NASA/TM-2014-218670

Is Model-Based Development a Favorable Approach for Complex and Safety-Critical Computer Systems on Commercial Aircraft?

Wilfredo Torres-Pomales
Langley Research Center, Hampton, Virginia

December 2014

NASA STI Program . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA scientific and technical information (STI) program plays a key part in helping NASA maintain this important role.

The NASA STI program operates under the auspices of the Agency Chief Information Officer. It collects, organizes, provides for archiving, and disseminates NASA's STI. The NASA STI program provides access to the NTRS Registered and its public interface, the NASA Technical Reports Server, thus providing one of the largest collections of aeronautical and space science STI in the world. Results are published in both non-NASA channels and by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA Programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counter-part of peer-reviewed formal professional papers but has less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services also include organizing and publishing research results, distributing specialized research announcements and feeds, providing information desk and personal search support, and enabling data exchange services.

For more information about the NASA STI program, see the following:

- Access the NASA STI program home page at http://www.sti.nasa.gov

- E-mail your question to help@sti.nasa.gov

- Phone the NASA STI Information Desk at 757-864-9658

- Write to:
  NASA STI Information Desk
  Mail Stop 148
  NASA Langley Research Center
  Hampton, VA 23681-2199

NASA/TM-2014-218670

Is Model-Based Development a Favorable Approach for Complex and Safety-Critical Computer Systems on Commercial Aircraft?

Wilfredo Torres-Pomales
Langley Research Center, Hampton, Virginia

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23681-2199

December 2014

Available from:

NASA STI Program / Mail Stop 148
NASA Langley Research Center
Hampton, VA 23681-2199
Fax: 757-864-6500

Abstract

A system is safety-critical if its failure can endanger human life or cause significant damage to property or the environment. State-of-the-art computer systems on commercial aircraft are highly complex, software-intensive, functionally integrated, and network-centric systems of systems. Ensuring that such systems are safe and comply with existing safety regulations is costly and time-consuming as the level of rigor in the development process, especially the validation and verification activities, is determined by considerations of system complexity and safety criticality. A significant degree of care and deep insight into the operational principles of these systems is required to ensure adequate coverage of all design implications relevant to system safety. Model-based development methodologies, methods, tools, and techniques facilitate collaboration and enable the use of common design artifacts among groups dealing with different aspects of the development of a system. This paper examines the application of model-based development to complex and safety-critical aircraft computer systems. Benefits and detriments are identified and an overall assessment of the approach is given.

1. Introduction

For decades now, the commercial aviation industry has seen a trend of adopting ever more sophisticated computer-based technology to realize aircraft systems that perform a wide variety of functions with different safety criticality levels. State-of-the-art aircraft avionics are highly complex, functionally integrated, network-centric systems of systems [1]. The design and analysis of embedded computer systems like the ones used on commercial aircraft are inherently complex activities. Ensuring that such systems are safe and comply with existing airworthiness regulations is costly and time-consuming as the level of rigor in the development process, especially the validation and verification activities, is determined by considerations of system complexity and safety criticality. A significant degree of care and deep insight into the operational principles of these systems is required to ensure adequate coverage of all design implications relevant to system safety.

The main drivers for the evolution of avionics architectures on commercial aircraft are the competition among airlines in the air travel market and the competition among airplane manufacturers to meet the demands from the airlines for more fuel-efficient and cost-effective airplanes that have more functionality and a higher level of functional sophistication [2, 3]. Over time, functionality has been added to improve airplane flight performance and safety, as well as to improve maintenance and passenger comfort. The greatest enabler of the evolution in avionics architectures has been advancements in electronics and computer technology, including microprocessors, operating systems, data networking, sensors, displays and design development tools [4]. As technology has improved, the cost of electronic hardware components has decreased, but so has their life-cycle duration and time to obsolescence. Simultaneously, the ever-increasing functional complexity is being realized mostly in software, and this has prompted greater interest in ways to simplify software development. System cost and considerations of hardware part obsolescence and software reuse have driven system developers toward layered and modular designs with standardized interfaces and the use of generic hardware and software commercial-off-the-shelf (COTS) components where practicable. As functionality has increased, software development and system integration have become the primary cost factors.

In addition to functional requirements, aircraft avionics systems must satisfy demanding quality requirements for performance, dependability and safety [5]. The main goal behind the system safety requirements is to ensure an acceptable and rational inverse relation between the probability and severity of functional failures [6]. These quality requirements must be satisfied under stated operational and environmental conditions.

