Criticality and Confidence Issues in Avionics

Youssef Laarouchi[1,2], Yves Deswarte[1,2], David Powell[1,2], Jean Arlat[1,2], Eric de Nadai[3]

[1]CNRS ; LAAS ; 7 avenue du colonel Roche, F-31077 Toulouse, France
[2]Université de Toulouse ; UPS, INSA, INP, ISAE ; LAAS ; F-31077 Toulouse, France
[3] EDYMI2 BP M03020, AIRBUS France, 316 route de Bayonne, F-31060 Toulouse, France
[1,2]firstname.lastname@laas.fr [3]eric.de-nadai @airbus.com

ABSTRACT

Ensuring safety in avionics has mostly been achieved through a complete separation between avionics software and open-world software, in order to avoid any interaction that could corrupt critical on-board systems. However, new aircraft generations need more interaction with off-board systems to offer extended services. The extent to which such interactions can be securely supported requires an in-depth characterization, analysis and control of potentially dangerous information flows. In this paper, we consider the safety aspect of such systems and detail the different viewpoints that justify the level of confidence that can placed on a system component.

Keywords

Dependability, safety, fault tolerance, critical systems, avionics.

1. INTRODUCTION

In avionics, every task is given a specific criticality, according to its failure severity. For instance, a task measuring altitude to calculate correct flight parameters is much more critical than one providing In-Flight Entertainment (*IFE*) services. Failure of the former may have catastrophic impact on the flight, while the failure of the latter would just interrupt media services. According to each failure severity, we have to justify the confidence that we must have in the concerned software. Once the different failure modes have been identified and classified, each component is attributed a certain integrity level, according to its criticality. Communication between heterogeneous integrity levels has to be tightly controlled.

2. FAILURE SEVERITY AND CONFIDENCE CONSIDERATIONS

As previously mentioned, the failure severity (*FS*) of a component defines its criticality (criticality of certain tasks or phases of a task). Consequently, criticality is a property that is intrinsic to the task performed by a software and hardware module. To perform a critical task, the assurance that the module satisfies its safety requirements must comply with a certain confidence level (*CL*): the higher the task criticality, the higher the required level of confidence. Moreover, if the outputs of a task are used by a more critical task, the confidence level assigned to these outputs must be raised to the confidence level required by the destination task. This view is consistent with the multiple application domain safety standard IEC 61508 [1] where Safety Integrity Levels (SILs) quantify the confidence levels of modules with respect to accidental faults. A SIL is a discrete level specifying the safety requirements of functions allocated to an electrical/electronic/programmable electronic safety-related system. To each SIL is associated a probability of failure (on demand or per hour).

In the context of safety in the avionics domain, the confidence that can be placed on a module is essentially dependent on three major viewpoints (Figure1): validation (conformance of the design/implementation with the specifications/requirements), credibility (belief in the source of data being processed) and integrity (absence of corruption of the data handled and of the underlying processing resources).

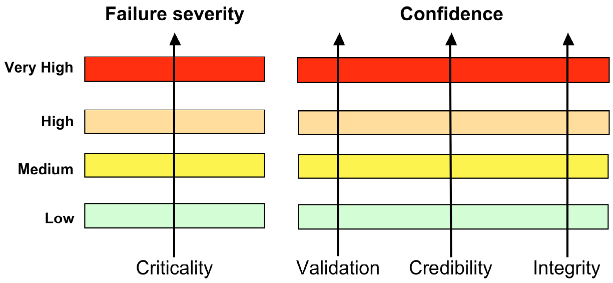

Figure 1. Failure Severity and Confidence

These viewpoints are further discussed in the subsequent sections.

2.1 Validation

By validation, we designate the system development effort devoted to providing assurance that a module respects its specifications. In avionics, ARP4754 [2] is a safety standard dealing with the system development process and how to show compliance to a regulator. This development process concerns both software development (covered in DO-178B [3]) and hardware development (covered in DO-254 [4]). DO-178B defines five software levels, according to the consequences of software failure. These levels are called Development Assurance Levels (DALs) in ARP-4754, and the failures are classified from "*catastrophic*" to "*no safety effects*". Following this classification, a flight control task is considered as DAL-A, while an IFE task is DAL-E (or possibly DAL-D).

2.2 Credibility

Credibility designates the confidence that we can have in the source of data being processed by a given module. The data source can be a human or another module. If the source is a human, its credibility may be defined according to the expertise of the operator in interacting with the module. If it is another module, we have to provide a technical justification of the credibility of the module's outputs. For example, a pressure sensor is a source of pressure data, but it may not be able to provide accurate pressure measures. Consequently, it may have to be considered as non-credible, and this non-credibility has to be taken into account while designing critical tasks using the sensor outputs. One solution may consist in using replicated sensors and implementing a voting mechanism, to raise the credibility level of the data transmitted to the critical tasks.

