Verification Scenarios for Databases in Safety-Critical Systems

Sarasuaty Megume Hayashi Yelisetty[1], **Johnny Marques**[1]

[1]Instituto Tecnológico de Aeronáutica (ITA)
São José dos Campos, SP – Brasil

sarasuaty@ita.br, johnny@ita.br

Abstract. *This paper presents comprehensive verification scenarios for databases in safety-critical systems, addressing the unique challenges of stringent regulatory environments. By proposing four distinct verification scenarios, this work ensures that databases used in these systems meet rigorous standards of completeness, correctness, and traceability, as required by RTCA DO-178C. The scenarios include testing databases with application software, using qualifiable data processing and error detection tools, and employing independent data processing tools. The proposed approach enhances the robustness and reliability of databases, contributing significantly to the safety and integrity of critical systems where data quality is paramount.*

Resumo. *Este artigo apresenta cenários de verificação abrangentes para bancos de dados em sistemas críticos de segurança, abordando os desafios únicos de ambientes regulatórios rigorosos. Ao propor quatro cenários de verificação distintos, este trabalho garante que os bancos de dados utilizados nesses sistemas atendam a padrões rigorosos de completude, correção e rastreabilidade, conforme exigido pela RTCA DO-178C. Os cenários incluem o teste de bancos de dados com software de aplicação, o uso de ferramentas qualificáveis de processamento de dados e detecção de erros, e a aplicação de ferramentas de processamento de dados independentes. A abordagem proposta aprimora a robustez e a confiabilidade dos bancos de dados, contribuindo significativamente para a segurança e integridade dos sistemas críticos, onde a qualidade dos dados é primordial.*

1. Introduction

Databases are critical components in software systems, acting as data collections that influence software behavior without altering the executable code, and are managed separately during system architecture definitions [Hernandes 2013]. The quality of this data is paramount, especially in safety-critical systems where poor data quality can severely impact software functionality. The challenge is compounded by the immense volume, rapid velocity, and variety of data sources, necessitating rigorous data validation to ensure high-quality data [Gao et al. 2016][Woodall et al. 2015].

Software errors can have catastrophic consequences in safety-critical environments, such as those governing aircraft, nuclear reactors, and medical devices, including loss of life [Marques and Cunha 2019]. As the complexity and reliance on such systems increase, ensuring the reliability and safety of both software and databases becomes essential. Regulatory agencies enforce stringent certification requirements to guarantee that these systems adhere to rigorous standards, demanding robust processes for verification, configuration management, and quality assurance [Rierson 2013][Marques and da Cunha 2017].

Critical systems, such as air navigation computers and mechanical respirators, rely heavily on their databases' integrity, completeness, and correctness from initial specifications to application software integration. This necessitates robust specification, development, validation, and verification processes [Barros et al. 2020][Xie et al. 2017].

The RTCA DO-178C [RTCA 2011a] is a critical standard for ensuring the safety and reliability of software used in airborne systems. Although it primarily focuses on software development and verification processes, it indirectly impacts the management of databases within these systems. Databases in safety-critical systems must adhere to rigorous standards to ensure data integrity, availability, and security. The RTCA DO-178C framework necessitates that databases are designed, implemented, and tested in a manner that aligns with the overall safety objectives of the software. This includes ensuring that database interactions do not introduce risks that could compromise system safety and that all data handling complies with the stringent requirements for traceability and verification.

RTCA DO-330 [RTCA 2011b]provides guidelines for certifying software tools used to develop airborne systems. It ensures that tools essential for tasks such as design, testing, and verification meet specific reliability and safety standards. By establishing criteria for tool qualification, RTCA DO-330 helps guarantee that these tools do not introduce errors or inconsistencies into the development process, thus supporting the overall safety and compliance of the final airborne system. This standard complements RTCA DO-178C by addressing the quality assurance of the tools themselves.

This paper aims to outline comprehensive verification scenarios for databases within safety-critical systems.

