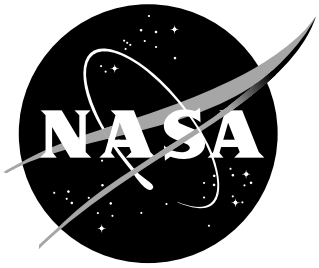
NASA/TM–2004–213006



Abstract Model of the SATS Concept of Operations: Initial Results and Recommendations

Gilles Dowek
Laboratoire d'Informatique, France

César A. Muñoz
National Institute of Aerospace, Hampton, Virginia

Victor A. Carreño
Langley Research Center, Hampton, Virginia

March 2004

The NASA STI Program Office ... in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) Program Office plays a key part in helping NASA maintain this important role.

The NASA STI Program Office is operated by Langley Research Center, the lead center for NASA's scientific and technical information. The NASA STI Program Office provides access to the NASA STI Database, the largest collection of aeronautical and space science STI in the world. The Program Office is also NASA's institutional mechanism for disseminating the results of its research and development activities. These results are published by NASA in the NASA STI Report Series, which includes the following report types:

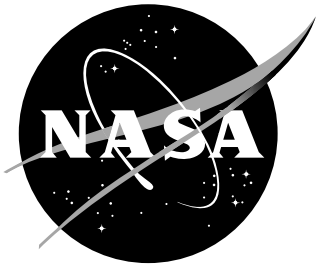
- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers, but having less stringent limitations on manuscript length and extent of graphic presentations.
- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.
- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.
- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.
- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.
- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services that complement the STI Program Office's diverse offerings include creating custom thesauri, building customized databases, organizing and publishing research results ... even providing videos.

For more information about the NASA STI Program Office, see the following:

- Access the NASA STI Program Home Page at ***http://www.sti.nasa.gov***
- E-mail your question via the Internet to help@sti.nasa.gov
- Fax your question to the NASA STI Help Desk at (301) 621–0134
- Phone the NASA STI Help Desk at (301) 621–0390
- Write to:
  NASA STI Help Desk
  NASA Center for AeroSpace Information
  7121 Standard Drive
  Hanover, MD 21076–1320

NASA/TM–2004–213006



Abstract Model of the SATS Concept of Operations: Initial Results and Recommendations

Gilles Dowek
Laboratoire d'Informatique, France

César A. Muñoz
National Institute of Aerospace, Hampton, Virginia

Victor A. Carreño
Langley Research Center, Hampton, Virginia

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23681–2199

March 2004

Available from:

NASA Center for AeroSpace Information (CASI)
7121 Standard Drive
Hanover, MD 21076–1320
(301) 621–0390

National Technical Information Service (NTIS)
5285 Port Royal Road
Springfield, VA 22161–2171
(703) 605–6000

AN ABSTRACT MODEL OF THE SATS CONCEPT OF OPERATIONS*

INITIAL RESULTS AND RECOMMENDATIONS

Gilles Dowek†, César A. Muñoz‡, and Víctor Carreño§

ABSTRACT

An abstract mathematical model of the Concept of Operations for the Small Aircraft Transportation System (SATS) is presented. The Concept of Operations consists of several procedures that describe nominal operations for SATS. Several safety properties of the system are proven using formal techniques. The final goal of the verification effort is to show that under nominal operations, aircraft are safely separated. The abstract model was written and formally verified in the Prototype Verification System (PVS).

1 INTRODUCTION

The *Small Aircraft Transportation System (SATS)* [3] program aims to increase access to small and medium sized airports. The great majority of small and medium airports in the US are underutilized for various reasons including limited use of general aviation aircraft, minimal or no commercial transport services, lack of facilities, etc. Airports lacking radar coverage and control tower facilities rely on procedural separation for access during Instrument Meteorological Conditions (IMC). Procedural separation uses a method of one-in/one-out. That is, only one aircraft is given access to the airport airspace at a given time. This method results in a significant reduction in potential airport throughput.

The objective of the SATS concept is to provide the capability for higher volume operations to these airports during IMC with a minimum of infrastructure and at a low cost. The concept of operation makes use of four main components: a designated airspace surrounding the airport called the Self Controlled Area (SCA); a centralized automated system called the Airport Management Module (AMM); aircraft to aircraft and aircraft to AMM data communication; and distributed, on-board navigation tools.

The concept is a significant departure from typical Instrument Flight Rules (IFR) operations where separation assurance services are provided by Air Traffic Control (ATC). In the SATS concept of operations, pilots accept responsibility for separation inside the SCA. This responsibility is supported by the rules of entry implemented by the AMM, the flight procedures, and the on-board navigation tools.

Aircraft separation is a major safety concern. Showing that the design of the rules and procedures are safe and correct are a top priority. This task shall be accomplished using

*For the first two authors, this work was supported by the National Aeronautics and Space Administration under NASA Cooperative Agreement NCC-1-02043.

†Laboratoire d'Informatique (LIX), École polytechnique, 91128 Palaiseau Cedex, France. Email: Gilles.Dowek@polytechnique.fr, Web: http://www.lix.polytechnique.fr/~dowek.

‡National Institute of Aerospace (NIA), 144 Research Drive, Hampton, VA 23666, USA. Email: munoz@nianet.org, Web: http://research.nianet.org/~munoz.

§NASA Langley Research Center, Hampton, 23666, VA, USA. Email: victor.a.carreno@nasa.gov, Web: http://shemesh.larc.nasa.gov/people/vac.

formal mathematical analysis. A high level mathematical model of the rules and procedures has been created and is described in this paper. The mathematical model is based on the SATS Higher Volume Operations concept of operation document, SATS-HVO-CONOPS [1]. In the abstract high level model, assumptions are made about physical properties and aircraft dynamics. These assumptions will be validated by other mathematical analysis and simulation.

The next section describes the concept of operations. Section 3 presents the abstract model and the transition rules for the model. Section 4 describes the desired and safety properties. Sections 5 and 6 are verification issues and conclusion.

2 CONCEPT OF OPERATIONS

The concept of operations is a collection of rules and procedures which, when followed, will support separation assurance during terminal area operations. The concept is implemented by four main components: the Self Controlled Area (SCA); the Airport Management Module (AMM); data communication; and on-board navigation tools.

2.1 The Self Controlled Area

The SCA is an airspace volume surrounding the airport facility. Figure 1 shows an example of an SCA volume.

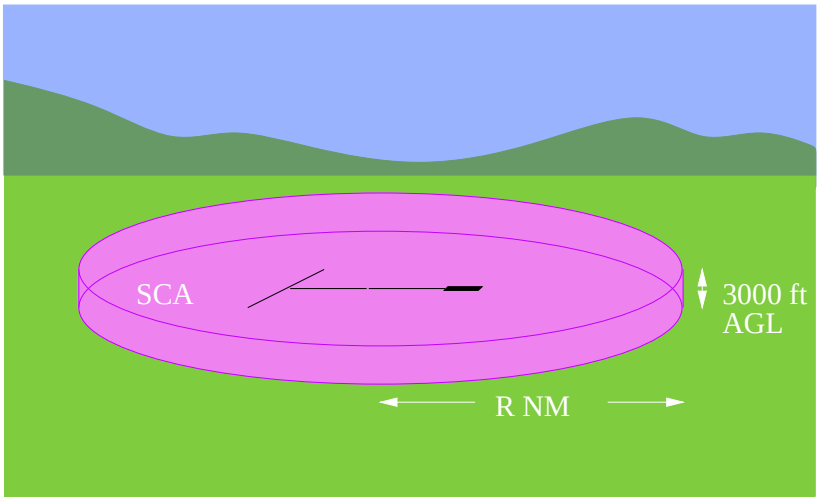


Figure 1: *Self Controlled Area airspace volume*

The design of the SCA is similar to a GPS T approach [2]. Figures 2 and 3 show a top view and side view, respectively, of a nominal SCA design. The top view shows the fixes and segments of the arrival and departure paths. The fixes are the initial arrival fixes (IAFs), intermediate fix (IF), final approach fix (FAF), and departure fixes (DFs). The side view shows the holding altitudes above the initial arrival fixes. The holding fixes at 2000 and 3000 feet AGL (above ground level) are inside the SCA. Holding fixes at 4000 and above are outside the SCA. The holding fixes inside the SCA at 2000 and 3000 also serve as missed approach holding fixes (MAHF). When an aircraft executes a missed approach, it

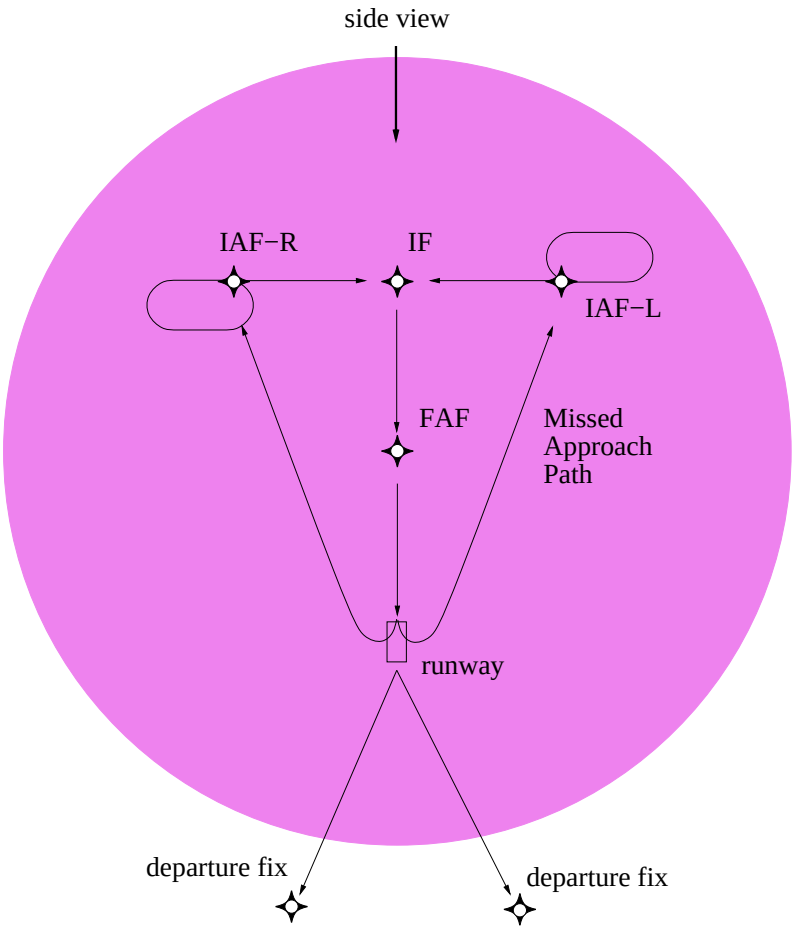


Figure 2: *Top View of SCA*

will proceed to one of the missed approach holding fixes, IAF-R or IAF-L, as specified by the operational concept.

There are two types of entry into the SCA: vertical entry and lateral entry. In a vertical entry, an aircraft flies to the IAF at an altitude above the SCA. The aircraft holds at the IAF above the SCA until entry is granted by the AMM. The aircraft then descends over the IAF flying a race track trajectory through 4000 to the 3000 feet holding fix. The aircraft descends from the 3000 to the 2000 holding when the 2000 altitude becomes available. In a lateral entry, an aircraft flies from the location where the lateral entry is granted to the IAF such that it arrives at the IAF at or above 2000 feet AGL. Once an aircraft is at an IAF at 2000 AGL, there is no distinction between a vertical and lateral entry. From the IAF at 2000, and after some spacing and leading aircraft criteria have been met, the aircraft proceed to the IF, FAF and runway.

