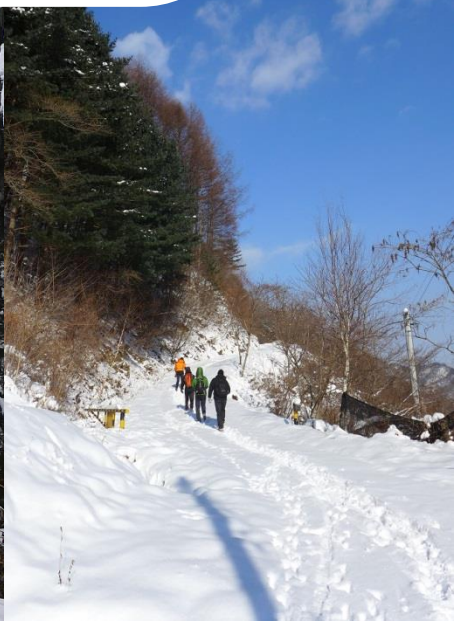




SA with IEEE 1471





목차

1. 개요
2. IEEE 1471 Key Concepts
3. IEEE 1471 Close-up
4. IEEE 1471 Review
5. SA Summary
6. 토의

1. 개요 – 아키텍처 범람

- ✓ 너무나 많은 아키텍트의 존재로 인해 모두가 혼란스럽군요.
- ✓ 인사 담당자는 아키텍트 직무가 너무 많아 혼란스럽고,