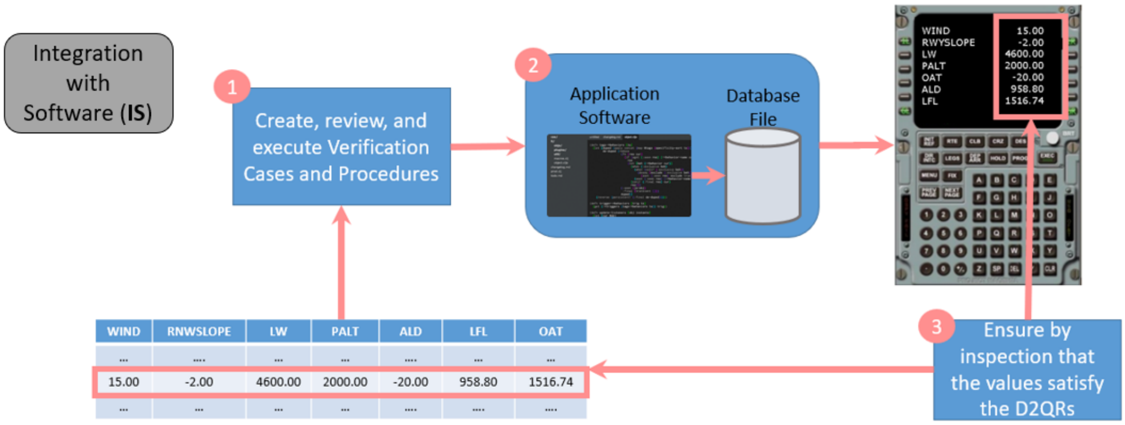
2. Verification Scenarios

The four scenarios are as follows:

- Verification Scenario 1 (VS1) – Testing the Database with the Application Software;
- Verification Scenario 2 (VS2) – Using a Qualifiable Data Processing Tool;
- Verification Scenario 3 (VS3) – Using a Qualifiable Error Detection Tool;
- Verification Scenario 4 (VS4) - Using Two Independent Data Processing Tools.

2.1. VS 1 – Testing the Database with the Application Software

According to [Institute of Electrical and Electronics Engineers 1987], testing involves running a system or component under specified conditions, monitoring the outcomes, and assessing aspects of the system's performance, reliability, or functionality. In VS1, the process involves executing the database with the application software under predefined conditions, observing the results, and evaluating how well the integration works. This is crucial for identifying defects, verifying system requirements, and ensuring the quality and effectiveness of the system. Testing under various scenarios provides valuable insights for further development and optimization. Figure 1 illustrates VS1 in three steps.