Aircraft departing the SCA must request, from Air Traffic Control, clearance to transition from the SATS airspace to a departure fix. Departure fixes are outside the SCA and they are under ATC control. After clearance is given to enter ATC controlled airspace and the aircraft is ready for departure, the arrival stream is monitored for a departure slot. In case

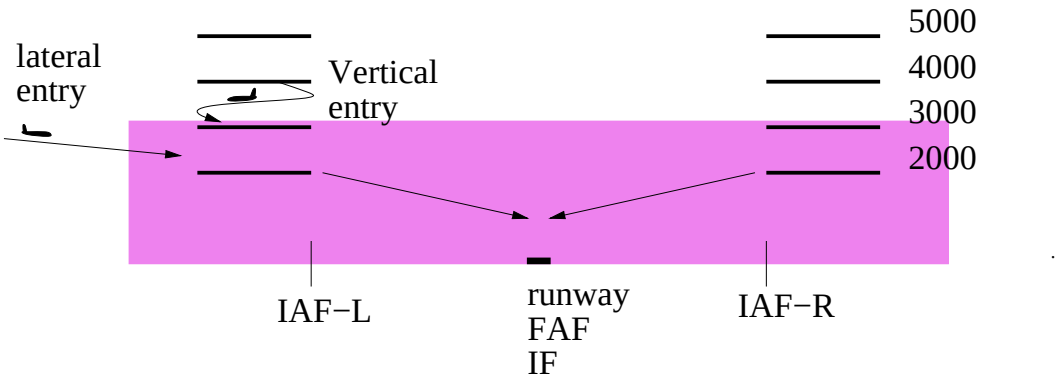


Figure 3: *Side View of SCA*

of multiple departure operations, a separation of 10 nautical miles is required for aircraft flying to the same departure fix. For aircraft flying to opposite departure fixes, a separation of 5 nautical miles can be used. On board navigation tools will assist the pilot in identifying a departure slot.

2.2 The Airport Management Module

The AMM is an automated system which will typically reside at the airport grounds. It serves as an arbiter and sequencer of the SCA. It receives state information from aircraft in the vicinity of the airport and communicates with aircraft via data link. The AMM is not intended to replace traditional air traffic control services. It is only one element of the concept to provide separation assurance which is a traditional air traffic control service.

The AMM implements the entry rules, provides follow notifications and assigns missed approach holding fixes (MAHF). In an informal and abbreviated way, the rules for assigning a leading aircraft, MAHF assignment, and entry are described in the following sections.

Leading aircraft

An aircraft that is granted entry into the SCA follows the aircraft that entered immediately before it. The AMM issues a follow notification (cleared to follow) after granting entry. If there are no aircraft in the SCA, the aircraft is cleared to runway (the leading aircraft is *none*).

Missed approach holding fix

Missed approach holding fixes (MAHFs) are at the IAFs. An aircraft that is granted entry into the SCA is given a MAHF assignment opposite to its leading aircraft. Therefore, aircraft in a stream have alternating MAHF. If there are no aircraft in the SCA, the MAHF given is the same as its entry IAF.

Vertical entry

A vertical entry over an IAF is permitted if there are less than 2 aircraft at that IAF or assigned to that IAF as a MAHF and there are no aircraft on approach assigned to that MAHF and there is no lateral entry in progress to that IAF.

Lateral entry

A lateral entry is permitted to an IAF if there are no aircraft at that IAF or assigned to that IAF as a MAHF.

These rules will be revisited in more detail and rigor in the following sections.

2.3 Navigation tools

There are several rules that pilots must follow inside the SCA and when transitioning to and from the SCA. The on-board navigation tools provide advisories to aid in following these rules. Example of these rules are: a pilot must determine if its leading aircraft is sufficiently ahead to initiate an approach from a holding fix; an aircraft must descend from a holding altitude to the lowest available altitude; a pilot must confirm that the holding altitude at 3000 feet is available before descending from 4000 to 3000 and entering the SCA. On a missed approach, a pilot must climb to the lowest available altitude at the MAHF. These rules will also be discussed in detail in the following sections.

3 ABSTRACT MODEL

The abstract model is a discrete representation of the rules and procedures of the concept of operations. It was written and formally verified in the theorem prover checker PVS (Prototype Verification System) [4]. There are assumptions made in the model about the performance and dynamics of the aircraft. The validation of these assumptions will be covered in future work.

Entry into the SCA is granted in the same order as the request order; the system is first come, first serve. Follow notifications provided by the AMM are encoded in the model by assigning a landing sequence to the aircraft within the SCA: 1 to the *first* aircraft, 2 to the aircraft following 1, and so forth. Landing sequences are not absolute values. When an aircraft lands or initiates a missed approach procedure, landing sequences change such that the aircraft with landing sequence 2 becomes the first aircraft, and the aircraft with landing sequence $n+1$ gets the landing sequence n. Note that landing sequences are artifacts of the model rather than an integral part of the SATS concept. Knowing each aircraft leader is of course sufficient to describe aircraft landing sequences, and vice-versa: the aircraft assigned to the landing sequence n is the *lead* aircraft of the aircraft assigned to the landing sequence $n+1$.

Aircraft that are cleared to enter the SCA receive, in addition to the follow notification, a MAHF assignment. Therefore, aircraft are described by a PVS record type `Aircraft` with fields `id` (identification), `seq` (landing sequence), and `mahf` (MAHF assignment).[1] For simplicity, aircraft identifications are encoded as natural numbers. Missed approach holding fix assignments are of type `Side`, which is an enumeration type of values `right` and `left`.

[1] PVS definitions are available in the Appendix Definitions.

3.1 SCA

The SCA is considered to be logically divided into 15 zones:[2]

Zone	Description
`holding3(right)`	Holding pattern at 3000 feet, right.
`holding3(left)`	Holding pattern at 3000 feet, left.
`holding2(right)`	Holding pattern at 2000 feet, right.
`holding2(left)`	Holding pattern at 2000 feet, left.
`lez(right)`	Lateral entry to IAF-R.
`lez(left)`	Lateral entry to IAF-L.
`base(right)`	Base segment right, IAF-R to IF.
`base(left)`	Base segment left, IAF-L to IF.
`intermediate`	Intermediate segment, IF to FAF.
`final`	Final segment, FAF to runway.
`runway`	Runway.
`maz(right)`	Missed approach path, runway to IAF-R.
`maz(left)`	Missed approach path, runway to IAF-L.
`departure(right)`	Departure path, runway to departure fix right.
`departure(right)`	Departure path, runway to departure fix left.

Table 1: *SCA Logical Zones*

Figure 4 illustrates the logical zones of the SCA. Note that geographically, these zones are not disjoint. For example, the lateral entry zone right and the missed approach zone right overlap. The SCA zones serve to define the state of an aircraft. An aircraft is defined by the three components of its `Aircraft` representation (`id`,`seq`,`mahf`) and the zone of the SCA where the aircraft is operating. A result of the verification presented in this paper is that overlapping zones do not contain aircraft simultaneously. For example, if there is an aircraft in the lateral entry zone right, then the missed approach zone right will be empty.

The global state of the SCA is represented in PVS by the record type `SCA`. Aircraft inside a zone are represented as a sequence (type `queue`) of aircraft descriptions (type `Aircraft`). The first element of this sequence is the first aircraft that entered the zone, and so forth. The record type `SCA` also includes fields for the next available landing sequence (`nextseq`) and alternated missed approach holding fix assignment (`nextmahf`).

3.2 Procedures

SATS procedures are encoded as transition rules that modify the state of the SCA. Under these rules, the zones behave as first-in first-out data structures: aircraft are removed from the head of one zone and added to the tail of the next zone. This representation guarantees that aircraft follow a logical trajectory according to operational and physical constraints. For example, an aircraft shall not overtake its leader. Moreover, it cannot go from

[2] As it is usually depicted, right and left are relative to the pilot facing the runway, i.e., opposite from the reader point of view.

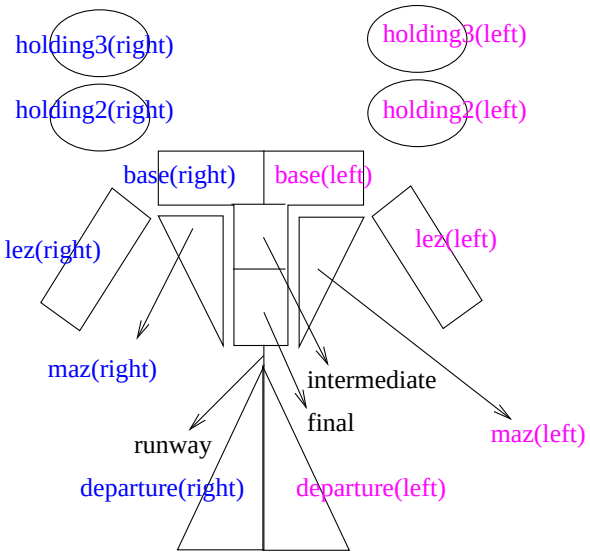


Figure 4: *SCA Zones*

`holding3(right)` to `final` without orderly going through `holding2(right)`, `base(right)`, and `intermediate`.

The application of a transition rule to a particular state of the SCA is restricted to the conformance of the state to the operational rule defined by the SATS concept. For instance, a vertical entry procedure at a given fix can be performed only if there is no aircraft on the approach assigned to the same fix as a MAHF.

Twenty four transition rules have been identified that correspond to different SATS procedures:

- Vertical Entry (right, left).
- Lateral Entry (right, left).
- Descend from 3000 to 2000 feet (right, left).
- Approach Initiation for Vertical Entry (right, left).
- Approach Initiation for Lateral Entry (right, left).
- Base Segment to Intermediate Segment (right, left).
- Departure from SCA.
- Intermediate Segment to Final Segment.
- Landing.
- Taxing.
- Missed Approach Initiation.

- Lowest Available Altitude (right, left).
- Departure Initiation (right, left).
- Takeoff.
- Departing from SCA (right,left).

3.2.1 Vertical Entry (right, left)

The Vertical Entry (right) procedure is illustrated in Figure 5. A vertical entry (right) is allowed into the SCA only if all the following conditions hold:

- There are less than 2 aircraft either at IAF-R or assigned to IAF-R as a MAHF.
- No aircraft currently on the approach assigned to IAF-R as a MAHF (zones base(right), intermediate, final).
- No aircraft executing a missed approach with IAF-R as its MAHF (zone maz(right).
- No aircraft performing a lateral entry to IAF-R (zone lez(right)).
- No aircraft at IAF-R holding at 3000 feet or transitioning to 2000 feet.

Conditions for vertical entry (left) are identical to vertical entry (right) with IAF-L and (left) in place of IAF-R and (right), respectively. An aircraft entering the SCA follows the last aircraft in the sequence; that is, the aircraft that entered the SCA immediately prior. It follows *none* if it is the first aircraft in the sequence. Aircraft are assigned MAHF opposite to the one assigned to its leading aircraft. If the entering aircraft is first in the sequence, its assigned MAHF is the same as its entry IAF.

