



데이터 기반 자동화된 진단 구현 검증방식

차량 진단

진단 프로토콜이 잘 구현되었는가를 검증하기 위해 자동으로 생성된 테스트를 사용하는 일은 수년 동안 실행되고 있다. 이러한 프로세스는 자동차 OEM 에서 제공하는 진단 디스크립션 파일(diagnostic description file)을 바탕으로 진행된다. 검증 가능한 방법을 사용함으로써 ECU 의 진단 인터페이스를 효율적으로 테스트할 수 있으며, 이를 통해 제품의 품질을 개선할 수 있다. 또한, 클라스(Claas)와 벡터의 협동 프로젝트에서 보여주고 있듯이 진단용 파라미터와 에러 코드의 검증을 자동화시키는 것 또한 가능하다. 물론 이것의 성공 여부는 진단 디스크립션이 얼마나 완전한지에 달려 있다.

과제

농업용 기계 제조업체인 클라스는 진단 데이터에 진단 파라미터와 ECU I/O 사이의 상호연관성(interrelationships)을 추가로 표현하고 있다. 새로운 방식으로 구현하기 위해 에러 설정 기준을 형식에 맞춰 기술하고 있다. 과거에는 이러한 정보는 수작업으로 테스트를 실시하거나, 테스트 담당 엔지니어가 특별한 테스트 케이스를 구현하는 데 사용되었다. 그러나 아직 이러한 방법으로 광범위한 테스트 범위를 모두 처리하는 것은 불가능했다.

클래스는 진단 파라미터와 ECU I/O 사이의 관련 정보가 기존의 네트워크(통신 매트릭스) 및 하드웨어 디스크립션(description)에 자동으로 연결되도록 벡터와 프로젝트를 수행하였다. CDD(CANdela Diagnostic Data) 또는 ODX(Open Diagnostic Data Exchange) 등과 같은 기존 사양의 데이터를 바탕으로 만든 테스트 환경 내에서 완전하게 자동화된 진단 구현 검증 테스트를 생성하여 수행하고 있다. ECU 를 자동으로 자극(stimulation)하는 환경을 이용하여 진단 기능과 연관된 하드웨어와 소프트웨어가 제대로 통합되었는지를 테스트한다. 이러한 작업은 버스 시뮬레이션을 통하여 신호값을 변경하거나 특정 하드웨어의 I/O 를 구동함으로써 수행할 수 있다. 따라서 진단용 파라미터가 제대로 설정되어 있는지, 에러 상태를 만든 후 그 에러가 메모리에 제대로 저장되는지를 테스트하는 것도 가능하다.

자동화된 테스트 생성

구현 검증 시험을 위한 테스트 생성 및 실행을 자동화하기 위해서는 진단 파라미터와 ECU I/O 를 서로 연관시켜야 한다. 진단 데이터 (ODX, CDD) 이외에도 간접적으로 참조할 수 있는 사양 데이터 (specification data) 또한 사용된다. 여기에는 네트워크 디스크립션(dbc, arxml)이나 HIL 시스템의 인터페이스 디스크립션 등과 같은 환경 설정이 포함된다. 이러한 시스템 정보는 테스트 환경에서 ECU 의 입력 및 출력을 자극하거나 측정할 때 사용할 수 있다.

오늘날 진단과 ECU 환경 사이의 상호 연관성을 표현하는 진단 데이터는 구체적으로 정의되어 있지 않은 상황이다. 이런 내용이 기술되는 경우에도 보통은 자연 언어로 기술된다. 이 때문에 자동화된 테스트 프로세스를 만드는 것이 불가능하다. 여기서 경험적 지식(heuristics)이 도움이 될 수 있으며, 적어도 이러한 지식을 테스트 자동화에 사용할 수 있다. 만약 OEM 이 ECU I/O 종류 및 범위를 구체적으로 기술하고 있다면, 이것은 진단 구현 검증 시험 방안을 만들어내는 데 사용할 수 있다. 특히 자동화된 진단 파라미터 테스트 및 폴트 메모리 테스트(fault memory test)를 가능하게 한다.

폴트 메모리 테스트

폴트 메모리 데이터의 구조 및 정형화된 내용은 DID(Data Identifiers) 레이아웃(layout)이나 DTC(Diagnostic Trouble Codes) 설정 조건과 같은 진단 데이터에서 가져온다. 그러나 DTC 를 정의하는 방식은 일반적으로 정형화되어 있지 않다. 만약 진단 데이터를 ECU 의 주변 기기들과 상호 연관시킬 수 있도록 정리할 수 있다면, DTC 가 폴트 메모리 내에 정확하게 저장되어 있는지 또는 적절한 조건을 따르고 있는지를 테스트하는 것도 가능하다. 또한, DTC 상태 전이를 확인하거나 DTC 를 적절히 소거하는 것도 가능하다.

