특집 07

차량 전장 시스템을 위한 신뢰성

한갑수 · 조정훈 (경북대학교)

목 차 »

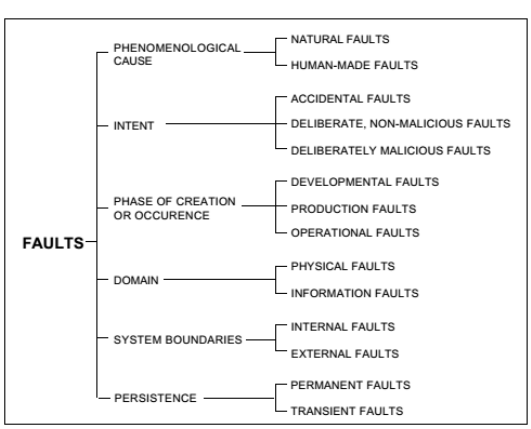
1. 서 론

신뢰성은 시스템이 특정 동작 조건에서 특정 기간 동안 만족할 수 있는 동작을 수행하는 확률이다. 일반적으로 확률은 상수 또는 무작위 고장 발생률(Random Failure Rate)를 사용하며 만족하는 범위는 시스템의 특성에 따라 100% 또는 이중화(Redundancy) 등으로 나타낼 수 있다. 일반적으로 중요한 내장형 시스템(Critical Embedded System)은 예외 상황이 발생하지 않는 경우 동작 기간 동안 정확한 동작을 수행해야 한다. 신뢰성에 대한 연구는 오래 전부터 과부하 상태의 고장과 단순한 기계 부품에 대해 수행되어 왔으며 1960년대 후반에서 1970년대 초반에 군사용 표준으로 채택되었다[1,2]. 오늘 날 많은 수의 내장형 시스템이 군사와 항공, 차량 등 여러 분야에 사용되고 있으며 이에 대한 신뢰성 연구가 꾸준히 진행 중 이다. 내장형 시스템의 하드웨어에 관한 신뢰성은 꾸준히 연구가 진행되어 Mean time to failure (MTTF), Failures in time (FIT) 등으로 표현할 수 있지만 소프트웨어 분야의 신뢰성은 정량화하여 예측할 수 가 없는 상태이다. 또한 내장형 시스템의 하드웨어와 소프트웨어 모두 일시적인 결함(Transient Fault)이 빈번하게 발생하여 정량화가 더욱 어렵다. 이 기고문은 신뢰성에 대해 간단히 기술하고 차량 전장 시스템의 소프트웨어 신뢰성을 평가하기 위해 소프트웨어 안전성과 신뢰성에 대한 연구 동향을 기술한다.

2. 내장형 시스템의 신뢰성

신뢰성을 높이기 위해서는 신뢰성에 대한 위협과 신뢰성의 속성, 신뢰성을 달성할 수 있는 방법 등을 고려해야 한다[3]. 일반적으로 내장형 시스템은 기능과 성능, 비용, 신뢰성으로 구성된 4 가지의 기본적인 특성을 가지며 신뢰성은 믿을 수 있는 서비스를 제공하는 내장형 시스템의 능력을 나타낸다. 신뢰성에 대한 위협으로는 결함과 오류, 고장을 들 수 있다. 고장은 정확한 서비스 상

태에서 정확하지 않은 상태로의 천이로 볼 수 있으며 정확하지 않은 서비스 상태에서 정확한 서비스 상태로의 천이는 서비스 복구로 볼 수 있다. 시스템의 기능이 사양에 맞지 않거나 사양이 기능을 명확하게 기술하지 않은 경우 내장형 시스템에 고장이 발생할 수 있으며 시스템은 고장 모드에 따라 항상 동일한 방식으로 고장이 발생하지 않는다. 고장은 관점에 따라 특정 고장 영역과 사용자에 의한 고장, 환경에 의한 고장으로 나눌 수 있다. 오류는 추후 고장의 원인이 되는 시스템의 상태이며 오류가 시스템의 서비스에 영향을 미치면 고장이 발생한다. 일반적인 오류의 분류는 고장을 나타내는데 사용되며 오류 값과 타이밍 오류, 불일치 오류, 오류의 중요성 등으로 나타낼 수 있다. 오류는 오류 메시지나 오류 신호로 시스템에서 검출될 수 있으며 검출되지 않을 수도 있다. 결함은 오류를 발생하게 하는 것으로 추측되는 원인이다. 일반적으로 결함이 발생하면 오류가 발생하거나 나타나지 않을 수 있고 결함의 원인은 매우 다양하며 일반적으로 발생 단계와 시스템 경계, 영역, 현상 원인, 의도, 지속 등의 주요 항목으로 나눌 수 있다. 신뢰성을 위협하는 결함과 오류, 고장에 대한 관계는 연쇄적으로