This procedure is encoded by the PVS function `VerticalEntry`.[3]

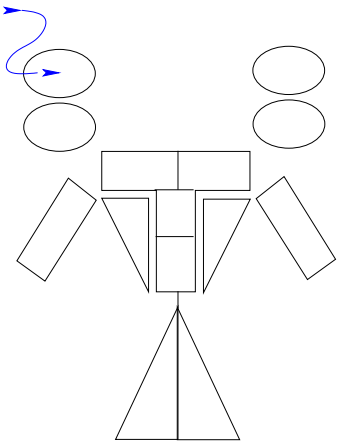


Figure 5: *Vertical Entry (right)*

[3] PVS transitions are available in the Appendix Rules.

3.2.2 Lateral Entry (right, left)

The Lateral Entry (right) procedure is illustrated in Figure 6. A lateral entry is allowed into the SCA only if all the following conditions hold:

- No aircraft at the IAF on the same side as the requested lateral entry.
- No aircraft assigned to that fix as a MAHF.
- No aircraft in lateral entry zone proceeding to the same IAF.
- No aircraft on missed approach zone at that fix.

The entering aircraft follows the last aircraft in the sequence (*none* if it is the first aircraft in the sequence). It is assigned to the opposite MAHF of its leader. If the entering aircraft is first in the sequence, it is assigned to the missed approach holding fix located at the same side as its initial approach fix.

This procedure is encoded by the PVS function `LateralEntry`.

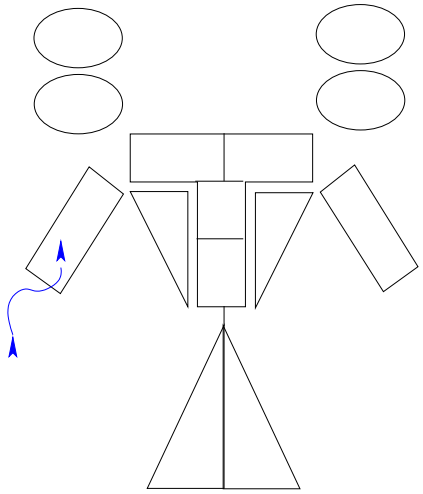


Figure 6: *Lateral Entry (right)*

3.2.3 Descend from 3000 to 2000 feet (right, left)

The Holding Pattern Descend (right) procedure is illustrated in Figure 7. An aircraft holding at 3000 feet is allowed to descend and hold at 2000 feet only if there is no aircraft holding at 2000 feet at the same IAF.

This procedure is encoded by the PVS function `HoldingPatternDescend`.

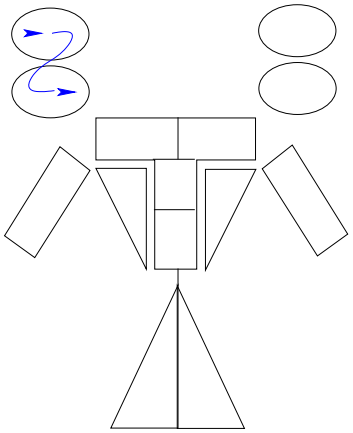


Figure 7: *Descend from 3000 to 2000 feet (right)*

3.2.4 Approach Initiation for Vertical Entry (right, left)

The Approach Initiation for Vertical Entry (right) procedure is illustrated in Figure 8. An aircraft holding at 2000 feet is allowed to initiate the approach only if the following conditions hold:

- It is the first aircraft in the sequence or its leader is already on the approach.
- There is at most one aircraft on base at the opposite side.

The second condition is a result of geometrical and spacing assumptions. The condition abstracts the details of the lengths of the base segment and spacing requirements. The

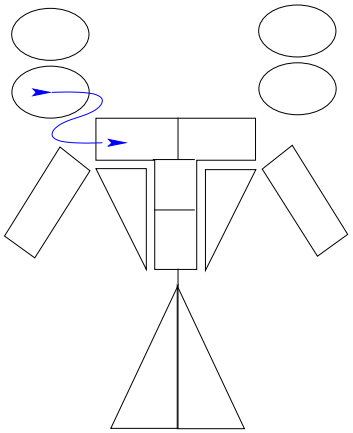


Figure 8: *Approach Initiation for Vertical Entry (right)*

rational for this condition is that the length of the base segment is approximately 5 NM and the spacing requirement is at least 3 NM. Figure 9 illustrates the case where two aircraft on base on one side of the T approach will not permit a third aircraft to initiate its approach.

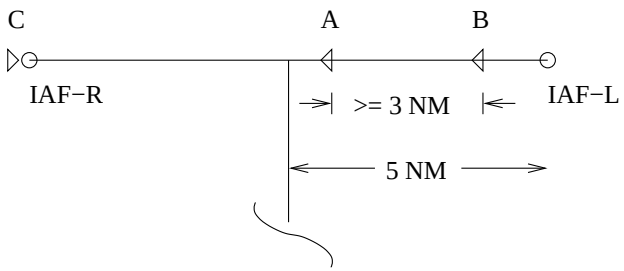


Figure 9: *Rational for Second Condition in Approach Initiation Rule*

In this example, aircraft A is the leader of aircraft B and B is the leader of C. Since there must be a spacing of at least 3 NM between all aircraft, the only possible way to have a 3 NM spacing between aircrafts B and C is for A to be past the IF and on the intermediate segment. Note that although aircraft B and C are at opposite IAFs, the spacing between these two aircraft is measured as the difference between their path distance to the runway threshold. This distance is sometimes called *Manhattan distance*.

This procedure is encoded by the PVS function `VerticalApproachInitiation`.

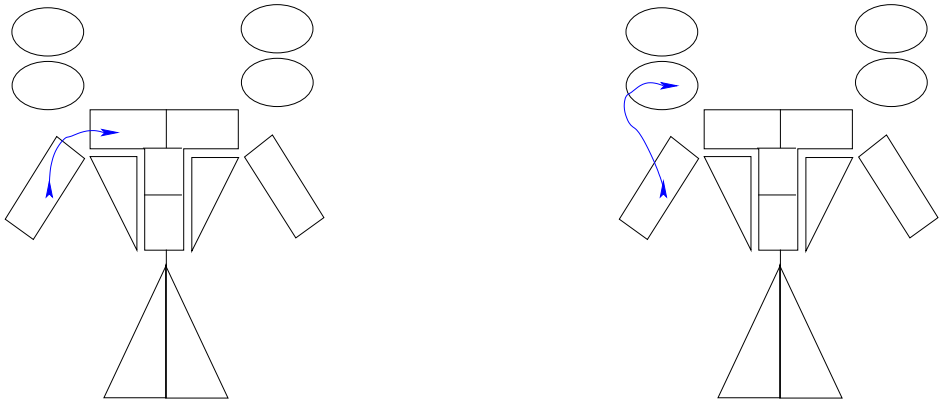
3.2.5 Approach Initiation for Lateral Entry (right, left)

The Approach Initiation for Lateral Entry (right) procedure is illustrated in Figure 10. An aircraft in lateral entry is allowed to initiate the approach only if the following conditions hold:

- It is the first aircraft in the sequence or its leader is already on the approach.
- There is at most one aircraft on base at the opposite side.

If one of these conditions is not met, the aircraft must hold at 2000 feet.

This procedure is encoded by the PVS function `LateralApproachInitiation`.



Leader on the approach and enough spacing Otherwise

Figure 10: *Approach Initiation for Lateral Entry (right)*

3.2.6 Base Segment to Intermediate Segment (right, left)

The Base Segment to Intermediate Segment (right) procedure is illustrated in Figure 11. An aircraft that makes the transition from the base segment to the intermediate segment is either the first aircraft in the sequence or its leader is already on the final approach.

This procedure is encoded by the PVS function `Merging`.

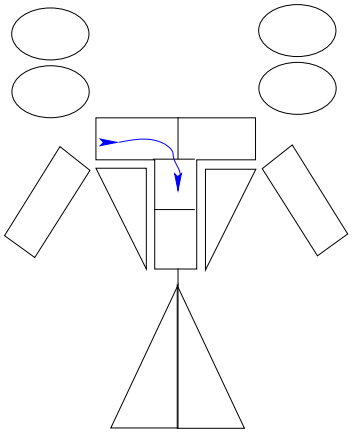


Figure 11: *Base Segment to Intermediate Segment (right)*

3.2.7 Departure from SCA from a Missed Approach

The Departure from SCA procedure is illustrated in Figure 12. An aircraft on the intermediate segment may exit the SCA only if it is the first aircraft in the sequence.

This procedure is encoded by the PVS function `Exit`.

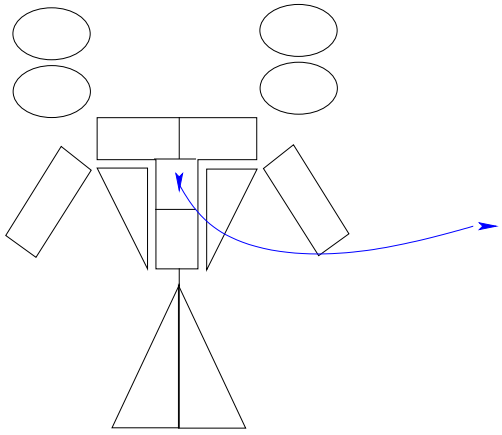


Figure 12: *Departure from SCA*

3.2.8 Intermediate Segment to Final Segment

The Intermediate Segment to Final Segment procedure is illustrated in Figure 13. An aircraft on the intermediate segment may go to the final segment and prepare to land.

This procedure is encoded by the PVS function `FinalSegment`.

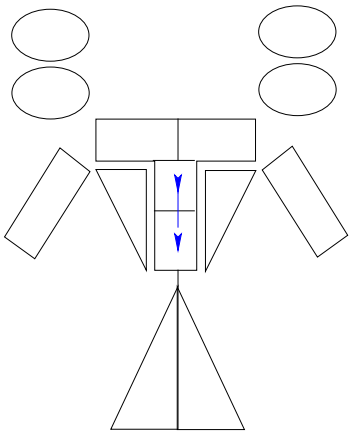


Figure 13: *Intermediate Segment to Final Segment*

3.2.9 Landing

The Landing procedure is illustrated in Figure 14. The first aircraft in the final segment may land if there are no aircraft on the runway.

This procedure is encoded by the PVS function `Landing`.

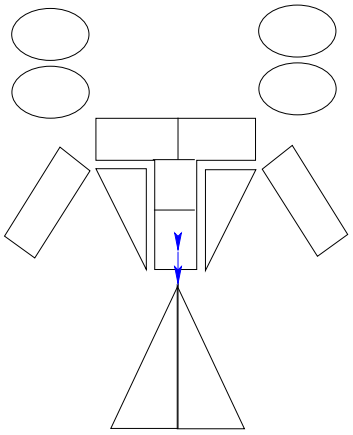


Figure 14: *Landing*

3.2.10 Taxing

The Taxing procedure is illustrated in Figure 15. An aircraft that has landed eventually leaves the runway.

This procedure is encoded by the PVS function `Taxing`.

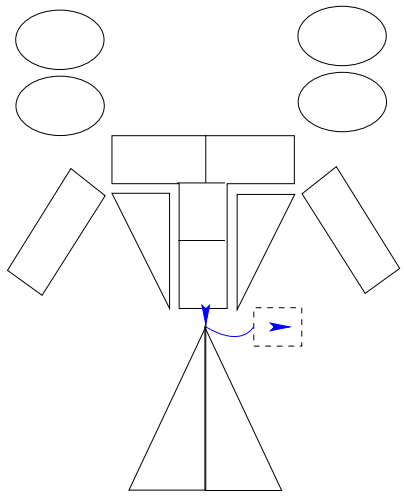


Figure 15: *Taxing*

3.2.11 Missed Approach Initiation

The Missed Approach Initiation procedure is illustrated in Figure 16. An aircraft may initiate a missed approach procedure if it is in the final segment and it is the first aircraft in the sequence. This rule is an abstraction of the operational rule which permits an aircraft to initiate a missed approach after crossing the missed approach point (MAP). The MAP physical location is generally at the end of the final segment or beginning of the runway. However, for the abstract model, even if physically the MAP is at the beginning of the runway, an aircraft performing a missed approach is considered to transition from the final segment to the missed approach zone.