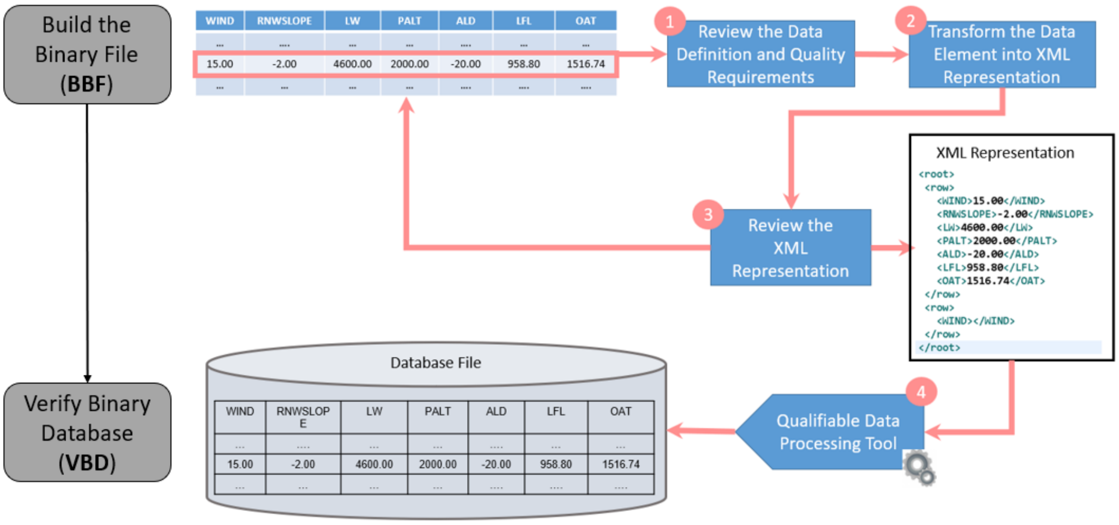

Integration with Software (IS)
1
Create, review, and execute Verification Cases and Procedures
2
Application Software
Database File
WIND
RWYSLOPE
LW
PALT
OAT
ALD
LFL
15.00
-2.00
4600.00
2000.00
-20.00
958.80
1516.74
3
Ensure by inspection that the values satisfy the D2QRs

WIND	RNWSLOPE	LW	PALT	ALD	LFL	OAT
...	....	...	...	....	....	...
15.00	-2.00	4600.00	2000.00	-20.00	958.80	1516.74
...	...	...	...	....	....	....

Figure 1. VS 1 - Testing the Database with the Application Software

2.2. VS 2 – Using a Qualifiable Data Processing Tool

In VS2, depicted in Fig. 2, a Qualifiable Data Processing Tool reads the values defined by Data Definition Quality Requirements (D2QR) using an intermediate representation, such as XML. This scenario ensures that data elements are implemented with correct values, aligning with Database Requirements.


Build the Binary File (BBF)
Verify Binary Database (VBD)
1
Review the Data Definition and Quality Requirements
2
Transform the Data Element into XML Representation
3
Review the XML Representation
XML Representation
4
Qualifiable Data Processing Tool
Database File

WIND	RNWSLOPE	LW	PALT	ALD	LFL	OAT
...	....	...	...	....	....	....
15.00	-2.00	4600.00	2000.00	-20.00	958.80	1516.74
...	...	...	...	....	....	....

```
<root>
 <row>
   <WIND>15.00</WIND>
   <RNWSLOPE>-2.00</RNWSLOPE>
   <LW>4600.00</LW>
   <PALT>2000.00</PALT>
   <ALD>-20.00</ALD>
   <LFL>958.80</LFL>
   <OAT>1516.74</OAT>
 </row>
 <row>
   <WIND></WIND>
 </row>
</root>
```

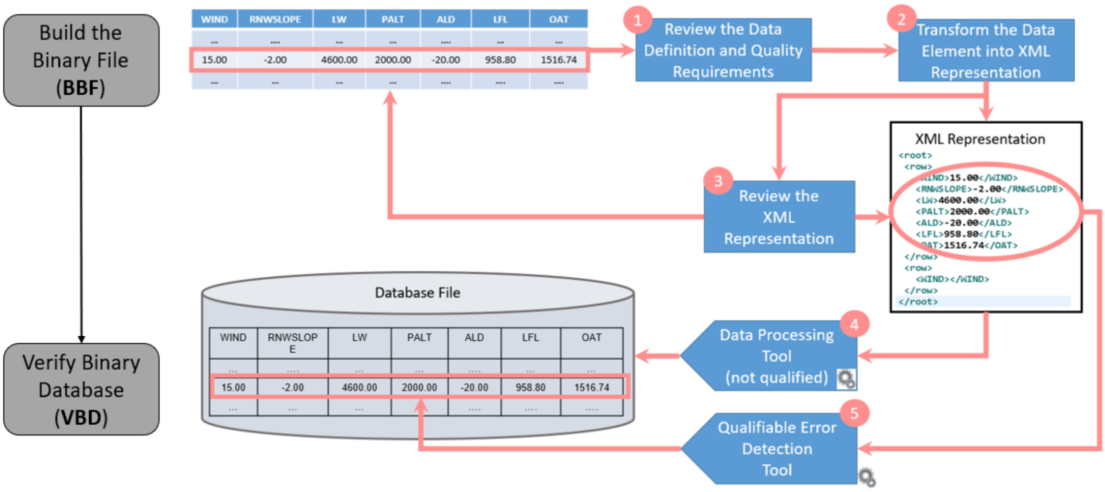
WIND	RNWSLOP E	LW	PALT	ALD	LFL	OAT
...	....	...	...	....	...	...
15.00	-2.00	4600.00	2000.00	-20.00	958.80	1516.74
...	...	...	...	....	....	....

