Practical approach in system engineering requirements between A-SPICE and ISO 26262

2018. 02.

Jerome Joo

System engineering process group v.s ISO 26262

* ISO 26262 is based on 2nd edition

[System engineering process group-SYS]

SYS. 1	Requirements Elicitation	BP 6
SYS. 2	System Requirements Analysis	BP 8
SYS. 3	System Architectural Design	BP 6
SYS. 4	System Integration and Integration Test	BP 9
SYS. 5	System Qualification Test	BP 7

[Product development at the system level –Part4]

4-5	General topics for the product development at system level	RE (6)
4-6	Technical safety concept	RE 6
SYS. 3	System architectural design	RE 8
SYS. 4	System and item integration and verification	**RE 4 (28)**
SYS. 5	Safety validation	RE 4

Traceability and consistency

- Engineering process group is divided with system engineering process group and software engineering process group (ENG → SYS, SWE)
- Traceability and bidirectional → Traceability and consistency
- Verification Criteria(SYS.2) : Verification measure and inputs for system qualification test (SYS.5)
- Evaluate (SYS.3) : Alternative architecture

Traceability and consistency especially in safety functions

- System level development initiation(Part 4) → overall product development(Part 2) and it is only informative, not requirements
- Only refer to cybersecurity concept, but it is other approach required (i.e. TARA etc.)
- Clarify the safety mechanism concept of latent faults
- 1st 2nd Both are addressing that safety and non-safety requirements are handled and satisfied in one process
- 1st 2nd Both are requiring safety analysis should be necessary but 2nd is not addressing quantitative analysis but only qualitative method.

Verification of outputs is emphasized on both A-SPICE and ISO 26262 by verification criteria, consistency and verification activities

What's a system?

What is a system in automotive E/E industries?

- It is very important concept in product development process but many R&D dose not define the its concept well
- System is real existence or the upper concept of implementation

Target object : Embedded automotive system

System :

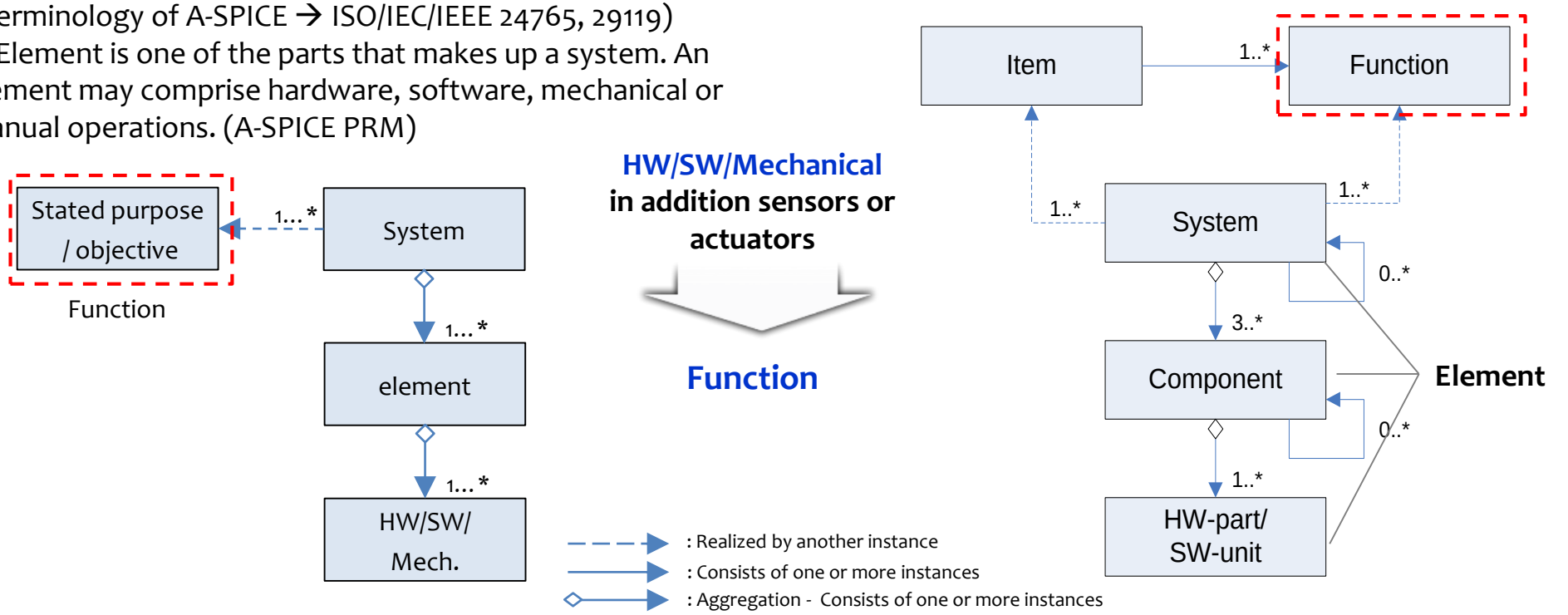
1. combination of interacting elements organized to achieve ore or more stated purpose.
2. something of interest as a whole or as composted of parts.
3. interacting of elements to accomplish a defined objective.