When an aircraft performs a missed approach, it goes to the missed approach zone corresponding to its MAHF assignment. The AMM computes a new landing sequence and missed approach holding fix assignment for the next approach operation of the aircraft. It follows the last aircraft to enter the SCA (*none* if the aircraft is first in the sequence). It is assigned to the opposite MAHF of its leader. If it is the first aircraft in the sequence, it keeps its original assignment.

This procedure is encoded by the PVS function `MissedApproach`.

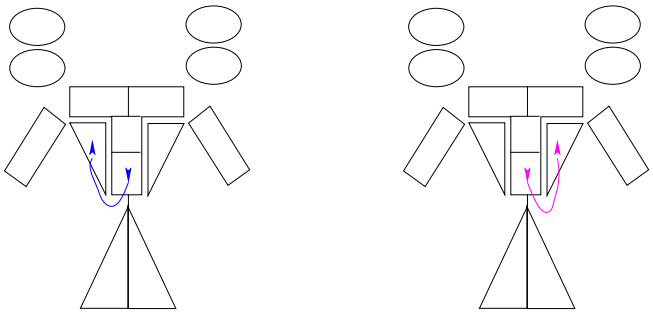


Figure 16: *Missed Approach Initiation*

3.2.12 Lowest Available Altitude (right, left)

The Lowest Available Altitude (right) procedure is illustrated in Figure 17. An aircraft in missed approach goes to its MAHF at the lowest available altitude. If an aircraft is holding at 3000 feet, but the holding pattern at 2000 feet is available, the lowest available altitude is 3000 feet.

This procedure is encoded by the PVS function `LowestAvailableAltitude`.

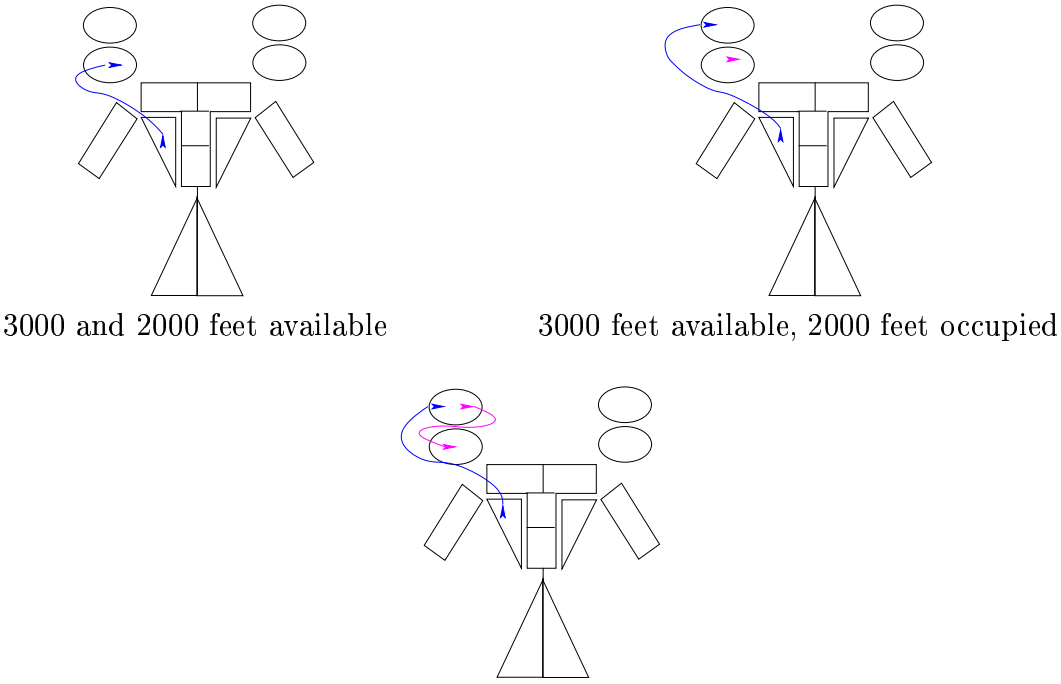


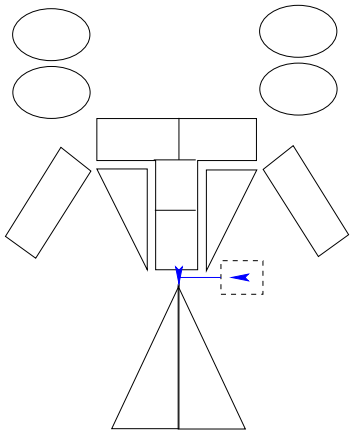
Figure 17: *Lowest Available Altitude (right)*

3.2.13 Departure Initiation (right, left) from the Runway

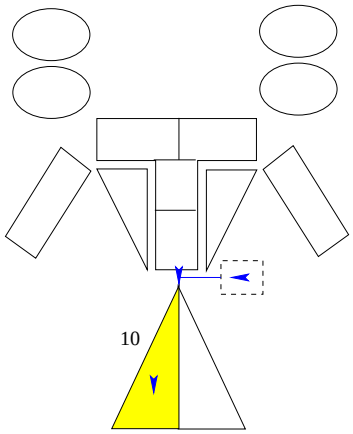
The Departure Initiation (right) procedure is illustrated in Figure 18. A departure operation is allowed only if no aircraft is on the final segment or waiting at the runway, and either one of the following cases holds:

1. No aircraft on departure zones.
2. Aircraft departing on the same zone are 10nm or more from the runway and aircraft departing on the opposite zone are 3nm or more from the runway.

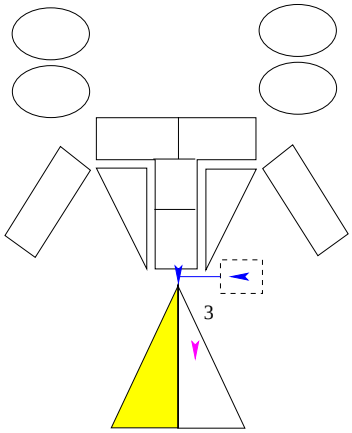
This procedure is encoded by the PVS function `DepartureInitiation`.



No aircraft on departure zones



One aircraft on same departure zone at 10nm



One aircraft on opposite departure zone at 3nm

Figure 18: *Departure Initiation (right)*

3.2.14 Takeoff

The Takeoff procedure is illustrated in Figure 19. Aircraft on the runway that have initiated a departure operation are allowed to takeoff and go to the zone corresponding to their departure fix.

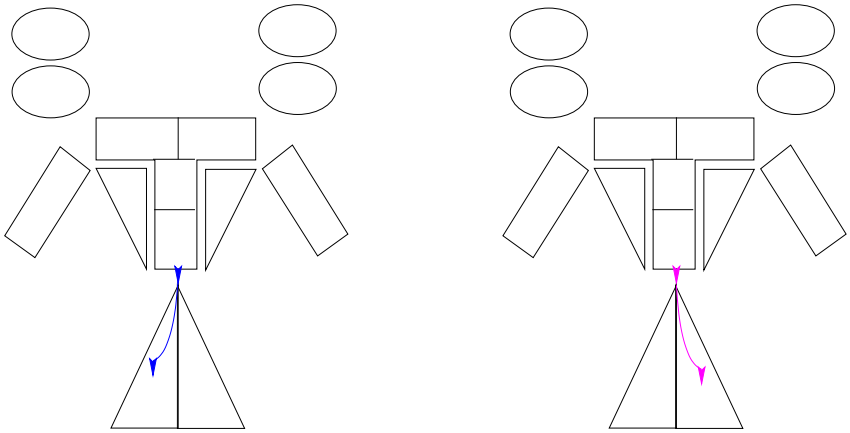


Figure 19: *Takeoff*

This procedure is encoded by the PVS function `Takeoff`.

3.2.15 Departing from SCA (right,left) from Takeoff

The Departing from SCA (right) procedure is illustrated in Figure 20. Aircraft in the departure zone eventually leave the SCA.

This procedure is encoded by the PVS function `Departing`.

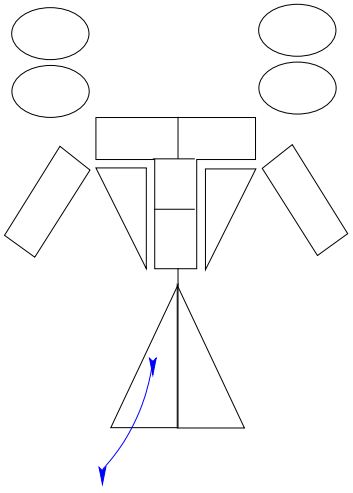


Figure 20: *Departing from SCA (right)*

4 PROPERTIES

Several properties of the operational concept have been identified and formally proven in PVS.

In the following examples, A_r denotes an aircraft A following *none* assigned to the right MAHF and A_l^B denotes an aircraft A following B assigned to the left MAHF.

4.1 Simultaneous Landing Operations

The SATS concept does not allow more than four simultaneous landing operations. The basis for the validity of this property is that there must always be a MAHF assignment available for an aircraft in the SCA. Since there are only two holding fixes and two possible altitudes, at most four simultaneous landing operations are allowed inside the SCA.

This property is encoded by the PVS predicate `four_landings`.[4]

4.2 Missed Approach Holding Fix Assignments

There are no more than two aircraft assigned to each MAHF. Moreover, there are no more than 2 aircraft at each side of the SCA (excluding the approach). This property is also a consequence of the fact that there is always a MAHF assignment available for an aircraft in the SCA.

Notice that the potential number of aircraft at or assigned to one side of the SCA may be greater than 2. For instance, in the nominal scenario depicted in Figure 21, aircraft A_r is on the approach, aircraft B_l^A is holding at 2000 feet at the right fix, and aircraft C_r^B is holding at 2000 feet at the left fix. In this case, all three aircraft are either on the right side of the SCA or have a missed approach fix at right. However, this scenario does not jeopardize the property on the actual number aircraft on the right side of the SCA as B has to leave the right side of the SCA before C can perform the approach. Only A and C can be in the right missed approach zone at the same time.

This property is encoded by the PVS predicate `well_assigned`.

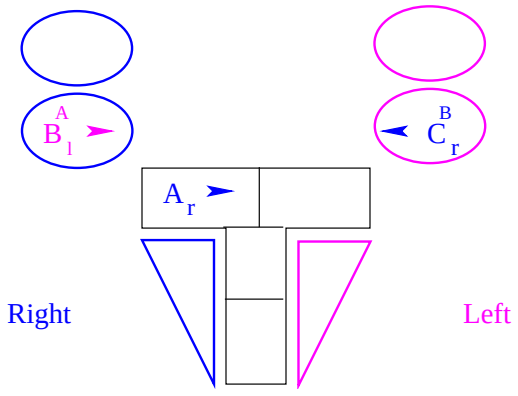


Figure 21: *Three Potential Aircraft at Right*

[4] PVS properties are available in the Appendix Properties.

