

항공전자시스템 하드웨어/소프트웨어 개발 및 인증 접근 방식에 대한

10가지 제안사항 - 6. 요구사항 기준의 리소스 최적화 달성

정수영^{1*}
수담연구소¹

10 Suggestions for Avionics System Hardware and Software Development and Certification Approach - 6. Resource Optimization based on Requirements

Suyoung Jeong^{1*}

Key Words : Aircraft/System Development(항공기/시스템 개발), Avionics System Certification (항공전자시스템 인증), Requirements(요구사항)

서론

국내에서 수행 중인 항공전자시스템⁽²⁾/HW⁽³⁾/SW⁽⁴⁾에 대한 개발 및 인증에 대해서 AAM/UAM 산업을 효과적으로 대응하고 이를 기반으로 향후 세계 시장까지 고려하는 '글로벌 스탠다드'를 기준으로 한 접근법에 대해서 지난 논문⁽¹⁾에서 10가지 접근법을 제안한 바가 있다.

본 논문에서는 그 중 여섯 번째 제안인 '요구사항 기준의 리소스 최적화 달성'에 대해서 좀 더 상세하게 논하고자 한다.

본론

1. 항공 인증과 요구사항 추적성 100% 달성

항공뿐만 아니라 어느 분야든 개발에 있어서 요구사항의 중요성과 역할에 대해서는 누구나 인지하는 부분이다. 특히 요구사항을 중심으로 설계, 구현, 통합, 시험 등의 모든 과정이 진행된다는 점에서 그 중요성은 이론의 여지가 없을 것이다.

그런데 항공 인증 관점에서 이를 바라보면 다른 분야와 다른 중요한 특징이 하나 있다. 그것은 바로 '요구사항'만 구현된다는 점이다.

어찌보면 너무도 당연한 말처럼 보이지만 항공 인증 관점에서 '요구사항'만 구현된다는 것은 실제 항공기/시스템/HW/SW 개발 및 인증 기준에서는 특별한 의미를 가지게 된다. 그것은 바로 '요구사항이 아니면 절대 구현되지 않는다'는 의미이다.

이를 가장 직관적으로 보여주는 것이 바로 항공기에 탑재되는 SW 요구사항이다. 일반적으로 항공기에 탑재되는 SW는 엄청난 분량의 소스 코드를 가지며 최종적으로는 그 모든 소스 코드가 항공기의 운항 중에 실행된다. 그런데 항공 인증 관점에서는 이러한 소스 코드 전체는 반드시 SW 요구사항과 연결되어야 한다. 즉 추적성이 100% 연결되어야 하는 것이다. 그런데 만약 흑시라도 SW 요구사항과 연결되지 않는 소스 코드가 존재한다면 항공 인증 관점에서는 해당 소스 코드(Dead Code)를 반드시 삭제할 것을 요구하고 있다.

정리하면 항공기/시스템/HW/SW 모두 요구사항에 대해서 '100% 추적성'이라는 속성이 적용된다. 다만 여기서 '100% 추적성'이라는 것은 예외적인 상황에 대한 명확한 근거 제시도 포함되므로 실무적인 관점에서는 단순히 정량적인 기준으로만 접근하는 것은 아니라는 점에 유의하자.

2. 항공 인증을 위한 요구사항의 정의

1장에서 항공 인증 관점에서 요구사항의 중요한 속성을 언급했지만 사실 그 자체로는 다른 분야에서 다루고 있는 요구사항의 속성과 크게 다른 점이 없다고 할 수 있다. 오히려 항공 인증 관점에서 요구사항에 대해 가장 주목해야 할 점은 바로 요구사항을 정의하는 '방법'에 대한 부분이다.

참고로 항공기/시스템/HW/SW 개발 및 인증을 위한 요구사항은 각각에 해당하는 지침서를 기준으로 볼 때 Table 1⁽²⁾⁽³⁾⁽⁴⁾과 같이 조금씩 다른 형태를 가진다.

Table 1. Aircraft/System/HW/SW Requirement

항공기/시스템 (ARP4754A)	HW (DO-254)	SW (DO-178C)
Aircraft/System Requirements	HW Requirements	High-level SW Requirements Derived High-level SW Requirements
Derived Aircraft/System Requirements	Derived HW Requirements	Low-level SW Requirements Derived Low-level SW Requirements

Table 1에서 주목해야 할 부분은 바로 모든 레벨에서 'Derived Requirements'가 존재한다는 점이다. 참고로 우리말로는 '파생 요구사항'으로 번역할 수 있다.

지침서를 기준으로 보자면 '파생' 요구사항은 상위 레벨과의 추적성을 가지지 않을 수 있는 요구사항으로 규정하고 있다. 그런데 이는 1장에서 언급했던 요구사항에 대한 '100% 추적성'과는 사실상 배치되는 개념이라고 할 수 있다. 한편 앞서 '100% 추적성'에는 예

외적인 상황도 포함한다고 했는데 이는 결국 예외적인 상황뿐만 아니라 ‘파생’ 요구사항도 존재할 수 있다는 것을 알 수 있다.

항공 인증 관점에서 요구사항에 대해 가장 주목해야 할 점으로 제시한 요구사항을 정의하는 ‘방법’에는 바로 이러한 ‘파생’ 요구사항의 정의에 대한 부분도 포함된다. 아울러 Table 1을 보면 SW에 대해서 ‘High-level SW Requirements’, ‘Low-level SW Requirements’라는 용어도 볼 수 있다. 참고로 이는 추적성에는 영향을 주지 않지만(100% 추적성에 해당) 요구사항을 정의하는 ‘방법’에는 상당한 영향을 미칠 수 있다.