(Terminology of A-SPICE → ISO/IEC/IEEE 24765, 29119)

4. Element is one of the parts that makes up a system. An element may comprise hardware, software, mechanical or manual operations. (A-SPICE PRM)

System or array of systems : to implement a function at the vehicle level

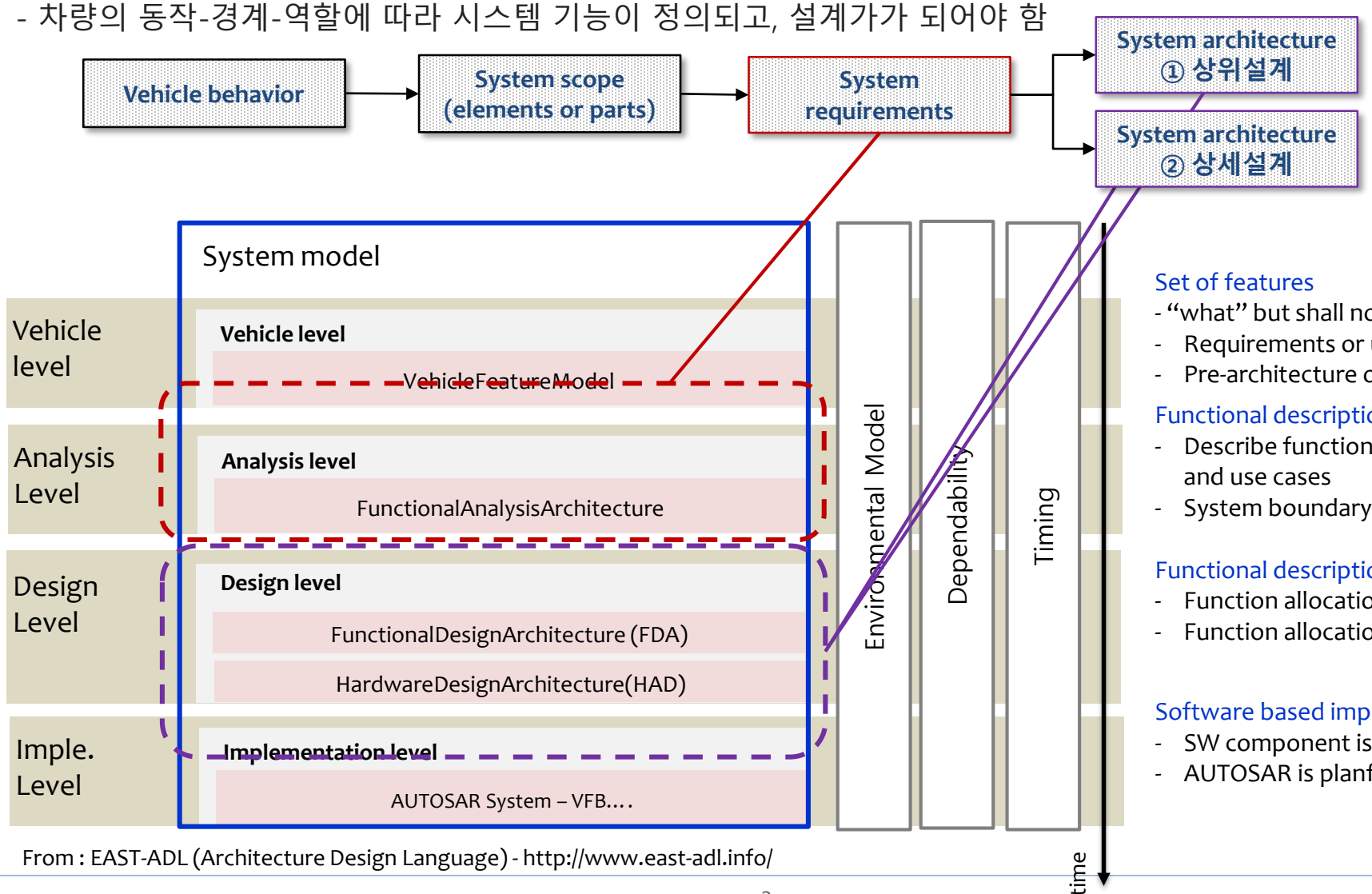
System : set of elements that relates at least a sensor, a controller and an actuator with one another and the related sensor or actuator can be included in the system, or can be external to the system. (ISO 26262)



How to specify the system requirements

시스템 레벨에서의 기능을 표현하는 것은 어려운 주제임 → System modeling 시도 (Based on UML)

- 차량 동작에서 시스템의 역할 및 시스템의 경계의 정의가 우선되 되어야 함
- 차량의 동작-경계-역할에 따라 시스템 기능이 정의되고, 설계가가 되어야 함



Set of features
- "what" but shall not fix the "how"
- Requirements or use cases
- Pre-architecture of vehicle level

Functional description of the E/E system
- Describe function from requirements and use cases
- System boundary of E/E system

