there must be some outgoing airways to be used for takeoff. The Zone enclosing airport must belong to a Sector, being controlled by a valid controller. The controller of Sector of airspace now controls the Aircraft.

<i>TakeOff</i>
$\exists \text{ATCSystem}$ $a?: \text{AircraftId}$
$a? \in \text{dom aircrafts}$ $a? \in \text{ran zoneStates}$ $\exists z: \text{Zone} \cdot z \mapsto a? \in \text{zoneStates} \wedge z \in \text{ran Airport} \wedge (\text{aircrafts } a?). \text{source} = z$ $\exists z: \text{Zone} \mid z \mapsto a? \in \text{zoneStates}$ <ul style="list-style-type: none"> • $\exists \text{cid}: \text{ControllId} \mid \text{cid} \in \text{dom controls}$ • $\exists c: \text{Controller} \mid \text{controls } \text{cid} = c$ <ul style="list-style-type: none"> • $z \in c. \text{sector} \wedge z \mapsto \text{OCCUPIED} \in c. \text{states}$ $\exists z: \text{Zone} \mid z \mapsto a? \in \text{zoneStates} \wedge z \in \text{ran Airport}$ <ul style="list-style-type: none"> • $\exists z1: \text{Zone} \cdot z \mapsto z1 \in \text{connections}$

Pre-conditions:

1. The input AircraftId must correspond to a valid aircraft.
2. The aircraft must reside in a valid zone having a state value.
3. The zone occupied by aircraft must enclose an airport, defined as its source.
4. The source airport must be within a zone of a controlled Sector.
5. The source airport must be connected with other zones i.e must have outgoing airway segments.

4.4.2 DEPARTURE OF AN AIRCRAFT

This operation is specified in a schema named *Departure*. After successful takeoff of Aircraft, represented by AircraftId given as input, a zone belonging to a controlled Sector is searched. It is important to note that the searched zone must be clear and connected with source airport. Aircraft is now handed off to the searched zone and previous occupied zone is made clear for further use.

Departure Δ ATCSystem Δ Controller $a?:$ AircraftId $a? \in \text{dom } \textit{aircrafts}$ $a? \in \text{ran } \textit{zoneStates}$ $\exists z: \textit{Zone} \cdot z \mapsto a? \in \textit{zoneStates} \wedge z \in \text{ran } \textit{Airport} \wedge (\textit{aircrafts } a?). \textit{source} = z$ $\exists z: \textit{Zone} \mid z \mapsto a? \in \textit{zoneStates}$

- $\exists z1: \textit{Zone} \mid z \mapsto z1 \in \textit{connections} \wedge z1 \in \text{dom } \textit{zoneStates}$

- $\exists cid: \textit{ControllId} \mid cid \in \text{dom } \textit{controls}$

- $\exists c: \textit{Controller} \mid \textit{controls } cid = c$

- $z1 \in c. \textit{sector} \wedge z1 \mapsto \textit{CLEAR} \in c. \textit{states}$

 $\Rightarrow \textit{zoneStates}' = \textit{zoneStates} \oplus \{z1 \mapsto a?\} \wedge \textit{states}' = \textit{states} \oplus \{z1 \mapsto \textit{OCCUPIED}\}$
 $\wedge \textit{zoneStates}' = \textit{zoneStates} \oplus \{z \mapsto a?\} \wedge \textit{states}' = \textit{states} \oplus \{z \mapsto \textit{CLEAR}\}$

Pre-conditions:

1. The input AircraftId must correspond to a valid aircraft.
2. The aircraft must reside in a valid zone having a state value.
3. The aircraft must be successfully taken off.
4. There must be a clear zone, connected with the source airport, belonging to a controlled Sector so that aircraft can be handed over to that zone.

4.4.3 HANDOVER AN AIRCRAFT

This operation is formalized in schema named *HandOff*. It is used in enroute phase of flight. Depending on the route, Aircraft (represented by AircraftId given as input) is handed off from current zone (input) to desired zone (input) provided capacity of receiving Controller is not violated. After successful handoff, the Aircraft closes its communication with previous Controller and communicates with receiving Controller.