(그림 1) 고장 분류

설명될 수 있다.

결함의 원인은 시스템에 따라 다양하게 발생하며 내장형 시스템의 경우는 하드웨어와 소프트웨어 부분으로 나누어 볼 수 있다. 하드웨어의 경우 제조 공정의 결함, 동작 환경의 진동과 열, 설계 오류 등이, 소프트웨어의 경우 설계 오류와 실행 중 오류 등이 원인이 된다. 이 외에 사용자나 천재지변 등이 원인이 될 수 있다.

3. 소프트웨어 안정성

기존의 안정성 관련 기법은 산업 분야에서 꾸준히 연구되어 왔으며 산업 분야에 내장형 시스템이 널리 사용되면서 내장형 시스템의 하드웨어와 소프트웨어에 관한 안정성이 문제가 발생하였다. 기존의 안전성과 관련된 중요한 고장은 다양한 방식으로 처리되어 왔다. Fail operational 방식은 고장이 발생해도 시스템이 계속 동작하는 방식으로 일반적으로 redundancy로 처리된다. Fail safe 방식은 안전한 상태를 식별하고 고장 발생 시 안전한 상태로 시스템의 상태를 변경하며 간혹 제한된 기능을 가질 수 있다. 이러한 방식은 시스템의 특정 부분에 고장이 발생하면 어떠한 문제가 발생하는지 예측해야하며 설계 단계에서 이중화 (Redundancy)를 고려해야 한다. 고장에 시스템에 미치는 영향을 분석하는 기본적인 기법으로 Failure mode and effects analysis (FMEA)와 Fault tree analysis (FTA)를 사용한다. FMEA는 내장형 시스템 설계에서 가장 일반적으로 사용하는 기법으로 특정 부분의 고장에 대한 결과를 분석하며 가장 치명적인 결과를 가지는 고장을 찾고 고장 확률을 낮추기 위한 방법을 제시하는 것을 목표로 한다. FTA는 위험 목록에 있는 위험의 모든 가능한 원인을 분석하는 기법으로 하나의

지점에서 발생하는 취약성을 제거하는 것을 목표로 한다. 그러나 실시간 특성과 시스템의 상태 또는 동작 상태를 고려하지 못하는 문제점이 있다. 또한 내장형 시스템이 여러 곳으로 분산되어 네트워크로 연결되면서 네트워크 고장 및 네트워크 패킷 오류 등의 네트워크 관련 문제가 발생할 수 있다. 패킷 손실의 원인으로는 높은 비트 오류율과 전기 모터 등으로 인한 전기적 잡음, 재전송 미지원, 충돌 기반 통신 프로토콜 등이 있으며 이로 인해 이벤트 기반 시스템의 경우 잘못된 상태로 동작하게 되고 시간 기반 시스템의 경우 제어가 불안정해 질 수 있다.

위험은 위험 발생 빈도와 심각성으로 정형화할 수 있으며 수학적으로 확률과 결과로 계산하여 위험의 우선순위를 정할 수 있다. 이는 위험 관리를 통해 위험을 식별하고 추적하며 위험 경감과 회피, 수용 등의 여러 기법을 적용할 수 있다. 민간 항공기 위험 분류는 다음과 같다. 일반적으로 적절한 이중화(Redundancy) 없이는 10^{-5}/hr보다 낮은 목표를 달성하기 어려우며 추가적인 기능을 위한 부하를 고려해야 한다. 일반적으로 항공기가 차량보다 안전하기 때문에 자율 주행 차량에 상용 항공기의 기술을 많이 사용하지만 여분의 하드웨어 비용과 소프트웨어의 완벽성 문제가 발생한다. 또한 다른 동작 환경과 반응 속도, 유지 보수의 어려움, 다양한 운전자 등의 특징을 가진다.

