Emerging COTS-Based Computing Platforms in Avionics Need a New Assurance Concept

Håkan Forsberg
School of Innovation, Design and Engineering
Division of Intelligent Future Technologies
Mälardalen University
721 23 Västerås, Sweden
Email: hakan.forsberg@mdh.se

Andreas Schwierz
Research Center:
Competence Field Aviation
Technische Hochschule Ingolstadt
85049 Ingolstadt, Germany
Email: Andreas.Schwierz@thi.de

***Abstract*— A new assurance concept for new upcoming COTS-based computing platforms have to be based on a framework that allows to respond to various assurance challenges of different types of COTS hardware technologies. Therefore, we propose to use the generic assurance approach of the Overarching Properties, currently under research, together with assurance case as a tool to get the needed flexibility in the way to argument that the COTS assurance objectives are met. Indeed, to achieve this, it is necessary to develop a concept about COTS assurance in general which is realizable with an assurance case-based Overarching Property approach. This we have already provided in [1]. In this paper we have refined our work to integrate COTS technology specific assurance objectives and explained how their demonstration can be made within this new assurance concept in a coherent way.**

Keywords—safety-critical avionics, assurance strategy, assurance case, COTS assurance, overarching properties, computing platforms

I. INTRODUCTION

When developing civil airborne electronic hardware (AEH), the use of the guidance document RTCA/DO-254 [2] has been the de facto standard since 2005 to ensure design assurance. Design assurance is the process of planned and systematic activities to be confident that errors in requirements, design and implementation have been taken care of to an adequate level for the specified system [3]. Electronic hardware is, however, usually designed with one or several commercial-off-the-shelf (COTS) hardware components not developed according to the specified process. Use of COTS components therefore need other assurance guidance.

RTCA/DO-254 indeed includes additional COTS assurance guidance in the additional considerations chapter not part of the main sections. However, when the RTCA/DO-254 document was developed, the use of COTS components was not common, and the components were not very complex. Therefore, several other COTS assurance guidance documents have been identified by the certification authorities during the years. See Section IV in [1] for a literature review. When new technology has been introduced, new assurance activities have been suggested by the authorities. Usually with sources from different research groups or reports.

The COTS assurance activities often depend on the complexity and functionality of the components. For instance, EASA's Certification Memorandum SWCEH-001 [4] (non-binding guidance material), Section 9, gives guidance for COTS integrated circuits (excluding the core processing unit), COTS microcontrollers and highly complex COTS microcontrollers including multicores. The assurance activities to be performed depend on design assurance level (DAL), complexity of the device and the device's service experience. In Section 10 in [4], COTS graphical processors (GPUs) are addressed, but only for airborne display applications. How about other applications using the GPU for numerical calculations? In addition, the guidance material for GPUs assumes a discrete graphical processor that has a very short life and therefore has an increased possibility of design errors, is complex, contains configurable elements, may exhibit performance variations over production time, and may completely lack empirical data on the actual failure rates experienced in avionics applications [4]. In this case, the guidance material explicitly assumes several low-level activities to be performed for all GPU devices, but nothing or very little concerns the use of the GPU for normal calculations. Furthermore, if a COTS intellectual property (IP) microprocessor core is used, other guidance material, closely related to RTCA/DO-254, should be used. The latest guidance from the certification authorities addressing COTS assurance (including COTS IP) is included in a Notice of Proposed Amendment [5], which is a joint FAA and EASA effort.

How about the use of emerging computing platforms? For instance, components integrating central processing units and graphics processors in the same chip, used in a transparent manner for the programmer. The graphics processors may be used for traditional floating-point calculations not aimed for display applications. Some of these devices also include a programmable logic area. If used, design assurance for the programmable part should follow RTCA/DO-254. Other new computing platforms include machine learning, which cannot be tested in the same manner as traditional deterministic hardware or software but rather needs to rely on other assurance, or microcontrollers developed for the automotive domain. For the latter, we claim there might be benefits for using these devices in avionics applications [6]. Furthermore, the more futuristic concept of approximate computing may even face some safety-critical airborne systems for certain applications. How about guidance for such computing platforms?

Clearly, there is a strong need for a new assurance concept for COTS-based computing platforms in avionics. Such

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

concept 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 demonstrate the use of an assurance case to structure COTS hardware components assurance for safety-critical avionics. G. Berthon [8] has used a similar approach, i.e. a structured assurance case for COTS airborne electronic hardware. Berthon suggests a design assurance level (DAL) based evidence approach for COTS hardware, to support six sub-claims, which origins from top level requirements from CS-25 certification specification and FAR 25 airworthiness standards. The use of assurance cases is in line with FAA's "Streamlining assurance process", i.e. 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 [9].

