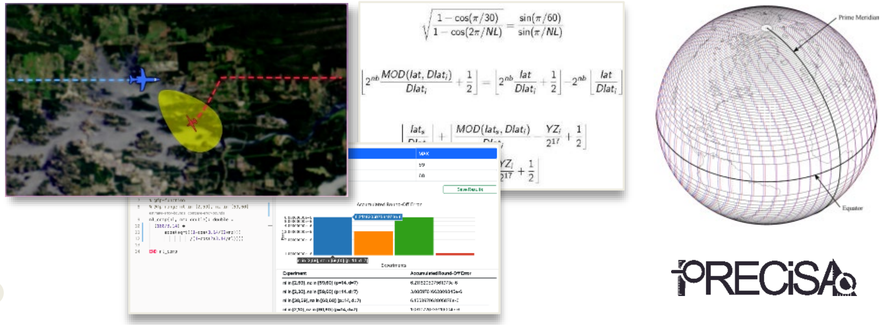
Formal Methods in the ~~Field~~ Air

Aaron Dutle*
NASA Langley Research Center
“Formal Methods in the Field” PI meeting,
11-13-2024

*In collaboration with many. Mariano Moscato, César Muñoz, and Laura Titolo most notably.

Safety-Critical Avionics Systems Branch

Maps Data: Google ©2021



Application and advancement of formal methods for specifying and verifying correctness and safety properties

Over 20 civil aviation standards and guidelines have D320 involvement

- RTCA DO-365 Detect and Avoid
- RTCA DO-260 ADS-B / TIS-S
- RTCA DO-178C SW Considerations
- RTCA DO-333 Formal Methods Supp.
- RTCA DO-276C Terrain & Obst. data
- RTCA DO-272D Aerodrome Mapping
- …

Work in progress

- SAE ARP4754B Development A/C Systems
- UL 4600 Standard for Safety for the Evaluation of Autonomous Products
- SAE ARP4761A Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment
- …

Safety assessment and assurance frameworks

Architectural and runtime V&V for assuring system-level integrity

Maps Data: Google ©2021

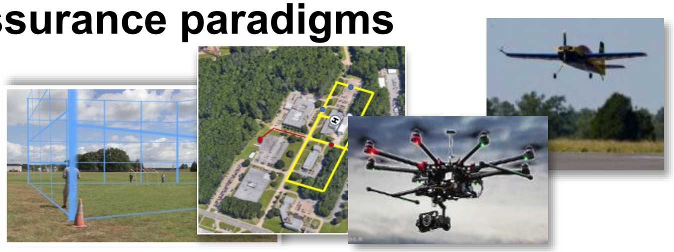
DANTi

Maps Data: Google ©2021

Highly-assured algorithms for aerospace applications

DAIDALUS

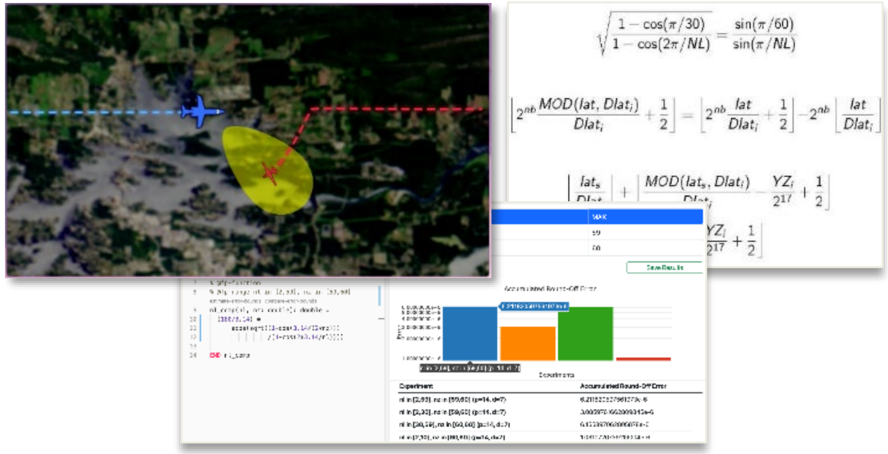
sUAS flight campaigns to validate environmental assumptions and test blended design and operational assurance paradigms

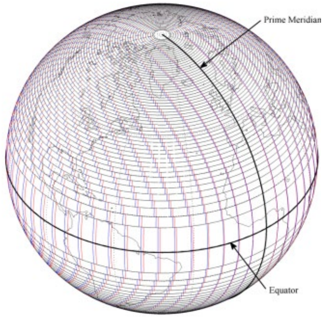


Maps Data: Google ©2021

Safety-Critical Avionics Systems Branch

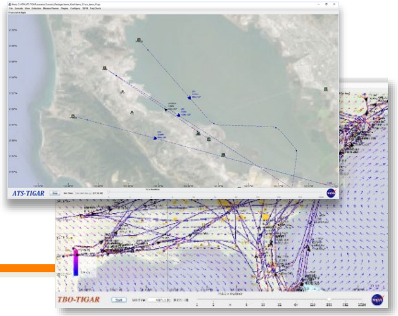
Maps Data: Google ©2021





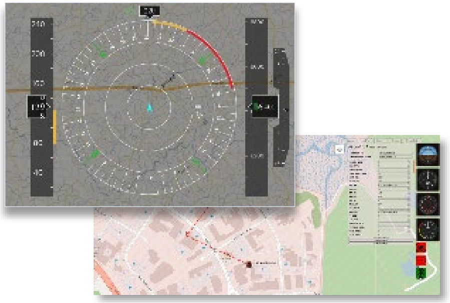
PRECiSA

Application and advancement of formal methods for specifying and verifying correctness and safety properties



ICAROUS
Integrated Configurable Algorithms for Reliable Operations of Unmanned Systems

DANTi



Maps Data: Google ©2021

Highly-assured algorithms for aerospace applications

DAIDALUS

Today's Success Story

Formal Methods

applied to

development of

Airspace Operations

See and Avoid

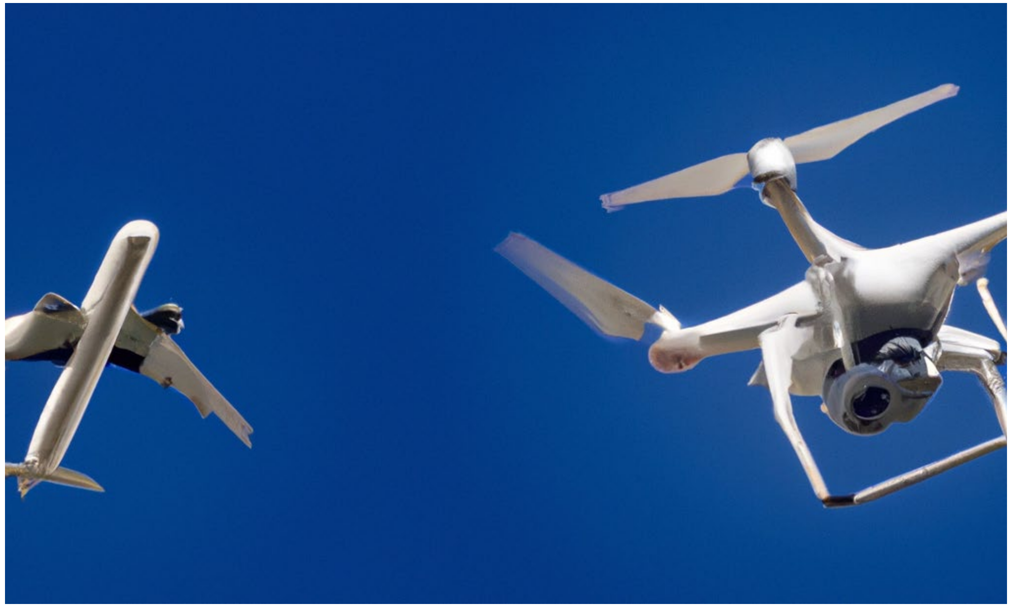
- Michael Huerta, Former Administrator, Federal Aviation Administration[1]:

*A bedrock principle of aviation is **see and avoid**. And if you don't have a pilot on board the aircraft, you need something that will substitute for that, which will **sense other aircraft**, and we can ensure appropriate levels of safety.*

Code of Federal Regulation requirements for operating in the National Airspace System include:

CFR 91.111: …not operate so close to another aircraft as to create a collision hazard

CFR 91.113: Vigilance shall be maintained…so as to see and avoid other aircraft…pilots shall alter course to pass well clear of other air traffic

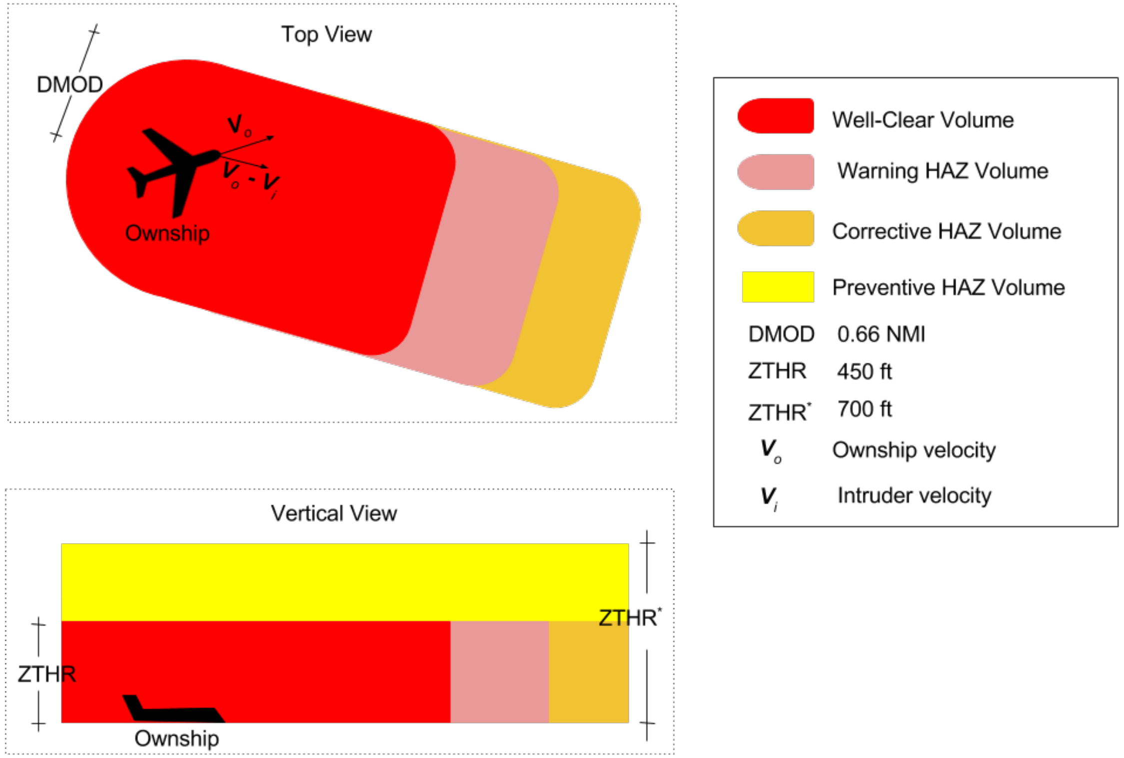


1: http://www.pbs.org/newshour/bb/drone-industry-grows-faster-flick-joystick-regulation-lags/

See and Avoid

Through the Sense and Avoid Science and Research Panel (SARP), NASA helped develop a geometric **well-clear** definition for uncrewed aircraft.

- Many stakeholders were involved in the panel.
- Several diverse candidates for the well-clear definition were considered.
- At first, experimental evidence, some pilot experiences, and other subjective measures were being considered…
- NASA developed **desired requirements**, and began **formally verifying or disproving** properties of candidate volumes.



Notional depiction of well-clear and alerting volumes

Some Desirable Properties for a Well-Clear definition

- **Symmetry**: Two aircraft in an encounter calculate the same well-clear status.

  Result: Neither aircraft is more burdened than the other.

- **Local Convexity**: In a non-maneuvering encounter, there is at most one time interval with a loss of well-clear.

  Result: A conflict or alert will not disappear and then reappear.

- **Extensibility**: Smaller parameter sets create a smaller (nested) volumes.

  Result: Alerts will progress monotonically in severity.

- **Convergence**: In a non-maneuvering encounter with a loss of well-clear, the interval includes the time of closest point of approach.

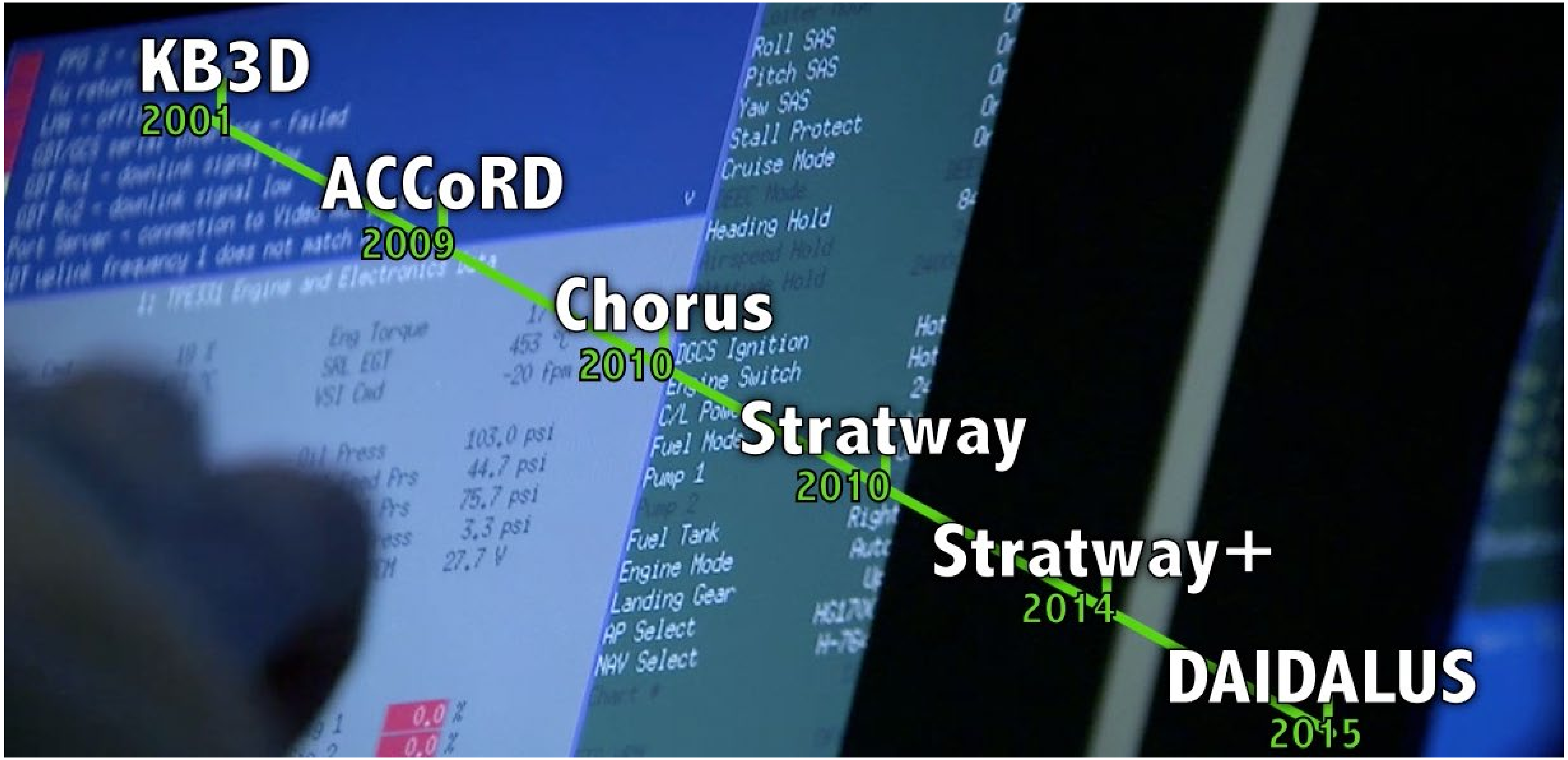
  Result: An alert will not disappear before the closest point of approach.

