특집 05

IT융합제품의 안전성 국제표준 인증을 위한 소프트웨어 테스팅 방법

이남희 ((주)솔루션링크)

목 차 »

1. 서 론

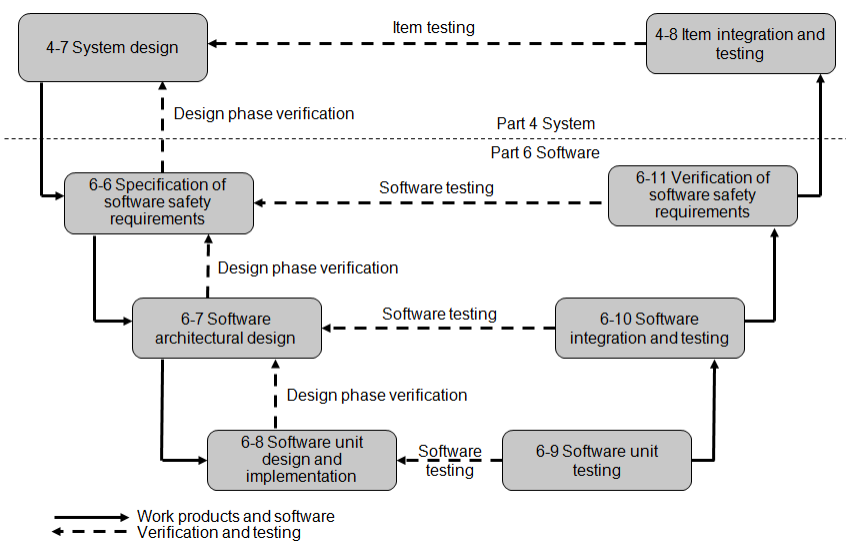
원자력발전소, 항공기, 차량 등의 제어시스템은 고장이 발생할 경우 사회·경제적으로 큰 문제를 야기할 수 있는 안전 필수 시스템 (Safety-Critical System)이다. 이러한 시스템들의 사소한 고장은 경제적 손실 뿐만 아니라 인명 피해와도 직접적인 연관이 있다. 1980년대 이후 컴퓨터 기술의 발전으로 이러한 제어시스템들이 아날로그 방식에서 디지털로 전환되고, 또한 많은 부분을 소프트웨어가 담당하게 됨에 따라 더욱 안전성을 보증하기 어려워 졌다. 특히, 최근의 토요타 차량 결함에 따른 리콜 건은 약 50조원 이상의 직·간접 손실이 발생한 것으로 알려져 있는데, 매트나 가속 페달 등 기계적인 결함이 원인이었다고 공식 발표되었지만 많은 전문가들은 소프트웨어 등 전자장치의 결함에 기인하고 있다고 의문을 제기하고 있다.

소프트웨어 등 전자장치에 의한 디지털 제어시스템은 아날로그 방식에 비해 더욱 복잡한 제어를 효과적으로 수행할 수 있는 장점을 갖고 있는 반면, 소프트웨어 자체의 복잡성 등으로 인해 그 정확성을 확보하기는 더 어렵다. 이에 1990년대 이후부터 이러한 안전성이 매우 중요한 IT융합 제품에 탑재되는 소프트웨어에 대해 개발 및 검증 과정 상에서 준수해야 할 절차 및 기법에 대한 표준을 제정하는 활동이 활발하게 진행되어 왔다. <표 1>은 안전성 국제표준의 현황을 보여주는 것으로, 가장 기본이 되는 IEC 61508 [1] 표준을 근간으로 각 IT융합 제품의 개발 및 사용상의 특성을 반영하여 전자장치의 안전성에 관한 표준이 제정되어 있거나 제정 중에 있다. 항공 부분에 대한 표준인 DO-178B [2]를 제외한 대부분의 표준에서 소프트웨어를 별도의 표준으로 분리하지는 않고 있다.

대부분의 표준이 (그림 1)에서 보는 것과 같은 일반적인 V-Model 기반의 소프트웨어 생명주기를 기반으로 표준 요건들을 정의하고 있다.