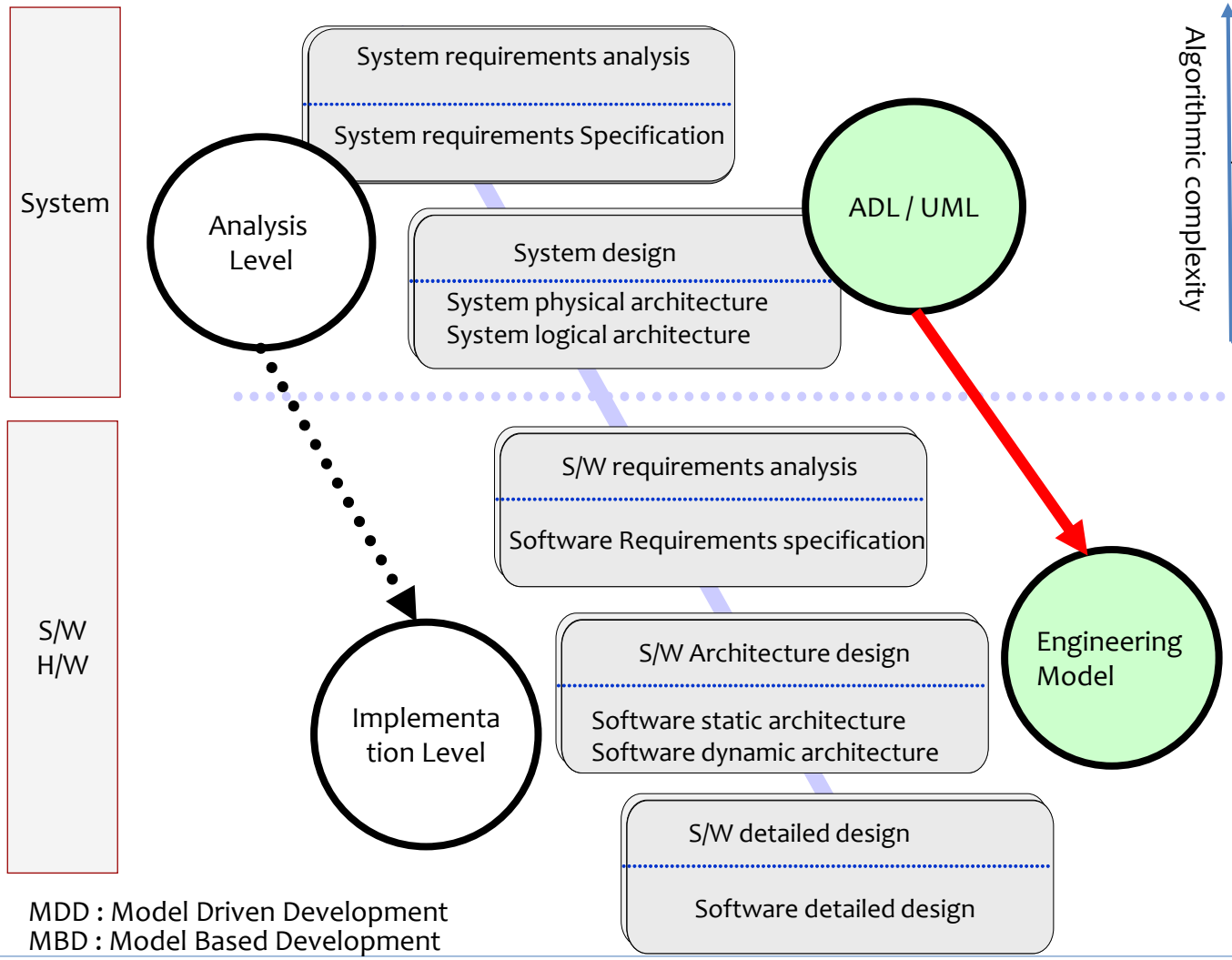
Functional description of ASW/HW
- Function allocation to HW
- Function allocation to SW

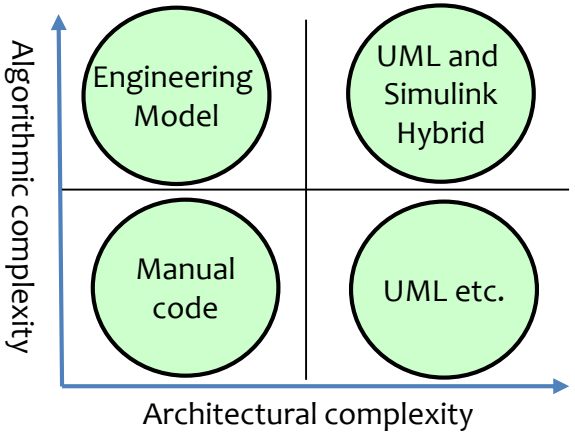
Software based implementation
- SW component is application
- AUTOSAR is planform

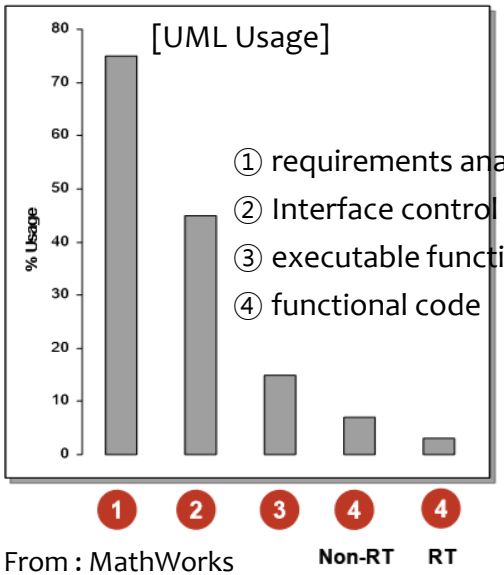
From : EAST-ADL (Architecture Design Language) - http://www.east-adl.info/

The method of system requirements' specification.

