International Journal of Innovative
Computing, Information and Control
Volume **10**, Number **4**, August **2014**

 ISSN 1349-4198
pp. 1401–1412

FORMAL MODEL OF AIRCRAFTS SAFETY SEPARATION

Nazir Ahmad Zafar

Department of Computer Science
King Faisal University
Hofuf, Saudi Arabia
nazafar@kfu.edu.sa

Received July 2012; revised November 2013

Abstract. *Modeling of an Air Traffic Control (ATC) has become a challenging problem due to its inherent complexity, introduction of new technologies and expansion of airways in the airspace. Minimum separation distance is an important part of defining flying rules in an ATC system. In this paper, formal analysis of safety properties of air cross and safe distance is provided using Z notation to keep, at least, minimum separation between any two aircrafts preventing collision in a controlled air space. Initially, we have presented a network model of airspace for traffic flow management then on-board system and ground-based controls are defined. For safety analysis, we have supposed that existence of two aircrafts in a smallest unit of airspace is a collision. The issue of air crossing approaching two aircrafts to the same point is also addressed. Graph theory is specified beneath the static part of the system then specification is transformed to Z notation for developing a rigorous model. Based on these definitions abstract safety properties are defined by introducing a notion of protected area of an aircraft. Further, the safety properties are analyzed and extended by introduction of computer based air traffic controls. The formal specification is analyzed using Z/Eves tool.*
Keywords: Air traffic control, Modeling, Safety properties, Z-Specification

1. **Introduction.** Air traffic control (ATC) system is a safety critical system because its failure may cause a huge loss. Ensuring safety of ATC has become a crucial issue due to increase of air traffic and introduction of new technologies [13]. There are various air traffic controllers responsible for monitoring the aircrafts from taking off to landing where their primary objective is to provide a safe, orderly and expeditious flow of air traffic [26]. Because of a large increase in capacity [32], next generation ATC systems are suggested to improve efficiency and achieve a required level of safety [8, 9, 10, 11, 12]. Although an automated support to ATC system is available but still it is heavily dependent upon human interaction causing accidents due to failure of communication and decision making [15, 30]. However, modeling and development of ATC system enabling aircraft to fly freely is an open issue because of its complexity [6].

In most of the existing work, safety criteria are developed by testing through simulation but unfortunately this approach is lacking in verifying the correctness of such systems. For example, the number of simulations increases exponentially to provide a required level of confidence due to their complexity. In addition, when a modification to the system is needed, regression testing will be required to perform which suggests that the complete set of simulations must be re-conducted to ensure that the change did not compromise with its safety and reliability. Therefore, it has become indispensable to apply advanced techniques, for example, formal approaches for safety verification of ATC system which has motivated us for embarking research in this direction.

In this paper, air cross and safe distance are selected to keep minimum separation between any two aircrafts in a controlled air space. The distance by which an aircraft avoids other aircrafts preventing collision is termed as a minimum separation distance, which is an important part of defining flying rules. The most important part of our contribution is formal description of the safety properties preventing collision of aircrafts. Initially, formal specification of network state space is described by graph theory. Then computer based controls and on-board control system are formalized. Next, safety properties are described for preventing collision of aircrafts in airspace. Formal specification is described using Z notation and analyzed by Z/Eves tool. Rest of the paper is organized as follows. In Section 2, most relevant work is discussed. In Section 3, an introduction to air traffic control system is presented. Formal analysis of ATC system's components and safety criteria is described in Section 4. Finally, conclusion and future work are given in Section 5.

2. **Related Work.** Formal analysis of ATC is proposed by Cerone et al. [3] in which an approach is proposed to identify the human errors based on patterns of recurring behavior. Jamal and Zafar have described formal model of ATC system using Z notation focussing on taking off and landing procedures [19, 20]. The PRISM tool is used for symbolic model checking for probabilistic timed automata to verify and analyze the properties of ATC system [25]. Artificial intelligence in terms of multi-agent system has been applied successfully to air traffic flow management (ATFM) [27]. The objective of the research is to show the applicability of agent based simulation in this area. NASA has developed collaborative air traffic flow management using multi-agent simulation, several simple strategies were used to select routes [34]. A fusion of intelligent computing methods is studied to solve the ATFM problem though the development of a new tactical system using advantages of the meta-level control approach [1, 5]. The applications of intelligent computing models for ATFM is presented in [33] in which the software agents are used to develop computational grid platform for congestions identification, conflicts resolution and agreements negotiation among the participating airports. Due to increasing air traffic density and relatively limited number of airways, the future solution for optimal airspace and safe air traffic control is proposed in [2]. A protocol-based multiple aircraft conflict resolution for a finite information horizon is proposed in [16, 17, 18] in which the communication range of an aircraft is finite. An effort is done on finding a predictable system to achieve free flight to choose an optimal path minimizing delay time rather than following pre-defined flight schedules in [21, 28, 35]. The performance of conflict detection and resolution on estimation of aircraft state is presented in [24].