각각의 DTC 에 대해서는 반드시 구체적인 설정 조건을 알고 있어야 한다. 이러한 조건에는 적어도 다음과 같은 것이 있다:

- > I/O 종류 (입력 또는 출력, 네트워크 또는 센서/액추에이터)
- > I/O 이름 (메시지 이름, 채널 이름)
- > 에러 패턴 (예를 들어, GND 단락회로).

에러 패턴은 표준화된 DTC(SAE J2012)의 FTB(Failure Type Byte)를 직접 사용할 수 있다. 에러 패턴에 따라 임계값, 설정 시간, 그리고 에러 정보와 같은 추가 정보가 필요할 수도 있다.

진단 파라미터 테스트

진단 파라미터 테스트(diagnostic parameter test)는 폴트 메모리 테스트와 유사한데, 이 테스트에서는 진단 파라미터와 ECU 핀 사이의 관계에 대해 알 필요가 있다. 진단값을 다음의 값과 비교하여 검증할 수 있다:

- > ECU 핀의 측정값
- > 버스 신호
- > CCP/XCP 신호

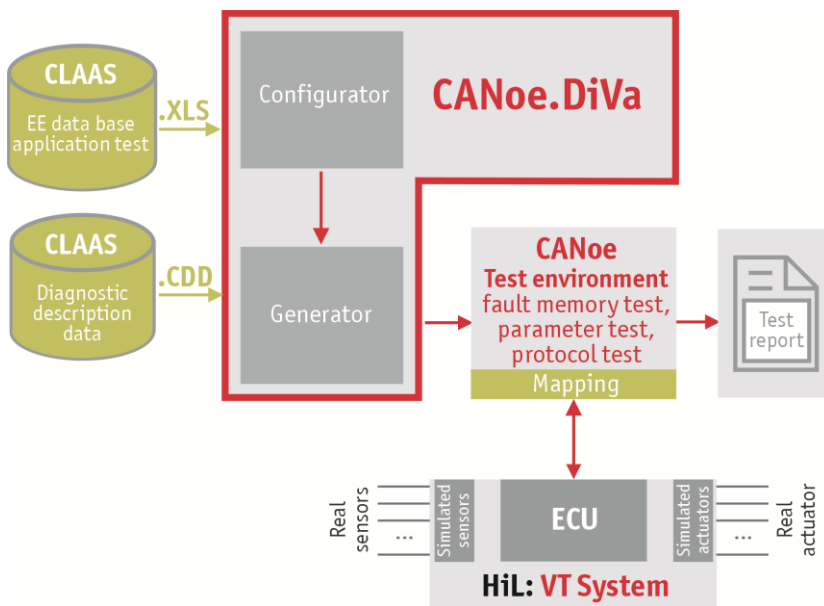
비교를 하기 위해서는 I/O 종류 및 명칭 이외에도 단위 변환 작업이 필요하다. 예를 들면, 센서의 저항값을 진단 파라미터인 온도값으로 변환하는 작업을 말한다. 진단 파라미터 또는 ECU 출력값의

업데이트 주기 또한 반드시 고려해야 한다. 더욱이 I/O 를 자극하기 위해서는 시험값이 필요한데, 그 이유는 이런 데이터는 진단 디스크립션 파일에 정의되어 있지 않고, 일반적으로 사양서도 적합한 값을 제공하고 있지 않기 때문이다.

플트 메모리 테스트나 파라미터 테스트 모두 테스트할 기능을 동작시키기 위한 전제 조건이 있는지를 확인할 필요가 있다. 예를 들어, 농업용 세단기(field chopper)의 칼날을 갈기 위해서는 메인 드라이브를 작동시켜야 하는 것과 같다. 테스트를 실행할 때에는 이러한 상호 연관 관계를 이해하고 고려해야 한다.

클래스의 목표 및 이행

클래스에서는 진단 구현 검증 테스트에 필요한 데이터의 상당 부분이 정형화된 상태로 기술되어 있다. 그러므로 여기서 목표는 이러한 정보를 바탕으로 테스트를 생성하고 실행하는 과정을 자동화시키는 것이다. 이러한 목표를 달성하기 위해서 클래스는 벡터의 CANoe.DiVa 툴을 사용한다. CANoe.DiVa 는 진단 데이터 파일(CDD)과 다른 소스의 I/O 정보를 임포트(import)하여 테스트 모듈을 자동으로 생성하기 위한 파라미터들을 설정한다. 그 후에는 CANoe 에서 DUT(Device Under Test) 테스트 환경과 함께 DiVa 테스트 모듈을 자동으로 실행시킨다. [그림 1 참조]



[그림 1: 시스템 구조: 클래스의 자동화된 진단 기능 구현 테스트]

클래스는 진단 구현 검증 테스트에 연관된 모든 데이터를 데이터베이스에서 관리하고 있다. 데이터는 또한 진단 디스크립션 파일로 임포트되므로, 보통 이 진단 디스크립션 파일은 파라미터 테스트를 생성하기에 충분하다. 추가적인 단위 변환 정보는 ECU 핀보다는 다른 단위를 사용하고 있는 I/O 용 진단 파라미터에 대해서 필요하다. 이 데이터는 변환 규칙에 따라 신호값과 CANoe 시스템 변수를 매핑 시키는 CANoe 매핑 작업에 사용된다.