〈표 1〉 IT융합 제품별 기능 안전성 국제표준 현황

Standards	Industry	Organization	First	Latest	Version
IEC 61508	E/E Systems	International Electrotechnical Commission	1998/12	2010/04	Ed. 2
IEC 61513 IEC 60987 IEC 62138 IEC 62349	Nuclear		2001/03	2007/08	2.0
IEC 60601	Medical		1977/01	2010/04	2.0
IEC 61511	Process		2003/01	2004/11	1.0
IEC 60335	Household		1970/01	2010/05	5.0
IEC 61784	Communication		2003/05	2010/07	3.0
IEC 62601	Machinery		2005/01	2010/07	1.0
DO-178B	Aviation	Radio Technical Commission for Aeronautics	1992/12	1999/03	B
EN 5012x	Railways	European Committee for Electrotechnical Standardization	2006	2010	2
ISO 26262	Automotive	International Organization for Standardization	2009/12		DIS



(그림 1) ISO/DIS 26262 [3]에서의 소프트웨어 생명주기

V-Model 왼쪽의 개발 단계에서는 안전 관련 요구사항을 개발하고 구체화하고 구현하는 과정 상에서 준수해야 하는 절차 및 기법 들과 검증 (Verification) 및 안전성 분석 (Safety Analysis) 요건 들을 정의하고 있으며, V-Model 오른쪽의 테스팅 단계에서는 대응되는 개발 단계의 수행 결과를 기반으로 확인 (Validation)을 수행하는 절차, 기법 및 환경에 대한 표준 요건을 정의하고 있다.

소프트웨어 테스트에서 국제표준을 만족하기 위해서는 계획수립, 준비, 실행에 대한 테스트 절차가 수립되고 이를 기반으로 테스트가 수행되어

야 하며, 개발 초기에 설정된 안전 무결성 등급(SIL:Safety Integrity Level)에 따라 요구사항 기반 테스트, 인터페이스 테스트 등 안전성 요구사항의 검증을 위한 테스트가 이루어져야 한다. 또한, 각 테스트에 대한 테스트 케이스 설계 기법과 테스트 수행 결과에 대한 구조적 커버리지(Structural Coverage)가 만족되어야 하며, 테스트 단계별로 요구되는 테스트 환경과 독립성, 그리고 사용되는 테스트 도구에 대한 신뢰성 등 다양한 요건들을 만족해야 한다.

이 중에서 특히 구조적 커버리지에 대한 요건이 기존에 수행해오던 테스트 활동에 비해 많은 시간과 노력을 추가적으로 필요로 하는 부분으로 대두되고 있다. 따라서, 각 표준에서 제시하고 있는 구조적 커버리지에 대한 상세 비교 분석을 통해 어떻게 효과적으로 인증 획득을 달성할 수 있을 지 알아보도록 한다.

2. 안전성 국제표준의 구조적 커버리지 요건 비교 분석

기능 안전성에 대한 국제표준은 IT융합 제품별로 별도로 존재하기 때문에, 본 원고에서는 모든 표준의 기준이 되는 IEC 61508, 항공기 분야의 소프트웨어 표준인 DO-178B, 그리고 최근 이슈가 되고 있는 자동차 분야에 대한 표준인 ISO/DIS 26262를 대상으로 각 표준에서 요구하고 있는 구조적 커버리지에 대한 요건을 상세히 살펴보았다.

2.1 IEC 61508

IEC 61508은 안전 시스템에 대한 요구사항 명세, 설계, 개발, 설치, 운영, 유지보수의 표준으로, 유럽・미국 등 해외 선진국에서는 2000년대 초부터 주로 안전 필수 시스템인 원전, 항공, 의료, 철도, 장치산업 등에서 활용되는 디지털 제어 시스템의 안전을 위해 IEC 61508을 근간으로 기능 안전성 (Functional Safety) 검증을 요구하고 있다. 최근에는 점차 모든 IT융합 분야로 확산돼 가는 추세이며, 유럽 등에 관련 제품을 수출하는 한국 기업에게도 IEC 61508 관련 인증을 요구하는 경우가 늘고 있다.

<표 2>는 IEC 61508에서 요구하는 테스트 방법을 나타내고 있는데, 여기서 R (Recommended)과 HR (Highly Recommended)은 필수적으로 수행해야 하는 요건을 의미한다. 이때, HR은 R에 우선하며, HR을 고려하지 않을 경우에는 그에 대한 사유가 계획단계에서 부터 명시되고, 인증 평가자의 동의를 받아야 한다.