A system is said to be complex when its operation, failure modes or failure effects are difficult to comprehend without the aid of analytical methods [7]. Most of the complexity in modern aircraft systems stems from requirements for high functional quality while protecting against potential failures due to physical or logical defects, or misuse [1]. The complexity of some systems is such that they cannot be analyzed and understood well enough to be managed effectively by any one individual or small group. In general, complex interactions between components (including both hardware and software components) have a higher potential for execution errors. This threat can be aggravated by coupling between components that allows the propagation of fault effects along paths of data and control information flow. With complex integrated systems, there is also the possibility of unintended coupling through shared resources and the environment in which the systems operate. Uncertainty about the interactions and coupling between components in complex computer-based aircraft systems, especially under failure conditions, is a recognized point of concern for certification authorities because of the possible safety implications [8].

The aviation industry and regulation authorities have developed many guidance documents on recommended practice for the development and certification of aircraft systems. A primary concern in the development of complex safety-critical systems is the generation of evidence to substantiate, to an adequate level of confidence, that errors in requirements, design, and implementation of a system have been identified and corrected and that any possible effects of residual errors are mitigated by the system architecture. To a large extent, current development practice is centered on process and product documents to manage the execution of the project and generate the required assurance data.

A model-based development (MBD) approach shifts the focus from the generation of documents to the generation of an integrated repository (or database) of project and product information (e.g., development schedules, design models, analysis models, etc.) used by all disciplines involved in a project [9]. A rigorous development process is still required for successful development, but the effort is directed mostly toward developing and integrating content into the central project repository rather than generating documents, which instead are automatically generated from the content of the project repository. There are many benefits to a model-based development concept, including the use of common design artifacts and the facilitation of collaboration among participants of a development effort.

The following sections present an overview of the development of safety-critical embedded computer systems for commercial aircraft, followed by an overview of the model-based development concept. This is followed by an analysis of benefits and detriments of model-based development. The paper ends with an overall assessment of the approach and conclusions drawn from the analysis.

2. Development of Safety-Critical Computer-Based Systems

The development of an aircraft begins with a concept generation phase that defines the operational concept (CONOP) (i.e., a context and how the vehicle will operate in it) and the overall characteristics of the vehicle (e.g., size, range, performance, etc.) [7]. The next step is to generate aircraft-level requirements, including the definition of the aircraft functional architecture consisting of behaviors such as flight control, navigation, ground steering, and engine control. The functional architecture is allocated to a system-level physical architecture with the resources needed to execute the functions. The system architecture defines a structure of components that has the performance and dependability attributes (e.g., reliability, integrity, etc.) required to meet the demands of the application. The development of the system architecture generates the functional and non-functional (i.e., quality) attribute requirements for the hardware and software components that will realize the system. A validation process runs in parallel with the requirements generation, design, and decomposition processes to ensure that the requirements at each level are complete and correct relative to the top-level CONOP and intended vehicle functionality. After the hardware and software items are designed and implemented, the formal process of verification and integration begins. Here the items are integrated into sub-systems, which are then integrated into full systems. The verification process proceeds in parallel with the integration process and design errors are corrected as they are discovered. Overall, the development flow consists of three parallel tracks of project management, development, and quality assurance to deal with all aspects of technical performance, cost, schedule, and development risk [9].

The definition and implementation of the development process for aircraft systems is influenced by considerations of safety criticality. A system is safety critical if its failure can endanger human life or cause significant damage to property or the environment. Safety is assessed relative to the level of operational risk, where risk is the combination of the likelihood (i.e., the frequency) of safety-relevant events and the corresponding level of severity. The overarching goal in the development of a safety-critical system is to ensure an inverse relationship between the likelihood and severity of system failures. In effect, system safety is determined not just by whether a system will fail, but also how it fails (i.e., its failure modes and effects) and the corresponding likelihood of occurrence. A system can fail due to development faults (i.e., errors or defects introduced at the time of development) and operational faults caused by internal physical conditions or the external environment (e.g., wear-out; lightning; excessive voltage, temperature, or vibration). It is the activation of these faults and the propagation of effects that can ultimately lead to the failure of the system. Safety analysis methods such as fault tree analysis (FTA), failure modes and effects analysis (FMEA), Markov analysis and common cause analysis are performed on the design of a system, especially its architecture, to assess its safety-relevant characteristics [10]. From the definition of safety in terms of risk, we can see that system safety constraints can be satisfied by controlling the severity of failures (i.e., the failure modes), their likelihood, or both. Architecture-level fault tolerance techniques (e.g., redundancy, independence, detection, isolation, recovery) can be applied to mitigate the effects of physical and design faults [11]. The failure rate of physical components can be controlled by proper selection of the quality of the components and the environment in which the system operates. Controlling the likelihood of design errors, however, requires carefully planned and systematic actions in the development process to ensure that errors in requirements, design, and implementation have been identified and corrected.