4.3 Occupancy

The operational concept imposes very strong conditions on how many aircraft may occupy a zone at the same time. For example, there is at most one aircraft at a holding fix for a given altitude, there are no more than 2 aircraft on a missed approach zone, and there are no more than 3 aircraft on base. Using the abstract model, it is possible to prove that there will never be more than 3 aircraft on the base segments. However, depending on how spacing constraints are implemented, most likely there will never be more than 2 aircraft and maybe only 1 aircraft in the base segments. Because the spacing constraints are not fully defined and the abstract model does not capture the spacing constraints, it is only possible to prove, as an upper bound, that there are at most 3 aircraft on the base segments.

This property is encoded by the PVS predicate `non_crowded_sca`.

4.4 Lateral Entry

Lateral entries are safe in the sense that there is at most one aircraft cleared for a lateral entry at a given fix. Furthermore, while an aircraft is performing a lateral entry, that side of the SCA (excluding the approach, but including the missed approach zone) remains empty.

This property is encoded by the PVS predicate `safe_len`.

4.5 Smooth Merging

The Smooth Merging property states that an aircraft on the base segment is either the first in the sequence or its leading aircraft is on base and will reach the IF first or its leading aircraft is on final. This property is a consequence of the operational concept requirement that an aircraft cannot initiate an approach until its leading aircraft has started the approach and that an aircraft cannot overtake its leading aircraft.

This property is encoded by the PVS predicate `smooth_merging`.

4.6 Safe Landing

The fact that overtaking is not allowed inside the SCA implies that aircraft land according to the leadership relation.

This property is encoded by the PVS predicate `safe_landing`.

4.7 Runway Incursion

Landing and departure initiation procedures ensure that runway incursions do not occur under nominal operations. It was established through the verification work presented in this paper that there will be no more than one aircraft on the runway at any time.

This property is encoded by the PVS predicate `no_incursion`.

4.8 Departures

The departure initiation procedure imposes 3 nautical miles separation between aircraft departing to opposite departure fixes and 10 nautical miles separation between aircraft departing to the same departure fix.

This property is encoded by the PVS predicate `safe_departure`.

5 VERIFICATION ISSUES

The dynamics of the SCA environment were encoded by a series of discrete events, i.e., transitions from one state to another, triggered by the operational rules. In order to accommodate all possible scenarios, due, for example, to different aircraft performances, the model is non-deterministic. For a given state of the SCA, all possible transitions are considered.

The set of possible states is theoretically infinite. However, using a simple exhaustive exploration technique, where aircraft identifications are disregarded, 2811 reachable states have been found from an initial state where the SCA is empty. Each one of these states represent a potential nominal SCA scenario. Checking by hand each one of them is clearly not feasible. Using the PVS ground evaluator[5], it has been checked that for all reachable states, all the properties of Section 4 hold. Hence, it is said that the model is *correct* (with respect to those properties).

The model has also been shown to be free of *deadlocks*, i.e., scenarios where the SCA global state cannot progress any longer. For instance, the situation depicted in Figure 22, where B_l^A is holding at 2000 feet while A_r is holding at 3000 feet, is not reachable and therefore can never occur under nominal operations.

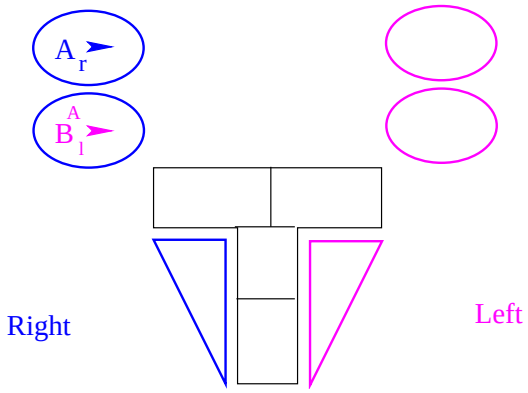


Figure 22: *Illustration of a Deadlock*

Finally, it has been checked that if the rules that allow aircraft to enter the SCA or depart from the SCA are suppressed, then the SCA eventually evolves into the empty state. From a practical point of view, this property means that the SCA can be effectively and properly sterilized if needed (for example, when non-equipped aircraft request clearance to enter the SCA or when an aircraft declares an emergency).

The absence of deadlocks and the fact that the SCA eventually evolves into the empty state (when no new aircraft are allowed) are called the *liveness* property.

This section is concluded with some discussions on the model and the verification that has been conducted.

[5]The ground evaluator is a PVS tool that produces efficient executable Lisp code from a PVS functional specification.

5.1 Non-Deterministic Model

The model of SATS dynamics is fully non-deterministic. For instance, in the scenario depicted in Figure 23, either A_r or B_l^A could potentially descend to 2000 feet. Determining which one of them will effectively be the first aircraft to descend in a real situation depends on several factors including aircraft performance and pilot preference. The operational concept precludes aircraft for hovering in a holding pattern once a lower altitude becomes available. However, the exact time when this will occur is not defined. The solution adopted is to avoid all these considerations by allowing either one of the transitions to occur first.

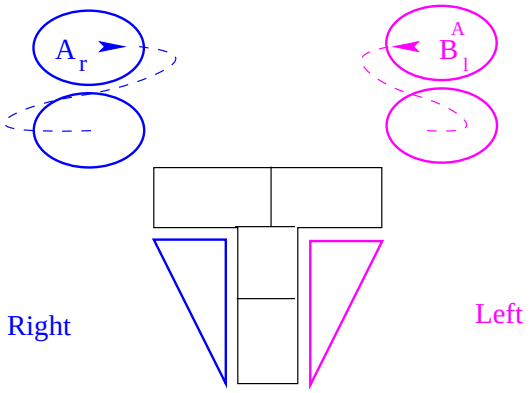


Figure 23: *Non-deterministic Behavior*

In such a non-deterministic model, some non-nominal sequences of events may occur where, for example, one aircraft changes zones several times, while another remains idle. Therefore, the model is more conservative than reality.

5.2 Simultaneous Transitions

From the discussion above, the fact that `LowestAvailableAltitude` (Section 3.2.12) is a simultaneous transition, potentially involving 2 aircraft, may be considered a weakness of the formalization. If simultaneous transition are suppressed from that rule, the scenario depicted in Figure 24 might result. In that case, an aircraft A_r^C is leaving the right missed approach zone while an aircraft C_r is holding at 3000 feet. If A goes to 3000 feet, there is a safety issue; if it goes to 2000 feet, it will result in a deadlock scenario.

In the model, the safety issue and the deadlock are avoided by forcing aircraft C to descend to 2000 feet and assuming 3000 feet as the lowest available altitude for aircraft A. As the model satisfies the correctness and the liveness properties, both problems have been effectively solved.

The fact that the simultaneous transition does not never occur under nominal operations is easily justified by the fact that the operational concept precludes an aircraft from hovering at a given altitude when a lower altitude is available (as it is the case of aircraft C). However, such property cannot be established by the model since it requires a finer time-space model of the SCA.

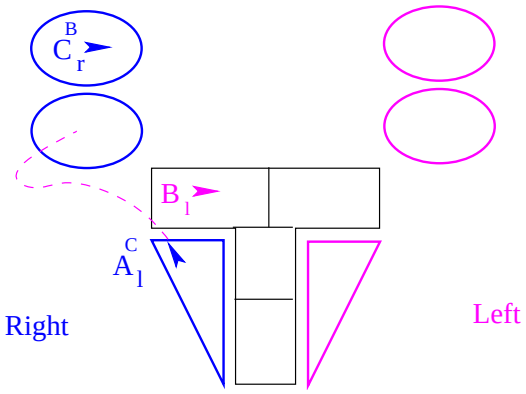


Figure 24: *A Potential Deadlock Scenario*

5.3 The Idle Effect

In the physical world aircraft do not remain idle (except, of course, aircraft on the ground). In the non-deterministic model this means that given a non-empty zone of a reachable state of the SCA, there is always a transition that effectively removes one aircraft from the head of that zone. However, this is not the always the case. In particular aircraft in the holding zones and aircraft in the base segments may, theoretically, remain idle.

The fact that aircraft on the holding zones remain idle does not defy physics. Indeed, that is the indented behavior of holding zones in the mathematical model. Since aircraft kinematics are not modeled in a holding pattern, from an abstract point of view, aircraft in a holding pattern *do* remain idle.

The case of base segment is different. The only rule dealing with that zone, `Merging` (Section 3.2.6), moves an aircraft from the base segment to the final approach only if its leader is already on final approach. From an abstract point of view, an aircraft in the base segment remains idle waiting for its leader to go first into the final approach. If this condition is removed from the rule, properties such as Smooth Merging cannot be proved.

Intuitively, the fact that the condition of the merging rule is always true and that, from a practical point of view, aircraft do not remain idle on the base segment is a consequence of the operational concept. Under the SATS concept an aircraft that initiates an approach has to be safely spaced with respect to its leader. Therefore, its leader always goes first into the final approach.

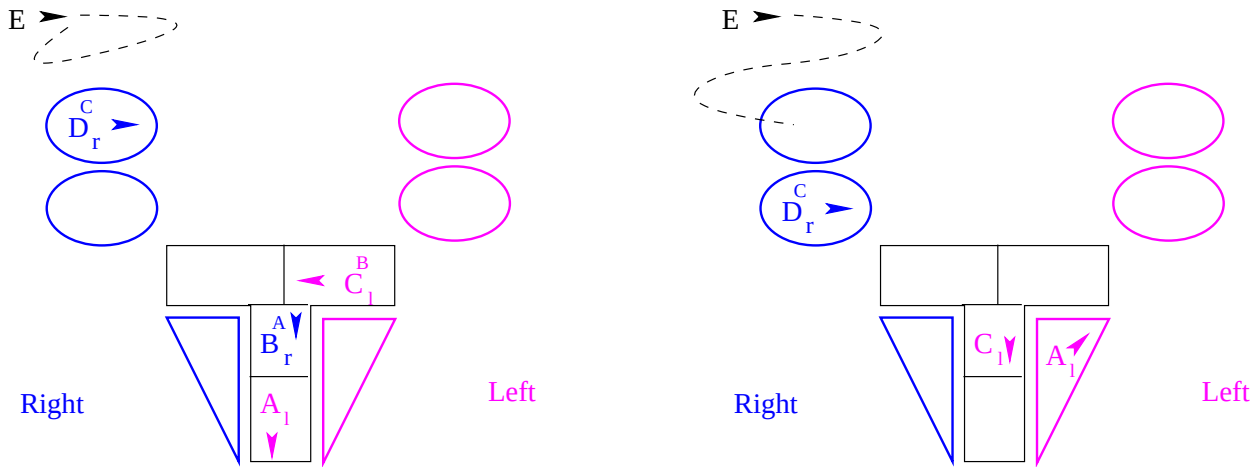
As in the case of the simultaneous transition in `LowestAvailableAltitude`, to formally verified the non-idle effect of `Merging` requires a more accurate time-space model of the SCA. Nevertheless, it is noted that the rules `LowestAvailableAltitude` and `Merging` are the only rules in the model that need this kind of justification.

5.4 Missed Approach Holding Fix Reassignment

The Vertical Entry rule in the original SATS concept [1] reads: *"The AMM rules that determine if a normal (vertical) entry into the SCA is permitted are: (1) There are less than 2 aircraft either at that fix or assigned to the fix, (i.e., as a missed approach holding fix),* **and** *(2) no aircraft assigned to that fix as a missed approach holding fix on the approach".*

Regarding the MAHF assignment, the original concept states that *"once the aircraft (in a missed approach) gets within the proximate area of the missed approach fix, the aircraft is automatically re-sequenced for another approach"*. Therefore, when an aircraft initiates a missed approach procedure, it keeps its MAHF until it arrives at the vicinity of the holding fix. Nothing is said about the leader assignment, but it may be implicitly understood that the aircraft loses it.