SysML, UML (MDD) are useful in system modeling which describes functional concept or requirements.
But it is not appropriate to implementation level → Engineering Model(MBD)
요즈음은 통칭 MBD라고 통용됨





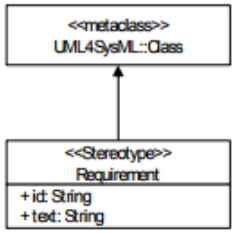


From : MathWorks

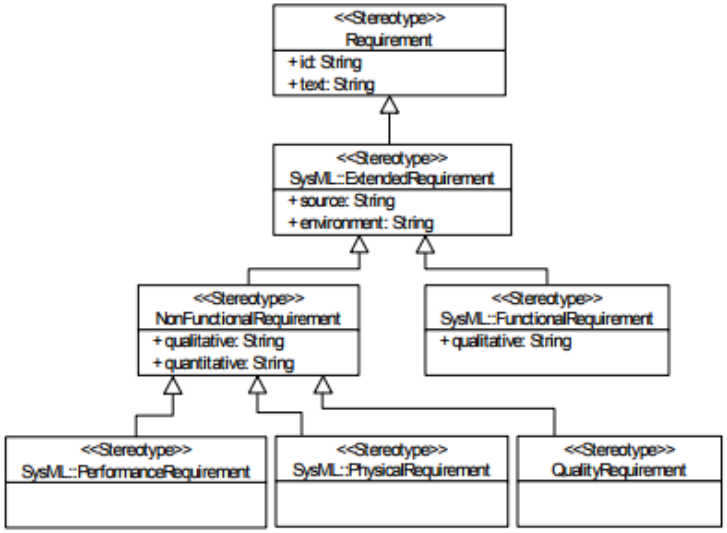
MDD : Model Driven Development
MBD : Model Based Development

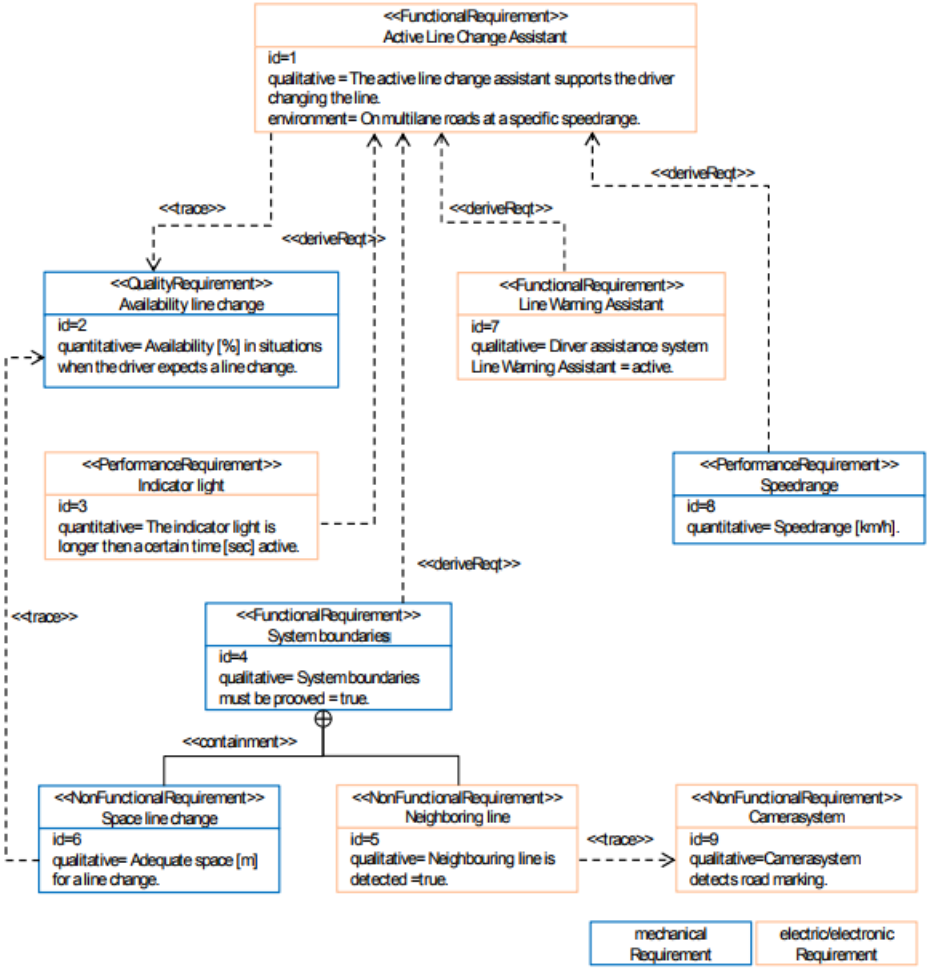
SysML description of lane change

System requirements include functional and non functional requirements in lane change function



- SysML Stereotype Requirements – metaclass





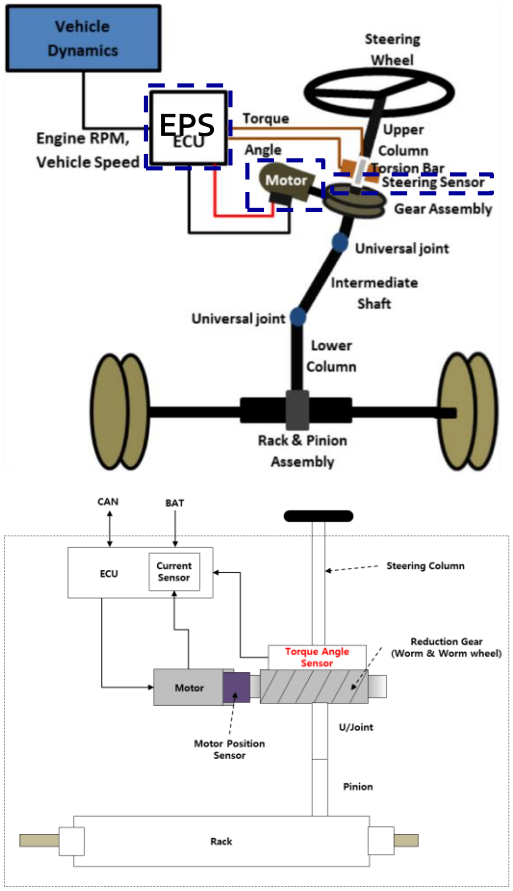
From : K.Gruber, J. Huemer (BMW Group) Integrated Description of Functional and Nonfunctional Requirements for Automotive Systems Design Using SysML