또한, 각 테스트 방법은 안전 무결성 등급에 따라 선택적으로 적용될 수 있는데, SIL은 안전 필수 시스템의 무결성을 나타내는 통계적 기준을 의미한다. SIL은 현재 4등급 (SIL1 ~ SIL4)으로 구분되어 있으며 SIL이 높을수록 시스템의 신뢰성이나 효율성이 더 높다. 기준별 장애 확률은 SIL1은 1/10~1/100 (1~100년 사이에 예상치 못한 장애 발생 가능), SIL2는 1/100~1/1,000 (100~1000년 사이에 예상치 못한 장애 발생 가능),

〈표 2〉 IEC 61508에서의 Dynamic Analysis and Testing 요건

Table B-2		Safety Integrity Level			
		SIL1	SIL2	SIL3	SIL4
1	Boundary value analysis	R	HR	HR	HR
2	Error guessing	R	R	R	R
3	Error seeding	——	R	R	R
4	Performance modeling	R	R	R	HR
5	Equivalence classes and input partition testing	R	R	R	HR
6	Structure-based testing	R	R	HR	HR

SIL3은 1/1,000~1/10,000 (1000~10000년 사이에 예상치 못한 장애 발생 가능), 그리고 SIL4는 1/10,000~1/100,000 (10000~100000년 사이에 예상치 못한 장애 발생 가능)을 의미한다.

<표 2>에서 보는 것과 같이 IEC 61508에서는 테스트 결과가 만족해야할 문장 (Statement) 커버리지나 결정 (Decision) 커버리지와 같은 구체적인 구조적 커버리지 기준을 정하지는 않고 있다. 즉, IEC 61508은 특정 IT융합 제품 영역을 대상으로 하지 않고 있기 때문에, 각 테스트 단계에 대한 수행 결과의 Pass/Fail 기준을 테스트 계획 수립 시 적절하게 설정하고 수행할 것을 요구하는 등 기법적인 준수 보다는 절차적인 준수 위주로 제시하고 있다.

2.2 DO-178B

RTCA/DO-178B는 유럽연합과 미국에서 산업 항공분야에 사용되는 소프트웨어에 대한 인증을 얻기 위해 사용되는 표준으로, 항공시스템에서 가져야 할 두 가지 주요한 목적 (소프트웨어의 항공운항 시스템과 장비들은 서로 호환되어야 하고, 내부 기능들이 안전하게 수행될 수 있도록 소프트웨어가 가져야 하는 지침을 제공해야 함)이 기술되어 있다.

DO-178B의 안전 무결성 등급은 A~D로 나누어지며 A가 제일 높은 등급이다. <표 3>은 DO-178B에서 요구하는 코드 커버리지 요건으로, ○와 ●는 모두 커버리지가 달성되어야 함을 의미한다. 단, ●는 테스트 자체가 독립적으로 수행되어야 한다.

DO-178B에서는 IEC 61508과는 달리 안전 무결성 등급에 따라 달성되어야 하는 구조적 커버리지 기준을 제시하고 있기는 하지만, 해당 커버리지가 어떤 특정 테스트 단계 (단위/통합/시스템)의 수행 결과로 만족해야 함을 요구하지는 않고 있다. 즉, 최고 안전 무결성 등급의 제품 개발 시 단위 테스트 단계에 MC/DC 커버리지를 만족하지 않아도 된다는 것을 의미한다.

〈표 3〉 DO-178B에서의 Verification of Verification Process Results 요건

Table A-7		Applicability by SW Level			
		A	B	C	D
1	MC/DC Coverage	●			
2	Decision Coverage	●	●		
3	Statement Coverage	●	●	○	
4	Data Coupling and Control Coupling Coverage	●	●	○	

2.3 ISO 26262

ISO 26262는 자동차에 탑재되는 SW의 오류로 인한 사고를 미연에 방지하기 위해 제정한 기능안전 표준으로, IEC 61508이 일반 전기전자 장치의 안전에 관한 포괄적 표준이라는 한계를 보완하기 위해 만들어졌다. ISO 26262는 현재 DIS (Distributed International Standards) 단계에 있으며, 2011년 6월에 정식 릴리즈될 예정이다. ISO 26262에서는 재난상황 노출 가능성 (probability of exposure), 잠재적 심각도 (potential severity), 그리고 통제 가능성 (controllability)을 고려하여 차량 안전 무결성 등급인 ASIL (Automotive SIL)을 결정한다. 이것은 자동차 제품의 특성을 반영한 것으로 ASIL은 최저 등급인 A부터 최고 등급인 D까지 4개 등급으로 구성되어 있다. BMW, GM, 보쉬 등의 글로벌 자동차 메이커 및 부품 공급업체들은 이미 ISO 26262를 자체 개발 프로세스 내에 적용하고 있으며, 국내 자동차 업체와 부품 공급업체 역시 도입을 서두르고 있는 상황이다.

