Aerospace Technology Congress, 8-9 October 2019, Stockholm, Sweden
Swedish Society of Aeronautics and Astronautics (FTF)

Assurance Strategy for New Computing Platforms in Safety-Critical Avionics

Håkan Forsberg, Andreas Schwierz*, and Kristina Lundqvist

School of Innovation, Design and Engineering, Mälardalen University, Västerås, Sweden[1]
E-mail: Hakan.Forsberg@mdh.se, Andreas.Schwierz@thi.de, Kristina.Lundqvist@mdh.se
*Research Center, Competence Field Aviation, Technische Hochschule Ingolstadt, Germany

Abstract

An assurance strategy for new computing platforms in safety-critical avionics has to be flexible and take into account different types of *commercial-of-the-shelf* (COTS) hardware technologies. Completely new COTS technologies are already being introduced and successfully used in other domains. Good examples are heterogeneous platforms, hardware-based machine learning and approximate computing. Current avionics certification guidance material cannot cope with next generation of devices. We suggest using the generic assurance approach of the *Overarching Properties* (OPs) together with assurance cases to argument that COTS assurance objectives are met and to achieve the flexibility required for future computing platforms. We introduce a novel assurance cased-based OP approach in [1] and refine the work into a framework in [2]. Within this framework we are able to integrate COTS technology specific assurance objectives using a five-step process. In this paper, we show through some representative examples of emerging computing platforms that our strategy is a way forward for new platforms in safety-critical avionics.

Keywords: safety-critical avionics, assurance strategy, assurance case, COTS assurance, Overarching Properties, computing platforms

1 Introduction

Over the last 15 years, RTCA/DO-254 [3] has been used as the guidance document to ensure design assurance for civilian airborne electronic hardware (AEH). Design/development assurance is "*All of those planned and systematic tasks used to substantiate, to an adequate level of confidence, that development errors have been identified and corrected such that the items satisfy a defined set of requirements*" [4]. AEH is often designed with several COTS components that are not developed according to RTCA/DO-254. The use of COTS components therefore requires other assurance guidance techniques to be used. The certification authorities have identified and produced several COTS assurance guidance documents, see Section IV in [1] for a literature review of these. When new technology has been introduced, new assurance activities have been suggested by the authorities. The latest guidance document from the certification authorities addressing COTS assurance (including COTS IP) is published in a Notice of Proposed Amendment [5], which is a joint EASA and FAA effort. This document is objective-based and is supposed to address all kinds of existing COTS components. It does not address new technologies such as e.g., hardware accelerated machine-learning. Objectives-based guidance benefits from being more flexible to adapt future technologies rather than activities-based documents. Still, both objectives-based and activities-based guidance documents suffer from the assumption that, if it is followed it is sufficiently assured that the COTS component operates with integrity according to its specification.

EASA's Certification Memorandum SWCEH-001 [6] (non-binding guidance material) is a mixed level activities-based guidance document. Section 9, in [6], gives guidance for COTS integrated circuits and microcontrollers while Section 10, in [6], addresses COTS graphical processors (GPUs). For integrated circuits and microcontrollers, the assurance evidence depends on the amount of service experience, complexity of the component, and the design assurance level (DAL). For GPUs, the guidance material [6] assumes a discrete graphical processor (opposite to several of today's integrated GPUs) that has a very short lifespan with an increased possibility of design errors, is complex, contains configurable elements, and is only used for graphical applications. The guidance material also assumes that the component may exhibit performance variations over production time and may completely lack empirical data on the actual failure rates experienced in avionics applications. To cope with all these uncertainties, the guidance material explicitly assumes several low-level activities to be performed for all kinds of GPU devices [2]. Guidance on activity level is not suitable for new hardware technologies.

[1] MDH's work in this paper is supported by the Swedish Knowledge Foundation within the DPAC project Dependable Platforms for Autonomous systems and Control.

DOI
10.3384/ecp19162015

The suggested approach for new architectures and other new COTS technologies are instead an argumentative approach allowing for flexibility to use more appropriate methods and also directly show how these methods contribute to meeting the assurance objectives [2]. One such approach could be based on assurance cases. An assurance case is a structured argument, backed-up with evidence, that a system operates as intended for a defined application in a defined environment [7]. In [1] we demonstrated the use of an assurance case to structure COTS hardware components' assurance for safety-critical avionics and in [2] we refined our work and introduced a five-step process (see Section 2.3 in this paper) to provide a concept to connect the demonstration of assurance objectives. The use of assurance cases is in line with FAA's process to streamline the certification process by delivering an approach (Overarching Properties) usable for both software and hardware development to ease the use of alternative means of compliance [8]. The main contribution in this paper is the integration of existing assurance objectives and representative examples from new COTS-based computing platforms using Overarching Properties and assurance cases.

The remainder of this paper is structured as follows: Section 2 explains our COTS assurance case concept, the used graphical notation, the Overarching Properties, and shows how current COTS assurance objectives can be integrated. In Section 3 we integrate assurance objectives from emerging computing platforms and in Section 4 we discuss our assurance case concept. Finally, in Section 5 we conclude the paper.

2 COTS assurance case concept

In the framework of assurance cases, assurance refers to the proven confidence that a top-level claim of an argument is true [2]. Figure 1 shows a graphical presentation of an assurance case.