- ✓ 프로젝트 관리자는 아키텍트가 너무 많이 필요해서 어려움이 있고,
- ✓ 엔지니어링 팀은 아키텍처 팀을 어떻게 구성해야 할 지 고민이 있습니다.

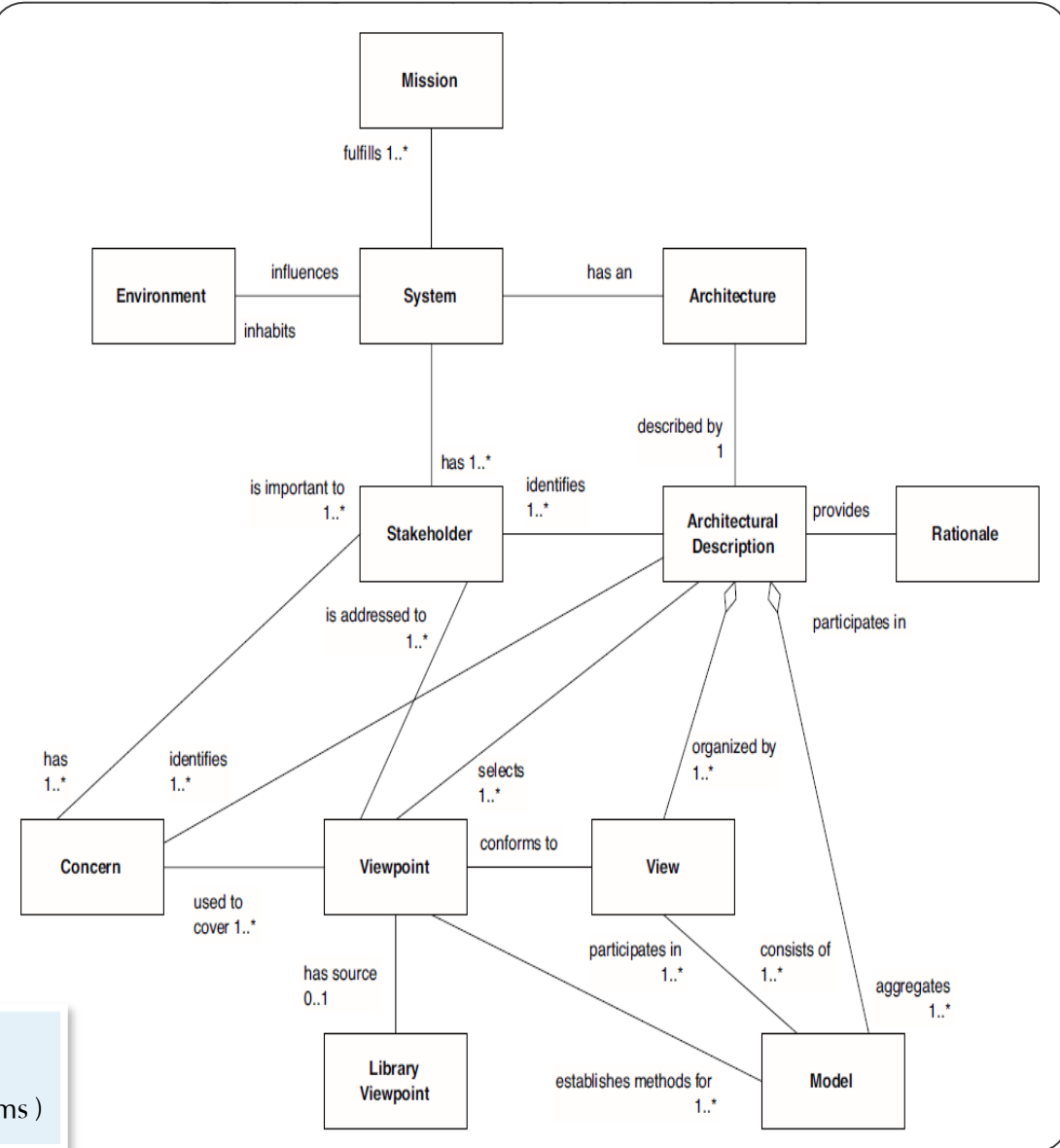
- ❖ 시스템 아키텍트
- ❖ 솔루션 아키텍트
- ❖ 보안 아키텍트
- ❖ 기술 아키텍트
- ❖ 네트워크 아키텍트
- ❖ 컴포넌트 아키텍트
- ❖ 애플리케이션 아키텍트