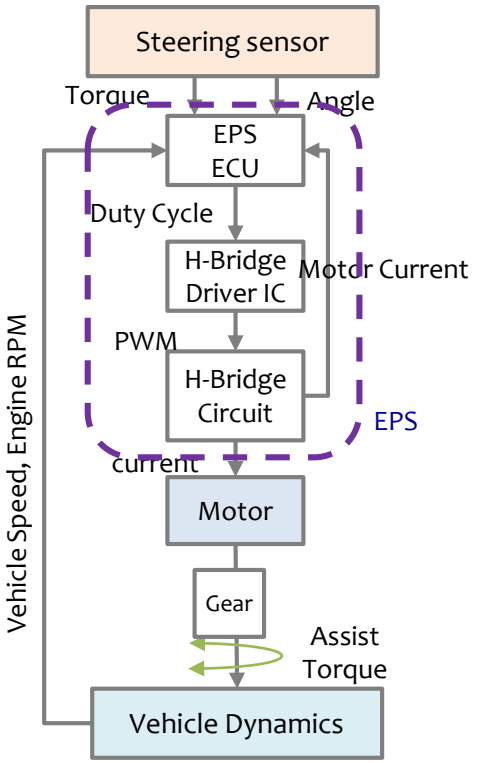
All engineering steps requiring the some level architecture

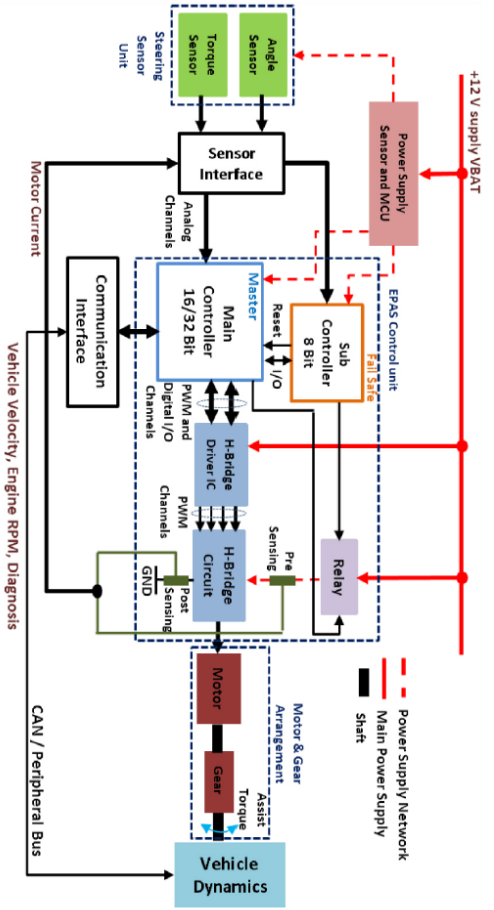
Requirements which have no concept of architecture are the fiction

Vehicle level -Sys → **Analysis level-Sys** → **Design level-Sys**

모든 레벨에서 아키텍처가 표현되어야 하며, 요구사항이 상세화 될수록 진화 되어야 한다.