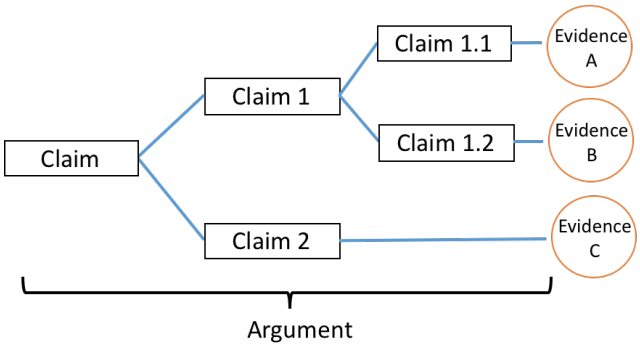

Figure 1. A graphical presentation of an assurance case. The top-level claim (leftmost) is decomposed until each sub-claim can be substantiated by evidence. The argument part, which is the focus of this paper, consists of strategies used to decompose claims and sub-claims.

Assurance cases in the specialized form as safety cases, have been successfully used for a very long time [9]. The strength with assurance cases is that it forces people to think deeper than usual [10] and motivates developers to formulate explicit arguments clearly targeting a top-level claim.

Structured assurances cases can be used to explain why a chosen assurance method is sufficient. They give the case writer the possibility to demonstrate (explicate) in a reviewable argument the assurance strategy in its entirety. This allows a third party (certification authority) to get the overview and full insight about how the item is assured and the justification why it is valid or acceptable to conclude that the item behaves with integrity in the system.

The question one can ask is - are assurance cases beneficial for emerging COTS-based computing platforms? Rinehart and Knight [9] have claimed several potential benefits for assurance cases in general. One of them is *assurance cases address modern certification challenges*. In [2] we interpreted all described benefits from Rinehart and Knight in the context of emerging computing platforms. Berthon [11] has used a structured assurance case for COTS AEH. Berthon suggests a design assurance level (DAL) based evidence approach for COTS hardware.

2.1 Assurance case notation

In this paper we use a graphical notation based on a subset of the Goal Structuring Notation (GSN). GSN is defined in [7]. Figure 2 shows the symbols we use in this article and Table 1 explains the used symbols.

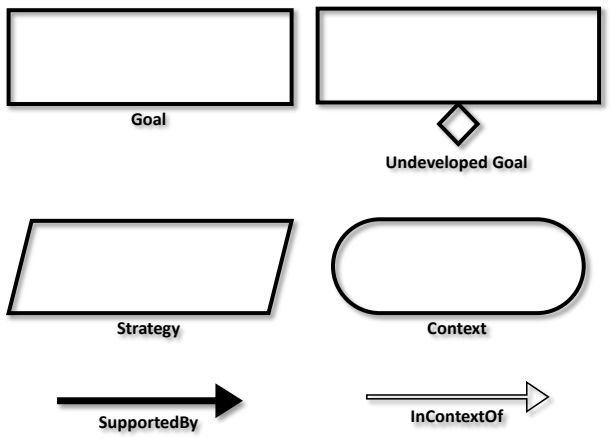

Figure 2. GSN symbol subset used in this paper

Table 1. Explanation of used GSN symbols

Goal	The *Goal* element illustrates claims and sub-claims supporting higher-level connected claims.
SupportedBy	The *SupportedBy* relationship creates a series of connected claims to establish an overall claim.
Undeveloped Goal	An *Undeveloped Goal* is left intentionally undeveloped for later investigations.
Strategy	The *Strategy* element helps explaining or argument the logic between a goal and its supporting goals.
Context	The *Context* element is used to clarify concepts mentioned in strategies.
InContextOf	A *Context* element has a corresponding *InContextOf* relationship.

2.2 Generic higher-level goals

In [1] we defined the top goal *COTS component operates demonstrably airworthy in its system context* for assuring a COTS integrated in safety-critical avionics. The top goal is based on applicable functional and safety certification specifications (CS) requirements, derived to the COTS component level by Berthon *et al.* [12]. Berthon *et al.* identified six key objectives applicable to all kinds of AEH based on CS requirements. These objectives form the context to our top goal. We then used a strategy to decompose our top goal in "time", i.e. argument over initial airworthiness and argument over continuous airworthiness. From now on we only consider the former. For initial airworthiness, we then used the strategy *Argument over isolated COTS component and integrated COTS component* to decompose our case into two sub goals, see Figure 3.

