Software System Safety & The NASA Aeronautics Blueprint

C. Michael Holloway; NASA Langley Research Center; Hampton, Virginia, U.S.A.

Kelly J. Hayhurst; NASA Langley Research Center; Hampton, Virginia, U.S.A.

Keywords: NASA, aeronautics research, automation, software safety, naiveté

Abstract

NASA's Aeronautics Blueprint lays out a research agenda for the Agency's aeronautics program. The word *software* appears only four times in this Blueprint, but the critical importance of safe and correct software to the fulfillment of the proposed research is evident on almost every page. Most of the technology solutions proposed to address challenges in aviation are software-dependent technologies. Of the fifty-two specific technology solutions described in the Blueprint, forty-one depend, at least in part, on software for success. For thirty-five of these forty-one, software is not only critical to success, but also to human safety. That is, implementing the technology solutions will require using software in such a way that it may, if not specified, designed, and implemented properly, lead to fatal accidents. These results have at least two implications for the research based on the Blueprint: (1) knowledge about the current state-of-the-art and state-of-the-practice in software engineering and software system safety is essential, and (2) research into current unsolved problems in these software disciplines is also essential.

Introduction

Software is ubiquitous. Automated teller machines, electronic mail, anti-lock braking systems, autopilots, on-line reservations systems, on-line banking, mobile telephones: the list is endless. It is nearly impossible to make it through a day anywhere within the developed world without coming into contact with something containing software. Systems and devices on which modern society depends, depend themselves on software.

The dependence of software is at least as pronounced in the aviation industry as it is in society in general. As David Hughes wrote in an *Aviation Week & Space Technology* commentary a few years ago, "The industry is being transformed from dependence on traditional manufacturing into something that looks more like IBM and Microsoft with wings" (ref. 1). Without making any explicit references to this transformation, the Aeronautics Blueprint (ref. 2) from the National Aeronautics and Space Administration (NASA) provides ample evidence for the reality of the transformation.

This paper is neither a summary nor a critique of the Aeronautics Blueprint. We will neither provide a full description of its contents, nor offer any opinions about its reasonableness or validity. Our purpose in this paper is simply to suggest how critically important software disciplines (particularly software system safety) are to the successful realization of the Blueprint.

The paper is structured as follows. First, the Aeronautics Blueprint is briefly described. Second, the results are presented of our analysis of whether each proposed technology solution within the Blueprint depends on software for success and safety. Third, two implications arising from these results are discussed. Finally, brief concluding remarks are made.

NASA Aeronautics Blueprint

NASA unveiled the Aeronautics Blueprint on February 5, 2002, with a press release titled, "NASA Develops Blueprint to Address Aviation Issues" (ref. 3). NASA leaders believe that the Blueprint accurately conveys the scope and nature of the aeronautics research that the Agency should be conducting in the coming years. Although the team that produced the Blueprint was formed by the previous NASA Administrator, the current Administrator fully endorsed the Blueprint upon its unveiling. The press release quoted him as saying, "The aeronautics blueprint identifies a new and revolutionary technology vision. Working in partnership with the Federal Aviation Administration, the Department of Defense and industry, this blueprint will transform NASA and create the excitement necessary to inspire and develop an engineering workforce that will enable a new era in flight."

To enable this "new era in flight," the Blueprint identifies four particular areas in which NASA will concentrate its aeronautics research: the airspace system, revolutionary vehicles, aviation security and safety, and a state-of-the-art educated workforce. For each of these areas, today's challenges are identified, and technology solutions to these challenges are proposed[1]. These relatively high-level challenges and solutions are grouped into several sub-areas and refined into additional detail.

For example, consider the airspace system area. Four basic current challenges are enumerated: overcome reduced throughput in bad weather, eliminate en route congestion and the 'domino effect' throughout the system, keep pace with demand for arrival and departures at benchmark airports, and increase situational awareness in the system. Each of these challenges constitutes a sub-area of research within the airspace system. These sub-areas are called weather; traffic optimization; high-flow airports; and communications, navigation, and surveillance (CNS). For each sub-area, refined challenges and technology solutions are discussed.