II. New Computing Platforms

Even though the avionics market is very conservative when it comes to bringing in new technology in safety-critical systems, the high pace of introducing new technology in the commercial sector affects the avionics business in the long run too.

This section introduces some particular technologies that might be introduced in the near future in avionics systems. Since no or little certification guidance exist today for these kinds of technologies, the certification authorities have to spend some effort in creating new guidance, with sometimes low-level activities, which may only be useful for few devices.

A. Heterogeneous systems architectures

Heterogeneous computing platforms exploit massive parallelism from non-traditional computing devices such as graphical processing units (GPUs) or digital signal processors (DSPs), to achieve high performance computations at low energy, and traditional central-processing units (CPUs) for latency-sensitive serial parts of the code [10]. Medical imaging, computational photography and fluid dynamics are areas where heterogeneous platforms have been successful [10]. New programming models and compilers, hardware/software interface, run-time support, load balancing and scheduling policies are all challenges to take advantage of the heterogeneous architectures [11]. One heterogeneous platform initiative is the Heterogeneous Systems Architecture (HSA) Foundation [17]. HSA uses a hardware platform and a software stack to allow seamless operation between several different types of processing elements including CPUs, GPUs, DSPs, crypto engines and various coding and encoding units, using only one unified application programming interface [10]. These heterogeneous COTS components do not have any certification guidance ready for the avionics market. If existing guidance would be allowed, probably most of it should be required, and yet not be sufficient for all certification aspects of these platforms, if used in higher design assurance levels.

B. ISO 26262 developed microcontrollers

In [6], we demonstrated design assurance benefits of using ISO 26262 developed microcontrollers for avionics applications. We did not consider new design assurance methods but relied on current aviation certification praxis. However, the concept of developing ISO 26262 microcontrollers, can itself be seen as an alternative design assurance approach, even for avionics. Not for all redundant units coping with physical errors, but for the requirements-based development and software controlling the low-level hardware. On a very low hardware level, software can and should be used for controlling correct operation of the microcontroller.

C. Architectures using approximate computing

Approximate computing is when some errors are allowed to happen in the computations, i.e computation accuracy is traded for better performance or energy consumption [12]. Leon *et al.* [13], for instance, have been able to reduce the energy consumption by 69% using a hybrid type of algorithm for approximate computing. In approximate computing, different types of reduction of computation accuracy can be used, e.g. reduced number of bits in the arithmetic operations, reduced number of loops in loop constructions (loop perforation), approximate findings of results from expensive function calls (approximate memoization), or relaxed synchronization in parallel applications [20, 23]. These kinds of algorithms may be used for applications such as machine learning, computer vision, and speech. Approximate computing techniques may also be used for future COTS based computing platforms, when CMOS technology moves to 7 nm processes and beyond with reduced reliability, where hardware faults might occasionally propagate to software [12]. From a COTS perspective, the following general guidelines should be followed:

- The computing algorithm used is static when deployed, i.e. no dynamic (self-adaptive) algorithms should be used;
- 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);
- it should be possible to demonstrate success via statistical simulation to some percent at a system level; and
- it should be possible to quantify the probability of an undetected, misleading error and show that the error is appropriate to the function

III. New Assurance Concept Based on Assurance Case

The provision of dedicated and comprehensive guidance material for every new upcoming technology, relevant for avionics systems, is an impractical endeavour. Especially, if it is supposed to be a guidance with a level of detail to assure an item in a process-based way as comparable to what is stipulated in RTCA/DO-178C [24] or RTCA/DO-254 [2]. This kind of classical prescriptive guidance explains a development approach based on the assumption, if it is followed it is sufficiently assured that the realized item will operate with integrity according to its specification. This approach is called *development assurance* as confidence about item's integrity is inferred from the adherence to a prescribed development approach based on commonly accepted practices.

A development assurance strategy is not adequate for all kinds of items. As COTS hardware components are already produced the application of development assurance is impractical for demonstration of its integrity in the envisaged avionics system context. That is why for these items and other new technologies often argumentative approaches are proposed allowing the flexibility to use other more appropriate methods by enabling to show directly how they contribute in meeting the assurance objectives. This is necessary because assurance is not demonstrated by "blind" adherence to prescribed process-based guidance.