3. **Air Traffic Control System.** Safety and efficiency are two core requirements in safe and normal operation of the system. Efficiency demands that aircrafts must be moved to and from the runways as expeditiously as possible and the time difference in flow of traffic must be minimum. On the other hand, safety requires a well-defined sequence of rules and patterns to prevent conflicts. To improve safety and efficiency, the next generation automated ATC systems are introduced [7]. For this purpose, ground based and automated airspace controllers are used generating efficient and conflict free traffic state space. The ground based controllers are for keeping separation distance between aircrafts to ensure the safety maintained in case of any failure in the air traffic controller or in on-board systems. In the new generation of air controls, the radar equipment requires the aircraft to provide the information related to its identity, altitude, speed and position. It is also possible to provide flight plans in advance after computer analysis that

would help to resolve conflicts in terminal areas in advance expediting the traffic flow and permitting direct routing from origin to destination saving fuel and time.

In problem analysis, the airspace is divided into different regions and sectors controlled by an ATC system. Further, the sectors are sub-divided into blocks in our model where block is a smallest unit of airspace used to define safety. Each ATC is responsible for the safe and expeditious flow of air traffic in a sector both horizontally and vertically. There are various type of controllers monitoring air traffic from departure to destination point. In this paper, a part of the system is described focussing on software component of en-route air traffic control system preventing collision. The en-route traffic controllers are linked with on-board systems preventing mid-air collisions of air traffic. A mid-air collision is defined as an incident associated with an aircraft in which a possibility of collision occurs as a result of violating the minimum separation with another aircraft.

4. **Formal Description.**

Formal description of safety properties is presented here. At first, an introduction to formal methods is provided. Few definitions used in the model are given. Certain assumptions, limitations and boundaries of the system are defined.

4.1. **Formal methods.**

Formal methods include specification techniques, modeling languages and formal verification procedures [22, 23, 36, 37, 38]. Model checking and theorem proving are two effective techniques for ensuring correctness of a program and involve a formulation of properties which can be verified using a suitable logic [25]. The verification algorithms used in model checking involve exploring the set of possible reachable states of a model to ensure the correctness of the formula [14]. For complex programs involving huge data such as trees and recursive definitions, model checking causes a state space explosion [4]. Theorem provers are used to prove program properties based on variations of Hoare logic and it does not need to exhaustively visit the program's state space. Consequently, a theorem proving approach can reason about infinite state spaces involving complex data types and recursion [29]. The Z notation is a model oriented approach, with theorem proving facility used for specification of abstract properties unlike a detailed description language [31]. The Z is used for specification by decomposing the system into its components and defining constraints over it. Z has allowed us to divide complex specification into smaller, more manageable and understandable parts using schemas. The Z/Eves is a powerful tool used to analyze the system specification.

4.2. **Safety model.**

Airspace, aircraft, on-board system and ground controls are major components of ATC system. The airspace is first modeled using graph relation. A smallest airspace unit named block which is a sub-division of a sector is denoted by a node and connectivity of two blocks is represented by an edge. An edge (u, v) in the graph means that an aircraft can move from sectors u to v. After representing the network by graph relation, it is transformed to Z notation. A block in airspace is represented by *Block* and connectivity of two blocks is supposed to be an edge of the graph. The set of edges of the graph is supposed as a description of the airspace denoted by *Links*. In the *Links* relation it is supposed that a block, being small enough, is not connected to itself.

$$[Block];\ Links == \{x, y : Block \mid x \neq y \bullet (x, y)\}$$

The state of a block is described by the schema *Blockinfo* consisting of variables *state* and *total*. The *state* has further two values, that is, clear and occupied. The *total* variable is used to represent total number of aircrafts in a block.

Blockinfo

$state : State$
$total : N$

$State ::= CLEAR \mid OCCUPIED$

An air crossing is specified by *Aircross* schema which is constituted by four variables, that is, *crossing* identifier, *airway1* as one airway, *airway2* as the other airway and *state* representing state of the crossing. It is noted that each airway of the air crossing is assumed as a set of connected blocks. In the predicate part of the schema, the invariants of air crossing are defined as: (i) The blocks identifying both airways of air crossing are different. (ii-v) For every ordered pair (a1, a2) of an airway, either a1 or a2 is the first block of the identifier and has no relation with the other block of the identifier. (vi) If an ordered pair (a1, a2) is in any airway of an air crossing, then (a2, a1) is also in the same airway.

Aircross

$crossing : Block \times Block$
$airway1, airway2 : \mathbb{P}\, Links$
$state : State$

$crossing.1 \neq crossing.2$
$\forall\, a1, a2 : Block \mid (a1, a2) \in airway1 \bullet$
$(crossing.1 = a1 \lor crossing.1 = a2) \land (crossing.2 \neq a1 \land crossing.2 \neq a2)$
$\forall\, a1, a2 : Block \mid (a1, a2) \in airway2 \bullet$
$(crossing.2 = a1 \lor crossing.2 = a2) \land (crossing.1 \neq a1 \land crossing.1 \neq a2)$
$\forall\, a1, a2 : Block \mid (a1, a2) \in airway2 \bullet$
$(crossing.1 = a1 \lor crossing.1 = a2) \land (crossing.2 \neq a1 \land crossing.2 \neq a2)$
$\forall\, a1, a2 : Block \mid (a1, a2) \in airway1 \bullet$
$(crossing.2 = a1 \lor crossing.2 = a2) \land (crossing.1 \neq a1 \land crossing.1 \neq a2)$
$\forall\, a1, a2 : Block \bullet$
$(a1, a2) \in airway1 \cup airway2 \Rightarrow (a2, a1) \in airway1 \cup airway2$