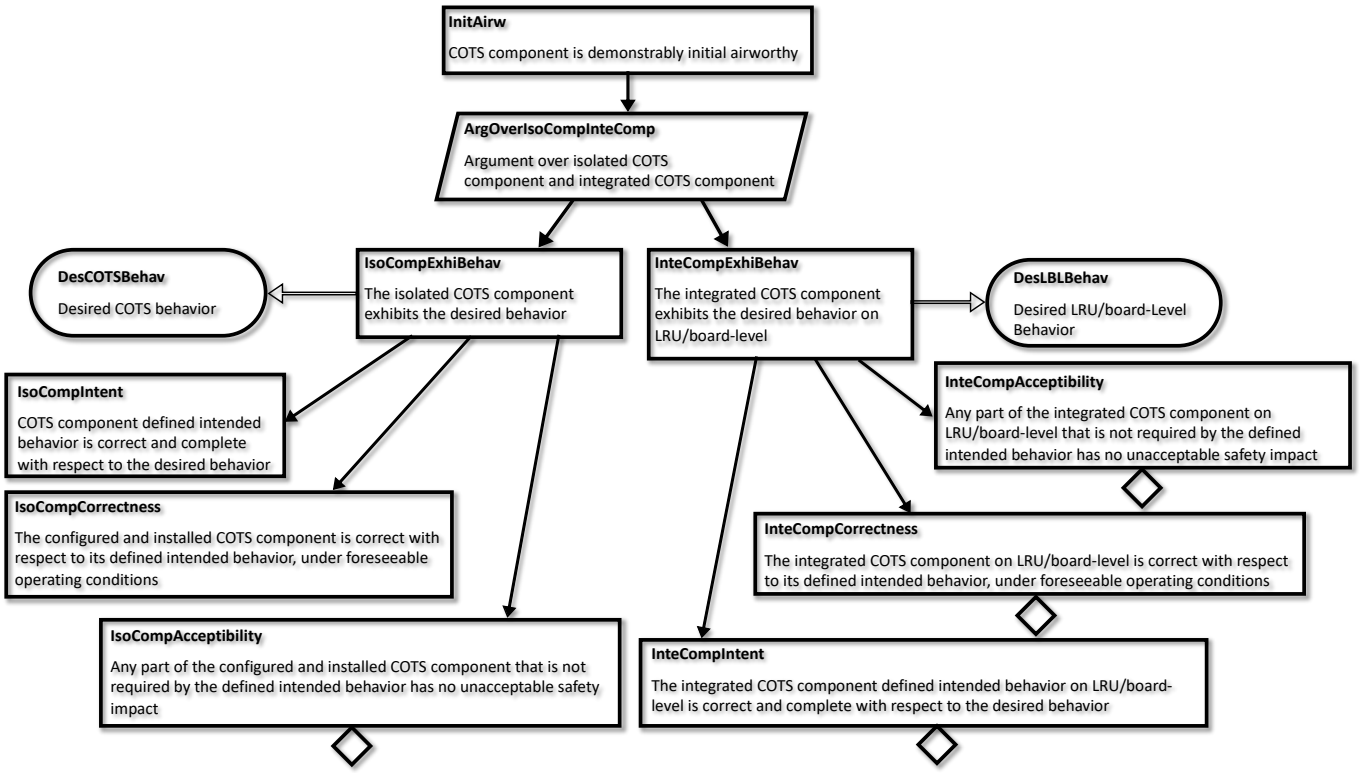

Figure 3. Initial airworthiness argument split into isolated COTS component and COTS component integrated on LRU/board level and further divided into the three Overarching Properties intent, correctness and acceptability.

Both sub-goals must be fulfilled to achieve the top-level goal for initial airworthiness, i.e.

1. the isolated COTS component exhibits the desired behavior and

2. the integrated COTS component exhibits the desired behavior on LRU/board level (note that certain behavior can only be verified on this level).

To build our next layer, we use the three Overarching Properties, informally written as [13]:

1. Intent – what the product is supposed to do is properly captured,
2. Correctness – the product does what it is supposed to do, and
3. Acceptability – the product does not cause harm since development decisions do not compromise the original safety assessment

See Figure 3 for proper implementation in the COTS element context. Note that both branches use the OP approach, i.e. the isolated and the integrated COTS component should demonstrate that the OPs are possessed on each level. With other words, it is not until all three properties on both levels have sufficient convincing arguments that the overall argument *COTS component is demonstrably initial airworthy* can be considered evidenced. The idea behind using OPs is to provide a unified method for the approval of different kinds of objects, i.e. such that this approach can be an alternative to already used assurance methods.

Up to this point the presented argument is aligned according to a generic layout which should be applicable for all kinds of COTS components. Even if the OPs are used to organize the assurance concept, it has to be shown that the COTS device meets the allocated specification adequately. The next step in the argument provides a strategy to enable the demonstration of the OPs together with considering the COTS technology dependent assurance.

To facilitate the demonstration of each OP, we use a separation into a primary and a confidence argument proposed by Hawkins *et al.* [14] and successfully demonstrated by Holloway and Graydon [15]. Compare it with safety cases where safety is the attribute of interest, where identification and mitigation of hazards to reduce risk should be in the primary argument.

Let us look at the goal *IsoCompCorrectness* from Figure 3. The primary argument should be "The configured and installed COTS component performs its intended behavior correctly, under foreseeable operating conditions," i.e., in this case, exactly the same as before the separation. The confidence argument should then produce the evidence that the primary conclusion is sufficiently creditable, i.e. the reviewer should believe the chain of transformation was correctly performed with sufficiently avoidance of errors. We thus define the confidence argument for *IsoCompCorrectness* as "Uncertainties in the correct transformation of the defined intended behavior to the configured and installed COTS component are sufficiently reduced."