HandOff Δ ATCSystem Δ Controller $a?: AircraftId$ $to?, from?: Zone$ $a? \in \text{dom } aircrafts$ $a? \in \text{ran } zoneStates$ $to? \mapsto from? \in connections$ $from? \notin \text{dom } zoneStates$ $\exists cid: Controlld \mid cid \in \text{dom } controls$

- $\exists c: Controller \mid controls \ cid = c$

- $to? \in c. sector \wedge to? \mapsto OCCUPIED \in c. states$

 $\exists cid: Controlld \mid cid \in \text{dom } controls$

- $\exists c: Controller \mid controls \ cid = c$

- $from? \in c. sector$

- $\wedge from? \mapsto CLEAR \in c. states \wedge c. aircrafts \in F \text{ aircrafts}$

- $\wedge \# c. aircrafts < c. capacity$

- $\Rightarrow zoneStates' = zoneStates \oplus \{(from? \mapsto a?)\}$

- $\wedge states' = states \oplus \{(from? \mapsto OCCUPIED)\}$

- $\wedge zoneStates' = \{to?\} \triangleleft zoneStates \wedge states' = states \oplus \{(to? \mapsto CLEAR)\}$

Pre-conditions:

1. The input AircraftId must correspond to a valid aircraft.
2. The aircraft must reside in a valid zone having a state value.
3. The current zone and destination zone must be connected.
4. The destination zone must be clear.
5. The current zone occupied by aircraft must belong to a controlled Sector.
6. The destination zone must be clear and within a controlled Sector. The controller of destination zone must have aircrafts less than its capacity limit.

4.4.4 DESCENT OF AN AIRCRAFT

This operation is specified in a schema named *Approach*. The Aircraft, represented by AircraftId given as input, requests arrival, near the destination Airport. During this phase, the Aircraft leaves the current zone and enter the zone enclosing destination Airport. After successful approach, the aircraft is successfully handed over to destination Airport's Controller.

Approach Δ ATCSystem Δ Controller $a?: AircraftId$ $a? \in \text{dom } aircrafts$ $a? \in \text{ran } zoneStates$ $\exists z: Zone \mid z \mapsto a? \in zoneStates$

- $\exists cid: ControllId \mid cid \in \text{dom } controls$
- $\exists c: Controller \mid controls \ cid = c$
- $z \in c . sector \wedge z \mapsto OCCUPIED \in c . states$

 $\exists z: Zone \mid z \mapsto a? \in zoneStates$

- $\exists z1: Zone \cdot z \mapsto z1 \in connections \wedge z1 \in \text{ran } Airport \wedge (aircrafts \ a?) . destination = z1$

 $\exists z: Zone \mid z \mapsto a? \in zoneStates$

- $\exists z1: Zone \mid z \mapsto z1 \in connections \wedge z1 \notin \text{dom } zoneStates$
- $\exists cid: ControllId \mid cid \in \text{dom } controls$
- $\exists c: Controller \mid controls \ cid = c$
- $z1 \in c . sector \wedge z1 \mapsto CLEAR \in c . states$

 $\Rightarrow zoneStates' = zoneStates \oplus \{(z1 \mapsto a?)\}$ $\wedge states' = states \oplus \{(z1 \mapsto OCCUPIED)\} \wedge zoneStates' = zoneStates \oplus \{(z \mapsto a?)\}$ $\wedge states' = states \oplus \{(z \mapsto CLEAR)\}$

Pre-conditions:

1. The input AircraftId must correspond to a valid aircraft.
2. The aircraft must reside in a valid zone having a state value.
3. The current zone occupied by aircraft must be within a controlled Sector.
4. The current zone occupied by aircraft must be connected with a zone enclosing destination airport.
5. The zone, enclosing destination airport, must be clear and within a controlled Sector, being controlled by a valid controller.

4.4.5 LANDING OF AN AIRCRAFT

This operation is specified in a schema named *Landing*. It deals with landing of Aircraft, given as input. During landing it is ensured that aircraft lands safely. The zone enclosing destination airport must belong to a controlled Sector.