For safe operation of aircrafts, there is always a horizontal and vertical distance between any two aircrafts. Because our objective is to present an abstract model therefore we do not consider such details. In the specification, state space of airspace is described by a schema *Airspace* consisting of six components, *zones*, *topology*, *undirected*, *directed*, *astates* and *crossings*. The *zones* is defined as a power set of *Zone* and *topology* is used to describe a controlled part of airspace. The *undirected* and *directed* is a division of topology used for both and one way flow of air traffic. The *astates* is a mapping from *Block* to *Blockinfo* defining state of a block. Finally, *crossings* is a power set of *Aircross* defining set of crossings in the airspace which may not be fixed.

Invariants: (i) The union of unidirectional and bidirectional airways is equal to the topology. (ii) The unidirectional and bidirectional topologies are disjoint. (iii) In unidirectional topology if an aircraft can move from block a1 to a2, then it is not allowed to move from a2 to a1. (iv) In bidirectional topology if an aircraft can move from a1 to a2, then it can move from a2 to a1. (v) For a block in the topology there exists a zone containing it. (vi) Both airways of an air cross are in the topology. (vii) Identifiers of two different crossings are not connected. (viii) For two different air crossing identifiers, the air crossings are disjoint. (ix) The domain of *astates* mapping is contained in the

topology. (x) If the identifiers of two zones are different their areas are disjoint.

$$Zone == \{x, y : Block \mid x \neq y \bullet (x, y)\}$$

Airspace

$$zones : \mathbb{P}\, Zone;\ topology, undirected, directed : \mathbb{P}\, Links$$
$$astates : Block \pfun Blockinfo;\ crossings : \mathbb{P}\, Aircross$$

$$topology = directed \cup undirected \land directed \cap undirected = \{\}$$
$$\forall\, a1, a2 : Block \mid (a1, a2) \in topology \bullet$$
$$(a1, a2) \in directed \Rightarrow (a2, a1) \notin directed$$
$$\forall\, a1, a2 : Block \mid (a1, a2) \in topology \bullet$$
$$(a1, a2) \in undirected \Rightarrow (a2, a1) \in undirected$$
$$\forall\, a1, a2 : Block \mid (a1, a2) \in topology \bullet \exists\, z : Zone \mid z \in zones \bullet (a1, a2) \in z$$
$$\forall\, x : Aircross \mid x \in crossings \bullet$$
$$\forall\, a1, a2 : Block \bullet (a1, a2) \in x.airway1 \cup airway2 \Rightarrow (a1, a2) \in topology$$
$$\forall\, x1, x2 : Aircross \mid x1 \in crossings \land x2 \in crossings \bullet x1 \neq x2 \Rightarrow$$
$$(x1.crossing.1, x2.crossing.1) \notin topology \land$$
$$(x1.crossing.1, x2.crossing.2) \notin topology \land (x1.crossing.2, x2.crossing.1)$$
$$\notin topology \land (x1.crossing.2, x2.crossing.2) \notin topology$$
$$\forall\, x1, x2 : Aircross \mid x1 \in crossings \land x2 \in crossings \bullet x1 \neq x2 \Rightarrow$$
$$\{x1.crossing.1, x1.crossing.2\} \cap \{x2.crossing.1, x2.crossing.2\} = \{\}$$
$$\forall\, a : Block \mid a \in \mathrm{dom}\, astates \bullet$$
$$\exists\, a1, a2 : Block \mid (a1, a2) \in topology \bullet a1 = a \lor a2 = a$$
$$\forall\, z1, z2 : Zone \mid z1 \in zones \land z2 \in zones \bullet z \neq z2 \Rightarrow \forall\, a1, a2, a3, a4 :$$
$$Block \mid (a1, a2) \in zones \land (a3, a4) \in z2 \bullet \{a1, a2\} \cap \{a3, a4\} = \{\}$$

If aircrafts in airspace are separated by controllers is called controlled airspace. If aircrafts can fly without ATC system is called uncontrolled. In controlled space, every aircraft has a well defined route which consists of blocks connected to each other. A path is defined by a schema *Path* consisting of three components. The first two, *nodes* and *edges* are used to represent airspace and the last one *path* is used to describe path using graph relation. The invariants are listed as: (i) The end points of an edge are nodes in the graph relation. (ii) For a block a in the range of *path* sequence there exists an edge in the graph relation such that one of the endpoints of the edge is block a. (iii) Any two consecutive blocks in the *path* sequence are connected in the relation.

