Towards Safety Assurance of Trusted Autonomy in Air Force Flight Critical Systems *

Jacob Hinchman
Air Force Research Laboratory
Wright-Patterson AFB
Ohio 45433
937-255-8682
Jacob.Hinchman@wpafb.af.mil

Matthew Clark
Air Force Research Laboratory
Wright-Patterson AFB
Ohio 45433
937-255-8483
matthew.clark3@wpafb.af.mil

Jonathan Hoffman
Air Force Research Laboratory
Wright-Patterson AFB
Ohio 45433
937-255-8489
Jonathan.Hoffman@wpafb.af.mil

Brian Hulbert
LinQuest Corporation
AFRL Subcontractor
2601 Commons Blvd, Suite 100
Beavercreek, OH, 45431
937-255-8275
brian.hulbert@wpafb.af.mil

Cory Snyder
Marathon Petroleum Co LLC
Former AFRL co-op
539 South Main Street
Findlay, OH, 45840
419-421-2614
cfsnyder@marathonpetroleum.com

ABSTRACT

While safety is not implicitely a security problem, a security compromise is a safety concern. The move to autonomy has brought this need to a national level. Every domain with security and safety critical systems is looking to advance the state of the art in certification including, aviation, transportation, information assurance, medical, and energy. Verification and Validation of these systems are the primary means today of assuring the robustness of both safety and security requirements of a new system. As unmanned/autonomous systems become more complex, the notion that systems can be fully tested and all problems presented by an uncertain and dynamic environment is becoming increasingly invalid. This paper discusses some of the efforts by the Air Force Research Laboratory, Aerospace Systems Directorate to reduce reliance on test using new advances in formal analysis and early design verification techniques.

Categories and Subject Descriptors

A.1 [**GENERAL**]: Introductory and Survey;
B.1.3 [**HARDWARE**]: Control Structures and Microprogramming—*Control Structure Reliability, Testing, and Fault-Tolerance*;
C.3.3 [**SOFTWARE**]: Special-purpose and application-based systems—*Real-time and embedded systems*

General Terms

NuSMV - New Symbolic Model Verifier, RTA - Run Time Assurance,PLC - Programmable Logic Controller,AFRL - Air Force Research Lab

Keywords

Verification and Validation of Complex Systems, Cyber Physical Systems (CPS), Formal Methods, Runtime verification and steering, Systems Engineering, Aircraft test and evaluation

1. INTRODUCTION

As autonomous systems become more complex, the notion that systems can be fully tested and all problems will be found is becoming an impossible task. This is especially true in unmanned/autonomous systems. Full test is becoming increasingly challenging on complex system. As these systems react to more environmental stimulus and have larger decision spaces, testing all possible states and all ranges of the inputs to the system is becoming impossible. While the Google autonomous cars have completed over 300,000 miles of testing without incident, are they safe for the general public [9]? It depends. How much of the software was actually exercised? How many of the inputs were covered? Were all interdependencies of the inputs covered? What were the test conditions? What unknown system behaviors still exist? Would you feel safe enough to take your family on a vacation road trip on highways that contained autonomous cars that had been tested over 500,000? What about 1 million miles? As systems become more complex, safety is really risk hazard analysis, i.e. given x amount of testing, the system appears to be safe. A fundamental change is needed. This change was highlighted in the 2010 Air Force Technology Horizon report [4], "It is possible to develop systems having high levels of autonomy, but it is the lack of suitable V&V methods that prevents all but relatively low levels of autonomy from being certified for use."

In addressing the challenge of certifying autonomy, the problem can be broken down into a set of questions. First, given

*DISTRIBUTION STATEMENT A: Approved for Public Release; Distribution Unlimited (Case Number: 88ABW-2012-5315)

perfect knowledge of the situation, will the autonomous systems make the right decision? Second, if something unexpected happens, will the autonomous system make a safe and reasonable decision? Third, can the probabilistic uncertainty or level of assurance be determined for the information used to feed the decisions making thus, stating what types of decisions can be made with the given information?