For the weather sub-area, two particular challenges are described: reduce disruptions of en route traffic due to bad weather; and eliminate delays in terminal area airspace by efficiently managing terminal area traffic flow and understanding wake vortex movement and dissipation. Two specific technology solutions are proposed for meeting these challenges: (1) Complete digital knowledge of the en route atmosphere, including precision forecasts, sensors, worldwide measurements, data processing, and information dissemination; (2) Precise local weather forecasts integrated with airport operations, including reliable prediction and conformation of wake vortices integrated with atmospheric conditions. For each of the three other sub-areas, refined challenges and specific technology solutions are also given in the Blueprint.

The revolutionary vehicles research area is also divided into four sub-areas: noise reduction, reduced emissions, improved safety, and enhanced capability. Four sub-areas are identified for aviation safety and security: aircraft hardening, flight procedures and monitoring, surveillance and intervention, and information technology. The educated workforce is divided into two sub-areas: approach to education, and accomplishing the education mission. All told, among the four main areas, fifty-two technology solutions are proposed; the full text for each is given in the appendix. Some duplication exists among the fifty-two (for example, two of the solutions involve "refuse to crash" technology), but for the purposes of this paper, we ignore the duplication, and count each as a separate technology solution.

[1] The terms *today's challenges* and *technology solutions* are not our invention; these terms are used in the Blueprint.

Software and the Technology Solutions

None of the fifty-two technology solutions listed in the Blueprint specifically uses the word *software*. In fact, the word appears only four times in the entire text. Nevertheless, our first reading of the text and associated presentation slides left us with the distinct impression that software and the disciplines associated with it underlay many of the proposed technologies. In an effort to confirm or deny this first impression, we conducted an analysis of the extent to which each of the technology solutions depends on software for successful and safe implementation. This analysis, which is described below, confirmed the validity of our first impression, and convinced us that we had not deceived ourselves (ref. 4).

Analysis Approach: We looked at the specific proposed technology solutions in each of the four main areas described in the Blueprint. For each solution, we considered two questions:

(1) Is software and its associated disciplines critical to the successful implementation of the proposed solution? That is, for the solution to be fully realized, must software be developed and implemented effectively and accurately? An affirmative answer to this question does not mean that software is the *only thing* critical for success, or even that it is the *most* critical thing, but simply that it is *one of* the critical things. A negative answer to this question means that it is possible to successfully implement the proposed solution without developing well-engineered software.

(2) Is software and its associated disciplines critical to the safe implementation of the proposed solution? That is, does the technology solution require using software in such a way that it may, if not specified, designed, and implemented properly, lead to fatal accidents? Similarly to the first question, an affirmative answer to this question does not mean that software is the *only thing* critical for safety, or even that it is the *most* critical thing, but simply that it is *one of* the critical things. A negative answer to this question means that no matter how poorly any software associated with this technology solution is developed, the software will not be able to contribute to fatal accidents.

In determining the answers to these two questions, we tried to err on the side of discounting the importance of software. Whenever we could formulate a plausible argument that software was not necessary for the success or safety of a particular technology solution, we answered the question (or questions) negatively. Thus, we believe that the results presented below, if anything, under-represent the importance of software to the fulfillment of the Aeronautics Blueprint.

It is impossible within the space constraints of this paper to describe the results of our analysis in detail. Instead in the rest of this section, we will simply give a summary of the results. In the next section, we will explain two implications that we believe arise from these results, and present a brief illustrative example of application of these implications to a technology solution.

Software and the Airspace System: Table 1 summarizes the results of the analysis for the Airspace System area. In this table, and in the other tables that follow, the technology solutions are listed in brief; please see the appendix for the full text. Of the fourteen technology solutions proposed in this area, all fourteen depend on software for success, and thirteen depend on software for safety. In other words, without the effective and accurate development of the appropriate software and software-based systems, not a single one of the proposed solutions will succeed.