The system development model described above is applicable whether the system is simple or complex. A system is simple if proper functioning and performance can be established by a combination of tests and analysis [12]. A system is complex when analytical methods and structured assessments are needed to comprehend the operation, failure modes, or failure effects. System complexity can be caused by characteristics such as the sophistication of the hardware and software components (items, units) and the number and intricacy of interactions between them [12]. In general, the likelihood of residual development defects in simple systems is remote or their failure modes can be sufficiently well understood such that their effects can be adequately mitigated. By definition, presently available test and analysis methods and techniques cannot feasibly establish the absence of defects in a complex system and significant uncertainties in the number, nature, and manifestations of defects can remain. This product-based approach must be complemented with a process-based approach to generate evidence that can substantiate with an adequate level of confidence that design errors have been identified and corrected and the system satisfies applicable regulations and policies. Development assurance frameworks have been created that specify the required level of rigor in a development process based on the worst-case severity of a system function failure [7]. A higher development assurance level is needed for systems with worse failure conditions. A comprehensive development for complex safety-critical systems includes processes for planning, requirements capture, safety assessment, development, implementation, validation, verification, configuration management, process assurance, and certification and regulatory authority coordination, all of which have multiple objectives and generate evidence documentation that must be carefully reviewed. The amount of product and process evidence that must be generated and submitted for certification credit is less for systems with lower safety criticality.

3. Model-Based Development

A model is a collection of one or more artifacts that represent a concept. When the concept is a system, the model is an abstraction whose form and content are chosen for the purpose of understanding, communicating, explaining, or designing aspects of interest of the system [13]. The scope, depth, and fidelity of a model must be chosen to fit the purpose. Models can be descriptive or analytical, and they can capture static properties of a system (e.g., hierarchical decomposition, interconnection) or dynamic properties (e.g., behavior). Characteristics of successful system models include:

- providing a framework to attack a problem in a coherent and consistent manner;
- having the power to show that a solution satisfies the needs of the stakeholders;
- providing integrity and consistency to the system; and
- providing insight into the problem and comparative advantages of different possible approaches and solutions.

Suitably chosen models can be used to capture operational, system and component concepts for any stage of the system life cycle.

Model-based development is the formalized application of modeling to support system development processes including generation of requirements; specification and design at mission, operational, functional, and physical levels; and validation and verification of components, their integration, and the system as a whole. Model-based approaches for validation and verification can be applied to the analysis, test, and review of the system. This development approach uses an integrated model repository and can support the whole development effort including modeling artifacts from disciplines working on different aspects of the system. Automatic means can be used to ensure traceability between source documents (e.g., descriptions of stakeholder needs, technical standards, and recommended practices), system requirements, design, and verification. Automatic means can also be used to propagate changes across development artifacts in the integrated repository, and to perform a broad range of checks to support reviews. The model repository and development support tools can be used for process activities such as requirements capture and validation, configuration management, development process assurance, and coordination with certification and regulatory authorities. Model-based development also allows for the possibility of automatically synthesizing software or hardware implementation code from the models, thus reducing the likelihood of implementation errors and increasing productivity.

Two critical elements in the use of models are the modeling language and the methodology used to solve problems. The modeling language should be specific to the problem domain to which it is applied. For example, there can be special languages for requirements and design for both behavior and structure, and languages for the various development disciplines including system, mechanical, electrical, digital electronics, and software. The languages should enable clear and precise definitions of concepts and their properties, as well as the relationships between components [14].

A methodology is a high-level problem-solving approach supported by a collection of processes, methods and tools [15]. A process is a particular sequence of tasks performed to achieve a particular objective, a method consists of the techniques to perform a task, and a tool is an instrument to enhance the efficiency of a task. Some examples of model-based system engineering methodologies include INCOSE Object-Oriented Systems Engineering Method (OOSEM), IBM Rational Unified Process for Systems Engineering (RUP SE), and ViTech Model-Based Systems Engineering Methodology [15].