Landing $\exists \text{ATCSystem}$ $a?: \text{AircraftId}$ $a? \in \text{dom aircrafts}$ $a? \in \text{ran zoneStates}$ $\exists z: \text{Zone}$

- $z \mapsto a? \in \text{zoneStates} \wedge z \in \text{ran Airport} \wedge (\text{aircrafts } a?) . \text{destination} = z$

 $\exists z: \text{Zone} \mid z \mapsto a? \in \text{zoneStates}$

- $\exists \text{cid}: \text{ControlId} \mid \text{cid} \in \text{dom controls}$

- $\exists c: \text{Controller} \mid \text{controls } \text{cid} = c$

- $z \in c . \text{sector} \wedge z \mapsto \text{OCCUPIED} \in c . \text{states}$

Pre-conditions:

1. The input AircraftId must correspond to a valid aircraft.
2. The aircraft must reside in a valid zone having a state value.
3. The aircraft must be within a zone enclosing destination airport.
4. The zone enclosing destination airport occupied by the aircraft must be within a controlled Sector.

Chapter # 5

Analyzing ATC Model

5. ANALYZING ATC MODEL

Firstly, this chapter introduces the Z/EVES tool-set which has been used for analyzing formal ATC system model. Then the chapter focuses some model exploration techniques offered by Z/EVES tool-set. Finally the results of analyzing ATC model are discussed.

5.1 Z/EVES TOOL-SET

The formal specification written in any formal language may contain potential errors ranging from obvious syntax errors to possible hazardous inconsistencies. The art of writing formal specification never assures the correctness and completeness of the system to be developed unless it is checked and analyzed with a powerful tool, which is capable of identifying even the potential errors in syntax and semantics of a formal specification.

Z/EVES is one of the most powerful tools for analyzing Z specifications. Z/EVES unites the important specification notation i.e Z notation with a leading automated deduction capability. Z/EVES supports the entire Z notation. "The Z/EVES Mathematical Toolkit includes the declaration of all the constants of the Standard Mathematical Toolkit as described by Spivey or the proposed ISO Standard for Z, and presents useful theorems about these constants" [19].

Z/EVES is based on the EVES system, and uses the EVES prover to carry out its proof steps [9]. The Z/EVES prover provides powerful automated support (*e.g.*, conditional rewriting, heuristics, decision procedures) with user commands for directing the prover (*e.g.*, instantiate a specific variable, introduce a lemma, use a function definition) [19]. It can be used for parsing, type checking, domain checking, schema expansion, precondition calculation, refinement proofs, and proving theorems.

Proofs available in Z/EVES work on a predicate known as the goal; each step of the proof transforms the goal into a new goal that is equivalent to the previous one. This transformation continues until a goal is evaluated to true by the tool-set.

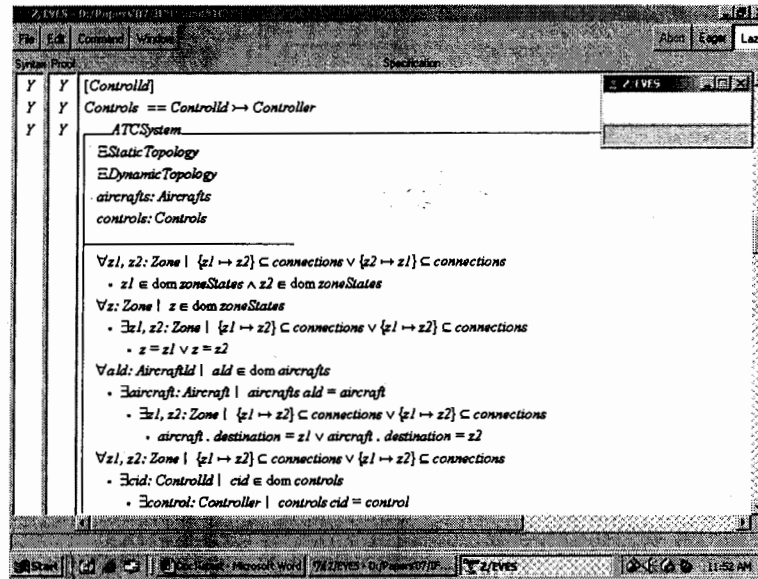


Figure 5.1
Snapshot of Z/EVES Tool-set

Figure 5.1 shows the snapshot of Z/EVES Tool-set. On the left hand side two columns can be viewed; one shows the status of syntax correctness and other shows the status of correctness of proof. The ‘Y’, in left hand side columns, corresponding to shown formal ATC Model schemas, illustrate that schemas are not only syntactically correct but also have correct proof. Therefore, proving the formal specifications using Z/EVES tool-set not only ensures syntactic correctness but also gives proof accuracy.