Table 1 – Software Importance for the Airspace System

Technology Solution	Software critical to success	Software critical to safety
Digital knowledge of atmosphere	Yes	Yes
Precise local weather forecasts	Yes	No
National airspace management	Yes	Yes
Interactive monitoring and goal setting	Yes	Yes
System-level traffic flow decision making	Yes	Yes
Integrated decision-support tools	Yes	Yes
All-weather situational awareness	Yes	Yes
New airport design and operation models	Yes	Yes
Smart non-towered airports	Yes	Yes
Airborne internet	Yes	Yes
Secure networked communications	Yes	Yes
Remote surveillance of all airspace	Yes	Yes
Satellite communications and surveillance	Yes	Yes
Digital broadband communication	Yes	Yes
14 total	14	13

Software and Revolutionary Vehicles: Table 2 presents the results for the revolutionary vehicles area. Seventeen technology solutions are proposed in this area; ten of these depend on software for both success and safety. At least six of them seem based on the premise that one way to improve the safety of air vehicles is to transfer much of the control of the vehicles from the humans in the vehicle to the humans who specify and develop software for the vehicle.

Table 2 – Software Importance for Revolutionary Vehicles

Technology Solution	Software critical to success	Software critical to safety
Improving designs of vehicles	No	No
Intelligent combustors	Yes	Yes
Increased fuel efficiency	No	No
Electric propulsion	No	No
Synthetic vision	Yes	Yes
"Refuse-to-crash" flight controls	Yes	Yes
Human-centered designs	Yes	Yes
Fault detection and reconfigurable systems	Yes	Yes
Self-healing systems	Yes	Yes
Precise knowledge of atmospheric conditions	Yes	Yes
Advanced modeling of air traffic	Yes	Yes
Nanostructures	No	No
Active flow control	No	No
Distributed propulsion	No	No
Electric propulsion	No	No
Integrated advanced control systems	Yes	Yes
Central "nervous system"	Yes	Yes
17 total	10	10

Software and Aviation Security & Safety: The results for the aviation security & safety area are shown in table 3. Of the fourteen technology solutions proposed for this area, twelve depend on software for success; software is also critical to the safety of these same twelve. As with the revolutionary vehicles area, several of these solutions are based on the premise that safety is improved by automation, which means that software developers will be given responsibilities that have traditionally been given to pilots. Even for the two technology solutions for which software does not, in our assessment, play a necessarily critical role (blast-resistant structures and self-extinguishing fuel), extensive use of software seems likely for modeling structural dynamics and fire propagation effects.

Table 3 – Software Importance for Aviation Security & Safety

Technology Solution	Software critical to success	Software critical to safety
Blast-resistant structures	No	No
Reconfigurable avionics	Yes	Yes
Self-healing systems	Yes	Yes
Recoverable computers with virus protection	Yes	Yes
Network intrusion prevention	Yes	Yes
Secure communications	Yes	Yes
Self-extinguishing fuel	No	No
Precise flight path management	Yes	Yes
Advanced modeling and evaluation of air traffic	Yes	Yes
Remote monitoring of flight path	Yes	Yes
Remote audio and visual links	Yes	Yes
"Refuse to Crash" flight systems	Yes	Yes
Real-time passenger threat assessment	Yes	Yes
Aviation security reporting system	Yes	Yes
14 total	12	12

Software and Educated Workforce: The final table (table 4) presents the results for the educated workforce area. None of the seven proposed technology solutions depend on software for safety. In fact, none of the technology solutions in this area have any direct safety dependencies of any kind. Five of the seven, however, depend on software for successful implementation. No one will be killed if the software developed for this area does not function properly; however, quite a few people may be disappointed, and the goals for the area will probably not be achieved without successfully executing software.