Finally, can the system be decomposed in such a way that it can be certified in pieces and there are no unintended interactions? To this end, AFRL's Verification and Validation of Complex Systems (VVCS) team has organized its Verification and Validation research into the following thrust: Enhanced Analysis - reducing the reliance on test through upfront system and software analysis; Run-Time Assurance - moving from *a priori* to online safety assurance; Information Integrity - making safety critical decisions from non-critical data; Systems of System Certification - reducing the necessity of system wide certification. These thrusts are not independent areas of research but overlapping research with complementary approaches and varying applications. For instance, run-time assurance uses analysis techniques from enhanced analysis to verify its run-time boundaries.

In summary, as system complexity increases, the need for advanced verification and validation techniques and methodologies also increases. The move towards more autonomous systems has lifted this need to a national level. Every safety critical domain is looking to advance the state of the art in certification including, aviation, transportation, information assurance, medical, and energy. While the applications are different, the underlying safety concerns are similar and the V&V technologies are similar. AFRL along with its partners are addressing many of these fundamental challenges in complex system certification.

2. ENHANCED ANALYSIS

Traditional safety-critical software verification requires that every condition of every branch of software is tested (DO-178 MC/DC). It also requires that every line of code and test be traced back to requirements, i.e. validated [8]. Through this process, one is testing to prove correctness of the software. With better software analysis techniques, software can be analyzed at design-time with the goal of finding software faults earlier. This analysis can also prove the absence of error or negative properties. As system complexity and functionality increase, complete testing is becoming impossible and enhanced analysis techniques will have to be used. Furthermore, many of these software techniques, such as model checking, can be used in analysis of requirements and system design to find conflicting requirements or logic faults before a single line of code is written saving more time and money over traditional testing methods [3].

2.1 Modeling of Requirements

Research over the past year has investigated how formal analysis tools can be integrated into a new or existing system engineering tool chain. Many development tools are used throughout the entire systems engineering process and adding several new tools to an already complicated process may not be desirable. However, many researchers are already working on formal methods that integrate well with current development tools such as Matlab's Simulink Verifier and Microsoft Visual Studio's Spec Explorer Power Tool, respectively. Our research has focused on the requirements definition and analysis portion of the system engineering process. The requirements generation stage is the most important step in the process as errors in the requirements will lead to costly errors in the design. A domain specific language was created begin to formalize requirements for gaining accuracy in the requirements generation step as well as the ability to analyze the requirements for errors before the system is developed.

2.2 Formal Methods Acceptance Study

Formal methods have had V&V successes previously in communities such as computer hardware and software security. However, these techniques have made few inroads into the safety critical software arena. A study was conducted to investigate the perceived barriers to the wide spread adoption of formal methods techniques in the aerospace domain. By identifying the largest barriers to adopting formal methods as reported by respected, domain leaders, it is easier to see which challenges could yield the most return on investment and show the most promise to help encourage the adoption of these enhanced analysis techniques. The majority of interviewees (15 out of 26) reported that the use of formal methods has increased within their organizations in the last 5 years. The top two categories of most identified barriers were education on how to use formal methods and the usability of formal methods tools. About half of the responses, 53 out of 105, fell into those two categories. Additionally, the interviewees were asked to rate the severity of the barriers found by the fmsurvey.org survey [2]. The two barriers rated as the highest barriers were that formal method tools were not user-friendly and that there was a lack of evidence to support adoption decisions.

2.3 Application of Formal Methods to an Industrial Design Challenge Problem

In order to gain understanding of and experience with formal methods, the team decided to select a formal method and a challenge problem to conduct in-house research. The team decided to use the New Symbolic Model Verifier (NuSMV) model checker on a Programmable Logic Controller (PLC) Industrial Design problem. The industrial system existed as a specification of the system with PLC design code. Although it would mean transposing the PLC code into Matlab Simulink and Stateflow for the formal method tool Gryphon from Rockwell Collins, this problem did come with a requirements specification which would be necessary to generate and derive properties to prove about the system. The industrial design system contains four asynchronously operating machines as well as human input. The machines consist of an inspection machine; a molding machine, a pack-out machine, and a machine to coordinate the operation of the three machines plus take input from the human operator, see figure 2.3.