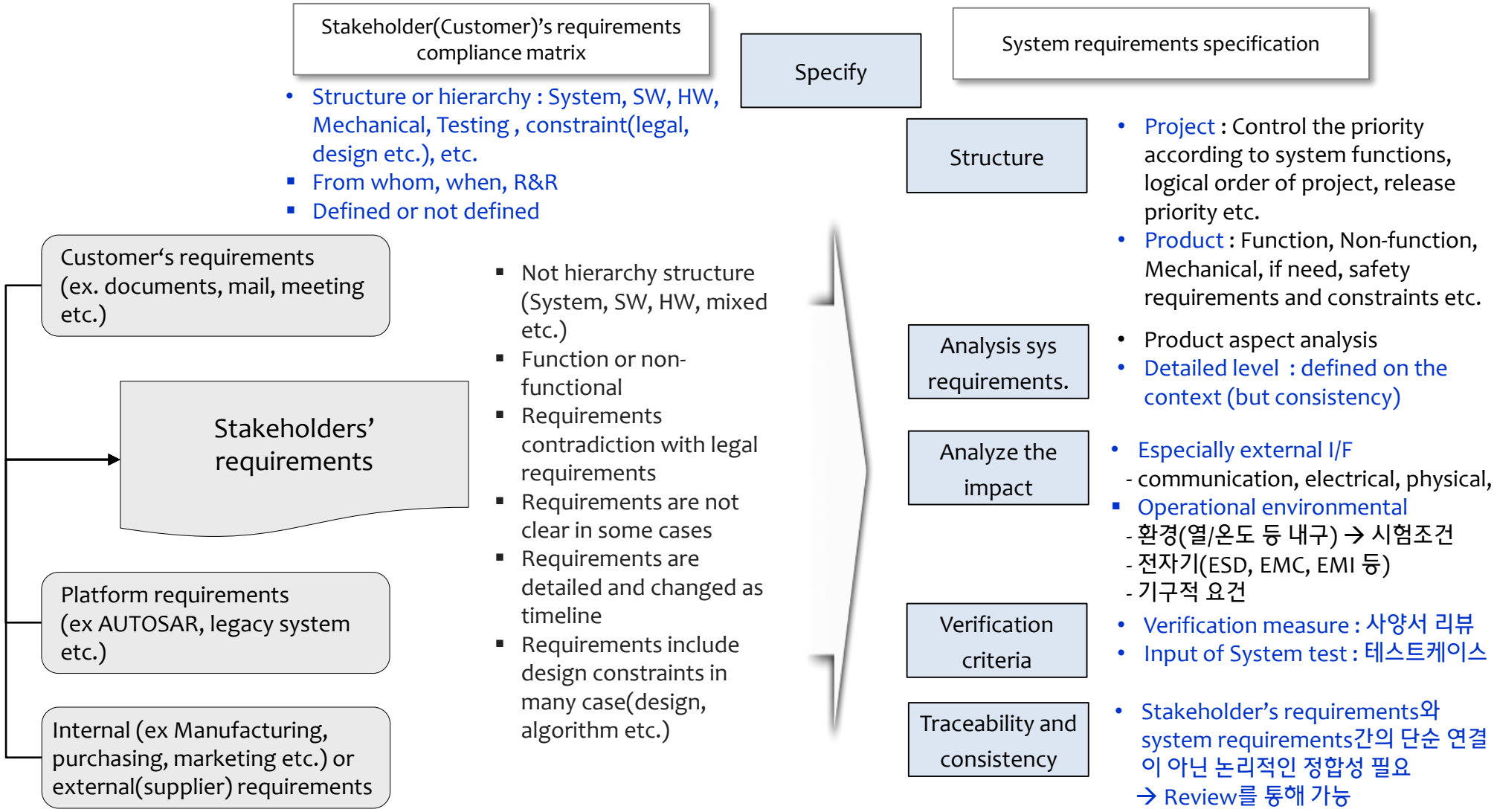




Refer from : Development of Electrical Power Assisted Steering (EPAS) Considering Safety and Reliability Aspects as per ISO 26262