The NASA Formal Methods group specified these properties in the Prototype Verification System (PVS), and verified that several variants of the eventually chosen candidate satisfied them.

DAIDALUS

Detect & AvoID Alerting Logic for Unmanned Systems

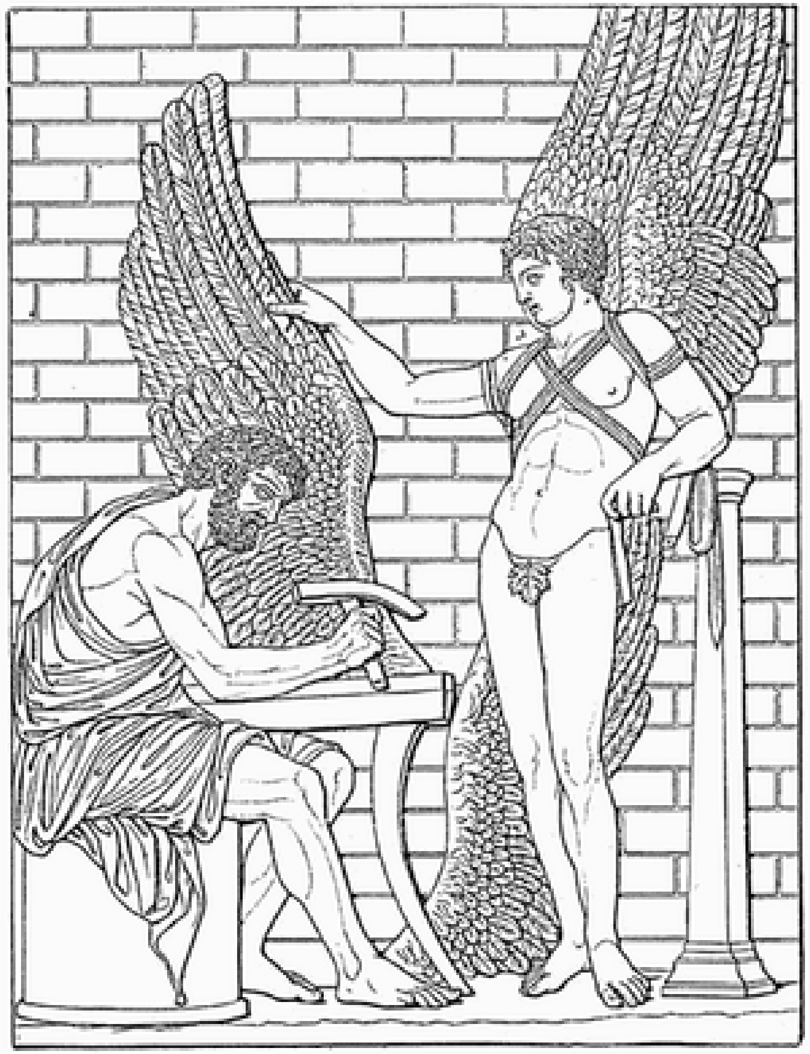
Not our first rodeo…



DAIDALUS is the product of years of research on formal V&V methods for the the safe integration of advanced air traffic concepts and algorithms in the National Airspace System.

What's in a name?

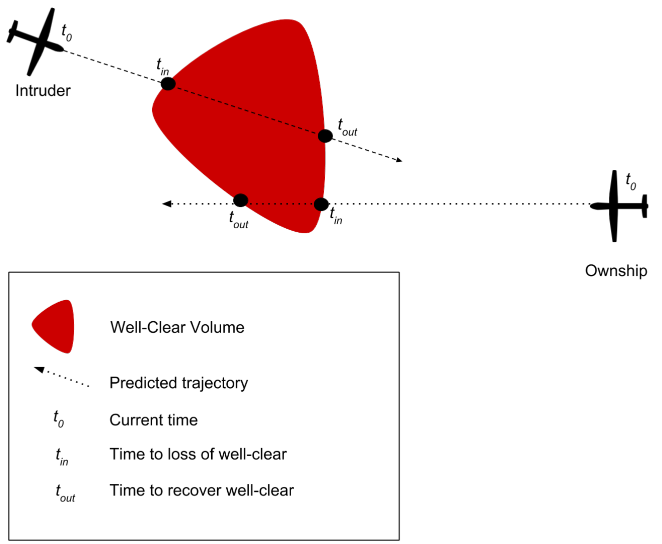
- DAIDALUS is a reference to the Greek myth.
- Daedalus advised Icarus not to fly too high, or his wings would melt in the sun, and not to fly too low, or his feathers would be soaked.



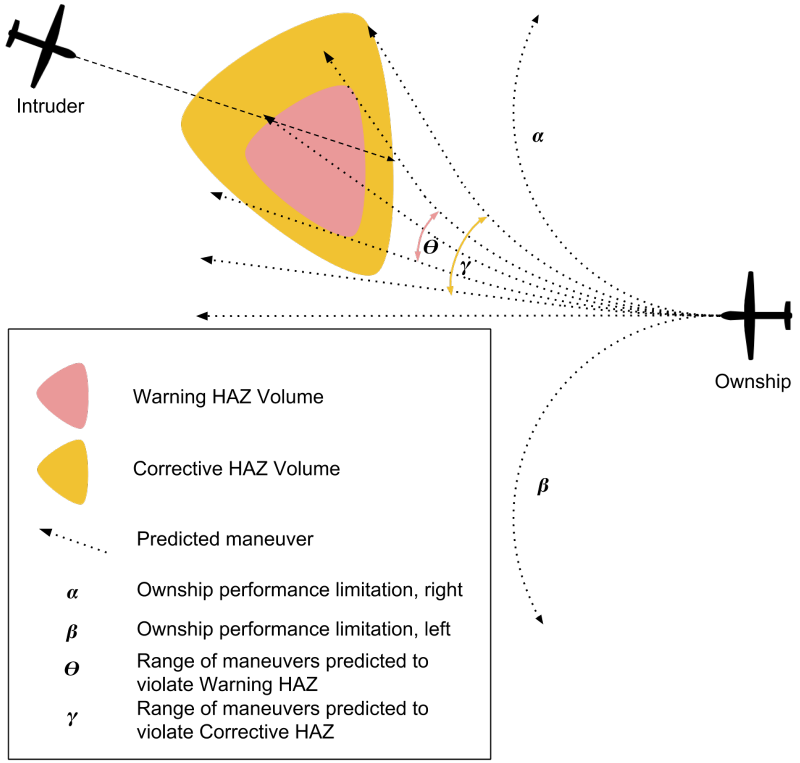
See page for author, Public domain, via Wikimedia Commons