Figure 2. VS 2 - Using a Qualifiable Data Processing Tool

The tool must be thoroughly evaluated to ensure it reliably generates databases that meet the specified D2QRs. This involves rigorous testing under various conditions to confirm that the tool's outputs are dependable and meet all necessary standards without requiring additional verification steps.

2.3. VS 3 – Using a Qualifiable Error Detection Tool

VS3 follows many principles from VS2 but involves using a Qualifiable Error Detection Tool instead of a qualified Data Processing Tool. This scenario addresses cases where the Data Processing Tool is not qualified. The Error Detection Tool independently verifies the equivalence of the XML representation to the Binary Database File, ensuring the integrity of the database. Figure 3 presents the five steps required to ensure that data elements conform to the D2QRs.


Build the Binary File (BBF)
Verify Binary Database (VBD)
1
Review the Data Definition and Quality Requirements
2
Transform the Data Element into XML Representation
3
Review the XML Representation
Database File
4
Data Processing Tool (not qualified)
5
Qualifiable Error Detection Tool

WIND	RNWSLOPE	LW	PALT	ALD	LFL	OAT
...	....	...	....	....	...	...
15.00	-2.00	4600.00	2000.00	-20.00	958.80	1516.74
...	...	...	....	....	....	....

XML Representation

```
<root>
 <row>
   <WIND>15.00</WIND>
   <RNWSLOPE>-2.00</RNWSLOPE>
   <LW>4600.00</LW>
   <PALT>2000.00</PALT>
   <ALD>-20.00</ALD>
   <LFL>958.80</LFL>
   <OAT>1516.74</OAT>
 </row>
 <row>
   <WIND></WIND>
 </row>
</root>
```

WIND	RNWSLOPE	LW	PALT	ALD	LFL	OAT
...	....	...	...	....	...	...
15.00	-2.00	4600.00	2000.00	-20.00	958.80	1516.74
...	...	...	...	....	....	....

Figure 3. VS 3 – Using a Qualifiable Error Detection Tool

RTCA DO-330 provides comprehensive guidance on tool qualification for both airborne and ground-based software. This guidance applies to various domains, including automotive, space, electronic hardware, aeronautical databases, and safety assessment processes. By adhering to RTCA DO-330, organizations ensure their tools are appropriately qualified, maintaining the integrity and reliability of their software and systems.

2.4. Verification Scenario 4 (VS4) - Using Two Independent Data Processing Tools

In VS4, depicted in Fig. 4, the data subset is converted into an intermediate format, such as XML (Step 1), to be processed by two different Database Generation Tools, neither of which is qualified (Step 2). After the final generation, the two independently produced Binary Database Files must be verified for equivalence (Step 3).