Model-based development offers a number of benefits over traditional document-based approaches. The ability to develop complex systems is enhanced by the use of validated abstract system models at multiple levels that require less effort to generate and that focus on irreducible inherent complexity aspects, as opposed to accidental complexity that can obscure critical aspects of a system. Other benefits include:

- increased development effectiveness through integrated multi-disciplinary analysis and design;
- increased efficiency of development time and cost through quicker capture and validation of requirements, reduced reliance on physical prototypes, reuse of models, automatic generation of design artifacts and documents, and less rework to correct errors;
- reduced development risk through higher predictability in program cost and schedule, as well as final system performance; and
- enhanced maintainability of system design by leveraging the central integrated repository for change impact analysis and trade studies.

Some of the detriments of model-based development include:

- the need for investment in development tools and model management infrastructure;
- technical models can hinder communication with stakeholders;
- the approach does not preclude the need for a rigorous development process; and
- a high degree of skill and experience may be needed to validate the models.

4. Application of MBD to complex and safety-critical aircraft systems

Modern computer-based aircraft systems have physically distributed and functionally integrated architectures that depend on the successful integration of contributions from different points of view such as system engineering, control engineering, mechanical engineering, electrical engineering, computer hardware and software engineering, as well as the system users. Model-based development implies a change in focus from a document-centered approach to a model-centered approach. As stated above, there are many benefits to this change in focus, but the most important consideration of any approach for complex and safety-critical systems is the ability to achieve high development assurance.

Feiler has identified two main points of concern in the development of modern aircraft systems [16]. One of these problems is the potential for multiple truths in the results of system analyses. Loose coupling among system development teams and between development and analysis activities can lead to inconsistencies between models of different aspects of a system and also between the system being developed and the one captured in analyses.

The second main problem of concern in the development of complex and safety-critical systems is the introduction of errors early in the development process and their discovery much later during the integration of components or system-level testing. This is a concern because, in general, the cost of correcting errors increases exponentially with the distance in the development process between the point where they are introduced and the point where they are identified [16]. This is because both the breadth and depth of implications of design decisions increases as the development process advances.

One significant source of errors and inconsistencies are mismatched assumptions about different aspects of a system. Assumptions of different sorts are leveraged to simplify and bound the design and analyses efforts, but for highly complex systems, successful development critically depends on the use of a consistent set of assumptions by everyone involved.

Another source of development errors is the inability to identify and understand all the implication of design assumptions and decisions. The difficulty in doing a thorough and precise examination increases with the complexity of the system.

Based on the description in the previous section, model-based development approaches with disciplined implementation of processes and methods, supported by suitable modeling and model management tools, enable a high level of confidence that errors of the sort mentioned here are highly unlikely to remain undiscovered at the completion of the development effort. Furthermore, frequent and automatic validation and verification activities facilitated by the comprehensive application of MBD should result in a considerable reduction in the time to identify development errors after they are introduction. The aggregate of MBD characteristics helps ensure the generation of product and process evidence needed for achieving high assurance of safety-critical systems.

5. Conclusions

Model-based development (MBD) is a formalization of product development with enhanced correctness, completeness, and precision of information throughout the process. The concept of using models to capture design information and to inform validation and verification activities is a well-established practice [9]. What differentiates and enables MBD is the exploitation of advanced information technology and tools to facilitate and integrate management, development, and quality-related activities. A common goal of development tools is to enable the users to think and work with domain specific concepts when designing and verifying their systems, and minimize the time to identify and resolve lower-level implementation issues. Model-based development expands and applies this idea to every aspect of the development process and all its participants: the developers focus on the problems within their domains of expertise and the tools automate and facilitate many of the integration, lower-level implementation, and quality assurance tasks.

The aviation industry has seen rapid increases in the complexity of computer-based systems onboard aircraft. Due to financial and business considerations, all development projects have finite resource budgets, which are summarized in terms of cost and schedule constraints. Competitive market forces drive the need for continuous increase in the level of complexity of the systems while preserving or improving the level of safety. Doing this with sustainable increases (or even reductions) in resource budgets requires commensurate increases in development productivity (i.e., efficiency). The success of the application of MBD to complex safety-critical aircraft systems will depend largely on its ability to meet this demand for enhanced productivity.

References

1. Beland, Steven C.: Assuring a Complex Safety-Critical Systems of Systems. AeroTech Congress & Exhibition, SAE Technical Paper 2007-01-3872. September 2007.