In the assurance case framework, assurance refers to the proven confidence that a top-level claim of an argument is true. Therefore, assurance cases work differently than traditional prescriptive methods or guidelines. A specialized form of assurance cases, safety cases, has been successful in certain environments for decades [15]. The strength with assurance cases is the motivation for developers to formulate explicit arguments clearly targeting a top-level claim. So why are assurance cases so beneficial for emerging COTS-based computing platforms? Rinehart and Knight [15] have claimed potential benefits for assurance cases. In the list below, we have explained these benefits in the context of emerging computing platforms:

- *Assurance cases are more suitable to address a broader variety of conventional methods and are not constrained to specific processes or techniques* - This is one of the main arguments for using assurance cases for emerging computing platforms. These platforms need the flexibility to use a mixed selection of methods for convincing arguments for using COTS.

- *Assurance cases address modern certification challenges* – prescriptive methods do not work well for new innovative systems where the complexity is high or emerging technologies are used. Assurance cases can address the use of new design assurance methods or a mix of old prescriptive methods and new assurance methods in order to reach a satisfactory level of assurance. Again, an important argument for using assurance cases.

- *Assurance cases offer a more effective certification path than other approaches* – Assurance cases can extend traditional prescriptive regulations to include more flexible certification paths in a structed manner, which is important for new technologies or novel systems.

- *Organization of information is improved by assurance cases* – Assurance cases explicitly show how the outcome of various conventional methods contribute in assurance, and thus help keeping the overview for both the system developer and the certification authority. The communication between these bodies is also streamlined.

- *The allocation of responsibility is improved by assurance cases* – emerging computing platforms are very complex and may benefit from apportioning parts of the design to responsible stakeholders. However, for the avionics business, it is still the system integrator that need to take the overall responsibility for the system and correct use of COTS components, and thus this benefit is of less importance for this field.

Another important benefit with assurance cases is that it forces people to think more deeply than otherwise [16]. This is very important for emerging technologies and would eventually remove some certification activities, i.e. those uncertain to help in design assurance for a particular platform. Compare with certification activities for GPU related platforms only useful for graphical applications.

To explain why a chosen assurance method is sufficient "structured assurances cases" are an adequate tool. They give the case writer the possibility to demonstrate or explicate in a reviewable argument its assurance strategy in its entirety. This allows a third party 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 integrated in the system.

Currently, the usage of assurance cases on item level in the avionics domain is not a practised method[1]. Therefore, no official guidance material is available at all that could be used as a basis to plan how the assurance strategy shall be created with an assurance case. But there are plenty of academic publications about usage of assurance cases for different applications. Also, some technology orientated references from certification authorities are available which should be addressed in assurance for a specific class of item. This material provides valuable knowledge to arrange the assurance case on widespread and acceptable concepts.

A. Assurance case notation

For the assurance case notation, we use a graphical notation based on a subset of the Goal Structuring Notation (GSN). GSN is defined in [7]. Figure 1 shows the symbols we use in this article.

The *goal* element illustrates claims and sub-claims supporting higher-level connected claims. The *SupportedBy* relationship creates a series of connected claims to establish an overall claim. A goal can be left intentionally undeveloped for later investigations, i.e. an *Undeveloped Goal*. A *Strategy* helps explaining or argument the logic between a goal and its supporting goals. Finally, to clarify concepts mentioned in

[1] For guidance how to structure an assurance case for another domain (Air Traffic Services), see [18].

strategies, a *Context* element with a corresponding *InContextOf* relationship are used.

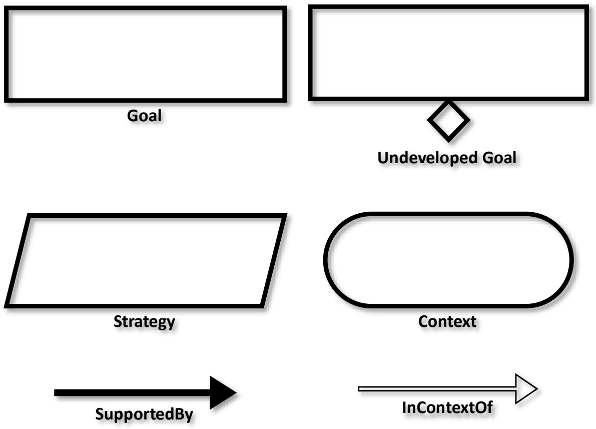

Fig. 1. Used symbol subset of the GSN

B. *Overarching properties as the root*

In previous research [1], we proposed to use an assurance case to structure the assurance of a COTS hardware component. In this paper, we extend previous research with a flexible framework that can respond to various assurance challenges of different types of COTS components. In [1] we defined a top goal for assuring a COTS integrated in safety-critical avionics. The top goal is – *COTS component operates demonstrably airworthy in its system context*. The top goal is based on the European certification authority's precondition that applicable functional and safety certification specifications (CS) requirements also have to be satisfied by COTS components. However, CS requirements are not directly applicable for COTS components but have to be derived for the current level and to what are important to be demonstrated at that level. This work has been performed by Berthon *et al.* [14]. They identified six key objectives applicable to all kinds of AEH based on EASA CS requirements. These objectives constitute context to our top assurance goal.