Path

$$nodes : \mathbb{P}\, Block;\ edges : \mathbb{P}\, Links;\ path : \mathrm{seq}\, Block$$

$$\forall\, a1, a2 : Block \mid (a1, a2) \in edges \bullet a1 \in nodes \land a2 \in nodes$$
$$\forall\, a : Block \mid a \in \mathrm{ran}\, path \bullet$$
$$\exists\, a1, a2 : Block \mid (a1, a2) \in edges \bullet a = a1 \lor a = a2$$
$$\forall\, i : N \mid i \geq 1 \land i \leq \#path - 1 \bullet (path(i), path(i + 1)) \in edges$$

An aircraft is specified by a schema *Aircraft*. Length of protected area depends on speed of an aircraft because an aircraft flying at a high speed requires in-front longer safe distance as compared to an aircraft flying at a low speed. The invariants of the *Aircraft* are listed in predicate part of the schema.

Aircraft

$source, destination : Block;\ speedlimit, currentspeed : N$
$minaltitude, maxaltitude, currentaltitude : N;\ protected, route : Path$

$currentspeed \leq speedlimit \land minaltitude \leq currentaltitude$
$currentaltitude \leq maxaltitude \land protected.path \neq \{\} \land route.path \neq \{\}$
$\text{ran}\ protected.path \subseteq \text{ran}\ route.path$
$\forall\ a1, a2 : Block \mid a1 \in \text{ran}\ route.path \land a2 \in \text{ran}\ route.path \bullet a1 \neq a2$

Invariants: (i) The speed of an aircraft does not exceed its limit. (ii) The altitude of an aircraft is within specified limits. (iii) Allocated route and protected airspace are non-empty. (iv) The protected route is contained in the allocated area. (v) Any two consecutive blocks in the route of aircraft are distinct.

The set of aircrafts is denoted by *Aircrafts* consisting *Airspace* and *aircrafts*. The *aircrafts* is a partial function because all of them may not be in the airspace. In predicate part of the schema, it is stated that for any aircraft, all blocks in the route must be in the topology.

$[AircraftId]$

Aircrafts

$Airspace;\ aircrafts : AircraftId \pfun Aircraft$

$\forall\ aid : AircraftId;\ craft : Aircraft \mid (aid, craft) \in aircrafts$
$\bullet\ \forall\ b : Block \mid b \in \text{ran}\ craft.route.path$
$\bullet\ \exists\ b1, b2 : Block \mid (b1, b2) \in topology \bullet b = b1 \lor b = b2$

In this research, only in air guiding system is considered. One ATC system is assumed for each section of the airspace. In Z notation, the control is defined as a schema *Control* which is composed of five components. The first one component is a *section* which defines a part of the airspace controlled by the ATC. The second one is *states* defining state of a block in the section. The third one is an *aircrafts* mapping which is a collection of aircrafts under this control. The *currentcapacity* represents the total number of aircrafts under the control. The *maxcapacity* represents the maximum number of aircrafts allowed under a control. In the predicate part of the schema it is stated that: (i) The state of a block of the section is known to the system. (ii) Every block in the domain of *states* mapping, is a block of section under control. (iii) The current capacity is always less than or equal to maximum capacity of the air traffic control system.

Control

$section : \mathbb{P}\ Path;\ states : Block \pfun State$
$aircrafts : AircraftId \fun Aircraft;\ currentcapacity, maxcapacity : N$

$\forall\ p : Path \mid p \in sections \bullet \forall\ a : Block \mid a \in \text{ran}\ p.path \bullet a \in \text{dom}\ states$
$\forall\ a : Block \mid a \in \text{dom}\ states \bullet \exists\ p : Path \mid p \in sections \bullet a \in \text{ran}\ p.path$
$currentcapacity \leq maxcapacity$

The set en-route controls is denoted by *Controls* consisting of *Airspace* and *controls*. The *controls* is defined as a partial function because we have considered only controls responsible for air traffic. It is stated that for every block there is a control monitoring it and every block is a part of the topology.

$[ControlId];$

Controls

$Airspace;\ controls : ControlId \pfun Control$

$\forall\, a1, a2 : Block \mid (a1, a2) \in topology$
$\bullet\ \exists\, cid : ControlId;\ c : Control \mid (cid, c) \in controls$
$\bullet\ \exists\, p : Path \mid p \in c.sections \bullet a1 \in \operatorname{ran} p.path \land a2 \in \operatorname{ran} p.path$
$\forall\, cid : ControlId;\ c : Control \mid (cid, c) \in controls$
$\bullet\ \forall\, p : Path \mid p \in c.sections \bullet \forall\, a : Block \mid a \in \operatorname{ran} p.path$
$\bullet\ \exists\, a1, a2 : Block \mid (a1, a2) \in topology \bullet a = a1 \lor a = a2$

The ATC system is denoted by *ATCS* which consists of *Aircrafts* and *Controls*. The *Controls* and *Aircrafts* also include *Airspace*. It is not a good modeling practice that schema *Airspace* is included in both *Aircrafts* and *Controls* but it can not be avoided because controls are the main drivers of aircrafts in airspace. Since aircrafts cannot fly without ground based controls and hence inclusion of *Airspace* in both schemas is required. In this way, the schema *ATCS* contains aircrafts, controls and air space. Relationship between these components is defined in the the predicate part of the schema as: (i) Every aircraft in a control is in the *Aircrafts*. (ii) Every aircraft in the *Aircrafts* is under some control.

