# AUTOSAR 모델 기반의 멀티 코어 시스템에 대한 타이밍 시뮬레이션

남동현 (MDS테크놀로지)

목차



## 1. 서 론

최근의 자동차 산업은 편의 기능과 정교한 기능, 안전성, 그리고 스마트 전자 장치 내 “사물 인터넷”과 같은 수요가 증가하고 있다. 이러한 수요를 감당하기 위해서 1억 줄 이상의 코드가 다수의 ECU에 매핑 되고 있다. 고급 차량의 경우 ECU가 100여 개에 달하기도 한다. 이렇듯 수많은 ECU로 인해 그 어느 때보다 복잡성이 증가하여 보다 막중한 엔지니어링 작업이 요구되고, 업데이트와 변경을 위한 유연성의 필요성이 요구되고 있다. 이에 대응한 솔루션으로 AUTOSAR는 자동차용 E/E 아키텍처 개발에 있어 사실상의 산업 표준으로 자리 잡고 있다.

최근 자동차 산업에서 더 높은 처리능력, ECU 통합을 통한 원가 절감을 위하여 멀티 코어 시스템이 지속적으로 도입되고 있다. 이에 따라 자동차 산업은 복잡한 개발 환경에 대응하고 새로운 기능 수행을 위하여 성능 향상을 위한 솔루션을 모색하고 있다. 그러나 멀티 코어와 같은 새로운 프로세서 기술을 적용하기 위해 대부분의 기존 애플리케이션은 상당한 수정 없이는 멀티 코어 아키텍처의 장점을 얻을 수 없다. 그래서 멀티 코어를 적용하는 것은 시스템 복잡도에 아주 큰 영향을 미치며, 이는 개발과정을 어렵게 한다. 예를 들어 추가된 코어들은 Task 할당에 대한 경우의 수를 기하급수적으로 증가시킨다. 동시에 Task 할당은 실시간성과 효율성, 데이터 통신 오버헤드에 직접적인 영향을 미친다. 우리는 이러한 문제를 해결하기 위해 타이밍 분석에 필요한 모든 정보를 AUTOSAR 모델로부터 얻을 수 있는가? 우리가 놓치고 있는 결정적인 부분은 없는가? 이러한 의문에 대하여 타이밍 시뮬레이션으로 접근하고자 한다.

## 2. 타이밍 시뮬레이션

타이밍 시뮬레이션은 그 이름이 함축하고 있듯, 스케줄링 동작의 시뮬레이션을 통해 소프트웨어 시스템의 동적 동작을 예측한다. 시뮬레이션은 해당 시스템에 대한 설명인 모델에 기반한다. 이 모델이 제공하는 시스템에 대한 필수 정보는 일반적으로 다음과 같다:

### 2.1 소프트웨어

애플리케이션 소프트웨어를 나타내는 부분. 이는 주로 우선순위와 같은 이들의 속성, 그리고 실행 조건을 가진 Runnable 들을 포함한 시스템 내부의 Task를 의미한다. Runnable은 실행 시간, 호출된 통신 API, 데이터 신호, 그리고 액세스된 자원과 같은 속성으로 기술된다.

### 2.2 하드웨어

하드웨어의 내부 절차를 나타내는 부분. 여기서 모델링 된 세부 정보의 수준은 코어의 수와 처리 성능에 따라 다양하며, 메모리 모듈, 캐시, 버스 네트워크, 크로스바와 같은 동작 설명과 상세한 메모리 토폴로지까지 포함할 수 있다.

### 2.3 운영 체제

커널을 나타내는 부분. 일반적으로 이는 스케줄링 알고리즘과 더불어 신호기, 이벤트 알림과 같은 리소스 관리 및 우선순위 상한 프로토콜(Priority Ceiling Protocol) 등 스케줄러에 대한 상세 설명을 포함한다.

### 2.4 Runtime Environment (RTE)

서로 다른 코어들과 네트워크 인터페이스 사이의 통신 백본을 나타내는 부분.

