BASING AVIATION SOFTWARE CERTIFICATION ON ASSURANCE CASES

Patrick J. Graydon
John C. Knight
Elisabeth A. Strunk

Department of Computer Science
University of Virginia
151 Engineer's Way, PO Box 400740
Charlottesville, VA 22904-4740

{graydon | knight | strunk}@cs.virginia.edu

Voice: +1 434.982.2216
FAX: +1 434.982.2214

NATIONAL WORKSHOP ON
AVIATION SOFTWARE SYSTEMS FOR THE SECOND CENTURY OF FLIGHT:
DESIGN FOR CERTIFIABLY DEPENDABLE SYSTEMS

Basing Avionics Certification On Assurance Cases

The availability of ever more powerful computers in ever smaller packages with ever smaller energy requirements has led to tremendous growth in the functionality and complexity of aviation software systems. Knowing that these systems will operate correctly in their intended environments is critical. Nevertheless, the state of the art in software engineering does not provide a technical basis for assurance that the systems we build meet both their functional and dependability requirements.

Many developers claim that their systems meet required dependability goals because they have followed a prescriptive development standard such as RTCA DO-178B. The unspoken argument supporting this claim is that: (1) the software being developed is of a certain type; (2) that developing software of that type in accordance with the given prescriptive standard will result in adequately dependable software; and (3) that the software was developed in accordance with the standard. While many of the techniques required by prescriptive standards are valuable, point (2) in the above argument is generally just an assumption. There is no scientific evidence that following a particular development standard ensures that software will have a particular level of dependability. The reason that this is the case is that, for most specific software techniques, the benefits of using that technique have been shown only for that technique in isolation. For example, system developers might use formal methods for the "critical parts of the system," but are usually unable to evaluate the ensuing effect on the dependability of the system as a whole.

The development, verification, and validation of *software* is often the largest contributor to *system* cost. At the same time, adherence to the prescriptive standard causes developers to undertake activities that might be of little value in demonstrating that the system is adequately safe to use. This problem is particularly apparent in the domain of unmanned aerial systems (UASs) where it is not clear that the techniques appropriate to the development and certification of large, fast, weapon-carrying UASs are equally appropriate to the development of ultralight camera platforms. In addition, it is not clear to what extent existing approaches for manned aircraft are applicable.

To deal with the myriad of software issues facing the avionics community, a transition should be made to the use of *assurance cases* for certification. By this we mean that the *certification criterion* for a specific system should be the presentation of an assurance case that is found to be acceptable by the certifying authority based on a comprehensive and detailed examination.

An assurance case is a comprehensive, rigorous argument that a particular system is fit for its intended use. Assurance cases are in common use in the form of *safety cases*. The assurance case brings together evidence from the development, verification, and validation of the aircraft's systems and the aircraft as a whole into an argument that the aircraft satisfies the goals of airworthiness, including (but not limited to) both *safety* and *security*.

In an assurance case, airworthiness is expressed in a manner independent of the design of any particular aircraft system, permitting developers to take different approaches to building systems with the required properties. A sub-goal associated with a flight-control subsystem, for example, could be met either by a system implemented mainly in hardware or a system implemented with adaptive software components. The assurance case mechanism allows the certification of systems using any development approach so long as the developers are able to demonstrate that the resulting system, as a whole, satisfies the airworthiness goals stated in the assurance case. Since particular techniques are not dictated for development, novel techniques such as those employing non-deterministic or adaptive approaches can be employed provided the developer can provide evidence that the techniques allow the airworthiness and safety goals to be met.

Developers should build the assurance case in parallel with the design and construction of the system. Having an up-to-date (but incomplete) assurance case during system development will allow developers to explore the interaction between their design decisions, the strength of their assurance argument, and the evidence that must be collected to complete the assurance argument. Since development cost will be driven by the system design and V&V costs will be driven by the assurance case, making the trade-off between

design and assurance case visible will allow the system developers to explore designs that minimize total system cost. Thus costs are justified directly by their impact on the airworthiness and safety goals.

Assurance cases as the basis of certification are applicable to all the application domains of interest including manned and unmanned vehicles, ground-based and on-board systems, and aircraft and spacecraft.

To support the use of assurance case technology as the basis for aircraft certification, we must meet three challenges. First, we must determine what the high-level goals of an airworthiness assurance case ought to be. Commercial air transports have prescribed reliability targets, but it is not clear that these targets are an appropriate definition of 'airworthiness' in the context of UASs. Second, we must characterize the assurance argument support that can be derived from existing technologies that we have developed for building, verifying, and validating aircraft systems. With this knowledge in hand, developers can structure their activities so as to minimize cost while assuring certifiers that the systems they develop are airworthy. Finally, we must develop a comprehensive process for assurance case creation, review, and maintenance. This process must make it possible for developers to create assurance cases with a reasonable effort, for reviewers to assess the strength of the argument with a reasonable effort, and for lessons learned from aircraft incidents and accidents to be incorporated into the process to forestall similar events.

In pursuit of these goals, we consider the following to be the three most important challenges that have to be addressed:

- The development of comprehensive and rigorous airworthiness criteria covering safety, reliability, availability and security for all types of vehicle based on the interests of all the stakeholders including the public, the government, regulatory agencies, and aircraft and component manufacturers.
- Characterization of the assurance-case support given by existing development and assurance technologies in the context of aircraft development, and the development of pattern libraries and processes for capturing both useful and fallacious assurance arguments.
- Accident and incident investigation techniques that can extract lessons learned about software development and assurance techniques.

These three challenges dictate a variety of specific research directions that we do not elaborate because of limited space. A roadmap for the next 5 to 10 years should include: (a) development of a preliminary certification approach using assurance cases; (b) creation of full-scale sample certification arguments based on assurance cases for purposes of evaluation; (c) development of training materials for all interested parties including development organizations and certification agencies; (d) assessment of the performance of the approach from the perspectives of the various stakeholders; and (e) development of an approach and schedule for integrating assurance cases in to both civilian and military production processes.

John C. Knight is a professor of computer science at the University of Virginia. His research interests include safety-critical software engineering for aerospace applications and techniques for securing critical networked information systems.

Patrick J. Graydon is a graduate student in the Department of Computer Science at the University of Virginia. His research interests include the use of assurance-case technology in the development of safety-critical systems.

Elisabeth A. Strunk is a research scientist in the Department of Computer Science at the University of Virginia. Her research interests include the use of assurance-case technology and the application of formal techniques in the development of safety-critical systems.