From our top goal, we identified a strategy splitting our case further in two branches [1]:

1. Argument over initial airworthiness
2. Argument over continuous airworthiness

For assuring COTS hardware's initial airworthiness it is mandatory to substantiate on a component level that the desired behavior is present. Therefore, we proposed to split the argument in two branches, see Figure 2:

1. The isolated COTS component exhibits the desired behavior.
2. The integrated COTS component exhibits the desired behavior on LRU/board-level.

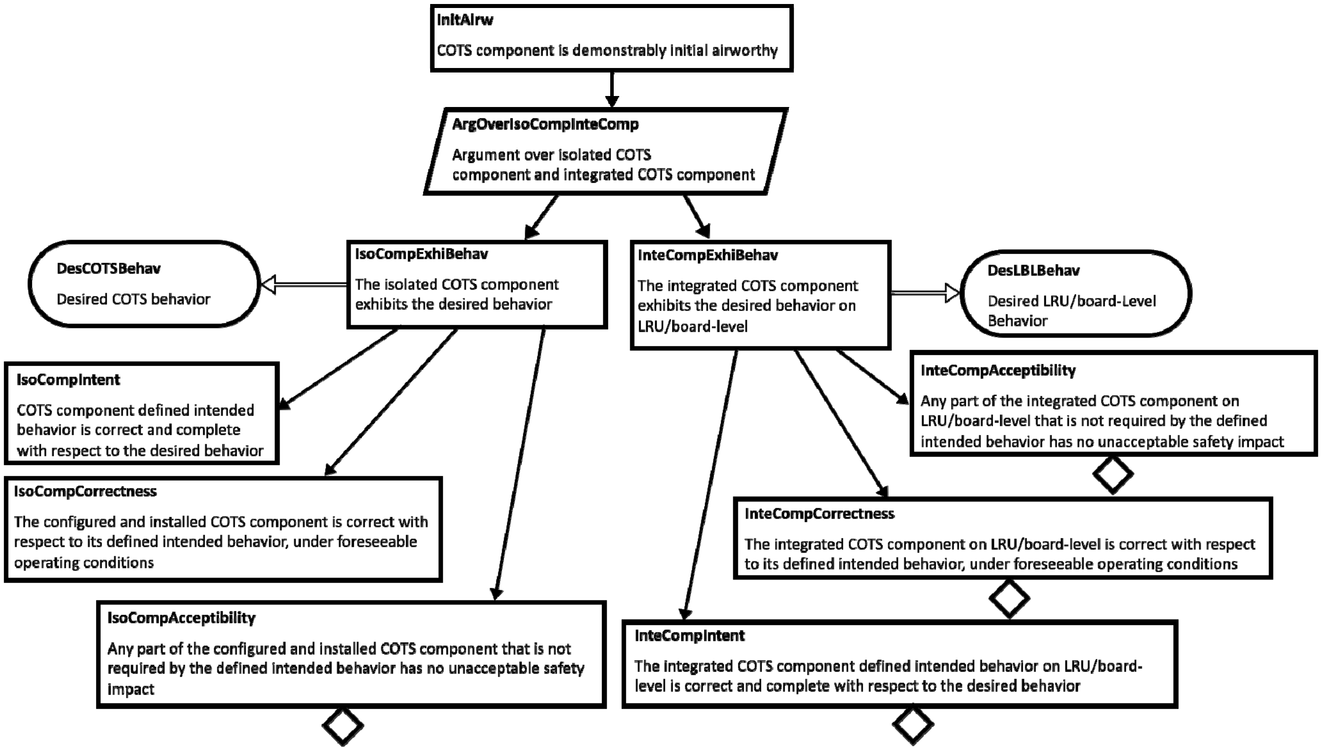
InitAirw
COTS component is demonstrably initial airworthy
ArgOverIsoCompInteComp
Argument over isolated COTS component and integrated COTS component
DesCOTSBehav
Desired COTS behavior
IsoCompExhiBehav
The isolated COTS component exhibits the desired behavior
InteCompExhiBehav
The integrated COTS component exhibits the desired behavior on LRU/board-level
DesLBLBehav
Desired LRU/board-Level Behavior
IsoCompIntent
COTS component defined intended behavior is correct and complete with respect to the desired behavior
InteCompAcceptibility
Any part of the integrated COTS component on LRU/board-level that is not required by the defined intended behavior has no unacceptable safety impact
IsoCompCorrectness
The configured and installed COTS component is correct with respect to its defined intended behavior, under foreseeable operating conditions
InteCompCorrectness
The integrated COTS component on LRU/board-level is correct with respect to its defined intended behavior, under foreseeable operating conditions
IsoCompAcceptibility
Any part of the configured and installed COTS component that is not required by the defined intended behavior has no unacceptable safety impact
InteCompIntent
The integrated COTS component defined intended behavior on LRU/board-level is correct and complete with respect to the desired behavior