System engineering process group

System's requirements analysis

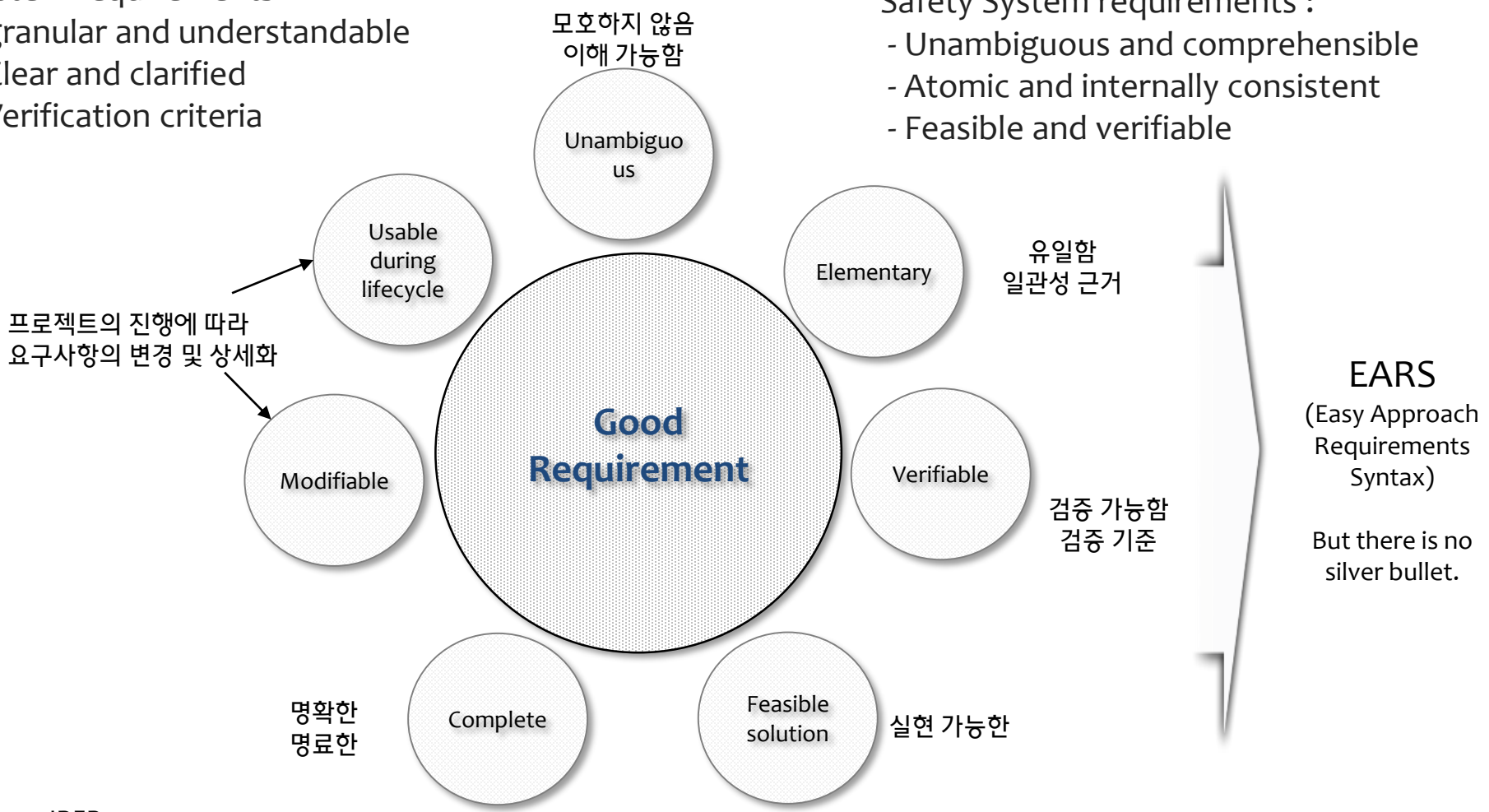


Requirements' attributes

System requirements shall have same attributes which is required in the requirements engineering

System requirements :
- granular and understandable
- Clear and clarified
- Verification criteria

Safety System requirements :
- Unambiguous and comprehensible
- Atomic and internally consistent
- Feasible and verifiable



From IREB

EARS method

System level에서 요구사항을 기술하기 위한 방법
- EARS can be converted to UML
- 7 requirements types

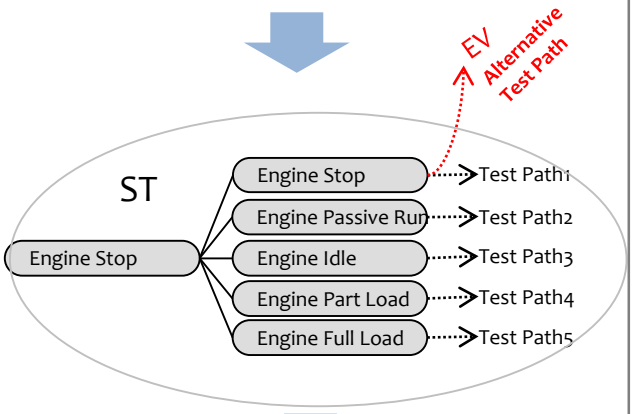
Req. Type	Description	Example
Info (Information) 정보	상황이나 이론을 설명하거나 요구사항에 대한 정보를 제공	
UB (Ubiquitous) 주요 일반 기능	항상 무엇(What)이 어떤 일(Functionality)을 수행하는지에 대해 기술	<entity>는 <functionality> **해야 한다**
		<entity>는 <entity>의 <functionality>를 **(위해)대해** <functionality> **해야 한다**
EV (Event-Driven) 이벤트 기능	시스템의 경계조건에서 이벤트가 감지되거나 입력되었을 경우(When)에만 반응(Functionality)	<precondition>**일 때 (발생하면)** <entity>는 <functionality>**해야 한다.**
		<entity>가 <functionality>**했을 때 (하면)** <entity>는 <functionality>**해야 한다.**
UW (Unwanted Behavior) 의도하지 않은 동작	의도하지 않은 상태나 동작(If precondition: 내부 결함, 외부 방해, 의존 고장 등에 의한)이 감지되었을 경우에 반응(Functionality)	**만약** <preconditions>**이면(하면),** <entity>는 <functionality> **해야 한다.**
		만약 <preconditions>**이면(하면),** <functionality>의 <functionality>는 <functionality>와 <functionality>**에 대해** <functionality> **해야 한다.**
ST (State Driven) 상태	특정 상태를 유지하기 위해 또는 특정 상태 동안에 수행해야 할 일(Functionality)	<in a specific state> **동안,** <entity>는 <functionality> **해야 한다.**
		<in a specific state> **동안,** <functionality>는 <functionality> **해야 한다.**
OP (Optional Features) 추가 기능	주요 기능 외에 추가적으로 포함되어야 하는 기능(Functionality)에 대해 기술	<feature is included> **경우에는** <entity>는 <functionality> **해야 한다.**
		<preconditions>**경우에는** <functionality>는 <functionality>**에 대해** <functionality> **해야 한다.**
HY (Hybrid) 복합 기능	Complex Requirements Syntax로써, 복합적으로 발생할 수 있는 상황 대처 기능(Functionality)에 대해 기술	<in a specific state> **동안,** <precondition>**일 때 (발생하면)** <entity>는 <functionality>**해야 한다.**
		<preconditions>**일 때 (발생하면), 만약** <precondition>**이면(하면),** <functionality>는 <functionality>**해야 한다.**
		<in a specific state> **동안, 만약** <precondition>**이면,** <functionality>는 <functionality>**해야 한다.**

EARS Method (Continued)

➢ **Hybrid Type : EARS에서 정의한 2개 이상의 Type이 결합된 형태이며 각 Type별 검증 방안을 결합하여 검증 진행**

특정 상태에서 이벤트 발생 (ST & EV)

ID	State <While>	Event <When>	Entity <Who>	Functionality <What>
REQ_07	차량 정지 상태 동안	역 추진력이 요구되면	제어기는	역 추진력을 발생시켜야 한다