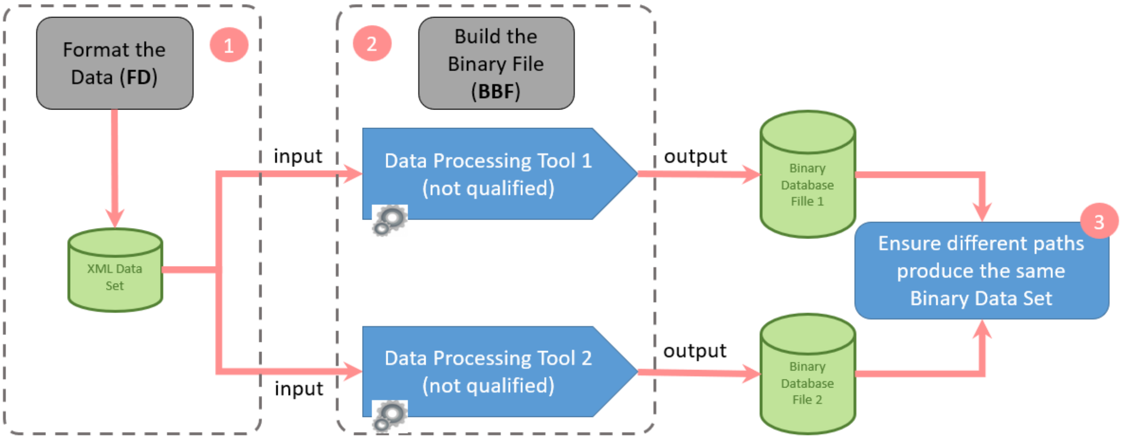

Format the Data (FD)
1
XML Data Set
input
input
2
Build the Binary File (BBF)
Data Processing Tool 1 (not qualified)
Data Processing Tool 2 (not qualified)
output
output
Binary Database Fille 1
Binary Database File 2
3
Ensure different paths produce the same Binary Data Set

Figure 4. VS4 - Using Two Independent Data Processing Tools

3. Conclusion

Developing safety-critical systems, such as those controlling aircraft, nuclear reactors, and medical devices, requires adherence to strict certification requirements due to the inherent risks involved. The software development process in these environments must be robust, with thorough verification, meticulous configuration management, and comprehensive quality assurance to prevent catastrophic outcomes. Ensuring the correctness and completeness of safety-critical software with the highest level of assurance is paramount.

Our work extends these stringent requirements to the databases used in safety-critical systems. We propose validation scenarios for these databases, ensuring compliance with regulatory standards and guaranteeing data completeness and correctness. Key contributions include ensuring traceability to RTCA DO-178C, promoting an organized approach to database construction, and mitigating the risk of errors from specification through integration with application software. Implementing these practices is crucial for minimizing risks and ensuring the integrity and reliability of data in safety-critical environments.

References

Barros, L., Hirata, C., Marques, J., and Ambrosio, A. M. (2020). Generating test cases to evaluate and improve processes of safety-critical systems development. In *2020 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*, pages 311–318.

Gao, J., Xie, C., and Tao, C. (2016). Big data validation and quality assurance– issues, challenges, and needs. In *2016 IEEE Symposium on Service-Oriented System Engineering (SOSE)*, pages 433–441.

Hernandes, M. (2013). *Database Design for Mere Mortals: A Hands-On Guide to Relational Database Design*. Addison-Wesley Professional.

Institute of Electrical and Electronics Engineers (1987). IEEE 610.2 Standard Glossary of Computer Applications Terminology.

Marques, J. and da Cunha, A. M. (2017). Verification scenarios of onboard databases under the rtca do-178c and the rtca do-200b. In *2017 IEEE/AIAA 36th Digital Avionics Systems Conference (DASC)*.

Marques, J. C. and Cunha, A. M. (2019). Ares: An agile requirements specification process for regulated environment. *International Journal of Software Engineering and Knowledge Engineering*, 29(10):1403–1438.

Rierson, L. (2013). *Developing Safety-Critical Software: A Pratical Guide for Aviation Software and DO-178C Compliance*. CRC Press.

RTCA (2011a). Do-178c software considerations in airborne systems and equipment certification.

RTCA (2011b). Do-330 software tool qualification considerations.

Woodall, P., Parlikad, A., and A. Koronios, A. (2015). *Classifying Data Quality Problems in Asset Management*. Springer International Publishing.

Xie, C., Gao, J., and Tao, C. (2017). Big data validation case study. In *3rd IEEE International Conference on Big Data Computing Service and Applications*.