Fig. 2. Initial airworthiness argument including both the isolated COTS component as well as when integrated on LRU/board-level

Usually, certain behavior of the COTS component can only be verified on a higher level of integration. In the second branch, see Figure 2, the COTS device has to be installed on the hardware item. Also, for the case that the COTS hardware assurance cannot be supported sufficiently by component level attributes, higher level mechanisms can be designed and argued to close this gap eventually.

At this point the Overarching Properties (OPs) initiative administrated by the FAA comes into the picture [19]. An entity that demonstrably possesses the three OPs labeled with *Intent*, *Correctness* and *Acceptability* will be granted an approval to be used on an aircraft. The informal meaning of the OPs is expressed as following [19]:

- Intent: "What the product is supposed to do is properly captured."
- Correctness: "The product does what it is supposed to do."
- Acceptability: "The product does not cause harm", since made development decisions do not compromise the original safety assessment.

The OPs build the next layer in the assurance case, see Figure 2. For the purpose of COTS assurance, they were adapted [1]. The isolated and the integrated COTS component should demonstrate that the OPs are possessed on each level[2]. This can be achieved by further development of the assurance case.

Up to that 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 has to provide a strategy to enable the demonstration of the OPs together with considering the COTS technology dependent assurance points of interest.

C. Primary and confidence arguments

The driving idea behind the OPs is that this approach shall provide a unified method for the approval of different kinds of entities. So that this approach can be an alternative to already used assurance methods. With this in mind, OPs have been devised in a way to enable its demonstration by application of the accepted development assurance methods for the triad of system (ARP4754A), hardware (RTCA/DO-254) and software (RTCA/DO-178C) together with an active safety assessment (ARP4761).

For demonstration of each adapted overarching property in Figure 2, the next level of argument structure has to be developed. The OPs and standards like RTCA/DO-178C are comparable on a conceptual level. Holloway and Graydon [25] have impressively shown how a transformation of the RTCA/DO-178C in an assurance case can look like. They used a proposed separation by Hawkins *et al.* [26] between a primary and a confidence argument. This concept helps to keep the primary argument focused and clear including only arguments that directly support the claimed attributes of interest. For a safety case, safety is the attribute of interest. According to Hawkins *et al.* this means in the primary argument only things should be added that show how system hazards are identified and mitigated in order to reduce the risk inherent in the product usage; this directly demonstrates safety on a product level. Development artefacts are only referenced in the case as supporting evidence or context. In case of the *IsoCompCorrectness* for example, see Figure 2, the attribute of interest is the *correctness* between the configured and installed COTS component and the defined intended behavior. So only arguments about the properties of the configured and installed COTS device and about the transformation chain from the defined intended behavior of it, with all intermediate artefacts, should be part of this argument. The realization of this chain of transformation together with its verification is the only direct way of reasoning allowing to conclude that the COTS component correctly behaves as intended.

In contrast, the confidence argument explains why the used methods to produce the evidence and the made conclusions in the primary argument are sufficiently creditable. In case of the *IsoCompCorrectness* the confidence should explain why a reviewer should believe that this chain of transformation was correctly done with sufficiently avoidance of error. The confidence argument can be driven by obvious or / and critical assurance deficits in the primary argument. Since, they have to be resolved in order to demonstrate the residual uncertainty is acceptable managed.

Figure 3 shows the further development of the *IsoCompCorrectness*. This kind of decomposition is adaptable for the other OPs under the *IsoCompExhiBehav* and the *InteCompExhiBehav* claim.