The system has 18 modes across the four asynchronous state machines and $2.0x10^6$ reachable states out of $2.6x10^{15}$ system permutations. The specification document provided many of the properties that were proven about the industrial problem. Functional properties, such as reachability to all 18

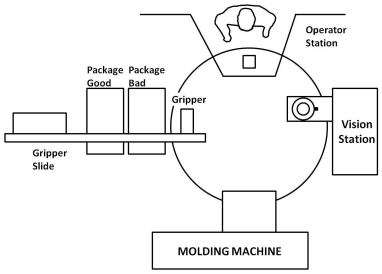

Figure 1: Industrial Automation Example

system modes, were proven about the model. A safety property ensuring that the table would not be in motion while the machines were operating was proven. Finally, the system requirements stating that good parts may only be placed into the 'good pack-out' basket and that bad parts may only be placed into the 'bad pack-out' basket were proven. A design flaw was discovered while checking for reachability. The logic design provided with the specification contains a bug in the startup sequence of the system. The 'main' state machine which coordinates the other state machines must assume that all of the other state machines contain a 'good' part in order to begin operation. This is a design error because the machine may be empty upon startup and therefore the other machines will not contain a 'good' part.

2.4 Enhanced Analysis Summary

Through early requirements analysis and incremental formal methods tool improvements, a comprehensive beginning to end analysis framework is being built. This framework will address many of the barriers to the acceptance of formal methods brought up in the study and will lead to an increased use in enhanced analysis techniques for software safety.

3. RUN TIME ASSURANCE (RTA)

While Enhanced Analysis attempts to reduce the amount of testing required to prove systems are correct prior to fielding the system, it may be impossible to prove everything *a priori*. However, if, through the use of a run time architecture, we can provably bound a system's behavior, then it may be possible to reduce the reliance on comprehensive off-line verification, shifting the analysis/test burden to the more provable run time assurance mechanism. Consider autonomy as "the ability to reason and make decisions to reach given goals based on a systems current knowledge and its perception of the variable environment in which it evolves [13]." Autonomous, safety critical software that relies on the perception of its environment to make decisions quickly becomes a large near infinite state problem. To that end, Run Time Assurance aims to enable certification for unverifiable functionality through dynamic, predictive bounding.

The goal of the RTA approach is to ensure the safe operation of a system that contains functional components, which may not be sufficiently reliable, or sufficiently verified, according to current development or certification standards. There may be multiple reasons for having such components in a system: under the normal conditions, they can provide improved performance or operational efficiency for the system, or enhance the user experience.

The core idea that enables the use of such components in a system is the presence of a safe, fallback mechanism that 1) reliably detects potential problems and 2) invokes a recovery/switching mechanism that can ensure safe operation of the system, possibly with reduced capabilities and performance. Development of the technology necessary to design and implement such mechanism and reasoning about its safety is, by and large, the scope of this thrust.

Within the aerospace domain, the following certification challenges were identified as only solvable at run time: unanticipated vehicle interactions, unanticipated external interactions, mission/battle management decisions with flight-critical consequences, untested system modes, and autonomous decision making control [13]. The desire is that unmanned aerial systems (UASs) should be able to use the same infrastructure as manned systems, with minimized uniqueness. They also must be made to be responsive to dynamic missions, adapting in real time to changes in environment, mission, etc. This creates an unsolvable offline certification challenge but an opportunity for run time certification techniques. Similarly, in the automotive sector, the Google autonomous car has successfully achieved over 300,000 miles of unattended driving in the streets of California[9]. For the autonomous vehicle scenario to become reality, the human monitor must be replaced with a certified bounding algorithm that is capable of providing absolute guarantees on the vehicle's safety in the highly dynamic environment such as urban streets. Within the power distribution industry, innovations in smart-grid technology consider decentralizing power distribution by creating stand-alone power units called "micro-grids"[1]. To enable the combined use of the micro-grid, highly adaptive autonomous systems would be needed to carefully manage energy production and consumption and would require a boundary mechanism to assure safety of the system.