Advice from DAIDALUS

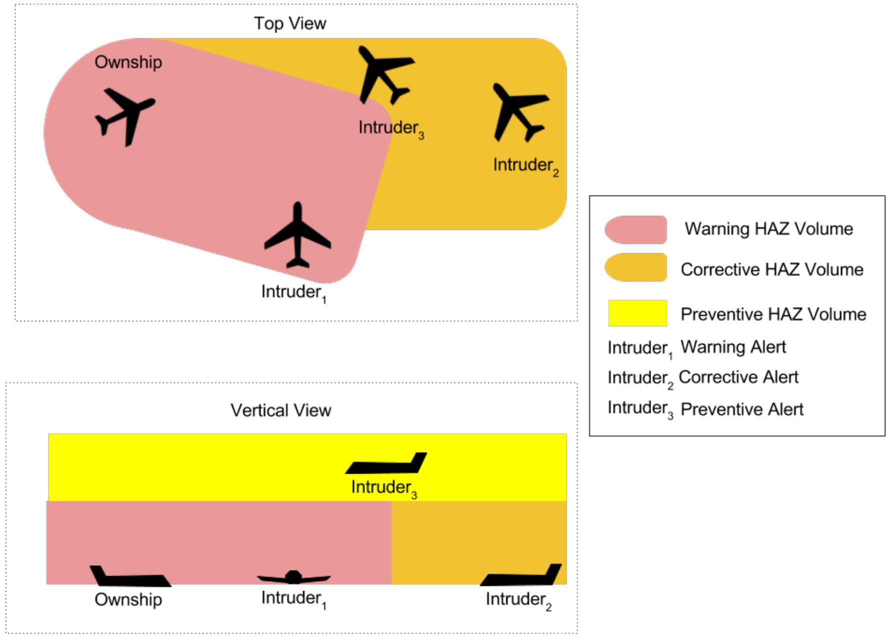
Detection determines the time interval where a loss of well-clear is predicted to occur



Maneuver Guidance finds ranges of maneuvers that lead to entering some hazard volume



Alerting computes a number indicating the severity of a potential loss of well-clear

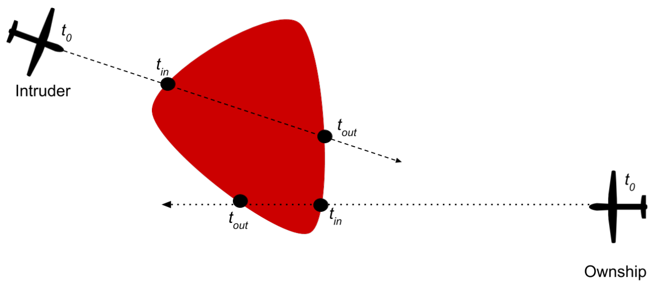


Formal verification of DAIDALUS in PVS

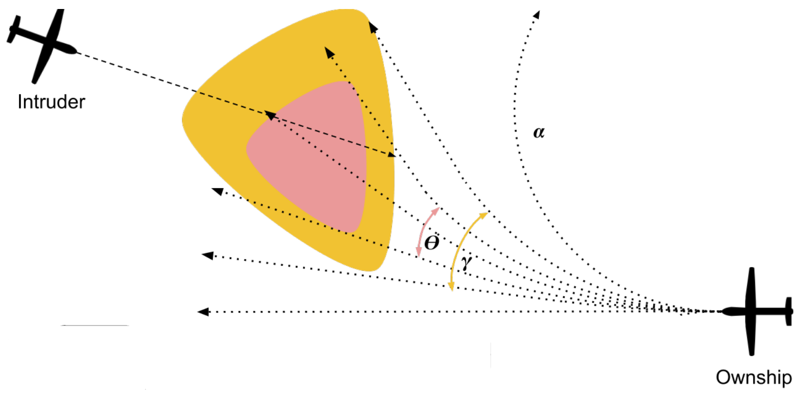
Detection determines the time interval where a loss of well-clear is predicted to occur

Maneuver Guidance finds ranges of maneuvers that lead to entering some hazard volume

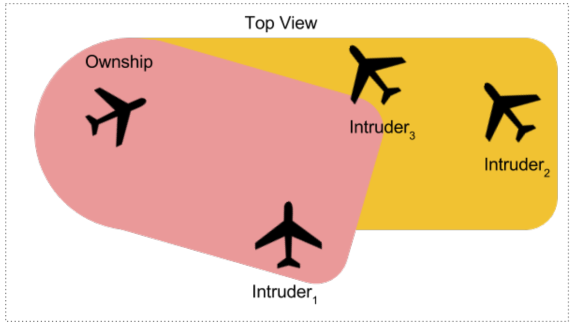
Alerting computes a number indicating the severity of a potential loss of well-clear



Lemma: (Detection Correctness) Time t ∈ [t_{in}, t_{out}] if and only if the projected aircraft will have a loss of well-clear in t seconds.



Lemma: (Conflict-free maneuver) If a maneuver is marked NONE, there will be no conflict along the path within the lookahead time.



Lemma: (Guidance coordination) An alert is issued if and only if maneuver guidance includes the current trajectory.

Formal Models are great, but…

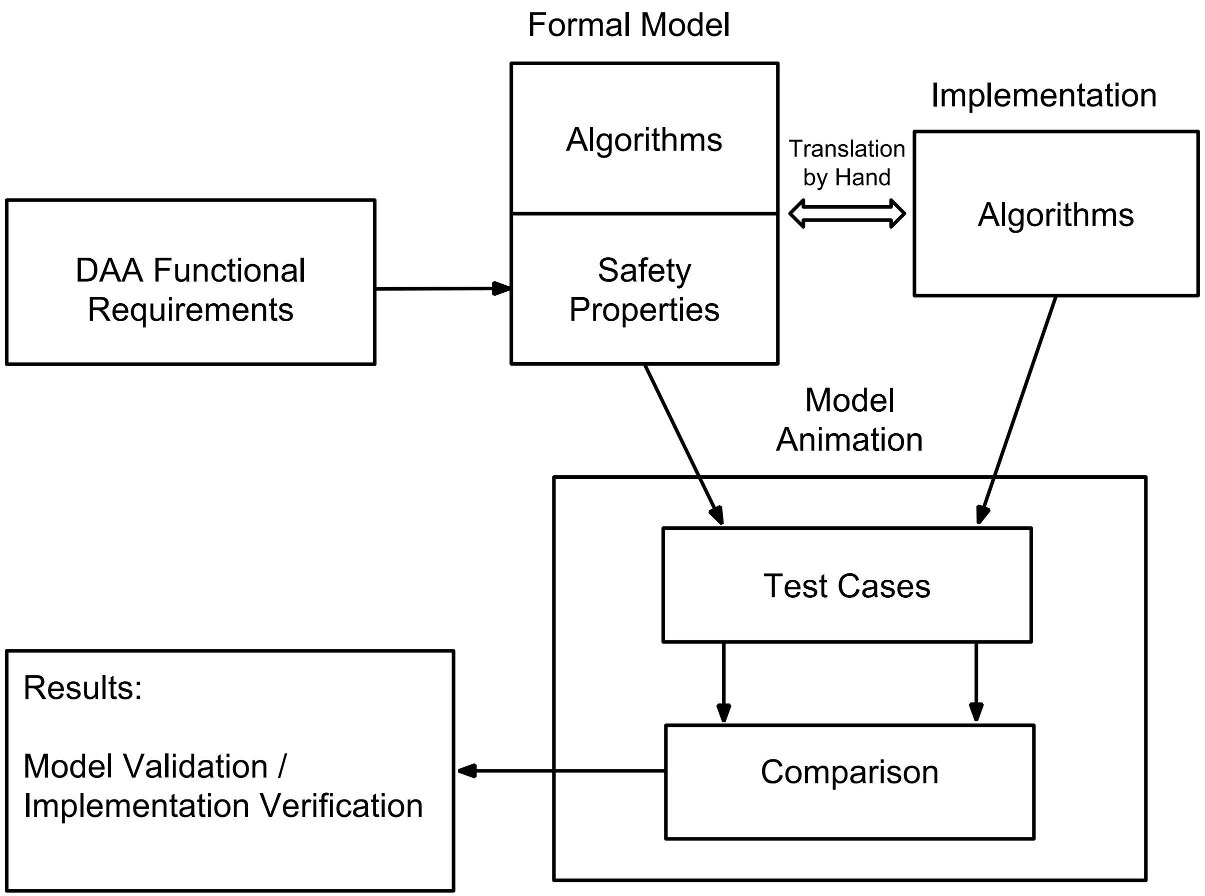
DAIDALUS model in PVS is for verification of core algorithms, not real-world use.

Our user-ready implementations are written in Java and C++.