### 2.5 Environmental Stimulation

시스템과 상호 작용하는 외부 입력을 나타내는 부분

타이밍 시뮬레이션의 큰 장점 중 하나는, 모델에 의해 나타나는 추상화의 수준이 고정되지 않고 변경될 수 있으며, 주어진 정보에 맞게 조정될 수 있다는 것이다. 이는 개발 과정의 모든 단계에서 시스템의 동작에 대한 평가를 가능하게 한다. 예를 들어 초기 개발 단계에서 타이밍 시뮬레이션은 소프트웨어 시스템의 필요 사양을 예측할 수 있다. 개발 중인 시스템에 대한 이러한 막연한 정보만으로도 설계에 대한 결정을 내릴 수 있다. 예를 들어 필요한 하드웨어에 대한 결정이나, 사용 가능한 하드웨어 플랫폼이 리소스 요구사항을 만족하는지 확인하는 것에 사용될 수 있다. 추후 개발 단계에서는 보다 정확한 정보를 활용하여 시뮬레이션 모델을 단계적으로 조정할 수 있다. 이에 따라 각각의 정보를 통해 실제 시스템의 동작을 보다 정확하게 평가할 수 있다.

## 3. AUTOSAR

AUTOSAR(AUTomotive Open System ARchitecture)는 자동차 산업을 위한 표준형 소프트웨어 아키텍처다. AUTOSAR의 주된 목표 중 하나는 기능의 재활용성을 증진시키기 위해 여러 공급업체의 소프트웨어들이 통합 가능하게 하는 것이다. 따라서 모듈 방식의 소프트웨어 아키텍처가 정의되었고, 기반하고 있는 하드웨어와는 독립적으로 소프트웨어 기능을 구현할 수 있게 되었다. AUTOSAR의 이러한 관점은 타이밍 제약사항을 위반하지 않는 한, 통신 대상이 동일한

ECU에 위치하는지, 아니면 차량의 다른 끝에 있는지는 문제가 되지 않는다는 것을 의미한다. 이는 보다 상세한 멀티 코어 매핑에 대한 결정에서도 동일하게 적용된다. 이를 달성하기 위해 개별 소프트웨어 기능이 소프트웨어 컴포넌트 안에 캡슐화된다. 그러나 각각의 컴포넌트는 표준화된 인터페이스를 통해서만 서로 또는 운영 체제와 통신할 수 있다. 이러한 상호 운용을 가능하게 하려면 각 소프트웨어 컴포넌트에 대한 정보를 포함하여 아키텍처를 구성해야 한다. 요구되는 정보는 아래와 같은 AUTOSAR 모델에 기술된다:

### 3.1 System Configuration Description / ECU extract of System Configuration

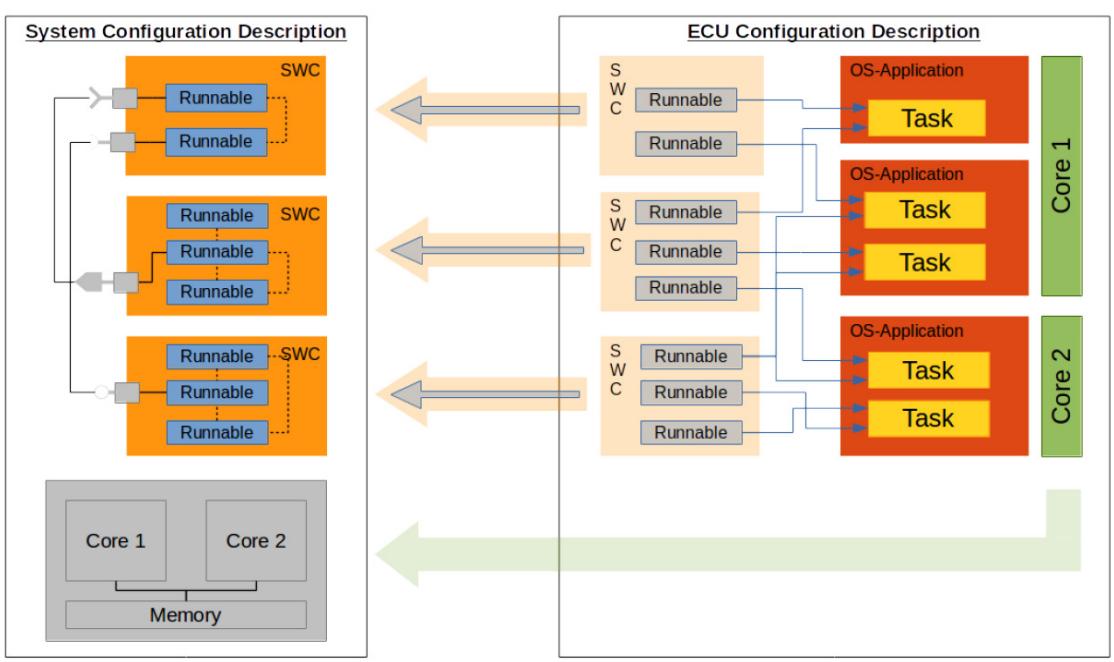
전체 시스템 또는 특정 ECU에 있는 소프트웨어 컴포넌트와 ECU, 그리고 시스템 제약사항에 대한 설명을 포함

### 3.2 ECU Configuration Description

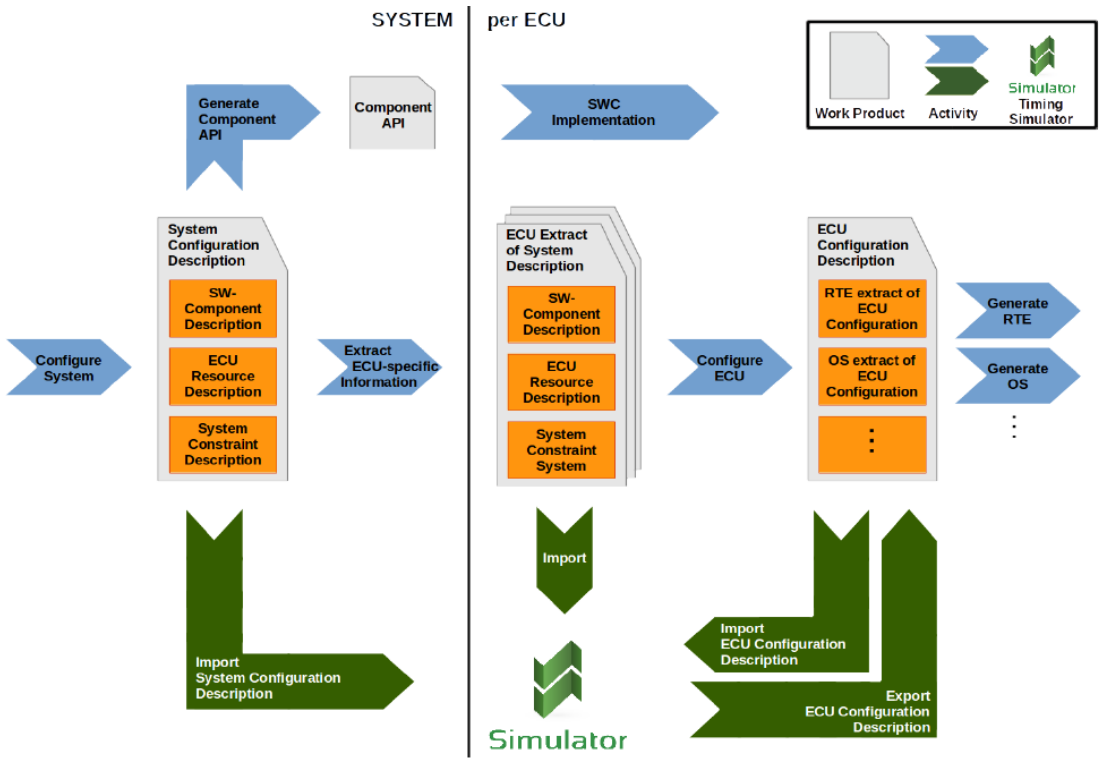
단일 ECU의 운영 체제와 런타임 환경을 포함한 기본 소프트웨어의 모든 구성 매개변수에 대한 값을 포함