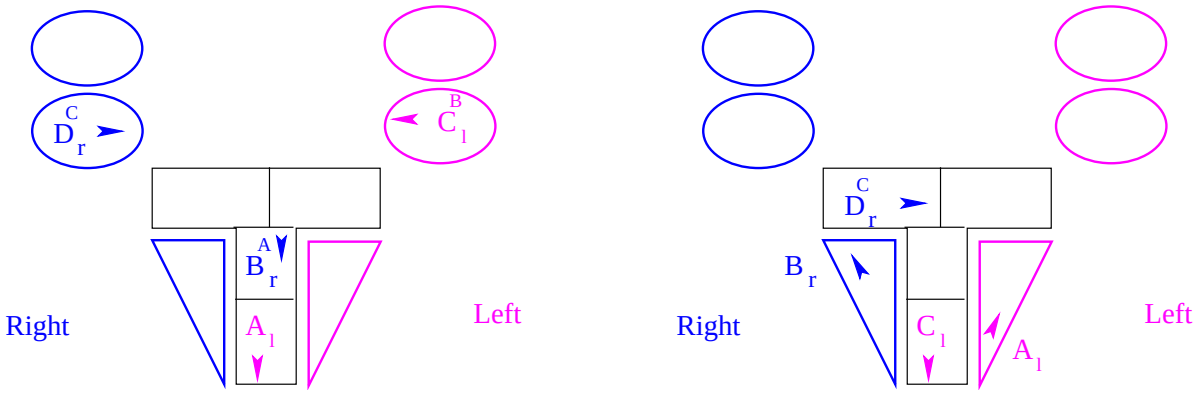
These rules, as they are originally stated, enable scenarios that violate safety hypotheses of the SATS concept. Indeed, several offending scenarios were discovered using the state exploration tool developed as part of this verification. For instance, Figure 25 (a) illustrates a nominal scenario with four aircraft in the SCA (A_l, B_r^A, C_l^B, and D_r^C) and one aircraft E requesting authorization for a vertical entry approach. The AMM denies the entry since there are already four aircraft in the SCA. Some minutes later A has performed a missed approach, B has landed, C is on the final approach, and D holds at 2000 feet (Figure 25 (b)). According to the vertical rule above, the AMM grants entry to aircraft E (there are less than 2 aircraft on the right fix and no aircraft on the approach is assigned to that fix). It gets the assignment E_l^D, i.e., E follows D and its MAHF is left. As result of this rule, , three aircraft, e.g.,A_l, C_l, and E_l^D, will be assigned to the left MAHF. This is a violation of a safety requirement of the operational concept.



(a) AMM denies entry to aircraft E (b) AMM grants entry to aircraft E

Figure 25: *Vertical Entry Issue*

A different scenario is illustrated by Figure 26 (a). Four aircraft are in the SCA: A_l and B_r^A on the final approach, and C_l^B and D_r^C holding at 2000 feet. Some minutes later A and B have both performed a missed approach, and C and D are on the approach (Figure 26 (b)). If aircraft B is assumed to be faster than the aircraft A, it arrives first to the proximity of its holding fix. According to the missed approach reassignment rule above, B gets the assignment B_l^D for its next approach, i.e., B follows D and its missed approach holding fix is left. As a result of this assignment, three aircraft, e.g., A_l, B_l^D, and C_l, are assigned to the left missed approach holding fix. This is a violation of a safety requirement of the operational concept.



(a) Full SCA (b) Aircraft B is faster than A

Figure 26: *Missed Approach Reassignment Issue*

To solve both cases, it has been recommended by the authors that when an aircraft initiates a missed approach procedure, it immediately gets a landing sequence and a missed approached holding fix assignment for the next operation (see Missed Approach Initiation procedure in Section 3.2.11). When the new rule is used in the first scenario (Figure 25 (b)), aircraft A gets the assignment A_l^D and aircraft E gets the assignment E_r^A. In this case, two aircraft are assigned to the left (A_l^D and C_l) and two aircraft are assigned to the right (D_r^C and E_r^A). In the second scenario (Figure 26 (b)), aircraft A gets the assignment A_l^D and aircraft B gets the assignment B_r^A. In this case, two aircraft are assigned to the left (A_l^D and C_l) and two aircraft are assigned to the right (B_r^A and D_r^C).

The MAHF reassignment is implemented in the current concept in a slightly different form: immediately after an aircraft performs a missed approach, the AMM computes a new landing sequence and MAHF assignment for the next approach. However, the aircraft itself keeps its MAHF for the current operation until it arrives at the vicinity of the holding fix. At that moment, the AMM issues the previously computed follow notification and missed approach holding fix assignment. The difference is of an implementation nature. From a human factors perspective, it is undesirable to reassign a MAHF to an aircraft that has not finished a missed approach procedure. However, from an algorithmic point of view, the logic of the operational rules is based on the AMM computed landing sequence and MAHF assignments, independently of the information that is available to the aircraft.

The new MAHF assignment rule is safe but incurs in some efficiency penalties. For instance, in the scenario depicted in Figure 26 (b)), B follows A although B is faster than A. It means that aircraft B has to hold at 2000 feet and cannot initiate the approach until A has initiated its own approach and both aircraft are safely spaced. This inefficiency occurs in other SATS operations as well. It is inherently related to the requirement that MAHF assignments are always alternated. Alternative solutions are considered where, for example, a faster aircraft remains unassigned and holds until the aircraft on the final approach lands or performs a missed approach. These solutions were discarded by the designers of the operational concept due to implementation complexities.

6 CONCLUSION

The SATS operational concept describes nominal operations inside the SCA. One of the key hypothesis of the concept's design is that nominal operations are safe, i.e., aircraft flying under nominal operation conditions are always separated. The formal validation of this hypothesis involves answering several kind of questions. Some of these questions involve structural issues such as "Do the rules cover all possible scenarios?", while other questions are more of operational or human-factors nature such as "Are the aircraft spaced appropriately on the final approach?", "Do pilot flight preferences affect the overall safety of the concept?".

The mathematical model presented in this paper addresses the first kind of issues: the structural and logical properties of the concept. Time and space dimensions are discretized to enable mechanical exploration of nominal scenarios. Moreover, by allowing non-deterministic behaviors, aircraft and pilot performances are abstracted away. For these reasons, the model is conservative, i.e., the set of scenarios described by the model is a superset of the set of nominal scenarios. However, such model yields a robust notion of safety, i.e., a notion that relies only on the logic of the concept and not on space-time properties such as the geometry of the SCA, physical constraints such as aircraft performances, or human factors such as pilot preferences.

It has been verified in PVS that the mathematical model satisfies several safety properties such as:

- There are no more than 4 simultaneous operations.
- There is always a MAHF available for an aircraft in the SCA.
- There are no more than 2 aircraft performing a missed approach to a given fix.
- There is at most one aircraft cleared for an entry at a given fix.
- Aircraft land in order.
- Runway incursions do not occur.

Furthermore, the model satisfies liveness properties such as:

- There are no deadlocks.
- Landing and departing aircraft eventually complete their operations.

Valid questions on this work are (1) weather or not the model accurately captures the *real* operational concept, and (2) how complete is the analysis. If the model is faithful, it can be claimed that *real* nominal operations satisfy all the properties that have been verified in the mathematical model. If the analysis is complete it can claimed that the *real* concept is indeed *safe*. These questions, of course, cannot be formally answered. However, the model has been validated with the designers of the SATS concept. The validation flowed in both directions. Indeed, as result of this formal work, nine issues have been identified related to the original concept and 10 recommendations presented to the SATS Concept of Operations working group. All the recommendations were accepted and implemented in the current concept. They are necessary to achieve the intended safe functionality in the SCA. On of the major

contributions of this work is the redesign of the missed approach holding fix reassignment rule as explained in Section 5.4.

On the other hand, the analysis is not complete. The time-space abstraction is not fine enough to handle safety properties that do require aircraft and pilot performances such as:

- Aircraft on the missed approach zone are separated.
- Aircraft on the approach are spaced.
- An aircraft in a holding pattern has enough time to initiate the approach before another aircraft needs that MAHF/altitude combination. (see discussion on Section 5.2).
- Aircraft have enough time to merge for the final approach (see discussion on Section 5.3).
- Aircraft are separated during the transition from one zone to the other.
- Departing aircraft are separated from landing aircraft.

Future work includes the construction of a refined model of the SATS concept, on top of the abstract one, that allows us to verify safety properties that cannot be discharged by the current approach.

ACKNOWLEDGMENTS

The authors are very thankful to the SATS-HVO research group at NASA Langley for providing technical support while conducting this investigation. The authors are specially thankful to Lou Glaab and Cathy Adams for reviewing this document and providing useful comments.

REFERENCES

[1] C. Adams, M. Consiglio, K. Jones, D. Williams, and T. Abbott. SATS HVO Operational Concept: Nominal Operations. NASA Langley Research Center, (Unpublished; available by request from Brian.T.Baxley@nasa.gov), 2003.

[2] *Federal Aviation Regulations/Aeronautical Information Manual*, 1999. Published by Jeppesen Sanderson Inc., Englewood, CO.

[3] SATS Program Office. Small aircraft transportation system program plan. Available from http://sats.larc.nasa.gov/documents.html, 2001.

[4] S. Owre, J. M. Rushby, and N. Shankar. PVS: A prototype verification system. In Deepak Kapur, editor, *11th International Conference on Automated Deduction (CADE)*, volume 607 of *Lecture Notes in Artificial Intelligence*, pages 748–752, Saratoga, NY, June 1992. Springer-Verlag.