<표 4>는 ISO 26262의 소프트웨어 단위 테스

〈표 4〉 ISO 26262에서의 Metrics for structural coverage at software unit testing 요건

Table 14		ASIL			
		A	B	C	D
1a	Statement Coverage	++	++	+	+
1b	Branch Coverage	+	++	++	++
1c	MC/DC (Modified Condition/ Decision) Coverage	+	+	+	++

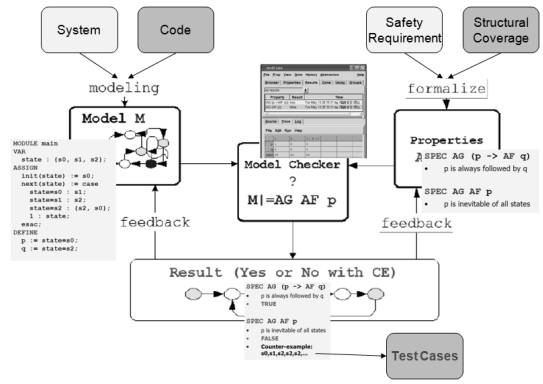
트에 대한 결과 검증을 위해 사용되는 코드 커버리지 기준에 대한 요건을 나타낸다. +와 ++는 각각 IEC 61508의 R과 HR에 해당한다.

ISO 26262는 DO-178B보다 더 엄격해서 특정 테스트 단계에서 달성해야 할 구조적 커버리지 요건을 제시하고 있다. 즉, 코드 기반의 단위 테스트는 물론, 모델 기반의 개발일 경우에도 모델에 유사한 구조적 커버리지 기준을 적용해야 하며, 통합 테스트 단계에서도 함수 (Function) 커버리지와 같은 구조적 커버리지를 적용한 결과를 요구하고 있다.

3. 소프트웨어 안전성 테스트 연구 동향

지난 수십년간 안전 필수 소프트웨어에 대한 모델링 방법과 안전성 요구사항에 대한 자동화된 검증 기법에 대한 많은 연구가 수행되어 왔고, 최근에는 매우 복잡한 시스템의 경우도 지원할 수 있을 정도로 성숙하면서 상용화된 자동화 도구도 많이 보급되고 있다. 이에 모델 기반의 테스트 케이스 생성 [4]과 모델 체커를 기반으로 테스트 케이스를 자동으로 생성하고자 하는 연구 [5],[6]가 다양하게 진행되고 있다.

[4]에서는 SysML/UML의 Activity Diagram에 구조적 커버리지를 적용하여 테스트 케이스를 자동 생성하였다. 모델 기반 개발은 복잡한 안전 필수 소프트웨어를 효율적으로 개발할 수 있도록



(그림 2) 모델 체커를 이용한 테스트 케이스 자동 생성

지원하는 방법으로 점점 더 그 영역이 넓어지고 있기 때문에, 모델을 기반으로 한 테스트 케이스 자동 생성 및 테스트 실행 도구를 사용하는 것은 안전성 국제표준에서 요구하는 구조적 커버리지 달성을 용이하게 한다.

[5]에서는 FSM (Finite State Machine)으로 모델링한 시스템과 LTL (Linear Temporal Logic)로 기술한 안전성 요구사항을 SPIN 모델 체커에 입력하여, SPIN에서 제공하는 counterexample 생성 기능을 이용하여 테스트 케이스를 자동으로 생성하였다. (그림 2)의 FSM 모델에 구현된 코드를 변환하여 입력하고, 구조적 커버리지 요건을 LTL 형식으로 기술하면 구조적 커버리지를 달성할 수 있는 자동화 도구로서 모델 체커를 활용할 수 있다 [6].