The question arose, what will it take to create a run time assurance framework for the cyber physical systems' vehicle space? A common, implementable framework required to reduce the reliance on offline verification has yet to be developed for the domain of safe and secure autonomous vehicles.

3.1 Run Time Assurance Investigation

To explore this question, a study was performed, investigating the key technologies available and needed to increase the reliance on run time assurance. To guide this research, four questions were provided to key researchers in the Controls and Computer Science domains. The goal was to investigate what technologies and research could apply to run time assurance framework and what challenges would arise in creating such a framework.

First, what algorithms can be used to guarantee safe bounds? For an autonomous system, certain assumptions about the known environment must be made given a set of known input and output states. Utilizing these assumptions to create a boundary for non-deterministic, adaptive systems, RTA

aims to achieve advanced performance with the assurance of safety constraints and failsafe operability. Hybrid Systems research has proven to be a viable area of research for provable RTA boundaries. Among other resources, a wiki was created by one of the researchers to catalog the hybrid analysis tools available and what types of problems they are capable of solving [12].

Second, how do we create a run time version of the algorithm that enables safe switching? Creating a mathematical boundary that accounts for all possible environmental scenarios becomes a highly computationally intensive problem. Such problems are difficult to calculate offline let alone provide assurance dynamically. Once the safety properties and switching conditions are identified, one needs to develop a monitor that will calculate the switching conditions and effect the switch. Therefore, the second domain of expertise needed to formulate the RTA framework is the ability to perform the computations at run time. The runtime verification community has done extensive research in this area providing a rich field of expertise to reference [15].

Third, how do we ensure timing constraints and worst case execution time are preserved? As run time methods and monitoring software is added, impacts to existing hardware and software interaction will need to be considered. For example, any run time approach for flight critical systems will need to address interactions between triplex redundant control architectures. Technologies need to be considered from a hardware timing, synchronization, and parallel monitoring approach to ensure timing is considered within and external to the system. Multiple processors, cores, or interacting systems of systems rely on consistent timing constraints being followed.

Finally, how can model based design/simulation enable quicker realization of an end product? Many formal verification and validation techniques emphasize correctness by construction and design for verification. These tag lines speak to the need to ensure the modeling and simulation environment is compatible with the current V&V techniques and formal methods, allowing an increase in validity of methods used earlier in the design process. A modeling and simulation environment must be able to connect different abstractions of not only the run time implementation but the environment of which it is protecting. Run Time Assurance must consider such environments in order to accelerate framework production, simulation, verification, and validation. A more comprehensive report detailing the findings should be published in fall 2012.

3.2 Hybrid Systems Verification

In concert with the questions presented above, it is necessary to find an analytical method to represent the discrete, linguistic (rule based), and continuous nature of an autonomous aerospace system. This system model must include not only the inner loop control dynamics but the higher level decisions and the bounding safety constraints. In an effort to create a general framework, Hybrid Systems modeling and verification has been a key concept within our research over the past ten years [13]. During that time, great advances in hybrid systems control theory and verification have been developed [14]. One of our initiatives is to understand this work and how it may apply to a general approach for boundary creation of a Run Time Assurance algorithm.