ATCS

$Aircrafts;\ Controls$

$\forall\, cid : ControlId;\ control : Control \mid (cid, control) \in controls$
$\bullet\ \forall\, aid : AircraftId;\ craft : Aircraft \mid (aid, craft) \in control.aircrafts$
$\bullet\ (aid, craft) \in aircrafts$
$\forall\, aid : AircraftId;\ craft : Aircraft \mid (aid, craft) \in aircrafts$
$\bullet\ \exists\, cid : ControlId;\ control : Control \mid (cid, control) \in controls$
$\bullet\ (aid, craft) \in control.aircrafts$

4.3. **Formal analysis of safety properties.** In our model, it is assumed that existence of two aircrafts in a block is a collision. Similarly existence of two aircrafts at an air crossing, one aircraft at each airway, is also a collision. If two aircrafts are moving towards air crossing at the same altitude then collision is prevented by forcing one aircraft to change the altitude by climbing up or moving down. Based on these definitions, the abstract safety properties for preventing collision are stated as:

1. There must be, at most, one aircraft in one block to avoid collision between aircrafts in the block.
2. There must be, at most, one aircraft at an air crossing to avoid collision between aircrafts at the air crossing.

At this level of specification, safety properties are defined based on the definition of *Airspace* because state of each block is defined there. The specification of the safe ATC system is denoted by *SATC* to prevent collision in a block and at an air crossing. In the schema it is specified that a block in air space can be occupied by only one aircraft at a time. Air crossing is an intersection of two air blocks. If there are two aircrafts at air crossing, one aircraft at each airway, then the above property will be satisfied but there is a possibility of collision at the air crossing. Consequently, safety property at air crossing is necessary for preventing collision. Collision at air crossing can be avoided if only one airway of the crossing can be occupied at a time. In the schema, it is specified that sum of the number of aircrafts at both the airways of an air crossing should not be greater than one. In the predicate part of the schema, it is stated that there can be at most one

aircraft in a block. If block is occupied then its state must be *OCCUPIED*. Further, the sum of the aircrafts in both blocks of the air crossing can not be more than one. If an aircraft is in any of the block of the air crossing then its state is occupied. The other approaching aircraft will climb up or move down to remove the conflict avoiding collision.

SATC

Aircrafts

$$
\begin{aligned}
&\forall\, b : Block;\ bi : Blockinfo \mid (b, bi) \in astates \\
&\bullet\ bi.total \leq 1 \land bi.state = OCCUPIED \\
&\forall\, x : Aircross \mid x \in crossings \bullet \forall\, bi1, bi2 : Blockinfo \\
&\mid (x.crossing.1, bi1) \in astates \land (x.crossing.2, bi2) \in astates \\
&\bullet\ bi1.total + bi2.total \leq 1 \land \\
&(bi1.state = OCCUPIED \Rightarrow bi2.state = CLEAR \land \\
&bi2.state = OCCUPIED \Rightarrow bi1.state = CLEAR)
\end{aligned}
$$

Preventing collisions is referred to as separation, that is, to prevent aircrafts from coming too close to each other. In fact, prevention of collision is not guaranteed by satisfaction of the above properties. This is because existence of one aircraft in one block does not give guarantee to avoid collision. On the other hand, the abstract safety properties are necessary as a foundation to develop the complete and consistent safety model which can be applied to a real ATC system. The properties are redefined by applying to the notion of safe area in front of aircrafts.

1. The intersection of safe areas of two different aircrafts is always empty.
2. An air crossing may be contained in the safe area of only one aircraft preventing collision at the air crossing.

The redefined safety system is denoted by the schema *SATCR*. In the schema, it is described that intersection of safe areas of any two different aircrafts is always empty. Further, it is specified that if a block of an air crossing is in the safe area of an aircraft then it cannot be in the safe area of another.

SATCR

Aircrafts

$$
\begin{aligned}
&\forall\, a1, a2 : AircraftId;\ craft1, craft2 : Aircraft \\
&\mid (a1, craft1) \in aircrafts \land (a2, craft2) \in aircrafts \\
&\bullet\ a1 \neq a2 \Rightarrow craft1.protected.path \cap craft2.protected.path = \{\} \\
&\forall\, a1, a2 : AircraftId;\ craft1, craft2 : Aircraft;\ cross : Aircross \\
&\mid (a1, craft1) \in aircrafts \land (a2, craft2) \in aircrafts \land cross \in crossings \\
&\bullet\ a1 \neq a2 \Rightarrow \\
&(\exists\, b1, b2 : Block \mid (b1, b2) \in cross.airway1 \cup cross.airway2 \\
&\land (b1 \in \operatorname{ran} craft1.protected.path \lor b2 \in \operatorname{ran} craft1.protected.path) \\
&(\forall\, b3, b4 : Block \mid (b3, b4) \in cross.airway1 \cup cross.airway2 \\
&\bullet\ b3 \notin \operatorname{ran} craft2.protected.path \land b4 \in \operatorname{ran} craft2.protected.path)) \\
&\land (\exists\, b1, b2 : Block \mid (b1, b2) \in cross.airway1 \cup cross.airway2 \\
&\land (b1 \in \operatorname{ran} craft2.protected.path \lor b2 \in \operatorname{ran} craft2.protected.path) \\
&(\forall\, b3, b4 : Block \mid (b3, b4) \in cross.airway1 \cup cross.airway2 \\
&\bullet\ b3 \notin \operatorname{ran} craft1.protected.path \land b4 \in \operatorname{ran} craft1.protected.path))
\end{aligned}
$$