5.2 MODEL EXPLORATION IN Z/EVES

The remarkable feature of formal specifications, which outclass all other traditional means of informal specification, is that a formal specification can be checked and analyzed for the presence of errors. It allows proving various properties of the system by proving the specifications using various exploration techniques. Since Z/EVES is a powerful tool it supports several means to explore the formal specifications. Some of exploration techniques offered by Z/EVES have been used to analyze our formal ATC model. The following description of these techniques is taken from Meisels et.al [19].

5.2.1 SYNTAX AND TYPE CHECKING

The Z language has quite a complex syntax, and the Mathematical Toolkit contains dozens of functions. It is easy, especially for an inexperienced Z user, to make a mistake. Z/EVES, like most Z tools, detects and reports such type errors. Unlike many other tools, however, Z/EVES can be used incrementally; as each paragraph of a specification is written, it can be immediately checked and, if necessary, corrected.

5.2.2 DOMAIN CHECKING

The Z notation allows one to write expressions that are not meaningful. There are two ways to do so. First, a function can be applied outside its domain, as in $1 \text{ div } 0$, $\max \mathbf{Z}$, or $\#\mathbf{N}$. Second, a definite description (μ -term) is not meaningful if there is not, in fact, a single value satisfying the predicate. Examples are $\mu x : \mathbf{Z} \mid x \neq x$, for which there is no possible value of x satisfying the predicate, and $\mu x : \mathbf{Z} \mid x > 0$, for which there are many possible values. Therefore, Z/EVES examines each paragraph as it is entered, and checks each function application and definite description for meaningfulness.

5.2.3 REDUCTION

The reduction commands traverse the current goal, accumulating assumptions and performing reduction on predicates and expressions in the goal. In a traversal, each formula and subformula of the goal is examined, and may be replaced by a logically equivalent formula that Z/EVES considers "simpler". Other replacements may occur, depending on the type of reduction being performed.

5.2.4 PROVE BY REDUCE

It is one of the techniques for analyzing Z specifications. The 'prove by reduce' performs repeated reducing on the current goal, until reduction has no effect. Therefore, this technique works in iterations.

In each iteration, this command does the following:

1. The current goal is prenexed.
2. The current goal is rearranged.
3. Equality substitution is performed.
4. The current goal is reduced.

5.3 EXPLORATION OF FORMAL ATC MODEL

The formal ATC Model and state operations defined on it are checked and strengthened using Z/EVES Toolset. The results of analyzing formal ATC model are described in table 5.1. The schemas are analyzed with four techniques of Z/EVES tool-set namely syntax and type checking, domain checking, reduction and prove by reduce. While proving the formal ATC system model in Z/EVES, two types of results were obtained. Firstly, some schemas were well written and proved automatically without any prove assistance of the tool (indicated with a ✓ mark in the table 5.1). Secondly, some schemas were proved using the prove assistance of the tool (indicated by ✓* symbol).

Schemas	Syntax & type Checking	Domain Checking	Reduction	Prove by Reduce
StaticTopology	✓	✓	✓	✓
DynamicTopology	✓	✓	✓	✓
Aircraft	✓	✓	✓	✓
Controller	✓	✓	✓	✓
ATCSystem	✓	✓	✓*	✓
TakeOff	✓	✓	✓*	✓
Departure	✓	✓	✓*	✓
HandOff	✓	✓	✓*	✓
Arrival	✓	✓	✓*	✓
Landing	✓	✓	✓*	✓

Table 5.1
Results of Exploration of Formal Model of ATC System

After checking our ATC system model using Z/EVES tool-set, it can be stated that our model is not only syntactically correct but also has correct proof. The assessment criteria, which was defined to evaluate existing models, has been fulfilled by our model. Table 5.2 shows the tabular representation of the comparison of our ATC model with existing models. Three types of symbols can be observed in table 5.2. A ✓ mark shows the model fully meets the assessment criterion, a ✓* symbol indicates that the model has abstract or partial representation of defined assessment criterion and a × symbol represents that the model do not meet the assessment criteria. It is clear from table 5.2 that our ATC system model meets all safety criteria defined by ICAO, therefore, it can be claimed to be safer than other existing models.