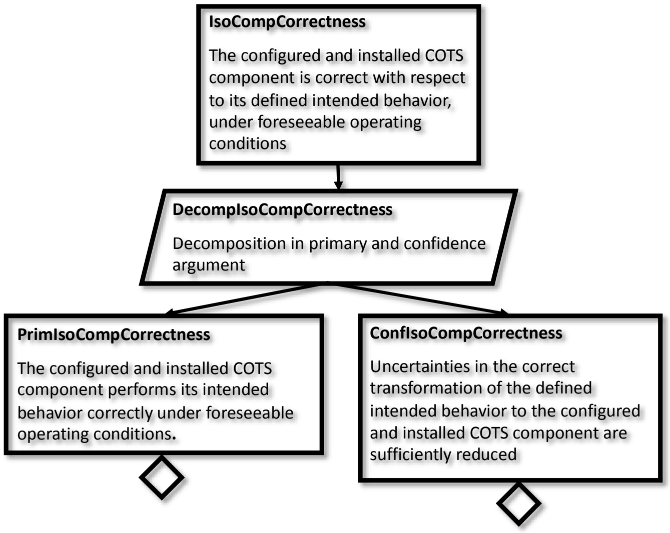
IsoCompCorrectness
The configured and installed COTS component is correct with respect to its defined intended behavior, under foreseeable operating conditions
DecompIsoCompCorrectness
Decomposition in primary and confidence argument
PrimIsoCompCorrectness
The configured and installed COTS component performs its intended behavior correctly under foreseeable operating conditions.
ConfIsoCompCorrectness
Uncertainties in the correct transformation of the defined intended behavior to the configured and installed COTS component are sufficiently reduced

Fig. 3. Primary and confidence decomposition for IsoCompCorrectness

[2] For further explanations about OPs w.r.t. COTS component assurance, see [1].

Chelini *et al*. (OP working group participant) gives an abstract example to demonstrate the compliance to the OPs with an assurance case with evaluation criteria [21]. They also decided to use the differentiation in the next layer of the case between primary and confidence argument.

D. *Integration of COTS assurance objectives*

Since COTS hardware components cannot be assured with a prescriptive development standard such as the RTCA/DO-254, certification authorities have provided dedicated guidance for COTS assurance. In the beginning, they were more concentrated on the description on what kind of activities should be performed. This approach has been changed, so that the latest guidance material, as in the Notice of Proposed amendment [5], is focusing on the adherence of certain assurance objectives.

Although assurance of new emerging COTS-based computing platforms can be addressed to a large extent with already available guidance, they have some peculiarities or critical assurance points that are only specific for themselves. The applicant has to analyze for such platforms if any additional assurance objectives have to be taken care of.

The demonstration that each objective is satisfied can be a very demanding endeavor. For that, the applicant usually needs the flexibility to use different kinds of methods to define a strategy which can be show sufficient.

With our framework we want to provide a concept to connect the demonstration of assurance objectives directly with a new COTS assurance concept based on OPs and assurance cases. To bring them together we developed the following process steps:

1. Choose the level on which the assurance objective has to be demonstrated.
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.

Figure 4 shows as example how we integrated the assurance objective *COTS-8* from the Notice of the Proposed Amendment [5]. It is about the mitigation of inadvertent alteration of critical configuration registers. According to our process at first it has to be decided if the assurance objective shall be demonstrated on the isolated COTS or on the LRU/board level. For our show case we assume that the COTS component includes a feature, like a CRC protection, to detect or mitigate inadvertent alterations of important registers. Because of that, the objective should be satisfied in the context of the isolated COTS component argument.

As the second step, the relevant OP has to be identified to which the assurance objective should be allocated. Therefore, it is necessary to think about, how the relevance can be determined. The informal statements about what the OPs seeking to achieve, from Section III-B, are very helpful here. From them we concluded that *COTS-8* constitutes a part of the desired behavior of the COTS component which have to be correctly transformed in the defined intended behavior of the COTS device. Since, one result of the demonstration of *COTS-8* will be requirements about how the COTS hardware should be configured.