APPENDIX

Definitions

```
% SATS Definitions
satsdefs : THEORY
BEGIN

  %% Sides in the SCA
  Side  : TYPE = right,left

  none  : MACRO nat  = 0

  %% Aircraft
  Aircraft : TYPE = [#
    id  : nat, % Identification. For printing only.
    seq : nat, % Sequence number, which encodes the leader relationship.
    mahf: Side % Missed approach holding fix assignment.
  #]

  % For a departing aircraft mahf is the departure fix and seq is the distance
  % in miles from runway (3,10).

  IMPORTING queue[Aircraft]

  %% SCA Zones
  Zone : TYPE = queue

  %% SCA State
  SCA : TYPE = [#
    holding3,    % Holding Pattern 3kft
    holding2,    % Holding Pattern 2kft
    lez,         % Lateral Entry Zone
    maz,         % Missed Approach Zone
    base,        % Base segment
    departure   :% Departure zone
    [Side->Zone],
    intermediate,% Intermediate segment
    final,       % Final segment
    runway      :% Runway
    Zone,
    nextmahf    : Side,  % Next missed approach holding fix
    nextseq     : posnat,% Next sequence number
    nextid      : nat,   % Next identification number. For printing only
    rule        : int    % Rule applied to get this sate. For printing only
  #]

  %% For Model-Checking
  Transition : TYPE = [SCA->list[SCA]]
```

```
Path       : TYPE = list[SCA]
Result     : TYPE = [#
  counterexample : Path,
  deadlocks      : list[Path],
  visited        : list[SCA]
#]

%% Variables
z    : VAR Zone
side : VAR Side
n    : VAR nat
s    : VAR SCA
a    : VAR Aircraft

%% Opposite side
opposite(side) : Side =
  IF side = right THEN left
  ELSE right
  ENDIF

%% First landing sequence
first : MACRO nat = 1

%% Sign of side
sign(side) : MACRO int =
  IF side=right THEN -1
  ElSE 1
  ENDIF

%% Is this the first aicraft in landing sequence ?
first?(a): MACRO bool =
  a`seq = first

%% Landing sequence of leader aircraft
leader(a): MACRO nat =
  IF a`seq = none THEN none
  ElSE a`seq - 1
  ENDIF

%% Is b the leader aircraft of a ?
leader?(a,b:Aircraft): MACRO bool =
  b`seq = leader(a)

%% Next landing sequence
next(a): MACRO nat =
  a`seq + 1
```

```
%% Number of aircraft in a zone to assigned to one side
assigned(z,side): RECURSIVE nat =
  IF empty?(z) THEN 0
  ElSIF first(z)`mahf = side THEN 1+assigned(rest(z),side)
  ELSE assigned(rest(z),side)
  ENDIF
  MEASURE z BY <<

%% Is any aircraft in zone z assigned to the mahf side ?
assigned?(z,side): bool =
  assigned(z,side) /= 0

%% Is aircraft with sequence number n in zone z ?
on_zone?(z,n) : RECURSIVE bool =
  IF empty?(z) THEN false
  ElSIF first(z)`seq = n THEN true
  ELSE on_zone?(rest(z),n)
  ENDIF
  MEASURE z BY <<

%% Decreases by one the sequence number of aircraft in zone z
decrease(z) : RECURSIVE Zone =
  IF empty?(z) THEN z
  ELSE
    LET a = first(z) IN
    add(a WITH [`seq := leader(a)],decrease(rest(z)))
  ENDIF
  MEASURE z BY <<

%% Is aircraft with sequence number n on the approach ?
on_approach?(s,n): bool =
  on_zone?(s`base(right),n) or on_zone?(s`base(left),n) or
  on_zone?(s`intermediate,n) or on_zone?(s`final,n)

%% Is any aircraft on the approach assigned to the mahf side ?
on_approach?(s,side): bool =
  assigned?(s`base(right),side) or assigned?(s`base(left),side) or
  assigned?(s`intermediate,side) or assigned?(s`final,side)

%% Acutal number of aircraft at one side (excluding the approach)
actual(s,side):nat =
  length(s`holding3(side))+length(s`holding2(side))+length(s`lez(side))+
  length(s`maz(side))

%% Virtual number of aircraft at one fix
virtual(s,side): nat =
  length(s`holding3(side)) + length(s`holding2(side))+
```

```
  length(s`lez(side)) + length(s`maz(side)) +
  assigned(s`holding3(opposite(side)),side) +
  assigned(s`holding2(opposite(side)),side) +
  assigned(s`lez(opposite(side)),side) +
  assigned(s`maz(opposite(side)),side) +
  assigned(s`base(right),side) +
  assigned(s`base(left),side) +
  assigned(s`intermediate,side) +
  assigned(s`final,side)

%% Number of aircraft assigned to a fix
assigned2fix(s,side):nat =
  assigned(s`holding3(right),side) +
  assigned(s`holding3(left),side) +
  assigned(s`holding2(right),side) +
  assigned(s`holding2(left),side) +
  assigned(s`lez(right),side) +
  assigned(s`lez(left),side) +
  assigned(s`base(right),side) +
  assigned(s`base(left),side) +
  assigned(s`intermediate,side) +
  assigned(s`final,side) +
  assigned(s`maz(right),side) +
  assigned(s`maz(left),side)

%% Total number of simultaneous landing operations
landing_op(s):nat =
  actual(s,right) + actual(s,left) +
  length(s`base(right)) + length(s`base(left)) +
  length(s`intermediate) + length(s`final)

%% New aircraft
aircraft(s,side):Aircraft = (#
  id   := s`nextid,
  seq  := s`nextseq,
  mahf := IF s`nextseq = first THEN side
          ELSE s`nextmahf ENDIF
#)

%% New aircraft for departure
departure(s,side):Aircraft = (#
  id   := s`nextid,
  seq  := 0,
  mahf := side
#)

%% Reassign aircraft
```

```
  reassign(s,a):Aircraft =
    a WITH [
      `seq  := s`nextseq,
      `mahf := IF s`nextseq - 1 = first THEN a`mahf
               ElSE s`nextmahf
               ENDIF
    ]

  %% Move a departing aircraft
  move(a):Aircraft =
    a WITH [`seq := IF a`seq = 0 THEN 3 ELSE 10 ENDIF]

END satsdefs
```

Rules

```
%% Formal Model of SATS Concept of Operations
sats: THEORY
BEGIN

  IMPORTING satsdefs

  %% Variables
  side      : VAR Side
  this,next : VAR SCA

  %% A vertical entry is allowed into the SCA only if
  %% 1. there are less than 2 aircraft either at the fix or assigned
  %%    to the fix as a missed approach holding fix,
  %% 2. no aircraft assigned to that fix as a missed approach holding fix on
  %%    the approach,
  %% 3. no aircraft on missed approach zone at that fix,
  %% 4. no aircraft on lateral entry zone at that fix, and
  %% 5. no aircraft holding at 3kft.
  %% The aircraft gets an aircraft to follow, the last aircraft in sequence,
  %% or none if it becomes the first aircraft in sequence.
  %% It also gets as missed approach holding fix the opposite side of its
  %% leader, or the same side as its initial approach fix if it becomes
  %% the first in sequence.
  VerticalEntry(side)(this):list[SCA] =
    IF virtual(this,side) < 2 &
       NOT on_approach?(this,side) &
       length(this`maz(side)) = 0 &
       length(this`lez(side))  = 0 &
       length(this`holding3(side))  = 0 THEN
      LET a = aircraft(this,side) IN
      LET next = this WITH [
        `holding3(side):= add(this`holding3(side),a),
```

```
      `nextseq       := next(a),
      `nextmahf      := opposite(a`mahf),
      `nextid        := this`nextid+1,
      `rule          := 1*sign(side)
    ] IN
    (: next :)
  ELSE
    null
  ENDIF

%% A lateral entry is allowed into the SCA only if
%% 1. no aircraft at the initial approach fix,
%% 2. no aircraft on lateral entry zone at that fix,
%% 3. no aircraft in missed approach zone at that fix, and
%% 4. no aircraft has been assigned to the fix as a missed approach holding
%%    fix.
%% The aircraft gets an aircraft to follow, the last aircraft in sequence,
%% or none if it becomes the first aircraft in sequence.
%% It also gets as missed approach holding fix the opposite side of its
%% leader, or the same side as its initial approach fix if it becomes
%% the first in sequence.
LateralEntry(side)(this):list[SCA] =
  IF virtual(this,side) =  0 THEN
    LET a = aircraft(this,side) IN
    LET next = this WITH [
      `lez(side):= add(this`lez(side),a),
      `nextseq  := next(a),
      `nextmahf := opposite(a`mahf),
      `nextid   := this`nextid+1,
      `rule     := 2*sign(side)
    ] IN
    (: next :)
  ELSE
    null
  ENDIF

%% An aircraft holding at 3kft is allowed to descend and hold at 2kft
%% only if there is no aircraft holding at 2kft.
HoldingPatternDescend(side)(this):list[SCA] =
  IF length(this`holding3(side)) /= 0 &
     length(this`holding2(side)) = 0 THEN
    LET a = first(this`holding3(side)) IN
    LET next = this WITH [
      `holding3(side):= rest(this`holding3(side)),
      `holding2(side):= add(this`holding2(side),a),
      `rule          := 3*sign(side)
    ] IN
```

```
    (: next :)
  ELSE
    null
  ENDIF

%% An aircraft holding at 2kft is allowed to initiate the approach only if
%% 1. there are less than two aircraft on base, and
%% 2. it is the first aircraft in sequence or its leader is already on the T
VerticalApproachInitiation(side)(this):list[SCA] =
 IF length(this`holding2(side)) /= 0 THEN
   LET a =  first(this`holding2(side)) IN
   IF length(this`base(opposite(side))) <= 1 &
      (first?(a) or on_approach?(this,leader(a))) THEN
     LET next = this WITH [
       `holding2(side):= rest(this`holding2(side)),
       `base(side)    := add(this`base(side),a),
       `rule          := 4*sign(side)
     ] IN
     (: next :)
   ELSE
     null
   ENDIF
 ELSE
   null
 ENDIF

%% An aircraft in lateral entry is allowed  to initiate the approach only if
%% 1. there are less than two aircraft on base, and
%% 2. it is the first aircraft in sequence or its leader is already on the
%%    approach.
%% Otherwise, the aircraft must hold at 2kft.
LateralApproachInitiation(side)(this):list[SCA] =
  IF length(this`lez(side)) /= 0 THEN
    LET a =  first(this`lez(side)) IN
    IF length(this`base(opposite(side))) <= 1 &
       (first?(a) or on_approach?(this,leader(a))) THEN
      LET next = this WITH [
        `lez(side) := rest(this`lez(side)),
        `base(side):= add(this`base(side),a),
        `rule      := 5*sign(side)
      ] IN
      (: next :)
    ELSE
      LET next = this WITH [
        `lez(side)     := rest(this`lez(side)),
        `holding2(side):= add(this`holding2(side),a),
        `rule          := 5*sign(side)
```

```
      ] IN
      (: next :)
    ENDIF
  ELSE
    null
  ENDIF

%% An aircraft that goes on the intermediate segment is either the first
%% aircraft in seqence or its leader is already on the final approach.
%% Note: This is not really a rule, but an hypothesis that the SATS
%%       environment satisfies. It has to be discharged in a geometric model.
Merging(side)(this):list[SCA] =
  IF length(this`base(side)) /= 0 THEN
    LET a =  first(this`base(side)) IN
    IF first?(a) or
      on_zone?(this`intermediate,leader(a)) or
      on_zone?(this`final,leader(a)) THEN
      LET next = this WITH [
        `base(side)  := rest(this`base(side)),
        `intermediate:= add(this`intermediate,a),
        `rule        := 6*sign(side)
      ] IN
      (: next :)
    ELSE
      null
    ENDIF
  ELSE
    null
  ENDIF

%% The first aircraft in the intermediate segment may exit the SCA only if
%% it is the first in sequence.
Exit(this):list[SCA] =
  IF length(this`intermediate) /= 0 & first?(first(this`intermediate)) THEN
    LET next = this WITH [
      `holding3(right):= decrease(this`holding3(right)),
      `holding3(left) := decrease(this`holding3(left)),
      `holding2(right):= decrease(this`holding2(right)),
      `holding2(left) := decrease(this`holding2(left)),
      `lez(right)     := decrease(this`lez(right)),
      `lez(left)      := decrease(this`lez(left)),
      `maz(right)     := decrease(this`maz(right)),
      `maz(left)      := decrease(this`maz(left)),
      `base(right)    := decrease(this`base(right)),
      `base(left)     := decrease(this`base(left)),
      `intermediate   := decrease(rest(this`intermediate)),
      `nextseq        := IF this`nextseq = first THEN first
```

```
                        ELSE this‘nextseq-1 ENDIF,
      ‘rule           := 7
    ] IN
    (: next :)
  ELSE
    null
  ENDIF

%% The first aircraft in the intermediate segment may go to the
%% final segment and prepare to land.
FinalSegment(this):list[SCA] =
  IF length(this‘intermediate) /= 0 THEN
    LET a =  first(this‘intermediate) IN
    LET next = this WITH [
      ‘intermediate:= rest(this‘intermediate),
      ‘final       := add(this‘final,a),
      ‘rule        := 8
    ] IN
    (: next :)
  ELSE
    null
  ENDIF

%% The first aircraft in the final segment may land if there is no
%% aircraft in the runway.
Landing(this):list[SCA] =
  IF length(this‘final) /= 0 & length(this‘runway) = 0 THEN
    LET next = this WITH [
      ‘holding3(right):= decrease(this‘holding3(right)),
      ‘holding3(left) := decrease(this‘holding3(left)),
      ‘holding2(right):= decrease(this‘holding2(right)),
      ‘holding2(left) := decrease(this‘holding2(left)),
      ‘lez(right)     := decrease(this‘lez(right)),
      ‘lez(left)      := decrease(this‘lez(left)),
      ‘maz(right)     := decrease(this‘maz(right)),
      ‘maz(left)      := decrease(this‘maz(left)),
      ‘base(right)    := decrease(this‘base(right)),
      ‘base(left)     := decrease(this‘base(left)),
      ‘intermediate   := decrease(this‘intermediate),
      ‘final          := decrease(rest(this‘final)),
      ‘runway         := add(this‘runway,first(this‘final)),
      ‘nextseq        := IF this‘nextseq = first THEN first
                         ELSE this‘nextseq-1 ENDIF,
      ‘rule           := 9
    ] IN
    (: next :)
  ELSE
```

```
    null
  ENDIF

%% A landed aircraft eventually leaves the runway.
Taxing(this):list[SCA] =
  IF length(this`runway) /= 0 & first?(first(this`runway)) THEN
    LET next = this WITH [
      `runway:= rest(this`runway),
      `rule  := 10
    ] IN
    (: next :)
  ELSE
    null
  ENDIF

%% The first aircraft in the final approach is the only aircraft allowed
%% to perform a missed approach. It goes to the missed approach holding
%% fix that has been previously assigned to it. The aircraft gets reassigned:
%% 1. new follow notification: last aircraft to enter the SCA or None if the
%%    aircraft is first in sequence, and
%% 2. new missed approach holding fix assignement: opposite side of its leader or
%%    it keeps its original assignement if the aircraft becomes the first
%%    in sequence.
MissedApproach(this):list[SCA] =
  IF length(this`final) /= 0 THEN
    LET a  = first(this`final) IN
    LET ra = reassign(this,a) IN
    LET next = this WITH [
      `holding3(right):= decrease(this`holding3(right)),
      `holding3(left) := decrease(this`holding3(left)),
      `holding2(right):= decrease(this`holding2(right)),
      `holding2(left) := decrease(this`holding2(left)),
      `lez(right)     := decrease(this`lez(right)),
      `lez(left)      := decrease(this`lez(left)),
      `maz(opposite(a`mahf)):= decrease(this`maz(opposite(a`mahf))),
      `maz(a`mahf)    := decrease(add(this`maz(a`mahf),ra)),
      `base(right)    := decrease(this`base(right)),
      `base(left)     := decrease(this`base(left)),
      `intermediate   := decrease(this`intermediate),
      `final          := decrease(rest(this`final)),
      `nextmahf       := opposite(ra`mahf),
      `rule           := 11
    ]  IN
    (: next :)
  ELSE
    null
  ENDIF
```

```
%% An aircraft in missed approach goes to its missed approach holding fix
%% at the lowest available altitude. If an aircraft is holding at 3kft,
%% but the holding pattern at 2kft is available, the lowest available
%% altitude is 3kft.
LowestAvailableAltitude(side)(this):list[SCA] =
  IF length(this`maz(side)) /= 0 THEN
    LET a = first(this`maz(side)) IN
    IF length(this`holding3(side)) = 0 & length(this`holding2(side)) = 0 THEN
      LET next = this WITH [
        `holding2(side):= add(this`holding2(side),a),
        `maz(side)     := rest(this`maz(side)),
        `rule          := 12*sign(side)
      ] IN
      (: next :)
    ELSIF length(this`holding3(side)) = 0 THEN
      LET next = this WITH [
        `holding3(side):= add(this`holding3(side),a),
        `maz(side)     := rest(this`maz(side)),
        `rule          := 12*sign(side)
      ] IN
      (: next :)
    ELSE
      LET next = this WITH [
       `holding3(side):= add(rest(this`holding3(side)),a),
       `holding2(side):= add(this`holding2(side),first(this`holding3(side))),
       `maz(side)     := rest(this`maz(side)),
        `rule         := 12*sign(side)
      ] IN
      (: next :)
    ENDIF
  ELSE
    null
  ENDIF

%% A departure operation is allowed only if no aircraft is on final
%% segment or waiting at the runway, and either
%% 1. no aircraft on departure zones, or
%% 2. only one aircraft on departure zone, opposite departure fix,
%%    at a distance of 3nm or greater from runway, or
%% 3. only one aircraft on departure zone, same departure fix,
%%    at a distance of 10nm or greater from runway.
DepartureInitiation(side)(this):list[SCA] =
  IF length(this`final)+length(this`runway) = 0 &
     length(this`departure(right))+ length(this`departure(left)) < 2 &
     (length(this`departure(right))+ length(this`departure(left)) = 0 or
      length(this`departure(opposite(side))) > 0 &
```

```
      first(this‘departure(opposite(side)))‘seq >= 3 or
      length(this‘departure(side)) > 0 &
      first(this‘departure(side))‘seq >= 10) THEN
    LET next = this WITH [
      ‘runway:= add(this‘runway,departure(this,side)),
      ‘nextid:= this‘nextid+1,
      ‘rule  := 13*sign(side)
    ] IN
    (: next :)
  ELSE
    null
  ENDIF

%% Aircraft that have initiate a departure are allowed to takeoff.
Takeoff(this):list[SCA] =
  IF length(this‘runway) > 0 THEN
    LET a = first(this‘runway) IN
    IF a‘seq = 0 THEN
      LET next = this WITH [
        ‘runway           := rest(this‘runway),
        ‘departure(a‘mahf):= add(this‘departure(a‘mahf),a),
        ‘rule             := 14
      ] IN
      (: next :)
    ElSE
      null
    ENDIF
  ELSE
    null
  ENDIF

%% Aircraft in the departure zone eventually leave the SCA.
Departing(side)(this):list[SCA] =
  IF length(this‘departure(side)) > 0 THEN
    LET a = first(this‘departure(side)) IN
    LET next = this WITH [
      ‘departure(side):= IF a‘seq >= 10 THEN
                           rest(this‘departure(side))
                         ELSE
                           add(move(a),rest(this‘departure(side)))
                         ENDIF,
      ‘rule           := 15*sign(side)
    ] IN
    (: next :)
  ELSE
    null
  ENDIF
```

```
END sats
```