2.3 Five step process to integrate COTS objectives

In [2], we created a framework that can connect assurance objectives directly with a new COTS assurance concept based on OPs and assurance cases. The framework should be performed in the following process steps:

1. Choose the level on which the assurance objective has to be demonstrated (isolated or integrated).
2. Assign the assurance objective to the relevant OP.
3. Reformulate it to a conclusion.
4. Demonstrate its satisfaction in the primary argument.
5. Explain in the confidence argument how you reduce the uncertainty in the primary argument.

Below we demonstrate the framework by integrating an already defined objective, *COTS-3*, from the Notice of Proposed Amendment [5]. In Section 3, we integrate two examples from emerging computing platforms.

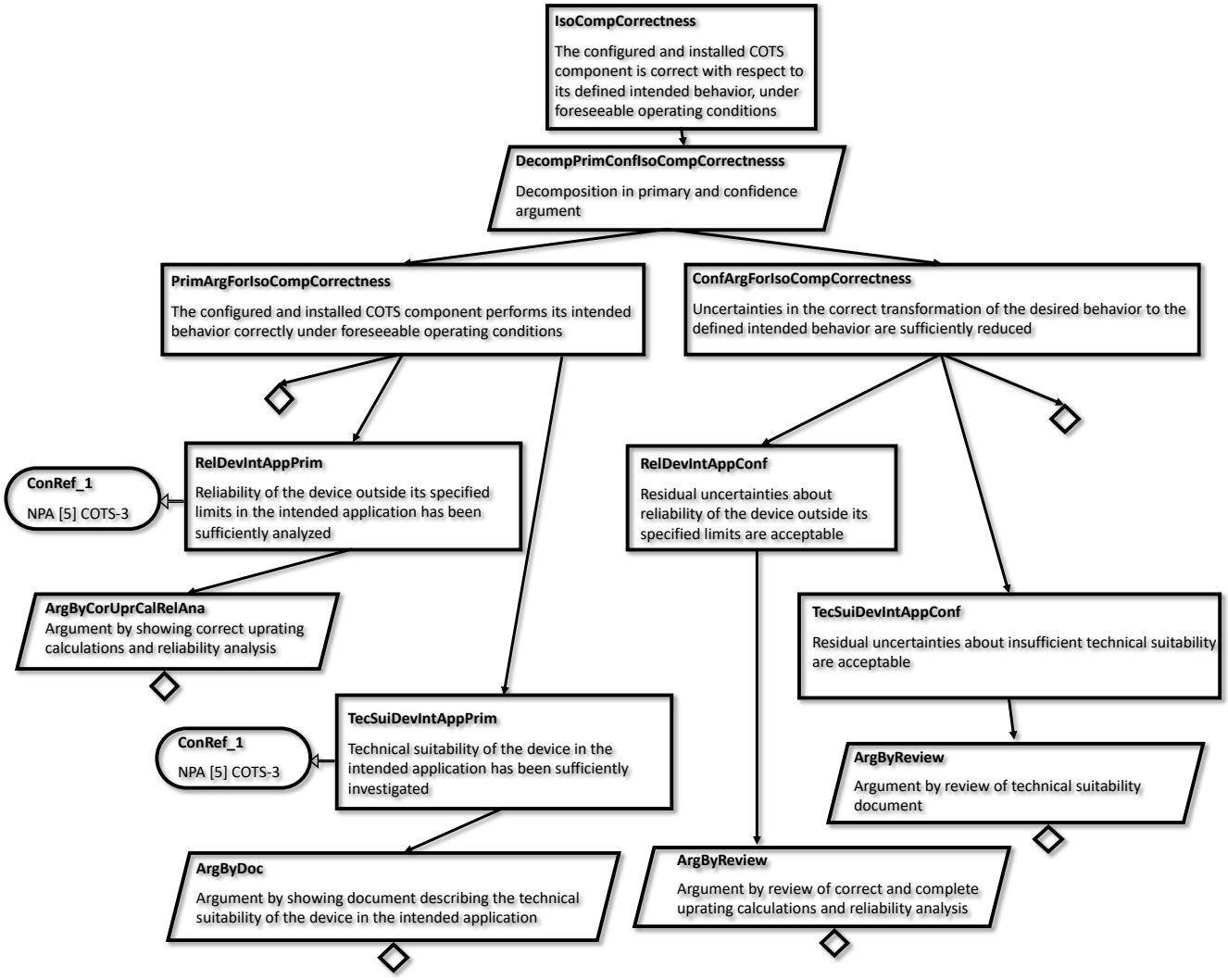

Figure 4. Integration example of the assurance objective COTS-3 from the Notice of Proposed Amendment [5]

The *COTS-3* objective in [5] states: "*When the complex COTS device is used outside the device manufacturer's specification (such as recommended operating limits), the applicant should establish the reliability and the technical suitability of the device in the intended application.*"

Step 1 - demonstration level – For *COTS-3*, the correct level of demonstration is the isolated COTS component level.

Step 2 – assignment to relevant OP – The appropriate OP is *IsoCompCorrectness*. See Figure 4. Even if the component is supposed to be used outside the device manufacturer's limits, it is under foreseeable operating conditions.