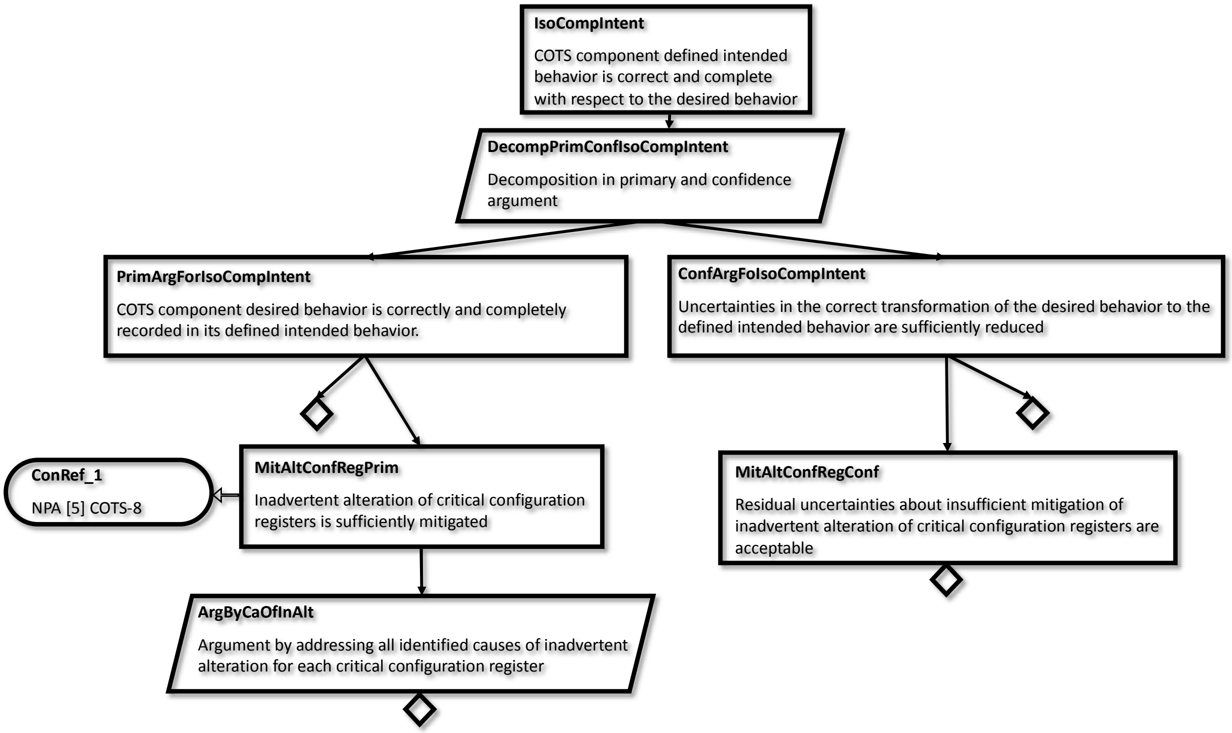
IsoCompIntent
COTS component defined intended behavior is correct and complete with respect to the desired behavior
DecompPrimConfIsoCompIntent
Decomposition in primary and confidence argument
PrimArgForIsoCompIntent
COTS component desired behavior is correctly and completely recorded in its defined intended behavior.
ConfArgFoIsoCompIntent
Uncertainties in the correct transformation of the desired behavior to the defined intended behavior are sufficiently reduced
ConRef_1
NPA [5] COTS-8
MitAltConfRegPrim
Inadvertent alteration of critical configuration registers is sufficiently mitigated
MitAltConfRegConf
Residual uncertainties about insufficient mitigation of inadvertent alteration of critical configuration registers are acceptable
ArgByCaOfInAlt
Argument by addressing all identified causes of inadvertent alteration for each critical configuration register

Fig. 4. Example of integration of the assurance objective *COTS-8* from the Notice of Proposed Amendment [5]

At next, the assurance objective has to be reformulated to expressing a goal or conclusion whose content can be substantiated with an assurance case. This have to be done for the primary and confidence argument.

In the primary argument of *IsoCompIntent* it has to be shown for many parts of the desired behavior, that they are completely and correctly recorded in the defined intended behavior. In the example in Figure 4 we considered only one assurance objective. The lonely square shaped like a diamond shall indicate, that also other assurance objectives have to be considered here. For *COTS-8* the applicant has to define a strategy to demonstrate that it is addressed adequately. This is totally dependent on the selected COTS product and the context in which it shall be integrated. That is why, the rest of the primary argument is only outlined.

Separated from the primary argument, the applicant have to show how the residual uncertainty about the demonstration of *COTS-8* is sufficiently low. This have to be done in the confidence argument *MitAltConfRegConf.*

IV. Related Work and Discussions

The concept of OPs is a topic that is under development. In parallel to this, the RESSAC (Re-Engineering and Streamlining the Standards for Avionics Certification) research project determined a prototype to examine how the OPs are applicable in a realistic environment[3]. In one case study about COTS AEH, it was included to demonstrate how a defined COTS assurance process can be applied to the OPs [22]. Unfortunately, results about the assessment with respect to the OPs are not included.