How do we give assurance that the implementation conforms to the model? With another acronym!

MINERVA

Mirrored Implementation Numerically Evaluated against Rigorously Verified Algorithms



- Is the ***reference implementation*** for Detect-and-Avoid in the US standards document maintained by RTCA special committee 228: DO-365 “Minimum Operational Performance Standards (MOPS) for Detect and Avoid (DAA) Systems.”
- Used by partner organizations for test and simulation, and as a benchmark for DAA capability.
- Integrated into currently in-use DAA solutions (e.g., NAVAIR’s Guardian system).
- As the DAA component of ICAROUS and Danti, NASA–developed software.

What is Compact Position Reporting?

The RTCA/EuroCAE publishes a standards document, DO-260C/ED-102.

**Minimum Operational Performance Standards for
1090 MHz Extended Squitter
Automatic Dependent Surveillance – Broadcast (ADS-B)
and
Traffic Information Services – Broadcast (TIS-B)**

- Describes a collection of algorithms called Compact Position Reporting (CPR) to compress, broadcast and recover the approximate location of an aircraft.
- Reports from pilots and manufacturers say that implementations can be inaccurate.
- FAA official suggested that the Formal Methods team could investigate.

Why use Compact Position Reporting?

Why don’t aircraft just send a GPS location?

- Earth is **BIG**.*[citation needed]*
- Position needs accuracy to be useful.
- The hardware and software used limit message size to 35 bits.

Image Credit: NASA/NOAA/GSFC/Suomi NPP/VIIRS/Norman Kuring
https://www.nasa.gov/sites/default/files/images/618486main_earth_full.jpg

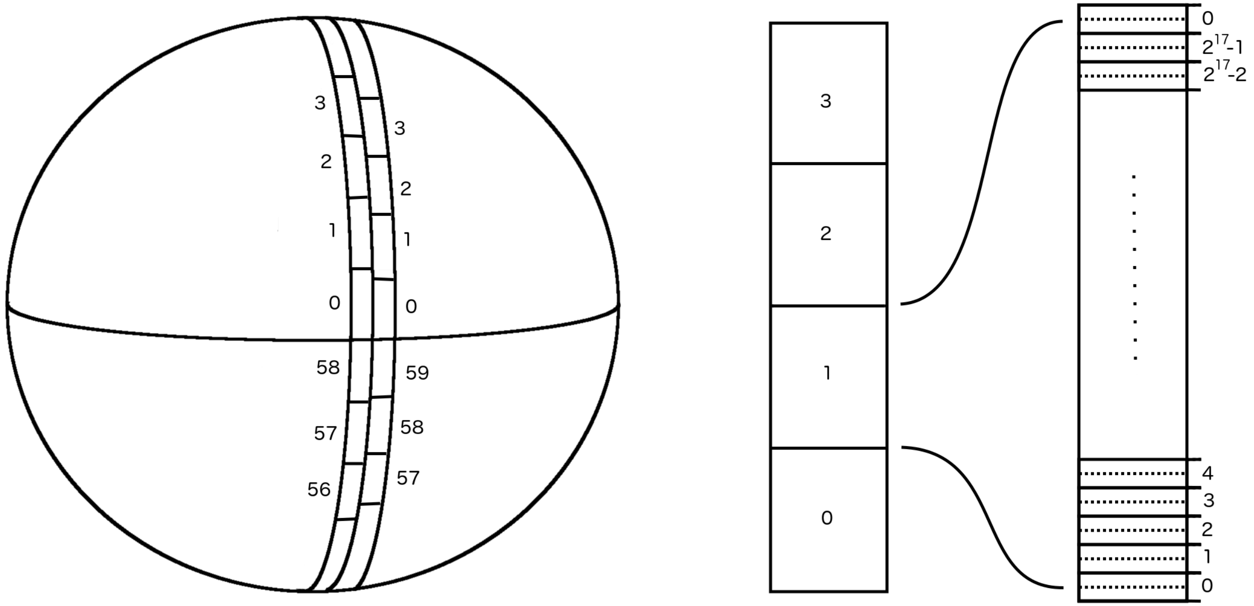
CPR in a nutshell

Focus on Latitude:

- Choose a format (odd or even)
- Divide latitude into equally sized zones (60 for even format, 59 for odd).
- Divide each zone into 2^{17} equally sized bins.
- Recovered position is intended to be the centerline of the bin.

Longitude is handled similarly, but the number of zones used decreases toward the poles.

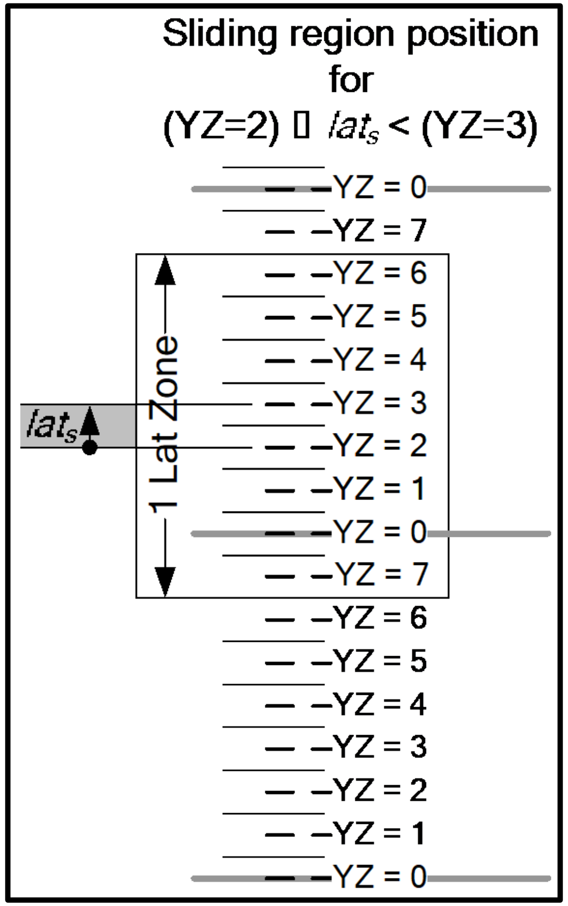
Message structure for CPR position



CPR coordinate system for Latitude encoding

A few further details

- A global decode is used to establish a first position.
- Global decoding does not compute all positions and compare them. It uses the Chinese Remainder Theorem to compute the zone.
- Local decoding can be used after establishing a position.
- There are distance limits for correct decoding:
  - Local Decoding: The reference point used should be *closer* than half of a zone length of the broadcast position.
  - Global Decoding: The pair of positions should be *closer* than half of a zone offset (difference between zone sizes).
- These distance limits are converted to time limits via maximum velocity assumptions.



A “one zone” decoding limit around a reference position for local decoding.

CPR Formal Analysis: Results and by-products

Analysis of the algorithm:

- Created a formal model of the algorithm in PVS (Prototype Verification System).
- Discovered incorrect requirements for decoding, and adjusted the requirements to allow for correct operation.
- Found numerous mathematical simplifications for formulas in the CPR specification, to reduce computational error.
  - Computing transition latitudes:

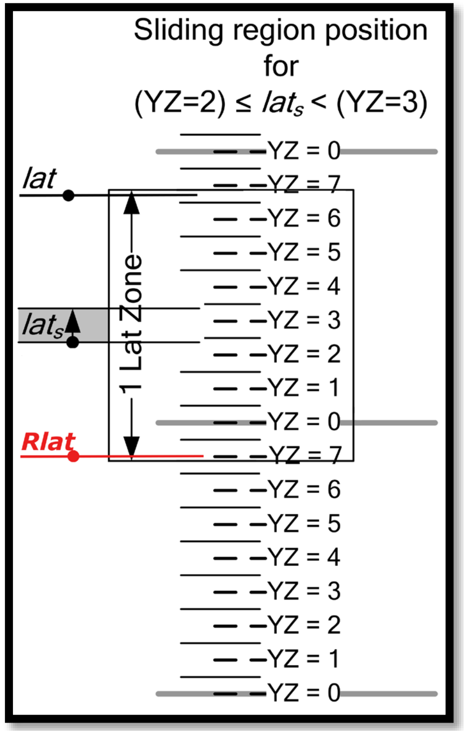
$$\sqrt{\frac{1-\cos(\pi/30)}{1-\cos(2\pi/NL)}} = \frac{\sin(\pi/60)}{\sin(\pi/NL)}$$

  - During encoding:

$$\frac{MOD(A,B)}{B} = \frac{A}{B} - \left\lfloor \frac{A}{B} \right\rfloor$$

  - During Local Decoding:

$$\left\lfloor \frac{lat_s}{Dlat_i} \right\rfloor + \left\lfloor \frac{MOD(lat_s, Dlat_i)}{Dlat_i} - \frac{YZ_i}{2^{17}} + \frac{1}{2} \right\rfloor = \left\lfloor \frac{lat_s}{Dlat_i} - \frac{YZ_i}{2^{17}} + \frac{1}{2} \right\rfloor$$



Previous CPR requirements could decode to an incorrect “zone”, resulting in aircraft being reported hundreds of miles from their actual position.