Step 3 – reformulate to conclusion – The objective should be reformulated to express a goal whose content can be substantiated with an assurance case. This concerns both the primary and the confidence argument. Here, we have chosen to split the *COTS-3* objective into two goals (the same two objectives as *COTS-3* actually addresses):

1. The reliability of the device outside its specified limits in the intended application has been sufficiently analyzed, and

2. The technical suitability of the device in the intended application has been sufficiently investigated.

Step 4 – demonstrate its satisfaction in the primary argument - In this case, we assume that the primary argument is to show correct uprating calculations and reliability calculations and by showing a document where the technical suitability has been documented and demonstrated.

Step 5 – explain in the confidence argument how you reduce the uncertainty in the primary argument - The uncertainty will be reduced by independent reviewing of uprating calculations and reliability analysis and by independently reviewing the document describing the technical suitability.

2.4 Integration of existing COTS assurance objectives

The latest proposed guidance document from the certification authorities (FAA and EASA) addressing COTS assurance is a Notice of Proposed Amendment (NPA) [5]. The COTS assurance part of the NPA consists of eight objectives (*COTS-1 to 8*) that must be fulfilled for the highest design assurance levels. In previous research we showed how to implement *COTS-8* in our framework. Above, we demonstrate the implementation of *COTS-3*. From a completeness standpoint, all eight *COTS-x* should cover the six sub-claims we defined, i.e. isolated COTS intent, correctness, and acceptability as well as integrated COTS intent, correctness, and acceptability. Our analysis show that all eight certification objectives indeed cover the three Overarching Properties on both isolated as well as integrated COTS level, with emphasize towards the former. Table 2 shows the mapping between the existing COTS assurance objectives in [5] and the relevant level and Overarching Property in our framework.

Table 2. Existing COTS objectives [5] and their mapping into isolated or integrated COTS level and Overarching Property

COTS objective – from [5]	Relevant level and Overarching Property
COTS-1 – assessment of complexity	Isolated - Intent
COTS-2 – electronic component management process	Isolated - Correctness
COTS-3 – usage outside manufacturer's specification	Isolated - Correctness
COTS-4 – non-qualified microcode	Integrated - Correctness
COTS-5 – assessment of errata	Isolated - Acceptability
COTS-6 – failure modes and common modes	Isolated - Acceptability & Integrated - Acceptability
COTS-7 -intended function of COTS device including interfaces	Integrated - Intent
COTS-8 – inadvertent alteration of critical configurations settings	Isolated - Intent

3 Emerging platforms implementation

In this section we introduce new emerging computing platforms that in the future might be introduced for the avionics industry and use two examples of potential assurance objectives from these platforms to show that our framework and five step process work for new objectives as well.

3.1 Heterogeneous platforms

Heterogeneous computing platforms use massive parallelism from non-traditional computing devices, e.g. GPUs or digital signal processors (DSPs) to achieve high performance computations at low energy. At the same time, they use traditional central-processing units (CPUs) for latency-sensitive serial parts of the code [16]. Medical imaging, computational photography and fluid dynamics are areas where heterogeneous platforms have been successful [16]. New programming models and compilers, hardware/software interface, run-time support, load balancing and scheduling policies are all challenges for heterogeneous architectures [17]. These heterogeneous COTS components do not have any specific certification guidance ready for the avionics market yet [2].

3.2 Hardware based machine learning

Machine learning in the form of deep neural networks (DNNs) has shown to be a promising alternative for object identification for several application domains [18]. In autonomous cars it is used as one of the primary sources for detection of pedestrians, cars, bicycles, animals, etc. Obstacle avoidance decisions are made from different types of objects and their movements. DNNs may also be successful in airborne systems. One such possible application is guided landing. To land autonomously without support from ground infrastructure requires advanced airborne systems including algorithms for detecting the runway. These systems are

safety-critical. The use of DNNs in safety-critical systems cannot rely on traditional design assurance techniques. Instead, other techniques have to be used. The main reason for this is that a DNN has to be trained with data sets of images (or other data) with objects it should be able to classify, but it cannot be trained with all possible inputs. Thus, misclassification of objects may appear. DNNs are also weak to adversarial inputs (the alteration of inputs which forces a trained DNN to misclassify) e.g. due to malicious attacks or external faults caused by the environment such as single event upsets. Several assurance techniques have been suggested for the use of DNNs in safety-critical applications [19, 20].

It is important for hardware-based machine learning to quantify the probability of an undetected misleading error and show that the error is appropriate to the function. It is not possible to perform this objective on isolated COTS component level since the component is trained for a certain purpose. We therefore define the following assurance objective ML-OBJ-1:

- *ML-OBJ-1* - the applicant should quantify the probability of an undetected, misleading error and show that the error is appropriate to the function.

We will now demonstrate this objective in our framework.

Step 1 - demonstration level – For *ML-OBJ-1*, the correct level of demonstration is isolated COTS component level.

Step 2 - assignment to relevant OP – The adequate OP is *InteCompAcceptability*, see Figure 5 below. Undetected misleading errors must be captured outside the device.

Step 3 - reformulate to conclusion – we reformulate the conclusion to "The probability for undetected misleading errors is quantified and the errors are appropriate to the function."

Step 4 - demonstrate its satisfaction in the primary argument – Our selected primary argument to fulfill the conclusion is to use statistical testing appropriate to the function.