- ❖ 비즈니스 아키텍트
- ❖ 정보 시스템 아키텍트
- ❖ 데이터 아키텍트
- ❖ EAI 아키텍트
- ❖ 포털 아키텍트
- ❖ 서비스 아키텍트
- ❖ 소프트웨어 아키텍트

[너무나 많은 아키텍트]

1. 개요 – Why IEEE 1471 ?

- ✓ IEEE 1471는 아키텍처 서술을 위한 권장 실행지침으로 표준에 준하는 위치를 가집니다.
- ✓ 아키텍트의 역할 논쟁을 표준을 근거로 조정하고 정리하는 것이 바람직합니다.
- ✓ IEEE 1471에서 아키텍처를 어떻게 정의하고 있는가로부터 출발합니다.
- ✓ [EA와 TOGAF]에서 정의한 IT 영역 개념도 아키텍처 역할 정리에 필요합니다.

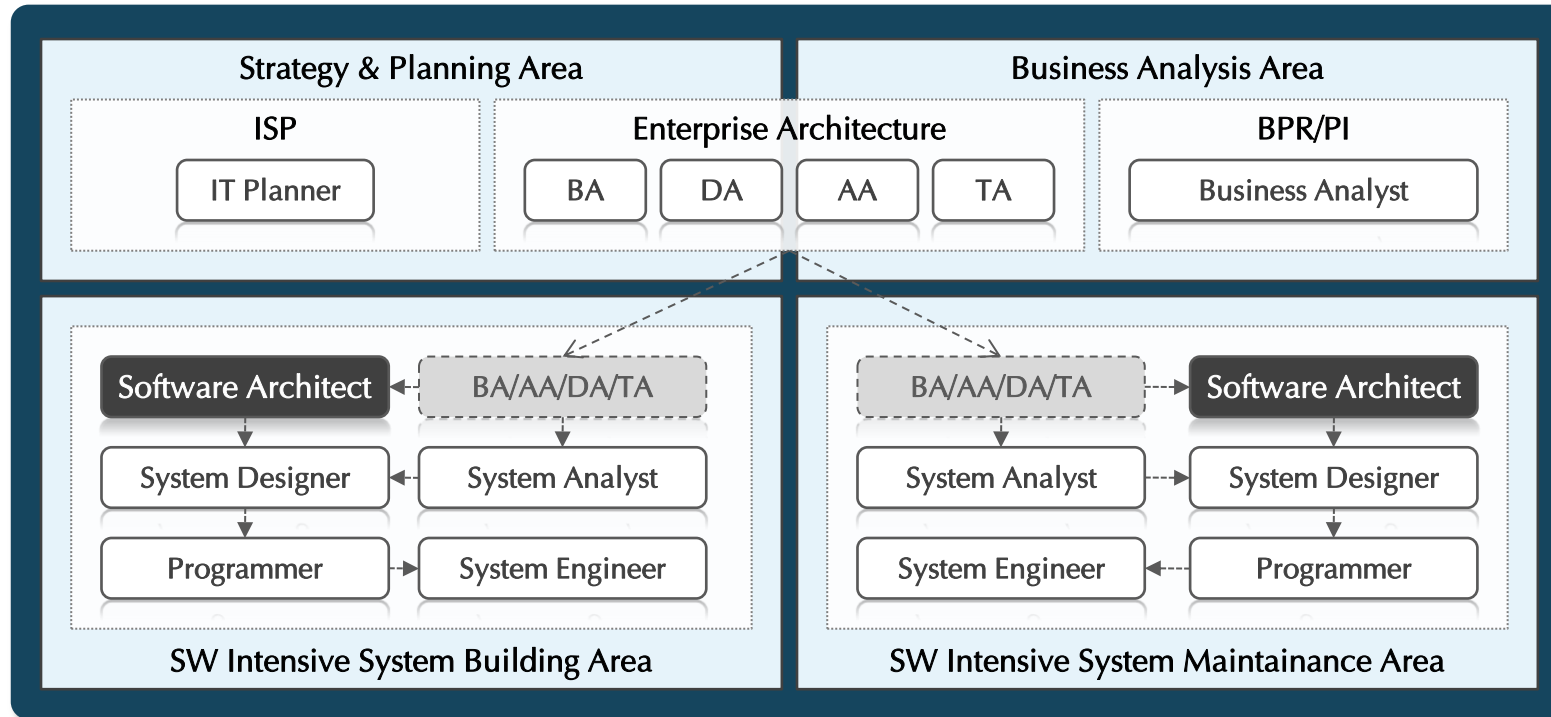


IEEE 1471 - 2000

(IEEE Recommended Practice for Architectural Description of Software-Intensive Systems)

1. 개요 - 혼란의 핵심

- ✓ 혼란의 핵심에는 EA에서 정의한 AA, DA, TA, BA가 있습니다.
- ✓ 일부 조직의 경우 Software Architect 는 없고 AA, DA, TA, BA 만 직무로 정의하고 있습니다.
- ✓ 특히 AA와 SA는 역할 중첩 논란이 항상 존재하고 있습니다.
- ✓ EA-AA는 거버넌스를 이유로 시스템 빌딩 영역에서도 지속적으로 활동을 해야 한다고 주장합니다.



1. 개요 – 소프트웨어 아키텍처 범위

- ✓ 소프트웨어 아키텍트는 [시스템 구축과 운영 영역]에서 정의하고 사용하며, 전 영역에서 참조하는 역할입니다
- ✓ 소프트웨어 아키텍처는 IEEE 1471에서 정의한 Software Intensive 시스템의 아키텍처를 의미합니다.
- ✓ 소프트웨어 아키텍트는 [Software Intensive System]을 위한 아키텍처를 설계하는 아키텍트입니다.
- ✓ 이러한 범위를 전제로 하고 IEEE 1471를 자세히 분해하여 이해하고자 합니다.