파라미터 테스트는 3 가지 종류로 만들어진다:

- > 입력 테스트 : 이 테스트 환경에서는 ECU 의 센서 입력단을 자극한다. 연관된 진단 파라미터를 읽은 다음 이를 해당 핀에서 측정된 값과 비교한다.
- > 출력 테스트 : ECU 에 새로운 값을 입력하는 진단 서비스 (I/O Control)로 액추에이터를 구동한다. 나중에 이 출력 값을 측정된 다음 진단 파라미터값과 비교한다.
- > 수동 테스트 (passive test) : 일부 신호는 진단 서비스에 의해 제어할 수 없으며, 단지 읽기만 할 수 있다. 이 신호들은 진단 레이어(diagnostic layer)와 상관없이 ECU 응용 프로그램이 제어하고 있기 때문이다. 이런 경우, 진단 서비스로 값을 읽고 이를 ECU 의 입력 또는 출력값과 비교하는 테스트를 생성한다.

파라미터 데이터와는 달리 클래스는 폴트 메모리 테스트를 수행하는 데 필요한 데이터 중 일부를 진단 디스크립션 파일에 정의하고 있지 않다. 그러므로 개발 데이터베이스에서 이런 데이터를 MS 엑셀 파일로 내보내고 DiVa 가 그 파일을 읽어 들여 테스트 파라미터를 설정한다.

이 데이터를 이용하는 테스트에서는 에러 상황을 재현하기 위해 ECU 의 I/O 값을 자극한다. 그다음 폴트 메모리에 정확한 DTC 가 저장되었는지를 확인한다. 이 에러로 발생한 DTC 상태 천이(DTC status transition)와 DTC 환경 데이터 또한 에러 상황을 정정하고 일정 시간 기다린 후 에러 상태를 다시 설정하도록 테스트함으로써 검증할 수 있다. 각각의 안전 수준에 따른 폴트 메모리가 초기화되었는지를 점검함으로써 폴트 메모리 테스트를 마무리한다.

ECU 핀의 전압 및 전류를 측정하고 자극하기 위해 클래스는 [그림 2]와 같이 HIL 시스템으로 벡터의 VT 시스템을 사용하고 있다. 진단 디스크립션 파일 및 개발용 데이터베이스의 데이터를 이용하여

자동으로 VT 시스템을 구동하기 위하여 VT 시스템 구성용으로 명명 규칙(name convention)이 정의되어 있다.



[그림 2: 벡터의 VT System 을 이용한 클래스의 ECU 전압 자극 및 측정 시스템]

테스트 범위의 확대

클래스는 진단 기능이 있는 ECU 를 콤바인(combine harvester) 및 트랙터에서 사용하고 있을 뿐만 아니라 제초기 및 베일 프레스(baling press)와 같은 기계에서도 사용하고 있다. 클래스의 최대형 기계 장치는 장비 옵션에 따라 ECU 를 최대 40 개까지 설치한다. 이런 모델 계열 전체에 대해서 반드시 구현 검증 테스트를 수행해야 한다. 이러한 테스트를 통해 클래스 진단 시스템 (CDS)은 ECU 모듈들을 검증한다. 클래스에서 설치한 최대형 ECU 는 75 개 이상의 I/O 를 지원하며, 설치된 기계의 옵션에 따라 최대 15 가지의 기능이 구현된다. 이러한 I/O 와 연관된 DTC 는 200 개 이상이며 반드시 테스트를 통해 검증되어야 한다.

보다 많은 구현 검증 테스트의 생성 및 실행을 자동화시키게 되면 ECU 검증에 드는 노력을 크게 줄일 수 있다. 클래스는 DiVa 의 기본 테스트 생성 기능에 클래스만의 확장 기능을 추가하여 자동화함으로써 플트 메모리 및 진단 파라미터에 대한 테스트 범위를 과거 55%에서 현재 95%로 크게 넓힐 수 있었다. 클래스는 자사가 생산한 모든 ECU 에 대해 CANoe.DiVa 툴을 이용한 자동화된 구현 검증 테스트를 적용한다는 목표를 설정했다.