Step 5 - explain in the confidence argument how you reduce the uncertainty in the primary argument – This one is really hard to cope with. How do you reduce the uncertainty of results from statistical testing? By testing more? We believe one solution can be the use of a diverse redundant architecture and to include timing aspects (to detect changes of objects in time). See Figure 5. This architecture is subject to future research.

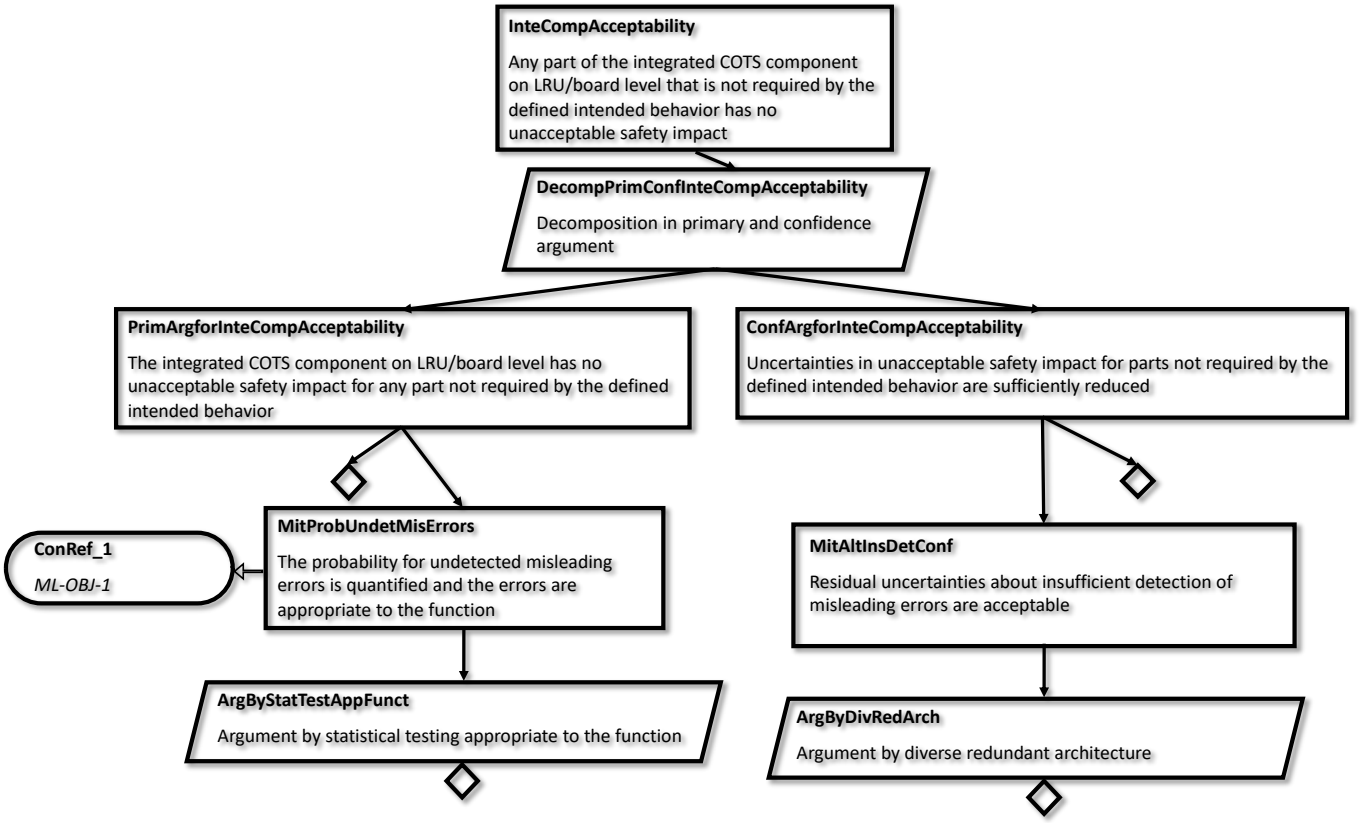

Figure 5. Integration example of an assurance objective most probable for machine learning architectures

3.3 Approximate computing architectures

Approximate computing is when computation accuracy is traded for better performance or energy consumption [21]. In approximate computing, different types of reduction of computation accuracy can be used, e.g. reduced number of bits in the arithmetic operations, approximate findings of results from expensive function calls (approximate memorization), reduced number of loops in loop constructions (loop perforation), or relaxed synchronization in parallel applications [22, 23]. These kinds of algorithms may be used for applications such as machine learning, computer vision, and speech.

In [2], we listed general guidelines that should be followed for architectures using approximate computing. We show one of them here:

- the computation should maintain integrity, i.e. using the same input data twice should show identical results (unless altered by physical phenomena, which must be detected)

We will now demonstrate this objective in our framework.

Step 1 - demonstration level – For *AC-OBJ-1*, the correct level of demonstration is integrated COTS component level.

Step 2 - assignment to relevant OP – The adequate OP is *IsoCompIntent*, see Figure 6. The device itself shall maintain integrity.

Step 3 - reformulate to conclusion – we reformulate the conclusion to "Using the same input data twice should show identical results."