### 3.3 모델 변환

AUTOSAR는 시스템 구성에 필요한 필수 정보들을 관리하기 위해 시스템 개발 중 수행되어야 하는 개발 단계에 대한 설명도 제공한다. 결국 모델은 AUTOSAR 방법론의 기초를 이루는 부분이다. 하지만 AUTOSAR모델에서 정확한 분석을 위해 필요한 모든 정보를 제공하는지에 대한 의문이 제기된다.




(그림 1) AUTOSAR의 System Configuration Description과 ECU Configuration Description




AUTOSAR 모델에서 어떠한 시뮬레이션 모델 요소들이 사용되는지에 대해 알아보면서 이 질문에 대한 답을 찾아보자. 아래 정보는 AUTOSAR 4.0 Rev 3으로 작성되었다. 이는 해당 버전이 최신 버전보다 업계에서 더 많이 사용되고 있기 때문이다. 이전 버전인 AUTOSAR 3.x 과 비교하여 4.x 버전은 중요한 두 가지 개념을 도입하였다. 멀티 코어 아키텍처에 대한 지원과 타이밍 제약사항을 정의하는 기능이 그것이다.

## 3.4 System Configuration Description

### 3.4.1 Runnable

AUTOSAR에서 Runnable은 독립적으로 실행되고 스케줄링 될 수 있는 일련의 명령들을 말한다. 각각의 소프트웨어 컴포넌트는 Runnable세트로 구성되어 있고 특정 ECU에 매핑 되어야 하기 때문에, Runnable들을 System Configuration Description으로부터 추출할 수 있다. Runnable에 대한 데이터 액세스 또한 정의될 수 있다. AUTOSAR는 두 가지 통신 패러다임을 허용한다. 즉, Sender-Receiver 및 Client-Server 통신이다. 이 패러다임의 접근 방식(Sender-Receiver 통신의 경우 암묵적/명시적 방식, Client-Server 통신의 경우 동기식/비동기식 방식)들은 읽기/쓰기 데이터 액세스 및 함수 호출로 전환될 수 있다. Runnable 내부에 데이터 액세스를 보호하기 위해 AUTOSAR는 Exclusive Area의 제약사항을 제공한다. 예를 들어 세마포어 등을 통해 리소스가 보호될 수 있다. 또한 동적 동작에 대한 정보도 System Configuration Description에 저장할 수 있다. 예를 들어, 특정 Runnable 실행에 대한 실행 시간을 명시할 수 있다. 이는 여러 단계의 품질로 수행될 수 있다. 실행 시간의 대략적인 예상에서

부터 정확한 실행시간의 분석까지, 시간의 정확한 범위를 찾기 위하여 다양한 분석 방법들이 사용된다.

#### 3.4.2 Runnable Execution Sequence

Runnable 이 소프트웨어 컴포넌트에 포함되어 있다 하더라도, 이들의 실행 순서에 대한 정보는 제공되지 않는다. 그러나 AUTOSAR 릴리즈 4.0의 Timing Extensions에서 소개된 Execution Order Constraint로, Runnable 의 실행 순서를 System Configuration Description에서 정의될 수 있다.