Initially, we looked at applying the reach-set theory for provably safe quadrotor back-flip maneuvers and to provably safe collision avoidance strategies [6]. The fundamental procedure relies on calculating the reachablity of Hybrid Systems by formulating the problem as a series of Hamilton-Jacobi partial differential equations (HJ-PDE) connected as hybrid modes. The modes are identified as reach (control modes that you want to safely achieve) and avoid (modes that are considered unsafe). The problem is setup to work backwards from an eventual safe mode, identifying what set of initial conditions will guarantee the entrance into that safe set. For simple problems, it may be possible to find analytical solutions to the HJ-PDE; however, most useful problems require relatively complex numerical solutions. A tool which is leveraged in many state of the art reachable-sets research studies to solve HJ-PDEs is the Level Sets Toolbox. The toolbox is implemented in MATLAB and uses level-set numerical methods to approximate HJ-PDE solutions[11]. However, there are several limitations to this approach. First, all the computations are done offline based on a known set of modes. Second, the method, depending on the the system model and the resolution of the grid is limited to only 3-5 controllable states. Third, the approach is highly reliant on the model. If the model varies too much from the actual system, the pre-determined reachability calculations are invalid.

Other tools explore the idea of forward reachability, enabling a faster calculation of an approximate safe mode that can be achieved in the future. One of those tools is SpaceEx, which integrates several tools to implement a forward reachable set solution [5]. The tool has promise in that it makes great advances in calculation time and the number of system states it can handle. However, the tool does not handle nonlinear dynamics. Future research will look at methods of creating piecewise affine approximations of our systems and implementing the tools via run time.

3.3 Run Time Assurance Summary

A goal of the verification and validation approach is to enable a technique that is so widely accepted within the community that it gains the same trust as test. To accomplish this task, future efforts will be placed on establishing a larger public domain community collaborating on Run Time Assurance technologies. As technologies and methods mature, greater the implementation of Run Time Assurance will enable greater advances in trusted autonomy.

4. SYSTEMS OF SYSTEMS CERTIFICATION

While Enhanced Analysis and Run Time Assurance look at improving single system verification and validation, today's systems are becoming so much more complex that there is a growing issue of unintended interactions on a macro-level within a system of systems (SoS) environment. As systems are composed into a larger system, behaviors begin to emerge that were not existent at the individual or loosely coupled level. Therefore, one can easily see how the whole SoS architecture is greater than the sum of the parts. As the complexity of these more advanced systems increases,

their non-linearity and non-deterministic qualities increase as well. This increased complexity can lead to instances of unintended interactions which may violate the safety, security, and certification constraints of the system being developed.

As systems become more tightly coupled, unintended interactions become more pronounced. As an example, the avionics on many of today's commercial aircraft have been designed using a federated architecture where each capability has its own resources. With this approach, there is very little interaction among the separate systems and unintended interactions between subsystems is eliminated. As a result and since there are very little common, shared resources, the certification of this style of architecture can be accomplished mostly independently for each of the avionics systems. However, due to the duplication of resources for many of the systems, this approach is extremely costly. Furthermore, today, current certification practice is to certify a system as a whole (i.e., there is no provision or basis for separate or modular certification).

On an even larger scale, the problem of multiple systems interacting safely, such as the Federal Aviation Administration (FAA) NextGen environment, can be achieved through maturation of this research. As such, the design of such cyber-physical systems is a major challenge. It has been stated that the verification and validation of critical avionics software alone is estimated to cost seven times as much as its software development costs[7].

The overall Air Force Research Lab (AFRL) vision for this architecture research area is to reduce reliance on system-wide certification through trusted, formalized, and safe interactions of certified systems with focus both on single systems and within a system of systems. Throughout the FY12 period, an initial literary search has been performed to heighten awareness of current practices and emerging trends and challenges. From this search there seems to be promising research out of MIT by Dr. Nancy Leveson in system theory and the analysis of systems of systems architectures [10]. In particular, the Systems–Theoretic Accident Modeling and Processes (STAMP) model provides an organized, methodical, and effective means to assess safety risk and develop appropriate hazard mitigations regardless of where in the life cycle the assessment is started. It incorporates three basic components: constraints, hierarchical levels of control, and process loops. To gain a deeper understanding of this area, in August 2012 Dr. Leveson, one of the leading American experts in system and software safety, presented a short course on this topic to help Air Force engineers gain a top-level understanding of the problem as well as new techniques (i.e., STAMP [Systems-Theoretic Accident Modeling and Processes], STPA [System-Theoretic Process Analysis], and CAST [Causal Analysis using System Theory]) that are currently in use in a wide variety of industries (e.g., space, aviation, medical, defense, nuclear, automotive, food, and other complex applications). One of the benefits of Dr. Leveson's teaching in this area is that it is in alignment with the current DoD standard practice guidance for system safety. Therefore, the impact of these techniques can be realized very quickly. Dr. Leveson's research is but one of several approaches to the challenges of certification of systems of systems architecture that need further research, evaluation, and application to real world systems.