Table 4 – Software Importance for Educated Workforce

Technology Solution	Software critical to success	Software critical to safety
Foster interest and excitement in aerospace	No	No
Virtual and collaborative learning environments	Yes	No
Life-long learning system	Yes	No
Long-term partnerships	No	No
Virtual collaborative research laboratories	Yes	No
Workplace virtual classrooms	Yes	No
Adaptive learning computer systems	Yes	No
7 total	5	0

Summary: Adding the numbers from the four tables yields the following results:

Forty-one of the fifty-two technology solutions (nearly 80%) proposed in the Aeronautics Blueprint depend for success, at least in part, on the effective and accurate development and implementation of software; and

For thirty-five of the technology solutions (nearly 70%), the safety of the implemented solution depends, at least in part, on software.

Implications of the Analysis

One need not know much about software to realize that the results just described have implications that affect the implementation of the Aeronautics Blueprint. Two of the most important implications are (1) knowledge about the current state-of-the-art and state-of-the-practice in software engineering and software system safety is essential, and (2) research into current unsolved problems in these software disciplines is also essential.

Software Knowledge is Essential: With nearly 80% of the proposed technology solutions relying on software systems for success, and with nearly 70% of them relying on software systems to ensure safety, the necessity of a broad and deep knowledge of software engineering and software system safety is clear. This knowledge, however, is elusive. The ubiquitous nature of software gives the appearance of maturity in software disciplines. For many common uses of software, the appearance matches the reality. Software disciplines are sufficiently mature to produce many useful systems and products. But for systems and applications with the complexity and criticality of the sort envisioned by the Blueprint, the appearance of maturity is simply an illusion.

Software systems, in general, are among the most complex artifacts that humans produce, and software development projects are among our most complex undertakings. The technology solutions proposed in the Blueprint further extend the degree of complexity beyond anything that has been successfully completed so far. With this complexity comes an increasing array of software and safety problems. Recognizing these problems, and understanding how to, and how not to, approach discovering solutions to these problems will be quite difficult.

Fred Brooks demonstrated over fifteen years ago that searching for a single "magic" solution (a.k.a. silver bullet) for software problems is futile: no one particular approach will be able to solve all the problems of software engineering (ref. 5). Fifteen years later, Edsger Dijkstra argued that software engineering challenges are yet to be met, although "there is widespread belief that computing science as such has been all but completed…This widespread belief, however, is only correct if we identify the goals of computing science with what has been accomplished and forget those goals that we failed to reach, even if they are too important to be ignored. I would therefore like to posit that computing's central challenge, 'How not to make a mess of it,' has *not* been met" (ref. 6).

For these reasons, understanding the limitations of the state-of-the art and practice in software engineering and software system safety will be essential to making wise choices while trying to fulfill the Blueprint. Wise choices in turn will be essential to avoid wasted money, failed projects, and, in the worst case, fatal accidents.

Software Research is Essential: Were all the relevant problems solved in the software disciplines, then knowledge of the current state-of-the-art and practice would be enough. That is, the

dependence on software of a very high percentage of the proposed technology solutions does not, in and of itself, imply that software research is needed. An equally high, or perhaps even larger, percentage of the technology solutions depend on calculus; but new research in calculus is not needed, because all the relevant issues in calculus have already been resolved. Not so with software engineering and software system safety. Many fundamental problems remain unsolved.

These fundamental problems include the following:

- How to efficiently, accurately, and completely determine and specify the requirements that a software system must satisfy.
- How to efficiently, accurately, and completely determine and specify the safety properties that a system must maintain.
- How to validate that the specified requirements and safety properties are the desired ones.
- How to best partition the implementation of requirements and safety properties among software, hardware, and humans.
- How to efficiently, accurately, and with a sufficiently high-degree of confidence verify that a software system satisfies all its requirements and maintains all its safety properties.
- How to demonstrate to others, such as certification authorities, that all necessary verification and validation has been completed.
- How to ensure the integrity and accuracy of all the databases on which a software systems depends.
- When accidents do occur, how to effectively diagnose the software contributions to the accidents, so that future systems will not be susceptible to similar accidents.

For many of these problems, "best practices" that provide partial solutions may exist, and be adequate for many current software systems. However, the Blueprint envisions complex and automated systems, which will require software that is far more complex and pervasive than exists today. Partial solutions to fundamental problems will not be enough.