4. 소프트웨어 안전성 테스트 방법

본 원고에서는 구조적 커버리지에 대한 가장 엄격한 인증 요건을 제시하고 있는 ISO 26262를 기준으로 효과적으로 인증을 획득하기 위해 테스트 시 고려해야 할 사항들을 제시하도록 한다. (그림 1)에서와 같이 ISO 26262에서는 소프트웨

어의 테스팅 단계를 Software unit testing, Software integration and testing, Verification of software safety requirements 로 구분하고 있고, 각 단계에 수행할 테스트 목적에 따라 테스트 케이스를 설계하는 기법과 수행 결과에 대한 커버리지 분석 보고서를 제출할 것을 요구하고 있다.

4.1 테스트 주도 개발

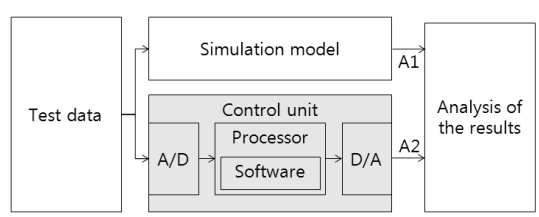
<표 4>에서 보는 것과 같이 ISO 26262의 최고 안전 무결성 등급인 ASIL D의 경우 MC/DC 100%의 구조적 (코드) 커버리지를 단위 테스트 단계에서 만족해야 한다. 하지만, 단위 테스트 프로세스는 극도로 노동 집약적이라서 개발 비용의 50% 이상을 소비하며, 그 중 75% 이상은 테스트 케이스를 만드는데 사용된다. 더구나 작성된 테스트 케이스는 코드가 변경될 때마다 지속적으로 변경되어져야 구조적 커버리지를 달성할 수 있는 유효한 단위 테스트 케이스로서 가치를 할 수 있다. 따라서, 단위 테스트 케이스는 코딩 후에 이를 기반으로 별도로 작성하는 것이 아니라, 테스트 케이스를 먼저 작성하고 코딩을 하는 방식의 개발이 도움이 될 수 있다. 즉, 가능한 빨리 테스트 케이스를 쓰기 시작해야 한다. 또한, 코드를 적기 전에 테스트 케이스를 먼저 적는 것은 오히려 요구사항을 향상 시키는데 도움을 주므로 더 나은 요구사항을 가질 수 있게 도와 준다.

4.2 요구사항 추적성 확보

테스트들이 잘 수행되면 복잡도에 따라 소프트웨어의 70~80%의 구조적 커버리지는 쉽게 달성되어야 한다. 좋은 요구사항이 쓰여지지 않았거나, 구체적인 기능 설계가 부족하거나, 부적절한 설계가 되었다면 좋은 테스트, 좋은 구조적 커버리지를 달성하지 못할 것이다. 따라서, 커버리지 분석 결과 100%를 달성하지 못했다면, 먼저 요구사항으로 돌아가야 한다. 100% 또는 그에 근접하지 못한 이유는 테스트 케이스를 쓰는데 좋은 요구사항을 가지지 않았기 때문이다. 즉, 기능으로 나타나는 더 많은 요구사항을 적고 모든 기능에 대한 상세 설계가 되고 추적되고 검증되었는 지를 확인해야 한다.

4.3 모델 기반 테스트

3장에서 소개한 모델 기반의 개발은 테스트의 많은 부담을 덜어 줄 수 있다. Matlab이나 SCADE, Rhapsody 등과 같은 상용화된 모델링 및 코드 자동 생성 도구는 모델 자체의 검증은 물론 (그림 3)과 같은 Back-to-back 테스트 환경을 쉽게 구성할 수 있도록 지원한다. 이때, 모델을 기반으로 생성된 테스트 케이스 또는 수작업으로 작성한 테스트 케이스를 back-to-back 테스트 환경의 테스트 데이터로 활용할 수 있다.