Assessment Criteria	Existing ATC system Models
C1. To have clearly defined performance standards, e.g. separation criteria.	M1. David Leadbetter et.al [12] M2. Peter A. Lindsay [45]
C2. Types of aircraft and their performance characteristics.	M3. Dines Bjorner [13] M4. J. C. Bicarregui et.al [22]
C3. Traffic density and distribution	
C4. Airspace complexity, ATC route structure and the classification of airspace.	
C5. Aerodrome layout.	

	M1	M2	M3	M4	Our ATC system Model
C1	✓*	×	×	×	✓
C2	✓*	✓*	✓*	✓*	✓
C3	×	×	×	×	✓
C4	✓*	×	✓	×	✓
C5	✓*	×	✓*	×	✓

Table 5.2
Comparison with existing models

Chapter # 6

Conclusion

6. CONCLUSION

In this chapter the final concluding remarks are given. Finally the portion of work that is left for future is discussed.

6.1 CONCLUDING REMARKS

FM is a promising field in research academia but there is a gap between academia and industry. Many practitioners are reluctant to use FM because of many baseless myths and misconceptions prevailing in market. But FM are very important for rigorous and concrete modeling of system. This has been observed in development of this formal ATC system model in which we resolved the ambiguities and gave the precise, complete and consistent definition of ATC system requirements.

The power of applying FM in modeling of a complex, highly safety critical system of ATC is shown, which was one of the objectives of our research. The Graph Theory, which is considered to be a convenient means of giving solution to various connectivity problems of real world, has been applied for the connectivity of different zones of airspace. Further formal ATC system, modeled as directed graph has been formalized in terms of Z notation and state operations are defined to ensure safety of an aircraft, which shows the strength of mathematics, in terms of FM, to model any complex system very precisely and unambiguously.

By applying FM, a deeper insight of system to be built has been achieved. The errors and inconsistencies those were found while describing formal specification of ATC system have been identified in early phase of development process those would have been detected in implementation or testing phase using Traditional Approaches. Therefore, the use of FM in this research has ensured making high quality, reliable and correct system specifications with respect to ATC system requirements.

Another objective of the research was to apply Z notation for modeling of ATC system because apart from other techniques, the rich mathematical notations offered by Z make it possible to reason rigorously and effectively about the behavior of specified system. Also the wide range of strong tool support offered by Z notation helps to spot out errors more effectively than its other counterparts. Therefore, analysis done using Z/EVES tool-set has given high confidence in our formal ATC system model.

6.2 FUTURE DIRECTIONS

Some interesting and new issues, relevant to this work, were cropped up during this research. Most of them could not be investigated due to time constraints. Some of them were analyzed but did not verify due to the same reason. The results those were not verified are not presented in this thesis and are reported in this section.

The network components of ATC system and the safety properties given at this level must be re-analyzed for implementing to real systems after further refinements. As, initially, we have taken some assumptions to make the model simple and consequently this model has some limitations, which can be relaxed in future work.

The ATC system is highly interactive system interacting with various other third-party components. ATC system also provides updated weather information to airports around the country, so aircraft can take off and land safely. For example, ATC uses various aircraft navigation and communication systems that use computers, radar, radios, and other instruments and devices to provide guidance to flying aircraft. But interfacing to humans and other components is outside the domain of our ATC system Model. Similarly, the communication between pilot and air traffic controller is also a tenet of ATC system but it is outside the scope of our ATC system Model. Therefore to implement our ATC system in real world the issues of interaction must be added to it

For safety in ATC system, Controller has also the job of Conflict Resolution but it is in itself a whole research area so we are not focusing on Conflict Resolution.

In an ATC system two types of flights are possible; Visual Flight Rule (VFR) and Instrument Flight Rule (IFR). VFR flight is restricted to altitudes below 18,000 MSL and does not require flight clearances from ATC. Whereas, IFR requires pilots have to be trained and certified in navigational methodologies. They must adhere to ATC clearances containing specific flight route and altitude directions. The main focus of our research was on IFR flights. However, the VFR flight provision exists in real world. Therefore to implement our ATC model the provision for VFR flights must be catered too.

Similarly refinements for airspace classes can be included; right now a Zone abstractly represents all of them. But to deal with the real world requirements in terms of various classes of airspace the model can be refined. Moreover the structure of zone is at abstract level of representation. In real world there are several tiers within a zone. It means that more than one aircraft can occupy a zone provided there is safe separation between all aircrafts. In our model it is assumed that only one aircraft can occupy a zone at a time. Therefore by inducing the concepts of tiers, efficiency of ATC system can be highly improved.