CPR Formal Analysis: Results and by-products

Analysis of the algorithm:

- Created a formal model of the algo
- Discovered incorrect requirements
  requirements to allow for correct d
- Found numerous mathematical sin
  specification, to reduce computatio
  - Computing transition latitudes:

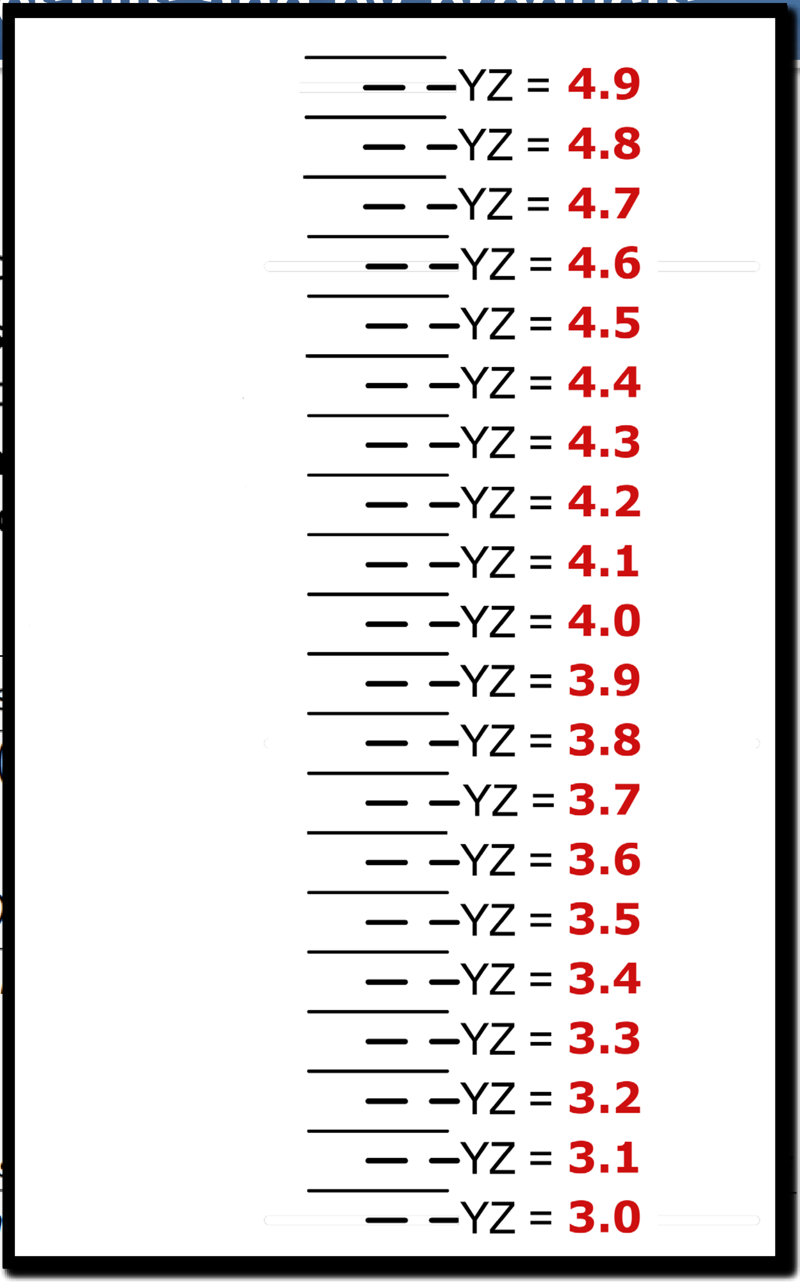
$$\sqrt{\frac{1 - \cos}{1 - \cos(}}$$

  - During encoding:

$$MOD$$

  - During Local Decoding:

$$\left\lfloor \frac{lat_s}{Dlat_i} \right\rfloor + \left\lfloor \frac{MOD(lat}{Dla} \quad + \frac{1}{2} \right\rfloor$$



Previous CPR requirements could decode to an incorrect “zone”, resulting in aircraft being reported hundreds of miles from their actual position.

CPR Formal Analysis: Results and by-products

Analysis of the algorithm:

- Created a formal model of the algo
- Discovered incorrect requirements requirements to allow for correct d
- Found numerous mathematical sin specification, to reduce computati
  - Computing transition latitudes:

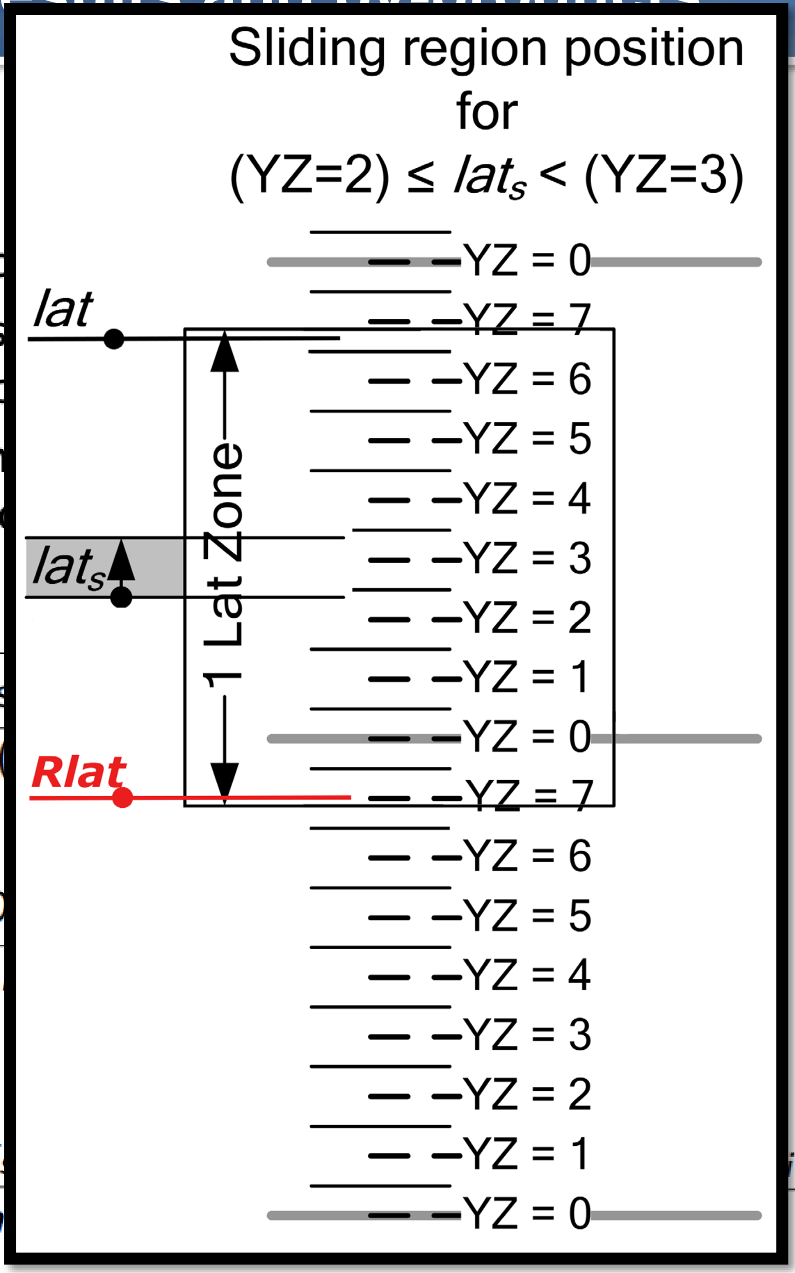
$$\sqrt{\frac{1 - \cos}{1 - \cos(}}$$

  - During encoding:

$$\frac{MOD}{}$$

  - During Local Decoding:

$$\left\lfloor \frac{lat_s}{Dlat_i} \right\rfloor + \left\lfloor \frac{MOD(lat}{Dla} \quad + \frac{1}{2} \right\rfloor$$