[기업 IT의 네 가지 영역]

2. IEEE 1471 Key Concepts (1/4)

- ✓ Software Intensive System의 범위(coverage)는?
 - 개인용 SW 애플리케이션
 - 기업 정보 시스템
 - 임베디드 시스템
 - SW Product Lines, Product Families
 - System-of- [Software intensive] System

What kind of systems does IEEE 1471 cover?

Its focus is software-intensive systems: any system in which software development and/or integration are dominant considerations (i.e., most complex systems these days). This includes computer-based systems ranging from individual software applications, information systems, embedded systems, software product lines and product families and systems-of-systems.

2. IEEE 1471 Key Concepts (2/4)

- ✓ Software Architecture란 무엇인가?
- ✓ CMU 사이트에 가면 아키텍처에 대한 정의를 60여가지 제시합니다.
- ✓ IEEE 1471에서 정의하는 아키텍처를 기준으로 합니다.
- ✓ 컴포넌트 == 아키텍처 요소(architectural element)

아키텍처는 시스템의 근간을 이루는 틀(fundamental organization)로써 시스템을 구성하는 컴포넌트, 컴포넌트 간의 관계, 컴포넌트와 환경 간의 관계, 그리고 설계와 개발 진행을 관리하는 원칙의 형태로 나타난다.

(The fundamental organization of a system, embodied in its components, their relationships to each other and the environment, and the principles governing its design and evolution)

IEEE 1471-2000

2. IEEE 1471 Key Concepts (3/4)

✓ Software Architecture란 무엇인가? (계속)

아키텍처는 시스템의 본질적(essential) 것이거나 통합적(unifying)인 것이다. 즉, 시스템의 구조(structure), 행위, 가치, 비용, 그리고 리스크를 결정하는 시스템 속성(properties) 집합이다. 제안 실행지침에서 정의는 “아키텍처는 시스템의 기반을 이루는 구조(fundamental organization)이며, 그 구성요소인 컴포넌트, 컴포넌트 상호 관계, 환경과의 관계, 그리고 설계와 개발진행을 이끌어 주는 원칙을 포함한다. 이 정의에는 서너 가지 핵심 아이디어가 있다.

첫째, 아키텍처는 시스템을 마음 속에 그린 것이다. 따라서, 아키텍처는 기록되지 않고 존재할 수도 있다. 표준에서 아키텍처와 아키텍처 서술을 구분했다. “지도가 영토는 아니다”라고 말하듯이 “아키텍처 서술은 아키텍처가 아니다.” 아키텍처 서술은 산출물로 작성된 것이거나 유형의 작업 결과물이다. 즉, 다른 사람들을 위해 마음 속에 그린 것을 표현하려는 시도이다. 표준의 초점은 아키텍처 서술 요구에 있다.

둘째, 아키텍처는 시스템의 토대가 되는 것(fundamental thing)들을 전체적으로 안고 있다. 토대가 되는 것들이란 시스템의 본질 또는 핵심에 해당하는 속성들이다.

셋째, 아키텍처는 고립 상태가 아니라 컨텍스트 안에서 이해되어야 한다. 시스템의 기반 속성(즉, 아키텍처)을 이해하는 것은 시스템이 환경과 어떤 관계에 있으며, 어떤 입장에 놓여 있는 지를 이해하는 것이다. 어떤 것을 위한 기반(fundamental)인지를 알지 못하고서는 시스템에 대한 기반이 무엇인지 알기 어렵다. 그러므로 “기반(fundamental)은 시스템 이해관계자의 컨텍스트와 그 환경 안에서 해석되어야 한다.

끝으로, 아키텍처는 시스템을 구성하는 물리적인 컴포넌트들로 구성된 구조 그 이상이다. 물리적인 구조가 흔히 시스템의 기본 모양(aspect)이긴 하지만, 반드시 있어야 하는 것은 아니다.