Finally, computer based controls are introduced for completing formalization of the properties. The definition of collision is the same as given above but with a difference that, here, aircrafts are under ATC system. In the further analysis, consistency of airspace

is also checked. By consistency we mean that state space of a block must be consistent in the entire system. A concept of protecting a block is introduced as well. The safety properties formalized in the previous section are augmented with this notion.

1. Intersection of protected areas of two different aircrafts under an ATC system is always empty. A block in the protected area of an aircraft must be in occupied state. The state of a block must be consistent in the entire state space of the system.
2. An air crossing can be in protected area of one aircraft under an ATC system. If any block of any airway of an air crossing is occupied then all blocks of the same crossing are occupied.

The schema *ATCFR* is a redefined form of the schema *SATCR* which includes *ATCS*. That is *Airspace*, *Aircrafts* and *Controls* are all included in first part of the schema *ATCFR*. In the *ATCFR* schema, it is specified that for any two different aircrafts under an ATC system, no aircraft can enter into the protected area of another aircraft. Further, it is described that for any two different aircrafts under an ATC system, if an air crossing is in the section under a control and is included in the protected area of one aircraft then it cannot be made available for any other aircraft under any control. The other aircraft has to change its height or speed to avoid the collision at the air crossing.

SATCFR

ATCS

$$
\begin{array}{l}
\forall\, cid : ControlId;\ control : Control \mid (cid, control) \in controls \\
\bullet\ \forall\, a1, a2 : AircraftId;\ craft1, craft2 : Aircraft \\
\mid (a1, craft1) \in control.aircrafts \land (a2, craft2) \in control.aircrafts \\
\bullet\ a1 \neq a2 \Rightarrow craft1.protected.path \cap craft2.protected.path = \{\} \\
\forall\, cid : ControlId;\ control : Control \mid (cid, control) \in controls \\
\bullet\ \forall\, a1, a2 : AircraftId;\ craft1, craft2 : Aircraft;\ cross : Aircross \mid \\
(a1, craft1) \in control.aircrafts \land (a2, craft2) \in control.aircrafts \land \\
cross \in crossings \bullet a1 \neq a2 \Rightarrow \\
(\exists\, b1, b2 : Block \mid (b1, b2) \in cross.airway1 \cup cross.airway2 \\
\land\ (b1 \in \text{ran}\ craft1.protected.path \lor b2 \in \text{ran}\ craft1.protected.path) \\
\bullet\ (\forall\, b3, b4 : Block \mid (b3, b4) \in cross.airway1 \cup cross.airway2 \\
\bullet\ b3 \notin \text{ran}\ craft2.protected.path \land b4 \in \text{ran}\ craft2.protected.path)) \\
\land\ (\exists\, b1, b2 : Block \mid (b1, b2) \in cross.airway1 \cup cross.airway2 \\
\land\ (b1 \in \text{ran}\ craft2.protected.path \lor b2 \in \text{ran}\ craft2.protected.path) \\
\bullet\ (\forall\, b3, b4 : Block \mid (b3, b4) \in cross.airway1 \cup cross.airway2 \\
\bullet\ b3 \notin \text{ran}\ craft1.protected.path \land b4 \in \text{ran}\ craft1.protected.path))
\end{array}
$$

We have supposed that if a block is in the protected area of any aircraft then it must be blocked. To describe formal specification of this property, a new schema *RouteBlocked* is introduced. In the predicate part, it is stated that for any control and for any aircraft under the control if a block is in the protected area of the aircraft then its state must be occupied. Further, for any control and for any aircraft under the control if any of the two different blocks constituting air crossing is in the protected area of the aircraft then states of both the blocks must be occupied. Finally, safety is defined by the schema *SafeATC* which is conjunction of *SATCFR* and *RouteBlocked*.