Safety Case는 소프트웨어를 위해 새롭게 도입된 개념으로 시스템이 안전하다는 것을 증명하는

〈표 1〉 민간 항공기 위험 분류

위험 분류	확률	비고
치명적	10^{-9}/hr	비행과 착륙
위험함	10^{-7}/hr	여러 승객에게 치명적인 부상
심각함	10^{-5}/hr	승객에게 치명적이지 않은 부상
심각하지 않음	10^{-3}/hr	적절한 해결방법
성가심	10^{-2}/hr	영향 없음

논리 정연한 상태이다[4]. 안전성을 증명하기 위한 다양한 기법들이 연구되고 있으며 그 중의 하나로 위험과 동작성 연구 (Hazard and operability study, HAZOP)가 있다[5]. HAZOP은 동작 중 발생할 수 있는 문제를 식별하고 평가하기 위한 구조적이고 체계적인 시험 기법으로 화학 플랜트와 핵발전소 등 복잡한 동작 환경의 안전성을 증명하는데 널리 사용되고 있다. 대상 시스템이나 소프트웨어에서 HAZOP을 통해 잠재적인 위험과 동작 중 발생할 수 있는 문제를 찾아내는 것을 목표로 한다.

이러한 방식이 다양한 경우에 대해 안전한지 결정하는데 어려움이 있지만 체계적인 소프트웨어 개발과정의 사용과 개발과정의 검토 및 관리, 안전 표준의 준수, 좋은 분석 기법의 사용 등으로 안전성을 확보할 수 있다[6]. 시스템을 더욱 안전하게 만들기 위해 위험을 명시적으로 표시하고 적합한 수준의 신뢰성과 안정성을 가지게 시스템을 설계하는 것이 중요하며 이렇게 설계된 시스템에 대해 검증과 인증을 수행하는 것도 잊지 말아야 한다.

4. 차량 전장 시스템을 위한 신뢰성

최근 차량에 내장형 시스템이 폭넓게 사용되면서 큰 쟁점으로 떠오르고 있다. 차량의 기계 부품과 달리 차량 전장 시스템의 소프트웨어는 기능을 완벽하게 수행하도록 사용되지만 실상은 그렇지 못하다. 실차 평가는 예상하지 못한 오작동을 찾아내는데 매우 중요하고 유용하지만 모든 항목을 평가하는데 한계가 있다. 그렇기 때문에 내장형 시스템과 소프트웨어 개발과정에서 사전에 결함을 예방하는 것이 중요하다. 차량 전장 시스템의 소프트웨어의 결함은 설계 과정의 문제이고 소프트웨어 고장 또한 직관적이지 않고 무작위로

발생한다. 뿐만 아니라 소프트웨어의 결함은 차량의 다른 부분의 파손으로 이어져 치명적인 고장을 발생시킬 수 있기 때문에 더욱 중요하다. 신뢰성 높은 소프트웨어를 개발하기 위해 새로운 기법을 개발하는 것보다 표준화 또는 문서화된 개발과정을 적용하여 검증을 거친 기법의 사용이 안전하며 새로운 기법의 사용을 위해서는 기존의 표준 절차에 맞게 구성 되었는지 평가가 필요하다. 이러한 안전과 관련된 차량 전장 시스템의 소프트웨어 신뢰성을 높이기 위해서 여러 개발과정 및 기법들이 연구되고 적용되고 있다.

