

TCAS Standards and OSS

Garfield Dean
EEC TCAS Team

Overview

- What is TCAS?
- A brief history of TCAS Development
- Compare and contrast with OSS
- Where could OSS be used in ATM?

What is TCAS?

- Traffic-Alert and Collision Avoidance System
- Equipment on an aircraft that tells a pilot to climb or descend to avoid collision with other aircraft.
- Uses a type of radar, (called a transponder),
 - measures distance to other aircraft and also gets information on altitude.
- Has a collision avoidance algorithm defined by pseudo-code and state charts.



One type
of ACAS
Display



Mode S
+ACAS
Control
Panel

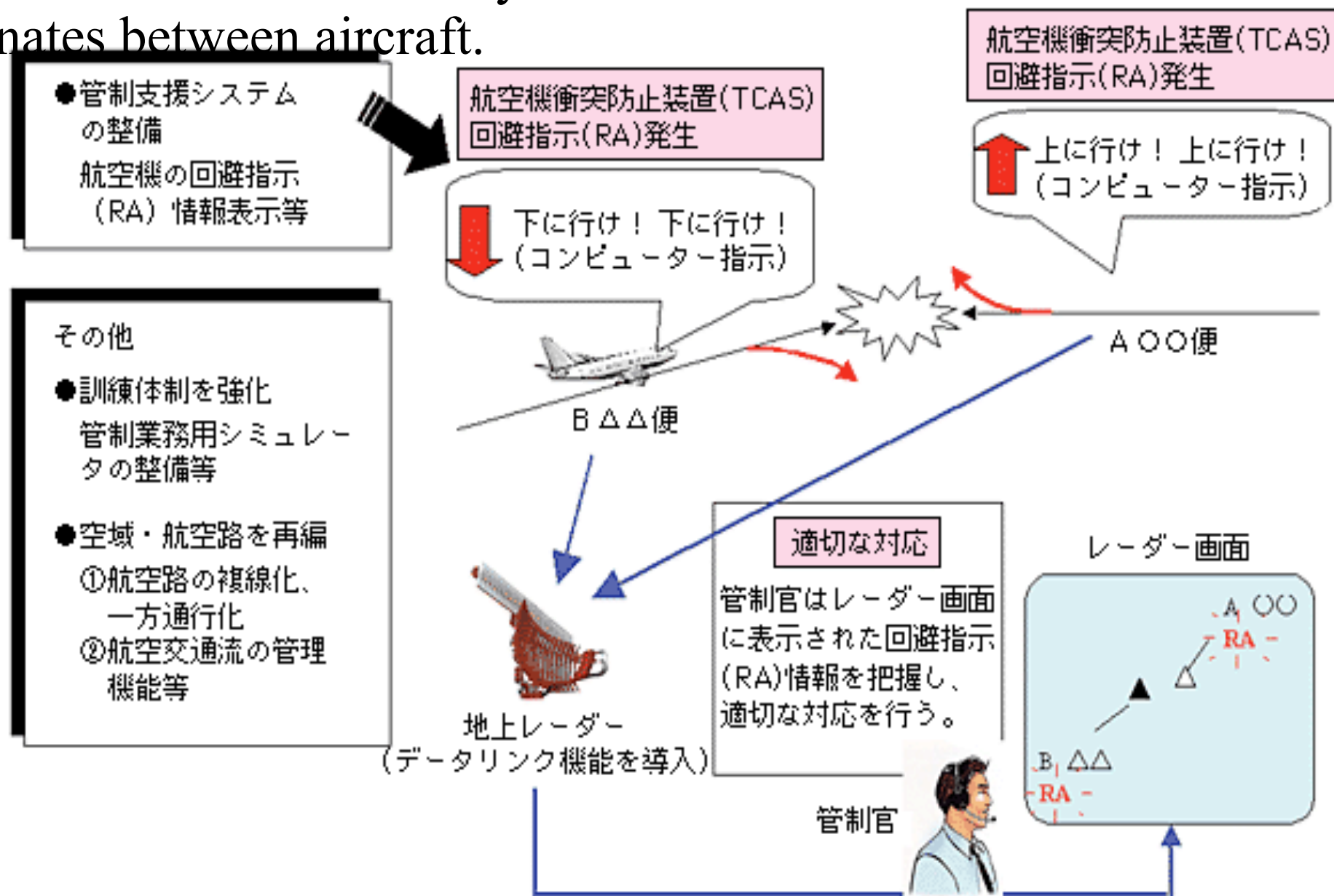
ACAS Black Boxes



How was TCAS developed?

- Initially (1950s - 1970s) just an FAA programme
- Early 1980s congressional mandate for a collision avoidance system
 - This would be put onto aircraft flown worldwide
 - Other countries and organizations become interested.
- Decision by FAA to develop standards
 - For regulators, via ICAO.
 - For manufacturers via RTCA.

TCAS is an International System – and coordinates between aircraft.

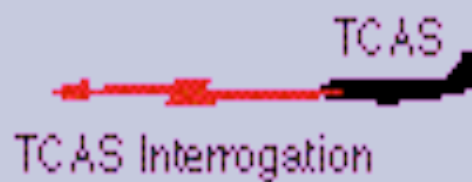


(注) TCAS: 航空機衝突防止装置 / Traffic Alert and Collision Avoidance System

RA : 回避指示 / Resolution Advisory

What are the ICAO ACAS standards?