RouteBlocked

ATCS

$\forall\, cid : ControlId;\ control : Control \mid (cid, control) \in controls$
$\bullet\ \forall\, a : AircraftId;\ craft : Aircraft \mid (a, craft) \in control.aircrafts$
$\bullet\ \forall\, b : Block;\ bi : Blocknfo \mid (b, bi) \in astates$
$\bullet\ b \in \operatorname{ran} craft.route.path \Rightarrow bi.state = OCCUPIED$
$\forall\, cid : ControlId;\ control : Control \mid (cid, control) \in controls$
$\bullet\ \forall\, a : AircraftId;\ craft : Aircraft \mid (a, craft) \in control.aircrafts$
$\bullet\ \exists\, xs : Aircross \mid xs \in crossings$
$\bullet\ \forall\, b1, b2 : Block \mid (b1, b2) \in xs.airway1 \cup xs.airway2$
$\bullet\ b1 \in \operatorname{ran} craft.route.path \lor b2 \in \operatorname{ran} craft.route.path$
$\Rightarrow xs.state = OCCUPIED$

$\text{SafeATC} \mathrel{\widehat{=}} SATCFR \land RouteBlocked$

5. **Conclusion.** In this paper, we have described the safety properties based on the critical components preventing in air collision of aircrafts. Initially, we have formalized the abstract properties which were not real safety properties. That is why we redefined the properties by applying the concept of protected area in front of an aircraft. Further analysis is done by introducing computer based controls to monitor the aircrafts and air state space. Z notation is applied in this concurrent and distributed natured system because of its rigorous and abstract characteristics. We observed that the complexity of the system was reduced by decomposing it into its critical components. The use of schema structure facilitated us in reducing complexity because of re-useability. Development from abstraction to detailed analysis made it easy to purpose a simple and understandable model.

There exists a lot of work on modeling of ATC; however, it needs much more effort to address next generation ATC systems achieving the required level of safety and efficiency. For example, the work of Tran and Hung [13] is close to ours in which the safety verification of ATC system is presented with probabilistic timed automata and analysis is done with PRISM model checker. Two major drawbacks of this approach are observed. Firstly, the probabilities in the model are assumed as artificial. Secondly, the disadvantages of model checking in comparison to theorem proving for verification of complex systems are realized.

In our experience, various benefits describing formal specification using Z notation of the ATC system were observed. For example, modeling each component of the system at hand provided us its complete characterization. Modeling at a high level of abstraction supported us in capturing the intuitive understanding of its behavior. On the other hand, if a system was specified at a detailed level the intuition may have lost. Compositional approach enabled us to give reasoning about the components and subsequently the entire system. After analysis and description of the system, an advantage of formal model can be achieved for further refinement. The detailed level model can be achieved after a series of successive refinements while guaranteeing the transformation of semantics of the previous model. Further, a clear set of assumptions and scope of the system was defined by producing a mathematical model. Finally, we were able to produce an absolute safety system under the stated assumptions. It is mentioned that this formal model can be applied to any ATC system after a further analysis. This is because we have modeled the system and defined the properties based on the requirements of like a real system. We

believe that this experience will be useful to model the other critical systems using the same approach.

Acknowledgment. This work is partially supported by Deanship of Scientific Research, King Faisal University, Saudi Arabia.

REFERENCES

[1] D. P. Alves, W. Li and B. B. Souza, Reinforcement learning to support meta-level control in air traffic management, *Reinforcement Learning: Theory and Applications*, ARS Publishing, pp.357-372, 2008.

[2] A. Cavcar and M. Cavcar, Impact of aircraft performance differences on fuel consumption of aircraft in air traffic management environment, *Aircraft Engineering & Aerospace Technology*, vol.76, no.5, pp.502-515, 2004.

[3] A. Cerone, P. A. Lindsay and S. Connelly, Formal analysis of human-computer interaction using model-checking, *The 3rd IEEE International Conference on Software Engineering and Formal Methods*, pp.352-361, 2005.

[4] E. M. Clarke, O. Grumberg, S. Jha, Y. Lu and H. Veith, Progress on the state explosion problem in model checking, *Informatics, LNCS*, pp.176-194, 2001.

[5] A. Marcio, F. Crespo, C. V. de Aquino, B. B. de Souza, A. Cristina and M. A. de Melo, Distributed decision support system applied to tactical air traffic flow management in the case of CINDACTA I, *Journal of the Brazilian Air Transportation Research Society*, vol.4, no.1, pp.47-60, 2008.

[6] N. E. Debbache, Toward a new organization for air traffic control, *Aircraft Engineering & Aerospace Technology*, vol.73, no.6, pp.561-567, 2001.

[7] H. Erzberger and R. A. Paielli, Concept for next generation air traffic control system, *Air Traffic Control Quarterly*, vol.10, no.4, pp.355-378, 2002.

[8] H. Erzberger, Transforming the NAS: The next generation air traffic control system, *Proc. of the International Congress of the Aeronautical Sciences*, 2004.

[9] H. Erzberger, Automated conflict resolution for air trafficcontrol, *Proc. of the 25th International Congress of the Aeronautical Sciences*, 2006.

[10] H. Erzberger and K. Heere, Algorithm and operational concept for resolving short-range conflicts, *Journal of Aerospace Engineering*, vol.224, pp.225-243, 2009.

[11] T. Farley and H. Erzberger, Fast time air traffic simulation of a conflict resolution algorithm under high air traffic demand, *Proc. of the USA Europe ATM Seminar*, 2007.

[12] Y. Guo, X. Cao and J. Zhang, Constraint handling based multiobjective evolutionary algorithm for aircraft landing scheduling, *International Journal of Innovative Computing, Information and Control*, vol.5, no.8, pp.2229-2238, 2009.