Finally, our model is based on graph theory, the various advanced concepts of Graphs like single source shortest path, all pair shortest paths, minimum spanning tree and flow networks can also be formalized to make the existing directed graph model of ATC system more efficient and reliable.

Moreover we have just focused the aspects of safety to evaluate some of the existing models of ATC system. The other two measures of dependability i.e reliability and availability are also vital for dependability of any safety critical system. Therefore the aspects of reliability and availability can also be focused for future work.

REFERENCES

- [1] Anthony Hall and David Isaac, "Formal Methods in a Real Air Traffic Control Project", *Software in Air Traffic Control Systems*, pp.1-4, 22 June 1992.
- [2] Anthony Hall, "Seven Myths of Formal Methods", *IEEE Software*, Vol.7 No.5, pp.104-103, September 1990.
- [3] Asaf Degani and Michael Heymann, "Pilot-Autopilot Interaction: A Formal Perspective", *Eighth International Conference on Human-Computer Interaction in Aeronautics*, Toulouse, France, 2000.
- [4] C. B. Jones, "Systematic Software Development Using VDM", Englewood Cliffs, NJ, Prentice-Hall, 1986.
- [5] Cliff B. Jones, Daniel Jackson and Jeannette Wing, "Lightweight Formal Methods", 0018-9162, Vol.29, Issue.4, pp.20-22, IEEE Computer Society Press, Los Alamitos, CA, USA, 1996.
- [6] Constance Heitmeyer, "On the Need for Practical Formal Methods", *Lecture Notes in Computer Science*, Vol.1486, pp.18-26, 1998.
- [7] Craig C. Freudenrich, "How Air Traffic Control Works", Website of How Stuff Works, <http://travel.howstuffworks.com/air-traffic-control.htm>, 2007.
- [8] D. Harel, "Statecharts: A Visual Formalism for Complex Systems", *Series of Computer Programming*, Vol.8, pp.231-274, 1987.
- [9] Dan Craigen, "Formal Methods Adoption: What's working, What's not!", *SPIN*, pp.77-91, 1999.
- [10] Daniel Jackson, "Dependable Software by Design", *Information Technology*, June 2006.
- [11] Daniel M. Berry, "Formal Methods: The Very Idea Some Thoughts About Why They Work When They Work", *Science of Computer Programming*, Vol.42, No.1, pp.11-27, January 2002.
- [12] David Leadbetter, Peter Lindsay, Andrew Neal, and Mike Humphreys, "Integrating the Operator into Formal Models in the Air-Traffic Control Domain", Technical report 00-34, November 2000.

- [13] Dines Bjorner, "Software Systems Engineering From Domain Analysis via Requirements Capture to Software Architecture", Proceedings of Software Engineering Conference, Brisbane, Australia, 1995.
- [14] Edmund M. Clarke and Jeannette M. Wing, "Formal Methods: State of the Art and Future Directions", ACM Computing Surveys, Vol.28, No.4, pp.626-643, 1996.
- [15] Emanuele Ciapessoni, Alberto Coen-Portisini, Ermani Crivelli, Dino Mandrioli, Piergiorgio Mirandola, Angelo Morzenti, "From Formal Models to Formally-Based Methods: An Industrial Experience", TOSEM, Vol.8, No.1, pp.79-113, January 1999.
- [16] Federal Aviation Administration (FAA) Publications, Aeronautical Information Manual, Official Guide to Basic Flight Information and ATC Procedures, February 16, 2006.
- [17] Federal Aviation Administration (FAA) Publications, Procedures for handling Airspace Matters, February 16, 2006.
- [18] Formal Methods Europe (FME) official website, http://www.di.uminho.pt/FME-SoE/fmesoe_all_in_one/fmesoe.htm
- [19] I. Meisels, and M. Saaltink, "The Z/EVES Reference Manual", TR-97-5493-03, ORA Canada, 1997.
- [20] International Civil Aviation Organization (ICAO) Publications, Safety Management Manual, 1st Edition, Doc 9859 AN/460, ICAO, 2006.
- [21] International Civil Aviation Organization (ICAO) Publications, Strategic Objectives of ICAO for 2005-2010, December 2004.
- [22] J. C. Bicarregui, J. S. Fitzgerald, P. A. Lindsay, R. Moore, and B. Ritchie, "Proof in VDM: A Practitioner's Guide", Springer-Verlag New York, New York, USA, 1994.
- [23] J. M. Wing, "A Specifier's Introduction to Formal Methods", IEEE Computer, Vol.23, No.9, pp.8-24, 1990.
- [24] J. M. Spivey, "The Z Notation: A Reference Manual", Englewood Cliffs, NJ, Prentice-Hall, 1992.
- [25] J. V. Guttag, J. J. Horning and J. M. Wing, "The Larch Family of Specification Languages", IEEE Software, Vol.2, No.5, pp.24-36, 1985.
- [26] Jim Woodcock and Jim Davies, "Using Z Specification, Refinement, and Proof", 0-13-948472-8, Prentice-Hall Inc., Upper Saddle River, NJ, USA, 1996.