하드웨어 I/O 를 이용한 진단 테스트는 자동화된 테스트 생성과 테스트 실행에도 불구하고 상당한 노력이 필요하다. 특히 테스트 환경을 설정하는데 많은 시간이 소요된다. 어떤 ECU 의 경우 여러 I/O 라인의 동작 연관성이 복잡하여 I/O 라인을 개별적으로 제어하며 동작시키는 것이 매우 어려운 경우도 있다. 이에 반해 네트워크 신호를 참조하여 진단 파라미터를 검증하는 것은 초기의 환경 설정이 쉬운 편이다. 가령, 필요한 테스트 환경을 수동으로 만드는 대신 네트워크 데이터베이스를 이용하여 잔여 버스 시뮬레이션(remaining bus simulation)을 자동으로 생성할 수 있고 이 환경을 테스트에서 그대로 사용할 수 있기 때문이다.

요약 및 전망

벡터에서 제공하는 CANoe.DiVa 툴을 이용하여 클래스에서 수행하고 있는 것과 같은 테스트 자동 생성 및 실행은 테스트에 대한 노력은 줄이면서도 테스트의 깊이를 늘릴 수 있는 상당한 잠재력을 제공한다. 현재, CANoe.DiVa 툴을 이용해 모든 테스트를 처리하기 위해서는 클래스에서 일부 기능을 수동으로 구성해야 하는데, 그 이유는 파라미터를 자동으로 설정하기 위해 필요한 추가적인 정보를 정의하는 방식이 아직 공식적으로 갖춰지지 않았기 때문이다. 예를 들면, 파라미터 테스트에서 제어하기 매우 복잡한 I/O 들이 있는데 이런 I/O 를 원하는 대로 설정하여 사용하기 위해서는 ECU 환경에 대한 일부 전제 조건(precondition)을 충족시켜야 하는 경우이다.

마찬가지로, 플트 메모리 테스트도 에러 모니터링 기능이 활성화되어 DTC 를 인식하고 플트 메모리에 저장하기 전에 충족되어야 할 전제 조건을 필요로 하는 경우가 있다. 현재 수행하고 있는 테스트에 자가 복구(self-healing) 기능 테스트, 플트 메모리 저장 장소의 우선순위 설정, 그리고

동일한 DTC 를 만드는 서로 다른 에러 상황 테스트와 같은 항목을 추가하여 보완할 수 있다. 테스트 솔루션은 향후 이 분야에서 지속적으로 발전하게 될 것이다.

자동차 산업에는 또한 전기/전자 아키텍처에 사용되는 개발 데이터를 병합하는 방향으로 나가려는 주목할 만한 경향이 있다. 여기서 데이터베이스와 툴은 직접 또는 간접적으로 정보를 수집하고, 수집한 정보는 진단용 소프트웨어를 통해 I/O 라인의 전기적인 연결 및 기능의 전자적인 통합 상태를 자동으로 테스트하는 데 사용될 수 있다.

초기에는 데이터의 형식화(formalization) 작업에 상당한 노력이 필요하다. 무엇보다도 데이터의 형식화는 자동화된 테스트의 최적화 가능성을 높이게 되어 개발 노력이 헛되지 않게 할 것이다. 표준화 분야의 진보(예를 들어 AUTOSAR)와 개발 툴의 통합 및 상호 운용성(interoperability)은 테스트 자동화 분야에서 새롭고 다양한 접근을 이끌어내고 있다. CANoe.DiVa 툴은 이러한 추세를 따르고 있으며, 이와 같은 새로운 기회를 진단 구현 검증을 위해 자동화된 테스트를 더 많이 생성하는 데 사용할 것이다.

데이터 기반의 자동화된 테스트가 갖는 성장 가능성은 매우 높으며, 이 분야는 앞으로도 활발하게 발전할 것이다.

그림 제공: Vector Informatik GmbH, Claas

링크:

벡터 홈페이지: www.vector.com

저자:



닐스 니더마크(Nils Niedermark)

독일 하르제빈켈에 소재한 Claas 의 전자 통합 개발 부서에서 개발 담당 엔지니어로 일하고 있다.

**프리드만 로브(Friedemann Löw)**

독일 슈투트가르트에 소재한 벡터 인포마틱의 자동차 진단용 제품 라인에서 개발 업무를 담당하고 있다.

**시몬 뮐러 (Simon Müller)**

독일 슈투트가르트에 소재한 벡터 인포마틱의 자동차 진단용 제품 라인에서 제품 매니저로 일하고 있다.

본 보도자료 배포 시 최종 인쇄물을 당사에 보내주시면 감사하겠습니다.
배포와 관련하여 문의사항이 있으시면 언제든지 연락해주시기 바랍니다.

벡터 코리아 편집자 연락처:

마케팅팀 김용성

서울특별시 용산구 한남대로 11 길 12 고와스 빌딩 5 층

Tel. 02-807-0600 Ext.5009, Fax. 02-807-0601

E-mail: Yongseong.Kim@vector.com