<http://www.iso-architecture.org/ieee-1471/ieee-1471-faq.html>

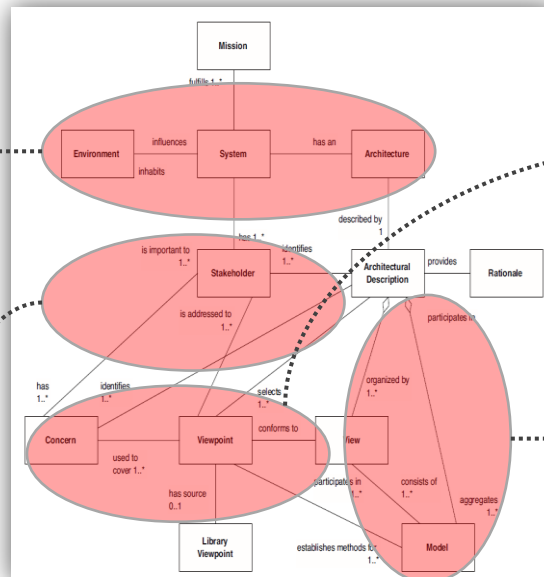
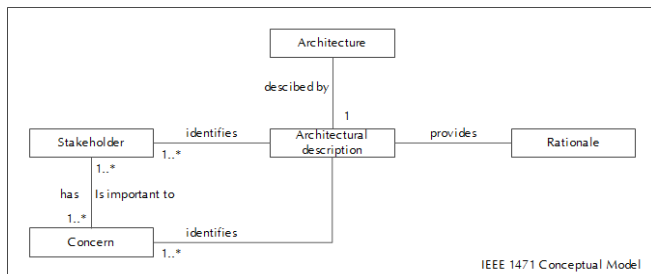
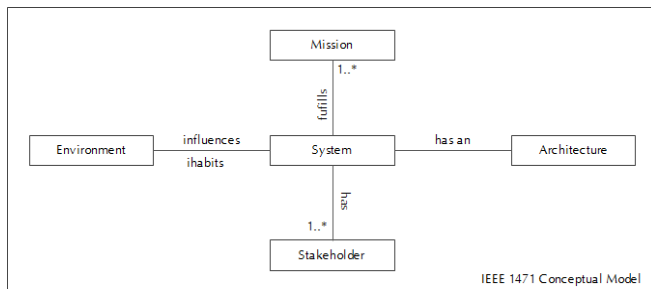
2. IEEE 1471 Key Concepts (4/4)

- ✓ 아키텍처 요소(Architectural Element)란 무엇인가?
- 시스템의 틀을 구성하는 요소를 의미합니다.
 - 아키텍처 설계 후에 이러한 아키텍처 요소를 획득(개발,구매,오픈소스선택)하여 실행 가능한 시스템을 구성합니다.
 - 모든 비즈니스 컴포넌트를 아키텍처 요소라고 할 수 있는가?

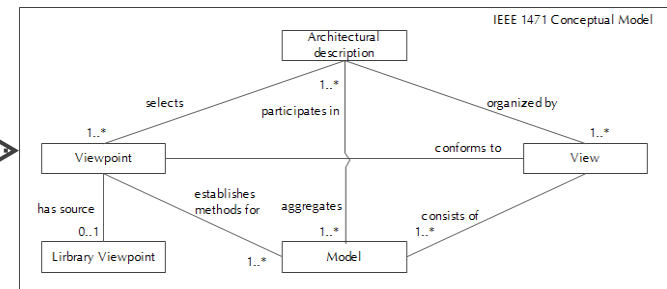
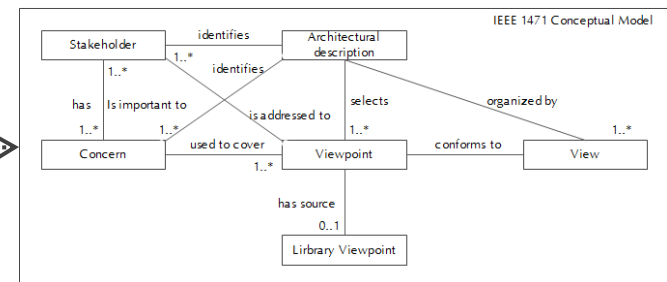
아키텍처 요소 타입 / 획득방법		자체개발	오픈소스	솔루션 벤더
비즈니스 컴포넌트		가입설계, 계약 컴포넌트	OMG Party컴포넌트	
프레임워크		자연처리 프레임워크, 예외처리	Struts, myBATIS	T사 프레임워크
서비스 서버		Mini-EAI서버, 상품 팩토리	Artifactory, SVN	T사의 암호화서버, IM/AM서버
플랫폼 소프트웨어	운영체제		Linux	HP-UX, Sun Solaris
	개발 언어		Physon, Groovy	Java, C, C++
	실행 플랫폼		JBoss, Spring, JVM	WebLogic, PaaS
	프로세스 플랫폼	EBOWorkflow	OSWorkflow, uEngine	BPM
	비즈니스 플랫폼		Compiere	SAP, Oracle ERP
관리용 소프트웨어 (제외)		컴포넌트 관리, 배포관리		Tivoli, OpenView