4.1 MISRA (Motor Industry Software Reliability Association)

MISRA는 영국 차량 관련 회사들이 협력하여 안전과 관련된 차량 전장 시스템 개발을 위한 기법들을 제공하는 협회이다. 차량을 위한 소프트웨어 개발 가이드라인은 신뢰성 높은 차량용 소프트웨어 개발을 위한 권장되는 기법을 담고 있으며 IEC 61508에 자동차 산업 관련 근거로 사용되었다[7]. 그 외에 전체적인 가이드라인과 소프트웨어 개발과 통합 및 평가에 대한 상세한 보고서를 제공한다. 또한 MISRA C라는 차량을 위한 소프트웨어에서 C 언어 사용에 대한 가이드라인을 제공하였으며 현재 많은 소프트웨어 개발 분야에서 사용되고 있다[8]. 그러나 출판된 지 약 20년 정도로 비교적 오래 되었고 X-by-Wire로 불리는 전자 제어를 효과적으로 지원하지 못한다. MISRA에서는 소프트웨어의 신뢰성과 안정성을 나타내는 의미로 완전성 (Integrity)를 사용하며 발생할 수 있는 위험에 따라 5단계의 Safety Integrity Level (SIL)로 나타낸다. 결함에서 안전하게 회복할 수 있는 확률에 따라 등급이 나뉘며 항공과 철도 안전 표준에서 사용하는 방식과 유사하다.

〈표 2〉 MISRA SIL 등급

SIL	제어가능성	수용할 수 있는 고장률
4	불가능	거의 발생하지 않음
3	어려움	매우 희박함
2	약화됨	희박함
1	산만함	낮음
0	성가심	적당히 가능

MISRA는 운전자의 능력과 상황이 전체적인 위험에 대한 중요한 요소 중 하나라고 판단하여 운전자 반응 시간, 상황 인지, 집중력, 운전 경험 등의 다양한 항목을 반영한다. 또한 SIL 등급에 따라 소프트웨어 개발과정마다 명확한 요구사항이 있으며 일반적으로 SIL 등급 3과 4를 중요하게 고려한다. SIL 등급을 만족하기 위해서는 하위 SIL에 안전 관련 행위를 추가하게 된다.

4.2 IEC 61508

IEC 61508은 전기, 전자 및 프로그래밍이 가능한 전자 안전 관련 시스템의 소프트웨어 신뢰성을 높이기 위한 새로운 통합적인 표준으로 각기 다른 분야의 소프트웨어 신뢰성을 높이기 위해 맞춤 제작을 하는 전략을 제공한다 [IEC 61508]. MISRA에서 제시한 SIL을 연속적이거나 빠른 주기로 동작하는 시스템에 대한 등급과 가끔씩 동작하는 시스템으로 나누었으며 새로운 유용한 권장 기법들을 추가하였다.

IEC 61508-7 기법과 평가는 소프트웨어 신뢰

〈표 3〉 IEC 61508의 SIL 등급

SIL	연속/빠른 주기 (시간당 위험한 고장)	가끔 동작 (동작 중 고장 확률)
4	$10^{-9} \sim 10^{-8}$ / hr	$10^{-5} \sim 10^{-4}$
3	$10^{-8} \sim 10^{-7}$ / hr	$10^{-4} \sim 10^{-3}$
2	$10^{-7} \sim 10^{-6}$ / hr	$10^{-3} \sim 10^{-2}$
1	$10^{-6} \sim 10^{-5}$ / hr	$10^{-2} \sim 10^{-1}$

성 기법에 대해 목표, 상세 정보 등의 많은 자료를 포함하고 있지만 모든 상황에 적합하지는 않으며 분산 시스템의 신뢰성에 대한 내용이 부족하다. 또한 MISRA에서 다룬 운전자에 대한 항목이 제외되어 있다.

4.3 ISO 26262

IEC 61508는 여러 분야의 소프트웨어 신뢰성을 폭넓게 고려하여 차량 전장 시스템의 신뢰성에 특화된 ISO 26262가 새롭게 제정되었다[9]. IEC 61508과 다르게 사람과의 인터페이스와 연속적인 동작만을 고려하였으며 일반적인 동작보다는 특정 안전 기능에 대해 초점을 맞추었다. Automotive SIL (ASIL)은 차량 전장 시스템의 안전 등급을 나타내며 Quality management (QM)은 MISRA의 SIL 0과 같이 품질 관리로 볼 수 있다.