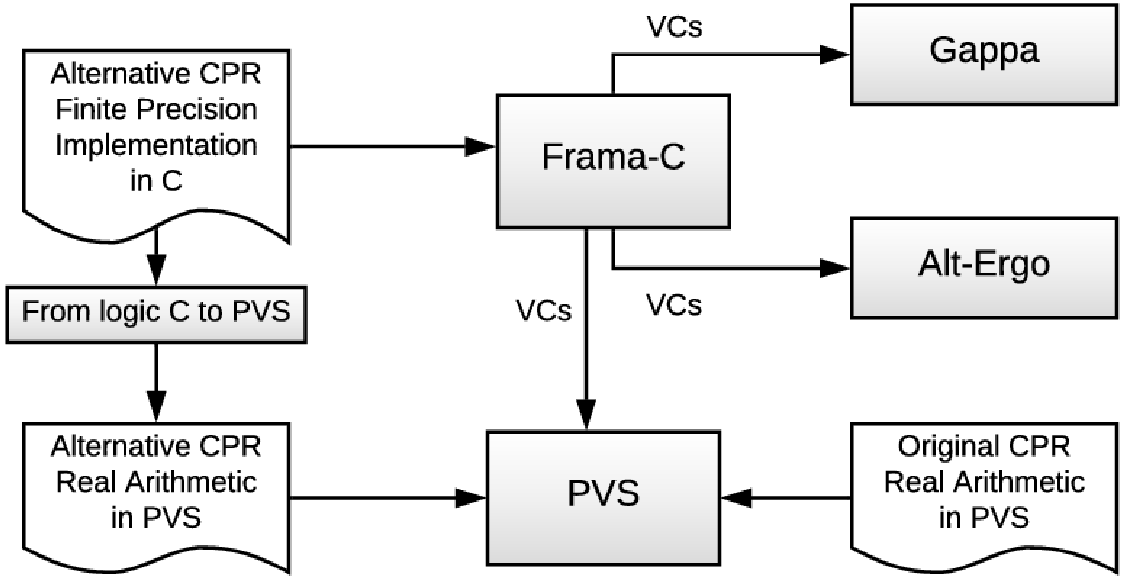


Previous CPR requirements could decode to an incorrect “zone”, resulting in aircraft being reported hundreds of miles from their actual position.

CPR Formal Analysis: Results and by-products

Analysis of CPR implementations

- NASA **developed** and **formally verified** versions of CPR in floating point (double precision) and fixed point (single precision).
- Analysis discovered additional **stressing test cases** that would catch some incorrect implementations of CPR.
- Led to development of **PRECiSA** and **REFLOW,** tools for analysis and rewriting of floating point programs.



The formal verification of CPR implementations involved the use and interaction of several formal methods tools.

The Work in CPR Leads to…

PRECiSA

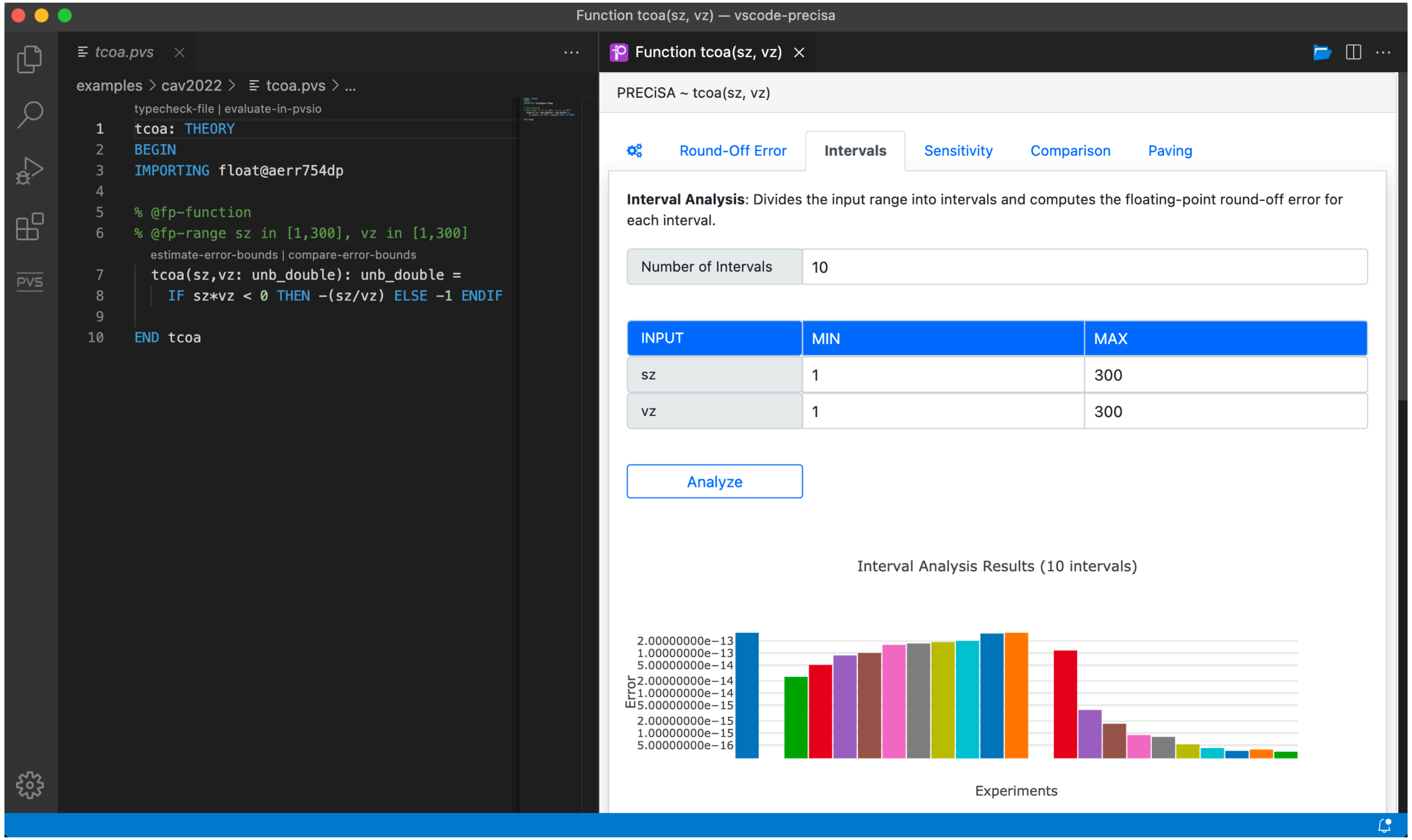
Floating-point Analysis:

- Bounds accumulated error.
- Identifies unstable conditionals.
- Produces verifiable certificates.
- VScode interface for experimentation