2.3 Integrity

We designate by integrity the fact that tasks are not corrupted in any way (either accidentally or maliciously) [5]. Integrity levels with respect to malicious attacks have been treated by many security models, such as the Biba [6] and Clark and Wilson [7] models, while Totel's model considers also accidental faults [8]. Integrity can be ensured by means of protection mechanisms. For security, the common criteria [9] define Evaluation Assurance Levels (EAL) to describe the depth and rigor of the evaluation of such mechanisms. In this case, the integrity level is defined by the validation level of theses mechanisms.

2.4 Confidence

Validation, credibility and integrity all affect confidence. Validation ensures that the developed module complies with its specifications, and is based on appropriate methods, recommendations and tools. Once the module is implemented, it needs to process correct data. This property is characterized by the credibility of the data source. The module needs also to continue to perform as specified, by protecting both task and processed data from corruption, which is characterized by integrity.

An interesting property of these viewpoints is that they are not intrinsic to the module, but they can be increased (to satisfy a criticality level) or decreased (to satisfy economic constraints). An optimal threshold could be established in order to satisfy both criticality and economic constraints, as detailed below.

2.5 Confidence level

For a given task *T*, the failure severity level *FSL* has to satisfy $FSL(T) \leq CL(T)$, where *CL* is the confidence level of the module performing the task. Equality represents the optimal deployment, economically speaking. In fact, increasing *CL(T)* requires more validation effort, more accurate sources and/or more efficient protection mechanisms, which is expensive too. If a task *T* is ensured by a module *M*, then it is possible to reduce the cost of increasing *CL* under some conditions. If *M* can be implemented by a fault-tolerant configuration of diversified components *Ci*, then for each *Ci*, DO-178B [3] proposes $CL(Ci) \geq FSL(T)$-1. This approach is interesting because it enables a tradeoff between the cost of a validation at a higher level and the cost of developing i) several diversified components *Ci* validated at a lower level and ii) the fault tolerance decision function, while fulfilling the same safety requirements. However, such a claim can only be justified if there is no significant common failure mode between the diversified components *Ci*.

3. CONCLUSION

In this paper, we have presented criticality as an intrinsic property of a task, and its confidence level as a function of three complementary parameters: validation, credibility and integrity. These parameters are different facets on which we can act to ensure the safety requirements.

In the ArSec project, we are investigating the use of virtualization to host a fault-tolerant configuration of diversified components to provide the increase in confidence level that is needed to allow open-world components to interact with onboard avionics software [10].

4. REFERENCES

[1] IEC 61508, "Functional safety of electrical /electronic /programmable electronic safety-related systems," 2007.

[2] ARP 4754, "Certification Considerations for Highly-Integrated Or Complex Aircraft Systems" 1996.

[3] Radio Technical Commission for Aeronautics (RTCA), "Software Considerations in Airborne Systems and Equipment Certification," *European Organization for Civil Aviation Electronics (EUROCAE), DO178-B*, 1992.

[4] Radio Technical Commission for Aeronautics (RTCA), *Design Assurance Guidance for Airborne Electronic Hardware, DO-254*, 2000.

[5] A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr, "Basic Concepts and Taxonomy of Dependable and Secure Computing," *IEEE TDSC*, vol. 1, no 1, pp. 11-33, 2004.

[6] K.J. Biba, "Integrity Considerations for Secure Computer Systems," *MITRE Co., technical report ESD-TR 76-372*, 1977.

[7] D. Clark and D. Wilson, "A Comparison of Commercial and Military Computer Security Policies," *Proc. 8th IEEE Annual Int. Symp. on Security and Privacy*, Oakland, CA, USA, IEEE Computer Society Press, pp. 184-194, 1987.

[8] E. Totel, J. Blanquart, Y. Deswarte, and D. Powell, "Supporting Multiple Levels of Criticality," Proc. 28th IEEE Annual Int. Symp. on Fault-Tolerant Computing (FTCS-28), Munich, Germany, IEEE Computer Society Press, pp. 70-79, 1998.

[9] "Common Criteria for Information Technology Security Evaluation," *V3.1, CCMB-2006-09-001, CCMB-2006-09-002, CCMB-2006-09-003.*

[10] Y. Laarouchi, Y. Deswarte, D. Powell, J. Arlat, and E. de Nadai, "Enhancing Dependability in Avionics Using Virtualization," Proc. *EuroSys Workshop on Virtualization Technology for Dependable Systems*, Nuremberg, Germany ACM Press, pp. 13-17, 2009.