Step 4 - demonstrate its satisfaction in the primary argument – Our primary argument is to use testing with equivalence classes.

Step 5 - explain in the confidence argument how you reduce the uncertainty in the primary argument – Here, you might be able to use advanced verification methods to ensure correct operation of the device, but you may also reduce the uncertainty by using an external monitor. In our case we used the latter. A simple monitor built upon a memory that stores previously received data and associated output results, and triggers only when a subsequent computation with identical input data results in different outputs.

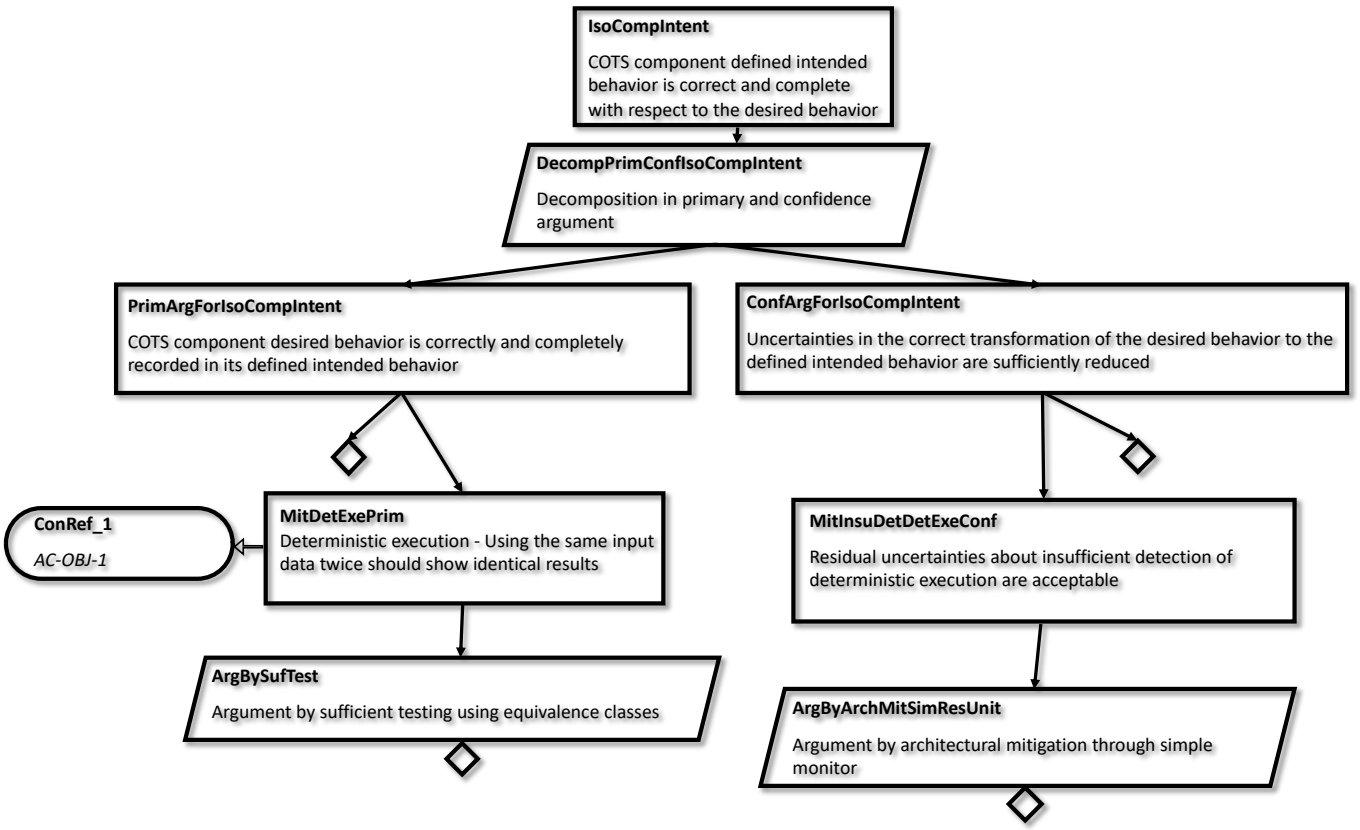

Figure 6. Integration example of an assurance objective most probable for approximate computing architectures

4 Discussions

In this section we particularly discuss two topics, the first regards the completeness of existing guidance and the second DAL modulation.

4.1 Existing certification guidance

One can argue that existing COTS assurance objectives in the NPA [5] alone is sufficient for implementing new COTS technology since all six sub-claims as described above are covered, but this is not true. New COTS technology may implement behavior which cannot be assured using existing guidance, e.g. statistical testing. The convincing arguments then need to address other assurance approaches such as architectural mitigation, see Section 3.2. The same is true for other new technologies. The convincing approach is thus first to implement the eight COTS objectives from [5] and then complement with sufficient arguments for the new (or emergent) technology to be used. This argumentation can be implemented using our framework. The hard part is to understand when the convincing arguments are sufficient.

4.2 Design assurance level modulation

In our approach we have not considered a DAL modulation so far. On a conceptual level it is reasonable to have such a mechanism so that the AEH manufacturer can indicate for which DAL the provided evidence is deemed to be sufficient. But the association of DALs to certain assurance activities is often debatable and a very subjective task that should be agreed with the certification authority. When the DAL modulation is integrated in our concept it will mainly affect only the confidence arguments.