결론적으로 항공기/시스템/HW/SW 요구사항을 정의할 때에는 앞서 언급된 ‘100% 추적성’과 ‘방법’에 대한 부분이 반드시 고려되어야 한다. 만약 이런 부분이 제대로 반영되지 않은 요구사항이 정의되는 경우에는 그 자체로도 문제가 될 가능성이 있고 그것으로 파생되는 문제점, 그 문제를 해결하기 위해 필요한 추가적인 활동, 산출물 생성, 그로 인한 불필요한 리소스 낭비, 내부 감사 혹은 인증 당국의 감사등을 통한 Finding 발생 가능성 증가, 이로 인한 전체 개발 및 인증 일정의 지연 가능성 등 그 여파가 상상할 수도 있다는 것을 인지할 필요가 있다.

그런데 이러한 부분을 모두 인지한다고 하더라도 결국 관건은 실무자가 제대로 된 요구사항을 작성할 수 있는냐이다. 이에 대해서 본 논문에서 제안하는 것은 바로 실무자의 요구사항 작성 경험을 늘리고 요구사항 작성 능력을 꾸준히 강화하는 것이다. 참고로 이를 위해서 Figure 1⁽⁵⁾과 같이 업계에서 통용되는 샘플을 적극적으로 활용하는 것도 추천하는 바이다.

System level:	The LRU shall transmit Flap Position with Label 204 at a 50.0 Hz +/- 0.1Hertz rate.
LRU level:	The ARINC 429 outputs shall transmit Flap Position with Label 204 at a 50.0 Hz +/- 0.1 Hertz rate.
Circuit Card level:	The ARINC 429 outputs shall output Flap Position with Label 204 at a 50.0 Hz +/- 0.1 Hertz rate.
PLD level:	The ARINC_429_N and ARINC_429_P outputs shall generate a message with Label 204 when a write operation is performed to address 0x30334024 on the processor data bus. The message output shall start within 1 microsecond after the completion of the write operation, and use the data from the write operation as the message payload.
Software level:	Label_204_FLAP_POSITION shall be written to ADDR(0x30334024) every 20 +/- 0.04 milliseconds.

Fig. 1. Requirements Samples

3. 항공 인증을 위한 리소스 최적화

본 논문에서 제안하는 ‘요구사항 기준의 리소스 최적화 달성’은 바로 1장, 2장에서 언급한 ‘요구사항’을 ‘제대로 작성’하는 것과 밀접한 관련이 있다. 특히 항공 인증 관점에서는 가장 근본적이고 핵심적인 부분이라고 할 수 있다.

사실 요구사항 정의와 리소스 최적화가 어떤 관계가 있는지 직관적으로 와닿지 않을 수 있지만 2장에서 잠시 언급한 것처럼 막상 요구사항 하나가 잘못 작성되었을 경우 그로 인해서 발생할 이슈들의 범위와 영향력을 고려한다면 항공 인증 관점에서는 ‘요구사항 작성’만큼 전체 리소스 결정에 막대한 영향을 미칠 수 있는 요인이 없다는 점을 이해할 수 있을 것이다.

참고로 1, 2장에서 직접적으로 언급하지는 않았지만 요구사항은 항공 인증에서 또 다른 핵심 활동인 안전성 평가(Safety Assessment)에도 상당히 중요한 영향을 미치게 되는데 Figure 2⁽²⁾를 통해서 관련된 활동을 유추해 볼 수 있다. 즉, 그림에서 최상단의 박스들이 각 레벨별 요구사항을 정의하는 활동이며 최상단 및 최하단 박스를 제외한 V모델 전체에 걸쳐서 표시되어 있는 작은 네모 박스들이 안전성 평가로써 상호간에 입/출력으로 영향을 주고 받는 것이다.

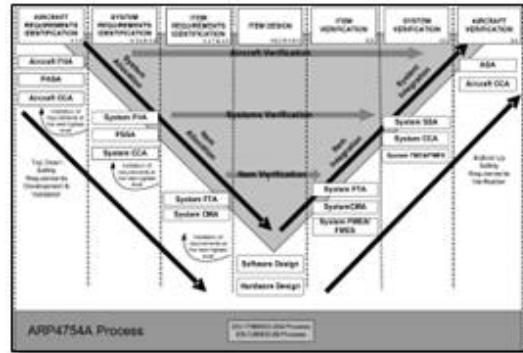


Fig. 2. Safety Assessment & Development

결론

지금까지 본 논문을 통해서 ‘요구사항 기준의 리소스 최적화 달성’과 관련된 요구사항 자체의 중요성과 요구사항 작성 ‘방법’에 대한 부분들을 구체적으로 살펴보았다.

항공 인증을 위한 요구사항 작성은 이러한 고유의 특성에 유의해서 개별 요구사항 하나하나를 신중하게 작성해야 하며 이를 통해서 궁극적으로는 항공 인증에 적합한 리소스 최적화를 달성할 수 있을 것이다.

참고문헌

- 1) Suyoung Jeong, “10 Suggestions for Avionics System Hardware and Software Development and Certification Approach“, ASSK, 2024.
- 2) SAE ARP4754A, “Guidance for Development of Civil Aircraft and Systems“, SAE Aerospace, 2010.
- 3) RTCA/DO-254, “Design Assurance Guidance For Airborne Electronic Hardware“, April 19, 2000.
- 4) RTCA/DO-178C, “Software Considerations in Airborne Systems and Equipment Certification“, December 13, 2011.
- 5) Randall Fulton, Roy Vandermolen, “Hardware Design Assurance - A Practitioner’s Guide to RTCA/DO-254“, CRC Press, 2015, pp. 87.