#### 3.4.3 Timing Constraints

AUTOSAR 릴리즈 4.0으로 Timing Extensions는 표준화 되어 가고 있다. 그 목적은 관련 타이밍 Dependency와 Constraint를 정의하기 위한 일관된 방법을 제공하는 것이다. 이는 전체 개발 과정에 걸쳐 시스템의 타이밍 동작을 분석하고 분석 결과를 시간 제약사항과 비교하여 검증할 수 있도록 하였다. 각각의 개발 단계에서 필요한 정보의 가용성에 따라 여러 가지 타이밍 제약사항을 이용하여 AUTOSAR 시스템의 타이밍 동작을 설명할 수 있다. 예를 들어, Runnable 의 시작 간의 최대 반복 시간이나 주어진 시간 내에서 연속되는 데이터 액세스 간의 최소 시간 차이가 될 수 있다. 또한 특정 이벤트 발생뿐만 아니라 Event Chain을 사용하여 시스템 런타임 동안 이들 간의 데이터 흐름 관계를 명시할 수 있다. 이러한 방법으로 전체 시스템 레벨로 특정 시나리오에 대한 타이밍 동작을 명시할 수 있는 것이다.

#### 3.4.4 하드웨어

AUTOSAR에서는 실제 하드웨어에 대한 설명을 제공할 수 있다. 계층적 방식으로 ECU의 개별 하드웨어 요소와 이들의 상호 연결을 설명할 수 있다. 이를 위해 하드웨어 요소에 대한 카테고리가 사전에 정의되며 특정 속성들을 제공한다. 예를 들어 마이크로 컨트롤러 내부 처리장치의 수를 결정할 수 있다.

### 3.5 ECU Configuration Description

#### 3.5.1 Task

Runnable과 유사하게 ECU Configuration Description은 각 Task에 대한 별도의 입력을 가진다. 이 입력의 매개변수는 우선순위, 대기열에 포함될 수 있는 최대 활성화 요청 개수, Task의 선점에 대한 정보를 포함한다. AUTOSAR에서의 Task-To-Core 할당은 OS-Application에 의해 명시되어 있다. OS-Application으로 Task 및 인터럽트들이 기능 단위로 묶여질 수 있다.

#### 3.5.2 Runnable Mapping

소프트웨어 컴포넌트의 모든 Runnable에 대하여, ECU Configuration Description 내에 별도의 입력이 존재한다. 여기서 Runnable에 대한 추가적인 매개변수가 설정된다. 예를 들어, Runnable Entity가 어떠한 Task에 매핑 되는지와 해당 작업 내에서 어떤 위치에 매핑 되는지가 정의된다. 추가적으로 Schedule Point 도 명시할 수 있다.

#### 3.5.3 Stimulation

각 Task에 대한 Activation 패턴은 Schedule Table에 의해 명시된다. 시간에 대하여 명세하기 위해 Schedule Table은 카운터에 의해 관리된다. Expiry Point는 운영 체제가 Task Activation또는 Event를 설정하는 시점을 정의한다. 또한 매개변수는 마지막 Expiry Point가 처리된 이후

Schedule Table을 다시 시작할지 명시한다.

#### 3.5.4 하드웨어

하드웨어에 대한 정보는 ECU Configuration Description에 간접적으로만 포함된다. 예를 들어, 해당 ECU에서 이용할 수 있는 처리 코어의 수를 매개변수에서 결정할 수 있지만 이는 선택적이다.

## 4. 한 계

앞에서 살펴본 것과 같이 시뮬레이션 모델을 얻는 데 필수적인 정보가 상당 부분 이미 AUTOSAR를 준수하는 개발 과정에 존재한다는 것을 알 수 있다. 그러나 정적으로 소프트웨어 통합 수행하는 AUTOSAR의 초기 의도 때문에 타이밍 시뮬레이션을 완벽하게 지원하는 데 몇 가지 제약이 있다. 세부적인 타이밍 동작을 반영하기 위한 동적 속성들은 여전히 제공되지 않는다. 이러한 한계점은 다음과 같다.