In [8] G. Berthon used an assurance case approach to clearly express the need from the certification specification for COTS assurance and how this need can be demonstrated. He showed that his proposed approach has similarities with the OPs but did not further elaborate with this relation.

A. Weaknesses of assurance cases

When using assurance cases there are certain things that have to be treated carefully. First of all, it takes long time to understand structured argumentation and to find the right arguments. Bad structured arguments and false conclusions can lead to incorrect assurance of a system [16]. Furthermore, assurance cases cannot replace all prescriptive methods, they rather extend the prescriptive methods and help understand the use of them on a higher integration level. Established activities for safety-critical development still have to be performed. In addition, to be useful for the industry, engineers need to understand the structured argumentation, otherwise, there is a big risk assurance cases will not be used. The training of engineers in using assurance cases also need to be relatively short. We see the potential of using assurance cases and a process to explicitly argument for relevant objectives for COTS assurance, but it has to be introduced gradually and in an explanatory way.

B. Selection of platforms

The selection of different types of computer platforms in this article were mainly based on diverse architectures such that our framework could be tested in a broader sense rather than choosing the most emergent architectures for avionics at this time. But we know that heterogeneous platforms are becoming popular in other domains than avionics for dependable platforms, and that some approximate computing ideas are already discussed for certain avionics applications. The choice of an ISO 26262 developed microcontroller was mainly to show that some artefacts following such COTS components, e.g. the safety manual, indeed may help when the COTS assurance objectives in the avionics industry have to be evidenced. We have not studied any functional suitability for avionics for these ISO 26262 developed microcontrollers.

V. Conclusions

Today's avionics systems frequently integrate COTS components, and in future they will rely even more on these devices. Since they are so inevitable for AEH the necessary assurance approach for such components has to keep pace with this trend.

We are convinced that future concepts for COTS assurance have to consider their specific assurance challenges. In this paper we have proposed a concept which strive to be aligned with the Overarching Properties (OPs) that represents an innovative approach to certify avionics. It gives the applicant more flexibility to adapt the assurance task for the current project needs. We have shown how these OPs are applicable for COTS assurance and proposed that their demonstration should be done with an assurance case. To address these COTS specific assurance objectives with our concept, we devised a five-step process. This helps the applicant demonstrating the objectives in a way compliant with the OP approach.

Our results are based on preliminary research since we have only added some COTS assurance objectives and not all. Furthermore, we have not considered all aspects of the OPs necessary for a complete case. We concentrated only on the tasks necessary to integrate the assurance objectives.

We believe that our results are a way forward to address the assurance of future COTS-based computer platforms. We will continue working on representative examples to further develop our approach and examine in more detail how it can be applied in an industrial environment.

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

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

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

[3] The project repository is accessible under: https://github.com/AdaCore/RESSAC_Use_Case

[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] A. Schwierz and H. Forsberg, “Assurance Benefits of ISO 26262 Compliant Microcontrollers for Safety-Critical Avionics,” 37th International Conference on Computer Safety, Reliability, & Security, 2018, pp. 27-41.

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

[8] 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.

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

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

[11] 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.

[12] M. Ammar Ben Khadra, “An Introduction to Approximate Computing,” arXiv:1711.06115v2, 2017.

[13] V. Leon, G. Zervakis, S. Xydis, D. Soudris, and K. Pekmestzi “Walking through the Energy-Error Pareto Frontier of Approximate Multipliers,” *IEEE Micro* vol. 38, no. 4, pp. 40–49, 2018.

[14] 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.

[15] D. Rinehart and J. Knight, “*Understanding what it means for assurance cases to “work’*”, NASA, Tech. Rep. NASA/CR{2017-219582, 2017.

[16] 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].

[17] HSA Foundation. [Online]. Available: http://www.hsafoundation.com/ [Accessed: 2019-07-08].

[18] Civil Aviation Authority, “*Air traffic services safety requirements*,” CAP670, Safety Policy, 3rd Issue, Amendment 1/2019, 1 June 2019.

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

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

[21] J. Chelini, J. Camus, C. Comar, D. Brown, A-P. Porte, M. de Almeida, and H. Delseny, “Avionics Certification: Back to Fundamentals with Overarching Properties,” ERTS 2018, Jan 2018, Toulouse, France, hal-02156109.

[22] M. De Almeida, *et al.,* ”To provide a process for COTS AEH, ready to apply Overarching Properties (OPs).,” RESSAC Case Study, Rev. 0, NT-2018-037.

[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.

[24] RTCA, DO-178C - Software Considerations in Airborne Systems and Equipment Certification, 2011.

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

[26] 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.