Take for example, the technology solution *"refuse to crash" flight controls*. For automated flight controls like this to accomplish their purposes (primarily, preventing accidental controlled flight into terrain and intentional crashing of airplanes), both the software requirements and the safety properties must be determined efficiently, accurately, and completely. These requirements and properties must be validated, and implementation of them partitioned among software, hardware, and humans. Efficient and accurate verification must be done to ensure with sufficient confidence that the implemented system meets the requirements and satisfies the properties. The appropriate certification authorities must be convinced that all the necessary verification and validation has been completed. The integrity and accuracy of all the databases on which the flight controls depend must be ensured and maintained. Finally, in the event that an accident occurs, diagnosing the software's contribution, which for a system such as this may be considerable, to the accident must be done. Techniques for accomplishing these things are, at best, immature, and in some cases, non-existent. Thus, software research is essential for the successful fulfillment of the Blueprint.

Concluding Remarks

NASA's Aeronautics Blueprint envisions a future with "on demand as well as scheduled air mobility, not just to hundreds, but to thousands of communities throughout the Nation and the world; traveling where we want, when we want, faster, safer, and with far fewer delays; having access to rural areas, no matter how remote; and having direct access to urban centers, no matter

how congested," using "air vehicles that pass overhead quietly with no emission of objectionable gases into our atmosphere" (ref. 2). Towards making this vision a reality, the Blueprint proposes over fifty technology solutions. In this paper, we have shown that nearly 80% of these solutions depend on software for success, and nearly 70% depend on software to ensure safety. An army may, as Napoleon Bonaparte is reputed to have said, travel on its stomach; but if NASA's Aeronautics Blueprint is fulfilled, society will most certainly travel on its software. For this travel to be possible, and to be safe, knowledge about the current state-of-the-art and state-of-the-practice in software engineering and software system safety is essential, and research into current unsolved problems in these software disciplines is also essential.

Appendix: Listing of Technology Solutions

The following is a full listing of the exact text of the specific technology solutions from the Aeronautics Blueprint, divided according to the sub-areas in which each is listed.

Airspace System: Weather

- Complete digital knowledge of the en route atmosphere, including precision forecasts, sensors, worldwide measurements, data process, and information processing
- Precise local weather forecasts integrated with airport operations, including reliable prediction and conformation of wake vortices integrated with atmospheric conditions

Airspace System: Traffic Optimization

- National airspace management through system level traffic planning and management
- Interactive monitoring and goal setting
- System-level (en route and local) traffic flow planning and decision making

Airspace System: High-flow Airports

- Integrated arrival, departure, and surface decision-support tools
- All-weather situational awareness and response
- New airport design and operation models
- Smart non-towered airports with autonomous sequencing and scheduling of aircraft

Airspace System: CNS

- Airborne internet
- Secure networked communications
- Remote surveillance of all airspace
- Satellite communications and surveillance
- Digital broadband communication

Revolutionary Vehicles: Noise Reduction

- Improving the design of engines, landing gear, and airframes through better understanding of the sources of noise, integrating emerging materials, structures, and flow-control technologies, and developing revolutionary vehicle designs

Revolutionary Vehicles: Reduced Emissions

- Intelligent combustors that use smart materials, and sensors and actuators to control the combustion process
- Increased fuel efficiency through the use of ultra-lightweight aircraft, dual-fan engines, and distributed propulsion
- Electric propulsion from fuel cells and global hydrogen generation

Revolutionary Vehicles: Improved safety

- Synthetic vision to provide visibility in all conditions
- "Refuse-to-crash" flight controls with digital terrain technology
- Human-centered designs

- Fault detection and reconfigurable systems
- Self-healing systems
- Precise knowledge of atmospheric conditions
- Advanced modeling of air traffic to identify and minimize risk

Revolutionary Vehicles: Enhanced Capability

- Nanostructures that are 100 times stronger than steel with 1/6 the weight
- Active flow control
- Distributed propulsion
- Electric propulsion, advanced fuel cells, and high-efficiency electric motors
- Integrated advanced control systems and information technology
- Central "nervous system" and adaptive vehicle controls