Properties

```
% SATS Properties
satsprops : THEORY
BEGIN

  IMPORTING satsdefs

  %% Variables
  side : VAR Side
  s    : VAR SCA

  % No more than four simultaneous landing operations.
  four_landings(s):bool =
    landing_op(s) <= 4 &
    s`nextseq-1 <= landing_op(s)

  % No more than 2 aircraft assigned to each missed approach fix. Moreover
  % no more than 2 aircraft at each side of the SCA (excluding the approach).
  well_assigned(s):bool =
    assigned2fix(s,right) <= 2 &
    assigned2fix(s,left) <= 2 &
    actual(s,right) <= 2 &
    actual(s,left) <= 2

  % At most one aircraft at a holding fix for a given altitude.
  % No more than 2 aircraft on a missed approach zone.
  non_crowded_side(side)(s):bool =
    length(s`holding3(side)) <= 1 &
    length(s`holding2(side)) <= 1 &
    length(s`maz(side)) <= 2

  % Each side of the SCA is non-crowded. Furthermore, no more than 3 aircraft
  % on base.
  non_crowded_sca(s):bool =
    non_crowded_side(right)(s) &
    non_crowded_side(left)(s) &
    length(s`base(right))+length(s`base(left)) <= 3

  % At most one aircraft on lateral entry. Moreover, if there is an
  % aircraft on lateral entry, that side of the SCA is empty (excluding the approach).
  safe_len(side)(s):bool =
    length(s`lez(side)) <= 1 &
    (length(s`lez(side)) > 0 IMPLIES
       length(s`holding3(side)) +
```

```
     length(s`holding2(side)) +
     length(s`maz(side)) = 0)

% An aircraft on the approach is either the first in sequence or its leader
% is already in the approach
smooth_merging(side)(s):bool =
  length(s`base(side)) > 0 IMPLIES
    LET a = first(s`base(side)) IN
    first?(a) or
    on_zone?(s`intermediate,leader(a)) or
    on_zone?(s`final,leader(a)) or
    (length(s`base(opposite(side))) > 0 &
     leader?(a,first(s`base(opposite(side)))))

% Aircraft land in sequence order.
safe_landing(s):bool =
  length(s`final) > 0 IMPLIES
    first?(first(s`final))

% At most one aircraft at the runway.
no_incursion(s):bool =
  length(s`runway) <= 1

% Simultaneous departures are separated (3nm if using different
% departure fix, 10nm if using the same departure fix).
safe_departure(side)(s):bool =
  length(s`departure(side)) > 1 IMPLIES
  (first(s`departure(side))`seq = 10 &
   first(rest(s`departure(side)))`seq = 0)

%% All the above properties
invariant(s):bool =
  four_landings(s)         &
  well_assigned(s)         &
  non_crowded_sca(s)       &
  safe_len(right)(s)       &
  safe_len(left)(s)        &
  smooth_merging(right)(s) &
  smooth_merging(left)(s)  &
  safe_landing(s)          &
  no_incursion(s)          &
  safe_departure(right)(s) &
  safe_departure(left)(s)

%  IMPORTING satsmc
%
%% All the states of the SAT model satisfy the invariant (via model-checking)
```

```
  %  correctness : THEOREM
  %    LET result = bdfs(invariant) IN
  %      length(result`counterexample)=0
  %
  %% SATS model does not have deadlock (via model-checking)
  %  liveness : THEOREM
  %    LET result = bdfs(invariant) IN
  %      length(result`deadlocks)=0

END satsprops
```

REPORT DOCUMENTATION PAGE	Form Approved OMB No. 0704–0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
01-03-2004	Technical Memorandum	

4. TITLE AND SUBTITLE
Abstract Model of the SATS Concept of Operations: Initial Results and Recommendations

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Dowek, Gilles; Muñoz, César A.; Carreño, Victor A.

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER
23-786-10-10

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
NASA Langley Research Center
Hampton, VA 23681-2199

8. PERFORMING ORGANIZATION REPORT NUMBER
L–18361

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Washington, DC 20546-0001

10. SPONSOR/MONITOR'S ACRONYM(S)
NASA

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
NASA/TM–2004–213006

12. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified-Unlimited
Subject Category 03
Availability: NASA CASI (301) 621-0390 Distribution: Standard

13. SUPPLEMENTARY NOTES
An electronic version can be found at http://techreports.larc.nasa.gov/ltrs/ or http://ntrs.nasa.gov.

14. ABSTRACT
An abstract mathematical model of the Concept of Operations for the Small Aircraft Transportation System (SATS) is presented. The Concept of Operations consists of several procedures that describe nominal operations for SATS. Several safety properties of the system are proven using formal techniques. The final goal of the verification effort is to show that under nominal operations, aircraft are safely separated. The abstract model was written and formally verified in the Prototype Verification System (PVS).

15. SUBJECT TERMS
Verification, SATS, General Aviation, PVS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			STI Help Desk (email: help@sti.nasa.gov)
U	U	U	UU	46	19b. TELEPHONE NUMBER *(Include area code)* (301) 621-0390