ASIL 등급은 고장의 심각성과 노출 빈도, 제어 가능성을 고려하여 산정되며 ASIL D가 가장 높다. 심각성은 차량 이용자의 부상 정도를 나타내며 심각성 0은 아무런 부상이 없는 것을 의미하고 심각성 3은 생명을 위협할 수 있는 부상과 치명상을 의미한다. 노출은 위험에 대한 노출 확률을 나타내며 노출 0은 거의 노출되지 않음을 노출 3은 자주 노출됨을 의미한다. 높은 ASIL 등급을 만족시키기 위해서는 낮은 ASIL 등급의 소프트웨어나 시스템으로 이중화 (Redundancy) 를 제공하여야 한다. 일반적으로 IEC 61508의 SIL 3 등급이나 ISO 26262 ASIL D 등급 이상에서는 멀티코어 등의 하드웨어 이중화가 필요하며 동일한 기능을 가진 다른 버전의 소프트웨어 등의 소프트웨어 이중화도 고려되어야 한다.

〈표 4〉 ISO 26262의 ASIL 등급

		제어가능성 1	제어가능성 2	제어가능성 3
심각성 1	노출 1	QM	QM	QM
	노출 2	QM	QM	QM
	노출 3	QM	QM	A
	노출 4	QM	A	A
심각성 2	노출 1	QM	QM	QM
	노출 2	QM	QM	A
	노출 3	QM	A	B
	노출 4	A	B	C
심각성 3	노출 1	QM	QM	A
	노출 2	QM	A	B
	노출 3	A	B	C
	노출 4	B	C	D

4.4 ISO/IEC 15504

ISO/IEC 15504는 성숙도 모델 (Maturity model)을 위한 참조 모델 (Reference model) 표준으로 소프트웨어, 시스템 등의 개발에 대한 능력을 결정하는데 사용된다[6]. 일반적으로 Software Process Improvement and Capability Evaluation (SPICE)로 불리며 초기에는 소프트웨어 개발과정에 대한 표준을 제시하였으나 프로젝트 관리, 설정 관리, 품질 보증 등의 분야로 확대 되었다. 2004년 개정에서 개발과정 참조 모델은 ISO/IEC 15504에서 제외되고 소프트웨어 생명주기 (Software Development Lifecycle)로 ISO/IEC 12207와 연계되었다[10].

〈표 5〉 ISO/IEC 15504 능력 등급과 프로세스 속성

능력 등급	상태	속성
5	최적화	프로세스 혁신, 프로세스 최적화
4	예측가능	프로세스 평가, 프로세스 제어
3	수립됨	프로세스 정의, 프로세스 배치
2	관리됨	성능 관리, 산출물 관리
1	수행됨	프로세스 성능
0	미완성	없음

4.5 Automotive SPICE

Automotive SPICE (A-SPICE)은 ISO/IEC

15504와 주요 차량 개발사의 의견을 바탕으로 한 차량 전장 시스템 개발을 위한 개발과정 평가 기법이다. Automotive SPICE는 개발과정 참조모델과 개발과정 평가모델에 사용되는 요구사항을 기술하는 성숙도 모델을 제공한다[11,12]. 이 모델은 개발 과정의 주요 부분과 6개의 능력 등급으로 구성되어 있다. 참조모델은 프로세스 카테고리와 프로세스 그룹, 프로세스로 구성되며 프로세스 카테고리 중 주요 생명 주기 과정들에서 엔지니어링 프로세스 그룹이 시스템과 직접 관련된 프로세스들을 나타낸다.

평가모델은 각 프로세스의 능력치를 평가지표에 따라 평가하는 것을 나타내며 프로세스 평가 모델 지표는 목표 달성 근거로 평가결과에 사용된다. 평가 지표는 능력 등급으로 표시되는 프로세스 능력 지표와 능력 등급 1에 적용되는 프로세스 성능 지표로 구성된다.