- [27] Jon Whittle, Jyoti Saboo and Richard Kwan, "From Scenarios to Code: An Air Traffic Control Case Study", Proceedings of the 25th International Conference on Software Engineering, pp.490-495, Portland, Oregon, 2003.
- [28] Jonathan Bowen and Victoria Stavridou, "Safety-Critical Systems, Formal Methods and Standards", IEE/BCS Software Engineering Journal, Vol.8, No.4, pp.189-209, 1993.
- [29] Jonathan Bowen, "Formal Methods in Safety-Critical Standards", Proceedings of Software Engineering Standards Symposium (SESS'93), Brighton, UK, 1993.
- [30] Jonathan Bowen, "Virtual Library of Formal Methods", Center for Applied Formal Methods, Museophile Limited, UK.
- [31] Jonathan P. Bowen and Michael G. Hinchey, "The Use of Industrial-Strength Formal Methods", Proceedings of 21st International Computer Software & Application Conference (COMPSAC'97), pp.332-337, August 1997.
- [32] Jonathan P. Bowen, "Ten Commandments of Formal Methods", IEEE Computer, Vol.28, No.4, pp.56-63, 1995.
- [33] Judy Bowen, "Celebrity Death Match Formal Methods vs. User-Centred Design", Proceedings of Computing Women's Congress, Student Papers, Hamilton, February 2006.
- [34] Len Bass, Paul Clements and Rick Kazman, "Software Architecture in Practice", 2nd Edition, 978-0-321-15495-8, Pearson Education Asia, India, 2003.
- [35] Marc Zimmerman, Mario Rodriguez, Benjamin Ingram, Masafumi Katahira, Maxime de Villepin and Nancy Leveson, "Making Formal Methods Practical", Proceedings of the 19th Digital Avionics Systems Conference, October 2000.
- [36] Maryam Jamal and Nazir Ahmad Zafar, "Formal Model of Computer-Based Air Traffic Control System Using Z Notation", Proceedings of 17th International Conference on Computer Theory and Applications, Alexandria, Egypt, 1st – 3rd September 2007.
- [37] Maryam Jamal and Nazir Ahmad Zafar, "Integration of Graph Theory and Z Notation for Modeling of Air Traffic Control System", Proceedings of 8th International Pure Mathematics Conference, Islamabad, Pakistan, 24th – 26th August 2007.
- [38] Maryam Jamal and Nazir Ahmad Zafar, "Modeling and Formal Specification of Air Traffic Control System Using Z Notation", 4th National Research Conference on Emerging Sciences, Islamabad, Pakistan, 12th January 2007.