- Very high level definition of Collision avoidance system.
 - Performance standards (open to different interpretations)
 - Interoperability requirements
 - Says almost nothing about displays
- Decided by committee needing consensus
 - Delegates appointed by ICAO member states/ organizations.
 - Very formal and slow
- Based on what the RTCA design could do



Directional Antenna

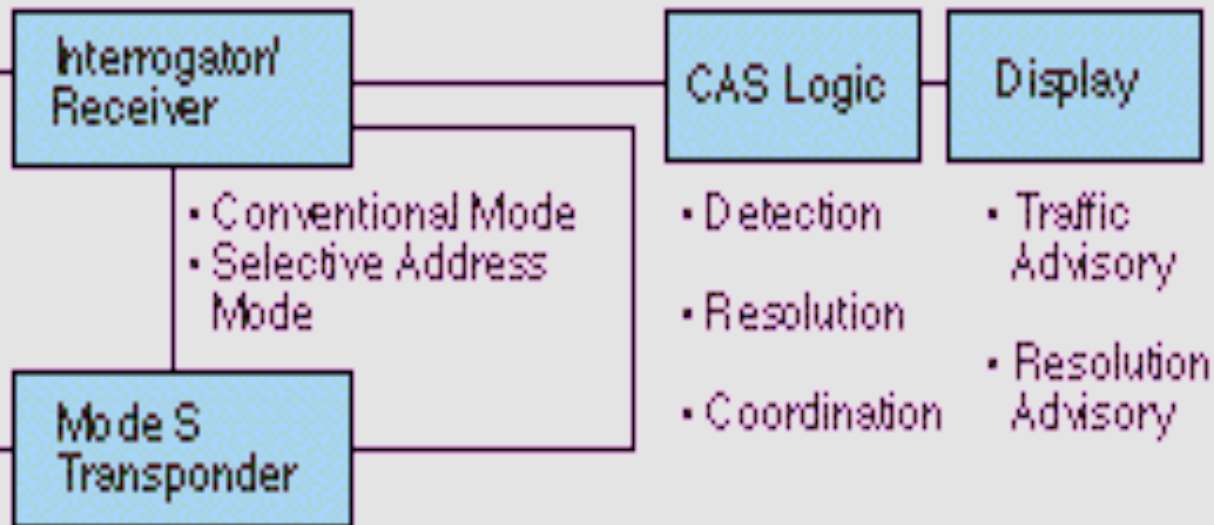
TCAS I: Traffic Advisory Only

TCAS II: Vertical Resolution Advisories (escape maneuvers provided by the TCAS unit to the pilot to solve a conflict)

Air Traffic Control Surveillance System



ATC



Airborne Collision Avoidance System

ACAS Overview

Oxidirectional

What are the RTCA TCAS standards?

- Detailed design requirements
 - Display descriptions
 - Architectural design
- Pseudo-code and State Charts
 - Hierarchically structured
 - Individual manufacturers implement the code
 - Code must pass stringent tests
- Decided by committee needing consensus
 - However, only chairman are appointed
 - Anyone is welcome – many different viewpoints
 - Initially very flexible.
 - Became rigid when widely implemented

Example of Pseudo Code

Example of State Charts

Comparison with OSS - 1

- Pseudo-Code is not code.
 - There are at least 5 different implementations of the same pseudo-code.
 - Therefore stringent testing standards are required
- Most/all of the participants are paid representing many different interest groups
 - Designers
 - Regulators
 - Manufacturers
 - Airlines
 - Pilots
 - Controllers

Comparison with OSS - 2

- The process was/is open for all to contribute
 - Even those who objected to the concept
- Although TCAS is a safety system, it is not a safety critical system
 - Would an OSS version of a safety critical system be acceptable?
- Responsibility for TCAS units is with the manufacturers
 - But the FAA absolves manufacturers of design flaws in the logic and the FAA cannot be sued!
 - Überlingen accident may test this in court.

Comparison with OSS – 3

- You have to pay to get the RTCA documents
 - A few hundred \$
 - Trivial if you are an organization, significant if you are an individual
- Feedback from user experience has been incorporated in upgrades
 - First version was not good (design bug).
 - Initial release + 2 upgrades. 1 or 2 more planned
- Individual manufacturers have added their own features (and mini upgrades)
 - c.f. Red Hat or Suze.
 - These additions are not fed back to RTCA

Responses to Questions - 1

- It's not just the motivation of developers that counts. Budget holders in airlines and ATM authorities still need to be motivated, even with “free” software.
- Hardware can sometimes break the software security symmetry.
- Collaboration facilitated by Teleconferences, Web site/ Wiki, email and regular meetings.
- Build peer review by welcoming criticism, and not taking it personally.

Responses to Questions - 2

- Substantial effort was used to develop TCAS evaluation criteria and software tools, but this was worthwhile. (How do you know you've improved something you can't measure).
- Need to accept many different motivations for working on OSS. Even those trying to break it, probably help.
- There must be an ultimate arbitrator for competing demands.

OSS in ATM?

- Only in non safety-critical systems?
 - Need clear lines of responsibility
 - Does not mean that safety critical software should not be open to review.
- Who will be in and run the steering committee?
- Initial suggestions
 - Safety / performance analysis tools
 - Real-time / Fast-Time simulators
 - Anyone for OpenESCAPE ?