〈표 6〉 A-SPICE Primary Life Cycle Processes 참조모델

프로세스 카테고리	프로세스 그룹	프로세스	설명
Primary Life Cycle Process	Acquisition	ACQ.3	계약 동의
		ACQ.4	공급사 검토
		ACQ.11	기술적 요구사항
		ACQ.12	법률과 관리 요구사항
		ACQ.13	프로젝트 요구사항
		ACQ.14	수요 조사
		ACQ.15	공급사 자격 검증
	Supply	SPL.1	공급 입찰
		SPL.2	제품 양산
	Engineering	ENG.1	요구사항 도출
		ENG.2	시스템 요구사항 분석
		ENG.3	시스템 구조 설계
		ENG.4	소프트웨어 요구사항 분석
		ENG.5	소프트웨어 설계
		ENG.6	소프트웨어 구성
		ENG.7	소프트웨어 통합 평가
		ENG.8	소프트웨어 평가
		ENG.9	시스템 통합 평가
		ENG.10	시스템 평가

〈표 7〉 A-SPICE 등급

지표	의미	범위(%)	설명
N	미달성	0~15	속성이 약간 또는 달성 되지 않았음
P	부분 달성	16~50	몇몇 속성이 달성되고 근거가 있음
L	대부분 달성	50~85	속성이 대부분 달성되고 체계적 근거가 있음
F	완전히 달성	86~100	속성이 완전히 달성되고 완전하고 체계적인 근거가 있음

5. 결 론

앞에서 살펴본바와 자동차 산업에서 차량 전장 시스템의 신뢰성을 높이기 위해 여러 개발과정 및 기법들이 연구되고 적용되고 왔다. 복잡한 환경에서 소프트웨어의 정상적인 동작을 고려하여 신뢰성을 높이기 위해서는 소프트웨어 제품의 품질을 높이는 MISRA, ISO 26262와 같은 기법이 적합하며 소프트웨어가 올바른 과정에 따라 개발되었는지를 고려하여 신뢰성을 높이기 위해서는 소프트웨어의 개발과정에 대한 Automotive SPICE 기법이 적합하다. 이는 소프트웨어 제품에 대한 신뢰성을 높이기 위한 방법과 개발 과정에 대한 신뢰성을 높이기 위한 방법으로 구분할 수 있으며 어느 하나만 적용하여 차량 전장 시스템의 소프트웨어 신뢰성을 확보하기는 어렵다. 차량 전장 시스템의 보다 높은 신뢰성을 달성하기 위해서 무엇을 해야 하는 지와 어떻게 해야 하는지를 함께 고려해야 할 것이다.

참 고 문 헌

[1] MIL-STD-721C, Military Standard: Definitions of Terms for Reliability and Maintainability,

June 1981.
[2] DOD-5000.3, Joint Test and Evaluation Procedures Manual, August 1988.
[3] A. Avizienis, J. Laprie, B. Randell, "Fundamental concepts of dependability", 2001.
[4] Defence Standard 0056 Part 1 Issue 4, Safety Management Requirements for Defence Systems, June 2007.
[5] British Standard IEC61882, Hazard and operability study - Application Guide, October 2003.
[6] ISO/IEC 15504, Information technology - Process assessment, 2008.
[7] MISRA, Development Guideline for Vehicle Based Software, November 1994.
[8] MISRA, Guideline for the Use of the C Language in Critical System, March 2013.
[9] ISO 26262, Road Vehicle - Functional Safety, 2011.
[10] ISO/IEC 12207, System and software engineering - Software life cycle processes, 2008.
[11] Automotive SPICE, Process Assessment Model, v2.5, May 2010.
[12] Automotive SPICE, Process Reference Model, v4.5, May 2010.

저 자 약 력

조 정 훈

이메일 : jcho@ee.knu.ac.kr

- 1996년 한국과학기술원 전기및전자공학과(공학사)
- 1998년 한국과학기술원 전자전산학과(공학석사)
- 2003년 한국과학기술원 전자전산학과(공학박사)
- 2003년~2005년 (주)하이닉스 / 선임연구원
- 2005년~현재 경북대학교 전자공학부 부교수
- 관심분야 : 차량 소프트웨어 플랫폼 및 신뢰성, 모델기반 개발/평가

한 갑 수

이메일 : kabus@knu.ac.kr

- 2000년 단국대학교 전자컴퓨터공학부(학사)
- 2002년 홍익대학교 컴퓨터공학과(석사)
- 2010년~현재 경북대학교 전자공학부 박사과정
- 관심분야 : 차량 소프트웨어 플랫폼 및 신뢰성, 모델기반 개발/평가