approach independently, the result would be chaotic and dangerous.

By disclosing its intended behavior, an airplane may join the set of aircraft managed by a ground-based system. There is much data to be accumulated, shared, and tracked to avoid possible conflicts. Static information must be uploaded to the plane describing the local terrain, airways, and other airport information. Dynamic information is uploaded as required throughout the flight, including weather, possible warnings, capacity constraints, and special use airspace schedules (e.g., military requirements). Given this information the pilot can produce a flight plan that results in a filed flight trajectory. This can be treated as an object, which will then be used by ground-based systems.

During flight, the pilot may wish to change the flight plans, but can only propose a change that must be approved by the ground-based system before it can be adopted. Furthermore, the actual trajectory is recorded and transmitted by the aircraft, so that the ground-based systems can track it as an object. The accumulation of this airspace data allows traffic density predictions to be calculated, and dynamic route structure objects to be produced [4].

These objects – produced, consumed and manipulated by computers – may be modeled and even implemented through some Object Oriented Technology. There are languages that support these concepts and provide a direct way of manipulating them. The implementation of the free-flight initiative has still not addressed such issues. The FAA is evaluating the problems of Object Oriented Technology.

Object Oriented Technology

There is pressure from industry to use object oriented paradigms in the development of safety critical software. The expectation is that, as in other industry sectors, such programming will lower the development costs. There is some reluctance by regulators to approve this type of programming as it introduces concepts of information hiding, polymorphism, and inheritance. This makes the coupling between code and data less obvious to an auditor. It may invoke run-time support code that creates and destroys these objects dynamically, depending on the scope of the objects during execution. The timing and resource usage of such run-time programs make the application less deterministic, complicating the analysis and approval of such systems. It is expected that ultimately some compromise will be reached and a subset of the object oriented programming paradigm will be adopted, thereby satisfying the concerns of determinism and providing the benefits of this new technology.

Conclusion

Although a number of challenges remain, the industry is very focused on safe air transportation. It is through tremendous vigilance and determinism that the industry has a good safety record. It can be improved, and these on-going initiatives will contribute to safer flight.◆

References

1. DO-178B. Software Considerations in Airborne Systems and Equipment Certification. RTCA, Dec. 1, 1992.
2. AC 25-1309-1A, Advisory Circular, Federal Aviation Administration.
3. DO-248A. Annual Report for Clarification of DO-178B. RTCA, Oct. 6, 1999. (DO-248B to be published in 2001.)
4. National Airspace System Concepts of Operations. RTCA, Dec. 13, 2000.

Resources

- For a complete listing of RTCA documents please see <www.rtca.org>.
- The FAA Flight Standards Service provides links to the regulatory Web sites at the following Web site <www.faa.gov/avr/afs/fars/far_idx.htm>.

About the Author

George Romanski has specialized in the production of software development environments for the past 30 years. Romanski was vice president of Technology at EDS/Scicon, vice president of Engineering at Alsys and director of Safety Critical Software at Aonix. Romanski also serves the safety-critical industry as a member of the HRG (Annex H Rapporteur Group) for the Ada95 ISO standard addressing safety and security issues as well as the Requirements and Technical Concepts for Aviation (RTCA)/SC-190 committee working to provide clarification of DO-178B for avionics and ground-based systems. Romanski is president of Verocel, a company specializing in the verification of software, and in the development of tools that help in this process.

Verocel, Inc.
234 Littleton Road, Suite 2A
Chelmsford, MA 01886
Phone: (978) 392-8860
E-mail: romanski@verocel.com