Additionally in late CY12, AFRL is preparing to release a Phase 1 Small Business Innovative Research (SBIR) contractual opportunity to conduct an evaluation of SoS certification research leading to the development of initial methodologies and analysis techniques for modeling and formally verifying Systems of Systems interactions.

5. CONCLUSIONS

Whether in early design, reduction of test, trust in unpredictable autonomy, or assuring safe interactions, our goal is to provide certification technologies that enable complex autonomous aircraft to interact with the world safely. As highly autonomous aircraft become more of a reality, trust in the pilot transfers to trust in highly complex software and systems. Quantifying that trust and then providing a certification argument is a daunting task both in the safety and security realm.

6. REFERENCES

[1] S. Balantrapu. Role of artificial neural networks in microgrid, 2010.

[2] J. Bicarregui, J. Fitzgerald, P. Larsen, and J. Woodcock. Industrial practice in formal methods: A review. *FM 2009: Formal Methods*, pages 810–813, 2009.

[3] D. Chandramouli and R. Butler. Cost effective use of formal methods in verification and validation.

[4] U. S. A. Force. Technology horizons a vision for air force science and technology during 2010-2030, 2010. `http://www.af.mil/shared/media/document/AFD-100727-053.pdf`.

[5] G. Frehse, C. Le Guernic, A. Donzé, S. Cotton, R. Ray, O. Lebeltel, R. Ripado, A. Girard, T. Dang, and O. Maler. Spaceex: Scalable verification of hybrid systems. In *Computer Aided Verification*, pages 379–395. Springer, 2011.

[6] J. Gillula, G. Hoffmann, H. Huang, M. Vitus, and C. Tomlin. Applications of hybrid reachability analysis to robotic aerial vehicles. *International Journal of Robotics Research*, 30(3):335–354, 2011.

[7] C. Hang, P. Manolios, and V. Papavasileiou. Synthesizing cyber-physical architectural models with real-time constraints. In *Computer Aided Verification*, pages 441–456. Springer, 2011.

[8] K. Hayhurst and L. R. Center. *A practical tutorial on modified condition/decision coverage*. National Aeronautics and Space Administration, Langley Research Center, 2001.

[9] F. Lardinois. Google's self-driving cars complete 300k miles without accident, Aug 2012. `http://techcrunch.com/2012/08/07/google-cars-300000-miles-without-accident/`.

[10] N. Leveson. *Engineering a safer world: Systems thinking applied to safety*. MIT Press (MA), 2012.

[11] I. Mitchell. The flexible, extensible and efficient toolbox of level set methods. *Journal of Scientific Computing*, 35(2):300–329, 2008.

[12] G. Pappas. Hybrid system tools, Feb 2012. http://wiki.grasp.upenn.edu/hst/index.php?n=Main.HomePage.

[13] L. Rudd and H. Hecht. Certification techniques for advanced flight critical systems. Technical report, WPAFB, 2008.

[14] S. Sastry and C. Tomlin. Hybrid systems—computation and control, Jan 2012. http://www-inst.cs.berkeley.edu/ ee291e/sp12/.

[15] O. Sokolsky. Runtime verification website, 2012. http://runtime-verification.org/.