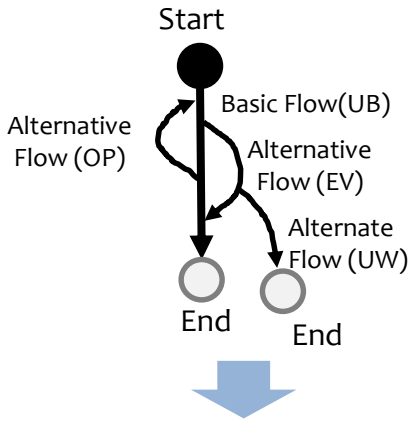


ST관련 Test Scenario에 EV 포함

ID \ Seq. #	1	2	3	4	5
TP1	VTC01	VTC02	VTC03	VTC04	
TP2	VTC05	VTC06	VTC07	VTC08	VTC09
TP3	IVTC01	IVTC02	IVTC03	**Alternative T/C**	
TP4	G01	G02	G03	G04	

이벤트 발생 시, 의도하지 않은 동작 (EV & UW)

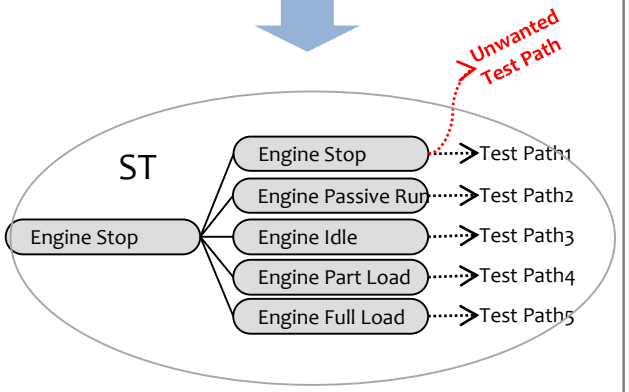
ID	Event <When>	PreCond <If>	Entity <Who>	Functionality <What>
REQ_08	배터리 SOC Low 발생시	배터리 충전 불가 상황 발생 시	제어 시스템은	HEV 모드로 주행한다



TC ID	입력조건 <EV> Battery SOC	입력조건 <UW> Motor Fault	기능목표 Battery Charge	예상 결과	테스트 결과	판정 결과
1	Low	Off	Failure			NG
2						OK

특정 상태에서 의도하지 않은 동작 (ST & UW)

ID	State <While>	PreCond <If>	Entity <Who>	Functionality <What>
REQ_09	차량 주행 상태 동안	역 추진력이 요구되면	제어 시스템은	역 추진력 발생을 금지시켜야 한다



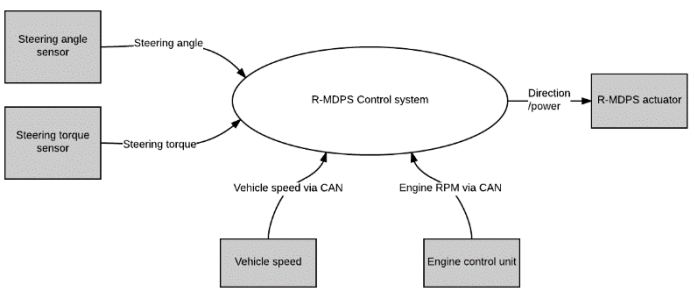
ST관련 Test Scenario에 UW 포함

ID \ Seq. #	1	2	3	4	5
TP1	VTC01	VTC02	VTC03	VTC04	
TP2	VTC05	VTC06	VTC07	VTC08	VTC09
TP3	IVTC01	IVTC02	IVTC03	**Unwanted T/C**	
TP4	G01	G02	G03	G04	

Next, system architecture

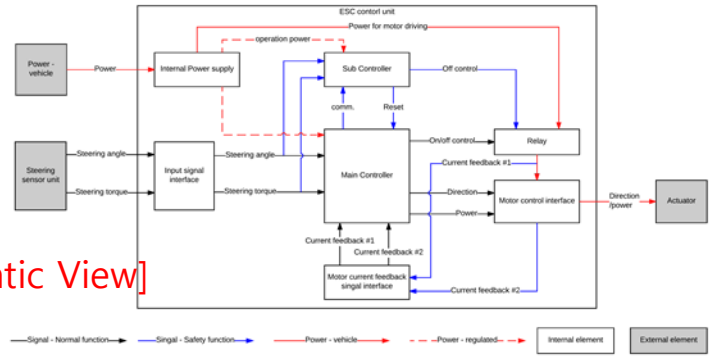
System architecture will be the agenda of the next seminar

[System Architecture level]



System Context

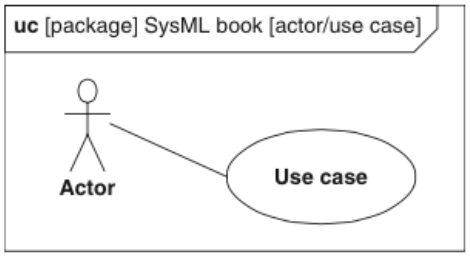
<<decompose into>>



[Static View]

System Structure (ibd)
with block function definition

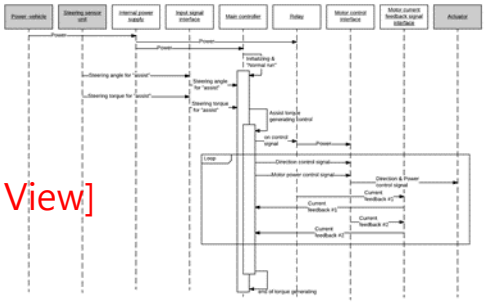
<<derived from>>



Use case and Scenario

<<satisfy>>

<<consistent with>>



[Dynamic View]

System Behavior

[System Requirements level]

Q&A

주백수
innospi.joo@espid.com
innospi.joo@gmail.com