REFLOW

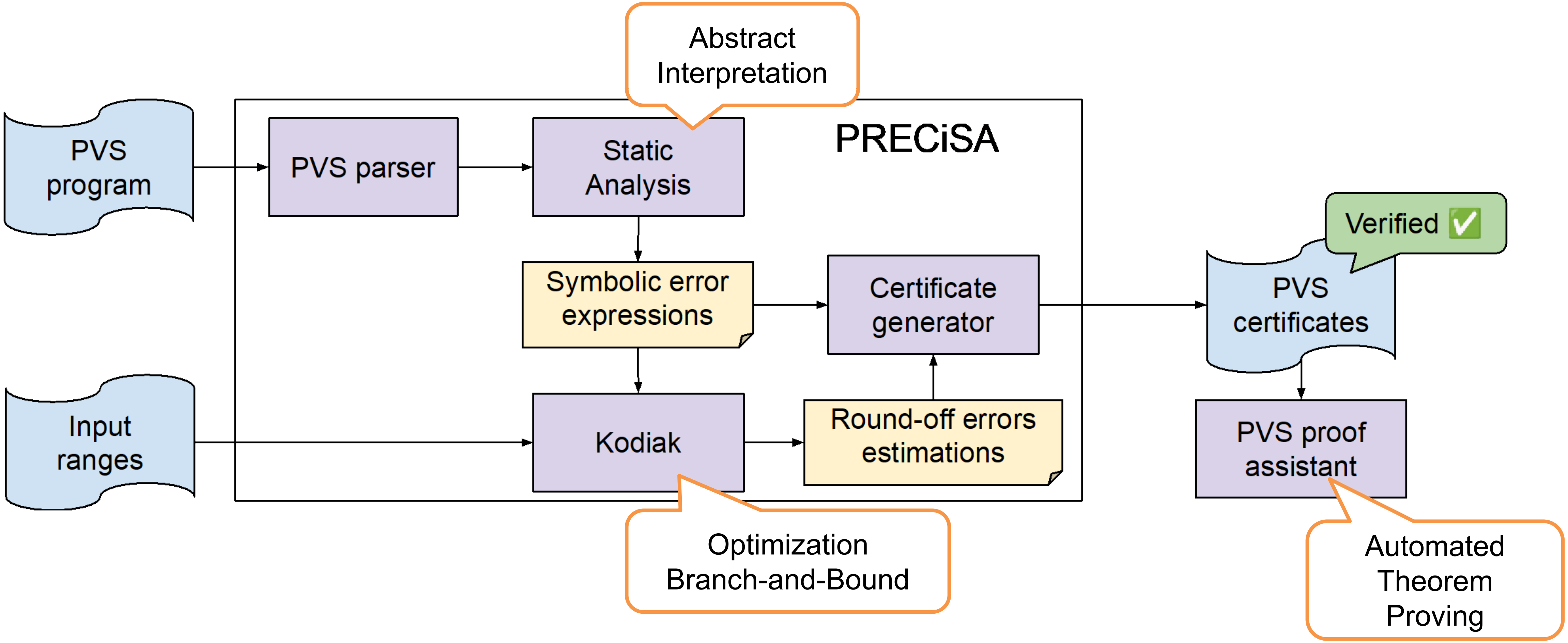
Program Transformation:

- Creates a guard-stable version of a program in C.
- Automatically verifies the correctness of the implementation.

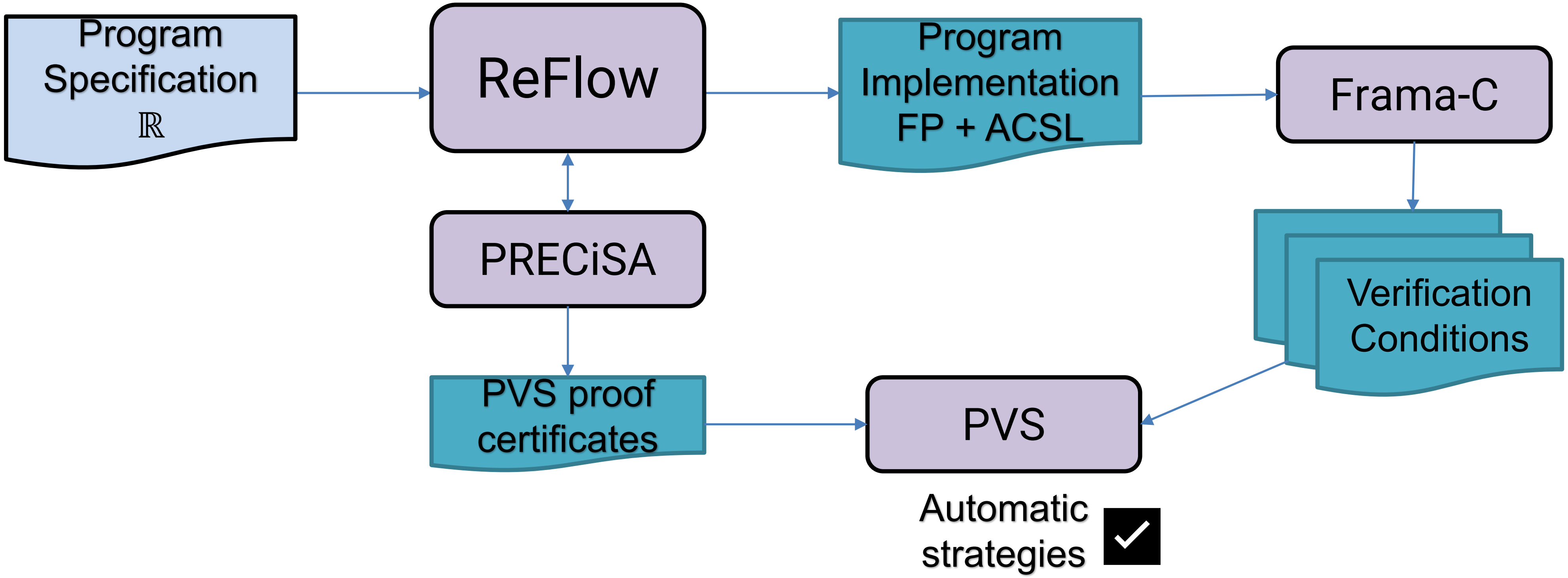


A view of PRECiSA in VScode allowing visual analysis of the accumulated error over a range of values.

Different Formal Methods Techniques Used in PRECiSA



ReFlow



ReFlow takes a real-number specification and generates

- Annotated floating-point C code.
- Proof certificates that verify bounds on error without user interaction.
- Ongoing integration with Herbie will rewriting the original program to lower numerical error.

A Virtuous Cycle

Ongoing development and enhancements to Formal Methods tools.

- PVS libraries and tools
- Floating point analysis
- Runtime assurance
- …

Assuring aerospace applications that are safety- or mission-critical.

- Advanced Air Mobility
- Uncrewed Aircraft Systems
- Air Traffic Management

For more information

Formal Methods as NASA:
https://shemesh.larc.nasa.gov/fm/

DAIDALUS
https://shemesh.larc.nasa.gov/fm/DAIDALUS/
CPR
https://shemesh.larc.nasa.gov/fm/CPR/
NASA PVS libraries
https://shemesh.larc.nasa.gov/fm/pvs/PVS-library/
PRECiSA and REFLOW
https://shemesh.larc.nasa.gov/fm/PRECiSA/

Come work with us!
https://intern.nasa.gov/

A Shameless Plug

Submit your paper!
https://shemesh.larc.nasa.gov/nfm2025/

For more information

Formal Methods as NASA:
https://shemesh.larc.nasa.gov/fm/

DAIDALUS
https://shemesh.larc.nasa.gov/fm/DAIDALUS/
CPR
https://shemesh.larc.nasa.gov/fm/CPR/
NASA PVS libraries
https://shemesh.larc.nasa.gov/fm/pvs/PVS-library/
PRECiSA and REFLOW
https://shemesh.larc.nasa.gov/fm/PRECiSA/

Come work with us!
https://intern.nasa.gov/