### 4.1 Runnable의 동적 동작

신호 액세스와 이들에 대한 보호는 AUTOSAR를 통해 명시할 수 있다. 그러나 모델은 아직 타이밍 세부 정보들을 포함하고 있지 않다. 예를 들어, Runnable이 데이터 신호에 얼마나 자주 액세스하는지, 또는 어떠한 조건에서 액세스하는지에 대한 정보가 없다. 모델은 또한 Runnable내에서 동작들의 순서에 대해 명시하지 않는다. 예를 들어, DataReadAccess와 더불어 DataWriteAccess를 가진 Runnable모델을 생각해보자. 데이터 신호가 먼저 읽혀지는지, 아니면 쓰여지는지가 분명하지 않다. 다른 한편으로 해당 데이터 신호에 대한 다수의 읽기 또는 쓰기 액세스가 있는지도 알 수 없다. Exclusive Area가 사용되는 경우에도 동일한 문제가 발생한다. 이 경우 AUTOSAR는 동일한 Exclusive Area를 가진 Runnable은 동시에 실행되지 않아야 한다는 것만을 명시한다. 그러나 Exclusive Area가 전체 Runnable의 실행을 보호하지 않는 경우, 이 Exclusive Area에 진입하고 나간 정확한 시간에 대한 정보는 모델에 존재하지 않는다.

### 4.2 운영 체제와 RTE의 동적 동작

Runnable에 실행 시간을 명시할 수 있지만, OS 및 RTE 실행 시간에 대한 정보는 AUTOSAR 내에서 모델링 될 수 없다. 응용 소프트웨어의 리소스 소비를 명시하는 방법이 있다 하더라도, 커널 및 통신 계층에서 생성된 오버헤드에 대한 정보는 정확한 시뮬레이션을 위해 필수적이다. 이를 해결하기 위해서는 운영 체제 벤더와 시뮬레이션 툴 벤더는 운영 체제의 런타임 동작에 대한 세부 정보를 교환할 필요가 있다. 이를 통해 시뮬레이션 툴 벤더는 오버헤드 정보를 시뮬레이션에 연동시킬 수 있고, 런타임 동작에 대한 더욱 자세한 분석을 제공할 수 있다.

### 4.3 하드웨어 설명

AUTOSAR가 실제 하드웨어 관련 특징을 설명하는 방법을 제공한다 하더라도, 이는 아주 일반적인 방법으로 수행된다. 개별적이고 표준화되지 않은 속성들은 여러 툴 벤더의 일반적인 활용을 제한한다. 예를 들어 마이크로 컨트롤러 코어를 나타내는 처리 장치를 생각해 볼 수 있다. 기본적으로 AUTOSAR는 이러한 카테고리에 대한 속성을 정의하지 않는다. 따라서 코어의 프리퀀시(Frequency)와 같은 속성을 일반 속성으로 정의해야 한다. 결국 이것은 서로 다른 해석 문제로

이어질 수 있다.

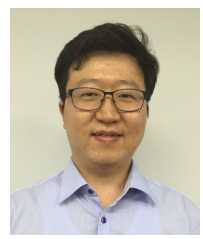
## 5. 결 론

지금까지 AUTOSAR에서 제공하고 있는 타이밍 정보의 부족한 부분에 대하여 설명했다. 현재로서는 AUTOSAR에서 제공하고 있는 타이밍 정보로 실제 ECU의 동적 동작을 예측하기 어렵다. 이러한 부분은 타이밍 시뮬레이션으로 ECU의 동적 동작을 예측하고 타이밍 문제를 예방할 수 있다. 타이밍 시뮬레이션과 같은 타이밍 분석에 대한 방법론은 AUTOSAR의 TIMEX와 Timing Analysis부분에 자세하게 설명되어 있다.

### 참 고 문 헌

[1] AUTOSAR: http://www.autosar.org/
[2] OSEK VDX: http://www.osek-vdx.org/

### 저 자 약 력



**남 동 현**

이메일: donghyun@mdstec.com

- 2008년 한양대학교 전자공학부 학사 졸업
- 2008년~현재 MDS테크놀로지 SES사업부 선임 연구원