[13] T. T. B. Hanh and D. V. Hung, Verification of an air traffic control system with probabilistic real-time model checking, *UNU-IIST, Report No. 355*, 2007.

[14] G. Holzmann, Trends in software verification, *Proc. of the Formal Methods Europe Conference*, 2003.

[15] J. Hu, M. Prandini and S. Sastry, Optimal maneuver for multiple aircraft conflict resolution: A braid point of view, *Proc. of the 39th IEEE Conference on Decision and Control*, no.4, pp.4164-4169, 2000.

[16] I. Hwang and C. Tomlin, Protocol-based conflict resolution for finite information horizon, *Proc. of the AACC American Control Conference*, Piscataway, NJ, USA, 2002.

[17] I. Hwang, J. Hwang and C. Tomlin, Flight-mode-based aircraft conflict detection using a residual-mean interacting multiple model algorithm, *Proc. of the AIAA Guidance Navigation, and Control Conference*, 2003.

[18] I. Hwang, H. Balakrishnan, K. Roy and C. Tomlin, Target tracking and identity management in clutter for air traffic control, *Proc. of the AACC American Control Conference*, 2004.

[19] M. Jamal and N. A. Zafar, Formal model of computer-based air traffic control system using Z notation, *Proc. of the 17th International Conference on Computer Theory and Applications*, 2007.

[20] M. Jamal and N. A. Zafar, Requirements analysis of air traffic control system using formal methods, *Proc. of IEEE International Conference on Information and Emerging Technologies*, pp.216-222, 2007.

[21] S. Kahne and I. Frolow, Air traffic management: Evolution with technology, *IEEE Control Systems Magazine*, vol.16, no.4, 1996.

[22] S. A. Khan and N. A. Zafar, Promotion of local to global operation in train control system, *Journal of Digital Information Management*, vol.5, no.4, pp.231-236, 2007.

[23] S. A. Khan and N. A. Zafar, Improving moving block railway system using fuzzy multi-agent specification language, *International Journal of Innovative Computing, Information and Control*, vol.7, no.7(B), pp.4517-4533, 2011.
[24] J. K. Kuchar and L. C. Yang, A review of conflict detection and resolution modeling methods, *IEEE Transactions on Intelligent Transportation Systems*, vol.1, no.4, pp.179-189, 2000.
[25] M. Kwiatkowska, G. Norman, J. Sproston and F. Wang, Symbolic model checking for probabilistic timed automata, *Joint Conference on Formal Modelling and Analysis of Timed Systems and Formal Techniques in Real-Time and Fault Tolerant Systems, LNCS*, vol.3253, pp.293-308, 2004.
[26] C. Livadas, J. Lygeros and N. A. Lynch, High level modeling and analysis of the traffic alert and collision avoidance system, *Proc. of The IEEE*, vol.88, no.7, pp.926-948, 2000.
[27] M. Nguyen-Duc, J. P. Briot, A. Drogoul and V. Duong, An application of multi-agent coordination techniques in air traffic management, *Proc. of the IEEE/WIC International Conference on Intelligent Agent Technology*, pp.622-628, 2003.
[28] M. S. Nolan, Fundamentals of air traffic control, *Brooks/Cole*, 3rd Edition, Wadsworth, 1998.
[29] M. Ouimet, Formal software verification: Model checking and theorem proving, *Embedded Systems Laboratory Technical Report ESL-TIK-00214*, MIT Cambridge, 2005.
[30] S. T. Shorrock and B. Kirwan, Development and application of a human error identification tool for air traffic control, *Applied Ergonomics*, vol.33, no.4, pp.319-336, 2002.
[31] J. M. Spivey, *The Z Notation: A Reference Manual*, Prentice-Hall International Series in Computer Science, (UK) Ltd., 1992.
[32] J. Villiers, ERASMUS – A friendly way for breaking the capacity barrier, *ITA*, vol.58, 2004.
[33] L. Weigang, M. V. P. Dib, D. P. Alves and A. M. F. Crespo, Intelligent computing methods in air traffic flow management, *Transportation Research Part C: Emerging Technologies*, vol.18, no.5, pp.781-793, 2010.
[34] S. R. Wolfe, P. A. Jarvis, F. Y. Enomoto and M. Sierhuis, Comparing route selection strategies in collaborative traffic flow management, *IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, Fremont, USA, pp.59-62, 2007.
[35] L. C. Yang and J. K. Kuchar, Prototype conflict alerting system for free flight, *Journal of Guidance, Control, and Dynamics*, vol.20, no.4, 1997.
[36] N. A. Zafar and K. Araki, Formalizing moving block railway interloking system for directed network, *Research Reports on Information Science and Electrical Engineering*, vol.8, no.2, pp.109-114, 2003.
[37] N. A. Zafar, Modeling and formal specification of automated train control system using Z notation, *IEEE Multitopic Conference, INMIC06*, pp.438-443, 2006.
[38] N. A. Zafar, S. A. Khan and K. Araki, Towards the safety properties of moving block railway interlocking system, *International Journal of Innovative Computing, Information and Control*, vol.8, no.8, pp.5677-5690, 2012.