Aviation Security & Safety: Aircraft Hardening

- Blast-resistant structures which can withstand damage and allow safe landings
- Fault detection and reconfigurable avionics
- Self-healing systems
- Recoverable computers with virus protection
- Network intrusion prevention
- Secure communications
- Self-extinguishing fuel

Aviation Security & Safety: Flight Procedures and Monitoring

- Precise flight path management including complex curved approaches and four-dimensional approaches
- Advanced modeling and evaluation of air traffic to identify and minimize risk
- Remote monitoring of flight path conformance to identify deviations and enable rapid intervention

Aviation Security & Safety: Surveillance and Intervention

- Remote audio and visual links to the aircraft cabin and cockpit
- "Refuse to Crash" flight systems that can correct pilot error and prevent sabotage

Aviation Security & Safety: Information Technology

- Real-time passenger threat assessment from reservation to boarding
- Aviation security reporting system that includes anonymous submission of security incidents and data mining to identify trends

Educated Workforce: Approach to Education

- Foster interest and excitement in aerospace-establish an exciting vision for aeronautics
- Stimulate curriculum change and virtual and collaborative learning environments that will enhance educational relevance and scope
- Create life-long learning system that links classrooms to laboratories and on-the job experiences

Educated Workforce: Accomplishing the Enterprise Mission

- Develop long-term partnerships between government, universities, and industry research entities
- Create virtual collaborative research laboratories working on multi-discipline projects
- Workplace virtual classrooms support lifelong and advanced distributed learning
- Adaptive learning computer systems for access to global scientific and technology knowledge

References

1. David Hughes. "Informational Technology: This Changes Everything." *Aviation Week & Space Technology*. December 21/28, 1998.

2. National Aeronautics and Space Administration. "NASA Aeronautics Blueprint: Towards a Bold New Era in Aviation." Washington, D.C., 2002. Available from <http://www.aerospace.nasa.gov/aero_blueprint/index.html> as of 7 April 2003.

3. National Aeronautics and Space Administration. "NASA Develops Blueprint to Address Aviation Issues, Release 02-23." Press release, Washington, D. C., 2002. Available from <http://www.nasa.gov/formedia/MP_Archive_02.html> as of 7 April 2003.

4. Gregory L. Bahnsen. "A Conditional Resolution of the Apparent Paradox of Self-Deception." Ph.D. dissertation., University of Southern California, 1978.

5. Fredrick P. Brooks. "No Silver Bullet: Essence and Accidents of Software Engineering." *IEEE Computer* 20, no. 4 (1987): 10-19.

6. Edsger W. Dijkstra. "The End of Computing Science." *Communications of the ACM* 44, no. 3 (2001): 92.

Biographies

C. Michael Holloway, NASA Langley Research Center, MS 130 / 100 NASA Road, Hampton VA 23681-2199, USA, telephone – 1.757.864.1701, email – c.m.holloway@larc.nasa.gov

C. Michael Holloway is a senior research engineer at the NASA Langley Research Center in Hampton, Virginia. The interests he is paid to have include accident analysis, software system safety, and foundations for high-integrity software development techniques. His real interests include theology, epistemology, Constitutional law, history, volleyball, baseball, and roller coasters. Mr. Holloway has a B.S. in computer science from the University of Virginia, and completed all-but-dissertation towards a Ph.D. from the University of Illinois. He is a member of the IEEE, the IEEE Computer Society, and the System Safety Society. Mr. Holloway is married and has two children.

Kelly J. Hayhurst, NASA Langley Research Center, MS 130 / 100 NASA Road, Hampton VA 23681-2199, USA, telephone – 1.757.864.6215, email – k.j.hayhurst@larc.nasa.gov

Kelly J. Hayhurst is a senior research engineer at the NASA Langley Research Center in Hampton, Virginia. Since 1988, she has worked extensively with the FAA on research projects involving verification and certification issues for aviation software. Kelly received her B. A. degree in Mathematics from Virginia Tech and M. A. degree in Mathematics and Operations Research from the College of William and Mary.