2. Bieber, P., et al: New challenges for future avionic architectures. Aerospace Lab Journal, issue 4, May 2012.

3. Tang, Christopher S.; Zimmerman Joshua D.; and Nelson, James I.: Managing new product development and supply chain risks: The Boeing 787 case. Supply Chain Forum: An International Journal, issue 19, volume 10, no. 2, pp. 74 – 86, 2009.

4. Moore, Jim: Advanced Distributed Architectures. In Avionics: Development and Implementation, Cary R. Spitzer, Editor, Chapter 13, CRC Press, 2007.

5. Hasson, Jeff; and Crotty, David: Boeing's safety assessment processes for commercial airplane designs. 16th Digital Avionics Systems Conference (DASC) , pp. 4.4-1 – 4.4-7, October 1997.

6. Federal Aviation Administration: Advisory Circular: System Design and Analysis. AC No. 25.1309-1A, June 21, 1988.

7. SAE International: Aerospace Recommended Practice: Guidelines for the Development of Civil Aircraft and Systems. ARP4754, Revision A, December 2010.

8. Bartley, Gregg and Lingberg, Barbara: Certification concerns of integrated modular avionics (IMA) systems. 27th Digital Avionics Systems Conference, pp. 1.E.1-1 – 1.E.1-12, October 2008.

9. Kasser, Joseph: Model-Based Systems Engineering: Back to the Future? Asia-Pacific Council on Systems Engineering (APCOSE) Conference. Yokohama, Japan, 2013.

10. SAE International: Aerospace Recommended Practice: Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment. ARP4761, December 1996.

11. Butler, Ricky W.: A Primer on Architectural Level Fault Tolerance. National Aeronautics and Space Administration, NASA/TM-2008-215108, February 2008.

12. Federal Aviation Administration: Advisory Circular: System Safety Analysis and Assessment for Part 23 Airplanes. AC No. 23.1309-1E, January 16, 2009.

13. Guide to the Systems Engineering Body of Knowledge (SEBok). http://www.sebokwiki.org/wiki/

Guide_to_the_Systems_Engineering_Body_of_Knowledge_%28SEBoK%29. Accessed March 16, 2014.

14. Lond, David and Scott, Zane: A Primer for Model-Based Systems Engineering. 2nd Edition, Vitech Corporation, 2011.

15. Estefan, Jeff A: Survey of Model-Based Systems Engineering (MBSE) Methodologies. International Council on Systems Engineering, INCOSE-TD-2007-003-02, June 10, 2008.

16. Feiler, P.: Challenges in Validating Safety-Critical Embedded Systems, *SAE Int. J. Aerosp.* 3(1):109-116, 2010, SAE Paper 2009-01-3284.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*
01- 12 - 2014

2. REPORT TYPE
Technical Memorandum

3. DATES COVERED *(From - To)*

4. TITLE AND SUBTITLE

Is Model-Based Development a Favorable Approach for Complex and Safety-Critical Computer Systems on Commercial Aircraft?

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
.
Torres-Pomales, Wilfredo

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER
534723.02.09.07

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
NASA Langley Research Center
Hampton, VA 23681-2199

8. PERFORMING ORGANIZATION REPORT NUMBER
L-20499

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Washington, DC 20546-0001

10. SPONSOR/MONITOR'S ACRONYM(S)
NASA

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
NASA-TM-2014-218670

12. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified - Unlimited
Subject Category 62
Availability: NASA STI Program (757) 864-9658

13. SUPPLEMENTARY NOTES

14. ABSTRACT

A system is safety-critical if its failure can endanger human life or cause significant damage to property or the environment. State-of-the-art computer systems on commercial aircraft are highly complex, software-intensive, functionally integrated, and network-centric systems of systems. Ensuring that such systems are safe and comply with existing safety regulations is costly and time-consuming as the level of rigor in the development process, especially the validation and verification activities, is determined by considerations of system complexity and safety criticality. A significant degree of care and deep insight into the operational principles of these systems is required to ensure adequate coverage of all design implications relevant to system safety. Model-based development methodologies, methods, tools, and techniques facilitate collaboration and enable the use of common design artifacts among groups dealing with different aspects of the development of a system. This paper examines the application of model-based development to complex and safety-critical aircraft computer systems. Benefits and detriments are identified and an overall assessment of the approach is given.

15. SUBJECT TERMS

Complex systems; Computers; Development; Model; Safety

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			STI Help Desk (email: help@sti.nasa.gov)
					19b. TELEPHONE NUMBER *(Include area code)*
U	U	U	UU	13	(757) 864-9658