(그림 3) Back-to-back 테스트 환경

4.4 신뢰성이 입증된 자동화 도구 사용

마지막으로 각 단계별로 목적에 맞는 테스트 도구를 사용해야 하며, 각 도구는 그에 합당한 신뢰성이 입증되어야 한다. 컴파일러와 같은 도구는 도구 자체의 오류가 제품에 미치는 영향이 크기 때문에 도구 신뢰성 수준인 TCL (Tool

〈표 5〉 소프트웨어 테스트 도구 및 TCL 예

Process	Usage	TCL	Tool Name
Software unit testing	Requirement-based test	2	TestDirector
	Model-based test	2	Simulink, SCADE, Reactis
	Fault injection test	2	Tessy
	Interface test (internal)	2	Simulink
	Statement, Branch, MC/DC	2	VectorCAST, LDRA
Software integration and testing	Requirement-based test	2	TestDirector
	Fault injection test	2	Tessy
	Interface test (external)	2	Simulink
	Back-to-back test	2	LIN emulators
	Measure test coverage	2	NCover, Bullseye

Confidence Level)이 3 이상이지만, 테스트 도구의 경우에는 TCL이 2 정도이다. TCL 2에 해당하는 도구의 경우에는 기존 제품에 적용되어 안전성에 관련된 문제가 없었음을 보이는 도구 사용 예와 도구 제공 회사의 개발 프로세스 성숙도를 이용하여 신뢰성을 입증하면 된다.

5. 결 론

각 IT융합 제품을 대상으로 하는 안전성 국제표준에 대한 인증 획득 요구는 선진국들이 개발도상국의 시장진입 장벽을 높이는 역할을 하게 될 것이다. 하지만, 짧은 개발주기를 맞추기 위해 개발이 거의 완료되는 시점에 집중적으로 테스트를 수행해 온 국내 IT융합 제품의 개발현실에서는 국제표준에서 요구하는 테스트 요건을 맞추기 어렵다. 이는 기존과 같은 프로젝트 수행 방식대비 최고 안전 무결성 등급의 경우 약 40~100% 이상의 노력을 추가로 요구하기 때문이다. 따라서, 국제표준을 만족하면서도 효과적으로 테스트를 수행하기 위해서는 제품 계획 초기 단계부터 더욱 더 철저한 테스트 전략 및 계획 수립이 필요하며, 모델 기반의 개발을 적극 적용할 필요가 있다.

또한, 이제는 제품이 개발되기 이전 또는 시작 시점부터 그 제품이 가질 수 있는 위험 또는 손상을 예측하고 분석하여 문제가 발생하지 않도록 개발하는 방향으로 개발 문화가 바뀌어야 한다. 즉, 단순히 테스터 또는 개발자 만의 개별적인 능력이나 판단이 아니라, 조직 수준에서 전체 제품 생명주기를 통해 프로세스가 준수되고 안전성 요구사항이 확인되어야 함을 의미한다.

참 고 문 헌

[1] IEC 61508, Functional Safety of Electrical/Electronic/Programmable Electronic Safety-Related System, Dec., 1998.
[2] RTCA/DO-178B, Software Considerations in Airborne Systems and Equipment Certification, Dec., 1992.
[3] ISO/DIS 26262, Road Vehicles Functional Safety, Dec., 2009.
[4] M. Hause, A. Stuart, D. Richards and J. Holt, Testing Safety Critical Systems with SysML/UML, IEEE International Conference on Engineering of Complex Computer Systems (ICECCS) 2010, Mar., 2010.

[5] G. Yu, Z. wei Xu and J. wei Du, An Approach for Automated Safety Testing of Safety-Critical Software System Based on Safety Requirements, International Forum on Information Technology and Applications, May, 2009.
[6] H.S. Hong, S. Cha, I. Lee, O. Sokolsky and H. Ural, Data Flow Testing as Model Checking, International Conference on Software Engineering, May, 2003.

저 자 약 력

이 남 희

이메일 : nhlee@sol-link.com

- 1991년 한국과학기술원 전산학과(학사)
- 1998년 한국과학기술원 전산학과(석사)
- 2003년 한국과학기술원 전산학과(박사)
- 1994년~1995년 (주)LG전자 미디어통신연구소/연구원
- 2003년~2005년 (주)삼성SDS 첨단소프트웨어공학센터 SW Quality팀 책임연구원
- 2006년~2007년 (주)삼성전자 CS경영센터 개발품질보증팀 과장
- 2007년~현재 (주)솔루션링크 컨설팅사업부 사업부장/CTO
- 관심분야 : 소프트웨어 공학, 임베디드 소프트웨어 테스팅, 기능 안전성, 정형 기법, 소프트웨어 프로세스 개선 (CMMI, SPICE, TMMi, TPI 등)