- [39] Maryam Jamal and Nazir Ahmad Zafar, "Requirements Analysis of Air Traffic Control System Using Formal Methods", Proceedings of IEEE International Conference on Information and Emerging Technologies, IEEE Catalogue No. 07EX1795, pp 216-222, Karachi, Pakistan, 6th – 7th July 2007.
- [40] Merlin Dorfman and Richard H. Thayer, "A Review of Formal Methods", Computer Society Press, 1996.
- [41] Milica Barjaktarovic, "The State-of-the-Art in Formal Methods", AFOSR Summer Research technical report for Rome Research Site, Formal Methods Framework-Monthly Status Report, F30602-99-C-0166, WetStone Technologies, January 1998.
- [42] N. A. Zafar, "Formal Model for Moving Block Railway Interlocking System Based on Un-Directed Topology", ICET06, pp.217-223, Peshawar, 2006.
- [43] N.A. Zafar and K. Araki, "Formalizing Moving Block Railway Interlocking System for Directed Network", Research Reports, Department of Computer Science and Communication Engineering, Kyushu University, Japan, 2003.
- [44] Navigation An Art and a Science, Aviation Navigation Tutorial.
- [45] Peter A. Lindsay, "A Tutorial Introduction to Formal Methods", Proceedings of 3rd Australian Workshop on Industrial Experience with Safety Critical Systems and Software, pp.29-37, Australian Computer Society, Australia, 1998.
- [46] Peter Gorm Larsen, John Fitzgerald and Tom Brookes, "Lessons Learned from Applying Formal Specification in Industry", IEEE Software, May 1996.
- [47] R. E. Fields, P.C. Wright and P. Marti, "Air Traffic Control as a Distributed Cognitive System: a study of external representations", Proceedings of Ninth European Conference on Cognitive Ergonomics (ECCE9), 1998.
- [48] R. W. Butler, "What is Formal Methods?", NASA LaRC Formal Methods Program, August 2001.
- [49] Rachelle L. Ennis and Yiyuan J. Zhao, "A Formal Approach to the Analysis of Aircraft Protected Zone", Air Traffic Control, Vol.12 No.1, pp.75-102, 2004.
- [50] Richard C. Baxter, Julian Reitman and Donald Ingerman, "Applying Simulation Techniques to An Air Traffic Control Study", Proceedings of the fourth annual conference on Applications of simulation, pp.39-44, New York, United States, 1970.
- [51] Ricky Butler, Victor Carreno, Gilles Dowek, and Cesar Munoz, "Formal Verification of Conflict Detection Algorithms", Lecture Notes in Computer Science, Vol.2144, pp.403-417, Livingston, Scotland, UK, 2001.

- [52] Roger S. Pressman, *Software Engineering A Practitioner's Approach*, Fifth Edition, 978-0072496680, McGraw-Hill, 2001.
- [53] Scott Johnson, Kristopher Zarns, Ritu Banerjee, Robert Ellingson, Travis Dazell, Ryan Langseth, and Tyler Mathwich, "A Study in the Analysis, Design and Implementation of an Air Traffic Control Simulation System Using UML", 38th Annual Midwest Instruction and Computing Symposium, Eau Claire, Wisconsin, 2005.
- [54] Seymour Lipschutz, "Schaum's Outline of Theory and Problems of Data Structures", McGraw-Hill Book Company, Singapore, 1998.
- [55] Steve Easterbrook, Robyn Lutz, Richard Covington, John Kelly, Yoko Ampo, and David Hamilton, "Experiences Using Lightweight Formal Methods for Requirements Modeling", *IEEE Transactions on Software Engineering*, Vol.24, No.1, pp.4-14, January 1998.
- [56] T. L. McCluskey, J. M. Porteous, Y. Naik, C. N. Taylor and S. Jones, "A Requirements Capture Method and its use in an Air Traffic Control Application", *Software-Practice and Experience*, Vol.25, No.1, pp.47-71, 1995.
- [57] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, "Introduction to Algorithms", 2nd Edition, 0-07-013151-1, The MIT Press, Massachusetts Institute of Technology, Cambridge USA, 2001.
- [58] VATSIM Europe Division - Training Department, *Air Traffic Controller Manual*.
- [59] Victor Carreno and Cesar Munoz, "Safety Verification of the Small Aircraft Transportation System Concept of Operations", *Proceedings of the 5th AIAA Aviation Technology Integration and Operations Conference*, Arlington, Virginia, September 2005.
- [60] W. Bruyn, R. Jensen, D. Keskar and P. Ward, "An Extended Systems Modeling Language, *Software Engineering Notes*", Vol.13, No.1, pp.58-67, 1988.
- [61] Wikipedia, the free encyclopedia, a registered trademark of Wikimedia Foundation, Inc., <http://en.wikipedia.org>
- [62] William Glover and John Lygeros, "A Stochastic Hybrid Model for Air Traffic Control Simulation", *HSCC*, pp.372-386, 2004.
- [63] Yves Bontemps, Patrick Heymans and Hillel Kugler, "Applying LSCs to the specification of an Air Traffic Control system", *Proceedings of the 2nd International Workshop on Scenarios and State Machines: Models, Algorithms and Tools (SCESM'03)*, 2003.