5 Conclusions

We have in this paper, via different examples, shown that COTS specific assurance objectives can be dealt with through assurance cases using Overarching Properties. We showed how already existing objectives [5] were successfully implemented and then we showed how examples of assurance objectives for new emerging computer platforms can be implemented. Through our framework consisting of five process steps, the applicant will have flexibility to adapt the assurance task for the current project needs. We believe that our results are a way forward to address the assurance of future COTS-based computer platforms. In future work, we will continue working on representative examples to show the strength of our framework.

References

[1] A. Schwierz and H. Forsberg, "Assurance case to structure COTS hardware component assurance for safety-critical avionics," in 2018 IEEE/AIAA 37th Digital Avionics Systems Conference (DASC), IEEE, 2018, Electronic ISBN: 978-1-5386-4112-5.

[2] H. Forsberg and A. Schwierz, "Emerging COTS-Based Computing Platforms in Avionics Need a New Assurance Concept," to appear in 2019 IEEE/AIAA 38th Digital Avionics Systems Conference (DASC), IEEE, 2019.

[3] RTCA, DO-254 Design Assurance Guidance for Airborne Electronic Hardware, 2000.

[4] SAE Aerospace, ARP4754A: Guidelines for Development of Civil Aircraft and Systems , Rev. A, 2010.

[5] EASA, Notice of Proposed Amendment 2018-09, "*Regular update of AMC-20:AMC 20-152 on Airborne Electronic Hardware and AMC 20-189 on Management of Open Problem Reports*," TE.RPRO.00034-006.

[6] EASA, "*EASA CM – SWCEH – 001 Development Assurance of Airborne Electronic Hardware*," EASA, Issue 1, Rev. 2. 2018.

[7] Origin Consulting, GSN Community Standard Version 1, 2011.

[8] J. Wlad, Verocel, "Certification initiatives ongoing for unmanned aircraft systems," *Military Embedded Systems*, April 26th, 2018.

[9] D. Rinehart and J. Knight, "*Understanding what it means for assurance cases to "work"*", NASA, Tech. Rep. NASA/CR-2017-219582, 2017.

[10] M. Holloway, *Understanding Assurance Cases: An Educational Series in Five Parts*, NASA, 2015. [Online]. Available: https://shemesh.larc.nasa.gov/arg/uac-all5.pdf [Accessed: 2019-07-08].

[11] G.-A. Berthon, "A Structured Assurance Case for Commercial Off-The-Shelf (COTS) Airborne Electronic Hardware (AEH)," SAE Technical Paper 2018-01-1939, 2018, doi:10.4271/2018-01-1939.

[12] G. A. Berthon, L. H. Mutuel, and C. Marchand, *DOT/FAA/TC-xx/xx: Final Report for System-Level Assurance of Airborne Electronic Hardware*, FAA, 2017.

[13] M. C. Holloway, DOT/FAA/TC-xx/xx: *Understanding the overarching properties: first steps*, Limited release document, September 2018.

[14] R. Hawkins, T. Kelly, J. Knight, and P. Graydon, "A New Approach to creating Clear Safety Arguments", in *Advances in Systems Safety*, C. Dale and T. Anderson, Eds., Springer London, pp. 3–23, 2011, ISBN: 978-0-85729-132-5.

[15] C. M. Holloway and P. J. Graydon, *DOT/FAA/TC-17/67: Explicate '78: Assurance Case Applicability to Digital Systems*, FAA, 2018.

[16] W-m. Whu, *Heterogeneous System Architecture: a New Compute Platform Infrastructure*. First edition, Amsterdam, Netherlands: Morgan Kaufmann, 2016. Print.

[17] H. Houcine, L. T. Yang, J. Xue, and E. Villar "Special Issue on: Heterogeneous Architectures for Cyber-Physical Systems (HACPS)," *Microprocessors and Microsystems,* vol. 52, pp. 333–334, 2017.

[18] LeCun, Y., Bengio, Y., & Hinton, G., "Deep learning". *Nature*, No. 521, 2015, pp. 436-444.

[19] C. Schorn, A. Guntoro, & G. Ascheid, "Efficient on-line error detection and mitigation for deep neural network accelerators," in International Conference on Computer Safety, Reliability, and Security, Springer, Cham, 2018, pp. 205-219.

[20] T. Gonschorek, M. Filax, & F. Ortmeier, "A very first glance on the safety analysis of selflearning algorithms for autonomous cars," in International Conference on Computer Safety, Reliability, & Security, 2018.

[21] M. Ammar Ben Khadra, "An Introduction to Approximate Computing," arXiv:1711.06115v2, 2017.

[22] A. Agrawal *et al.*, "Approximate computing: Challenges and opportunities," 2016 IEEE International Conference on Rebooting Computing (ICRC), San Diego, CA, 2016, pp. 1-8.

[23] M. Samadi, D. A. Jamshidi, J. Lee, and S. Mahlke, "Paraprox: Pattern-based approximation for data parallel applications," in *ACM SIGPLAN Notices*, vol. 49, no. 4, Feb., pp. 35-50, 2014.