3. IEEE 1471 Close-up

- ✓ IEEE 1471 개념 모델은 추상화 수준이 높은 모델로 한 번에 이해하기 어렵습니다.
- ✓ 따라서, 각 부분 별로 확대하여 부분별로 이해한 후 전체를 보는 것이 편리합니다.
- ✓ IEEE 1471 개념 모델을 네 개의 부분으로 나누어서 이해하고자 합니다.

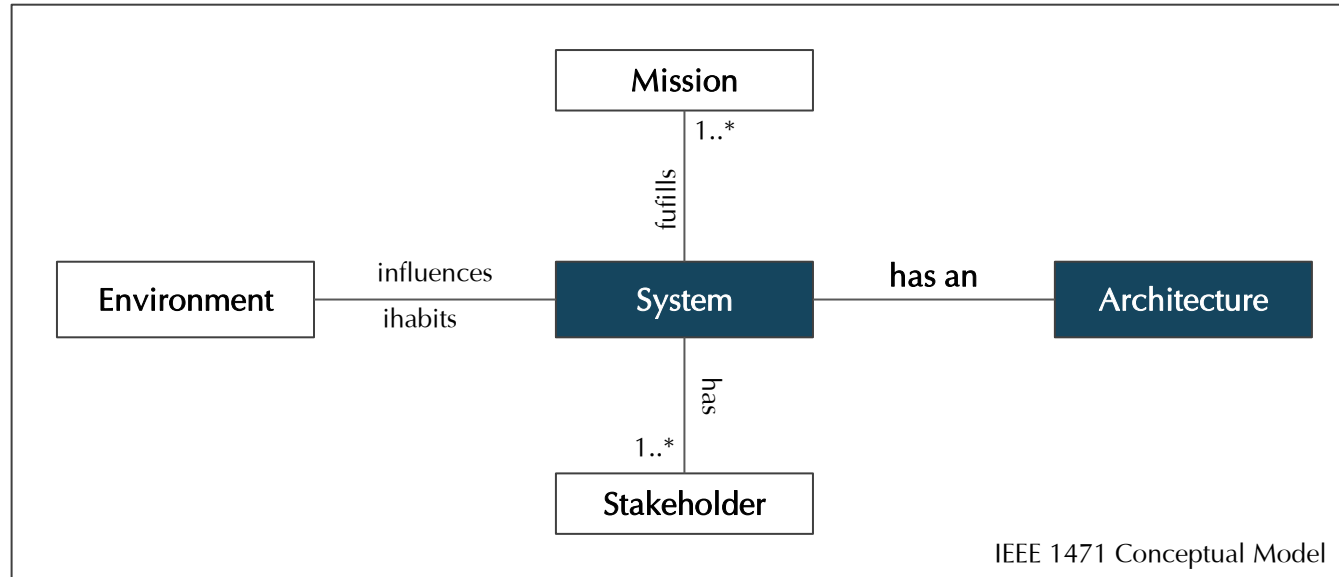


[IEEE 1471 개념 모델 Close-up]



3. IEEE 1471 Close-up – System has an architecture.

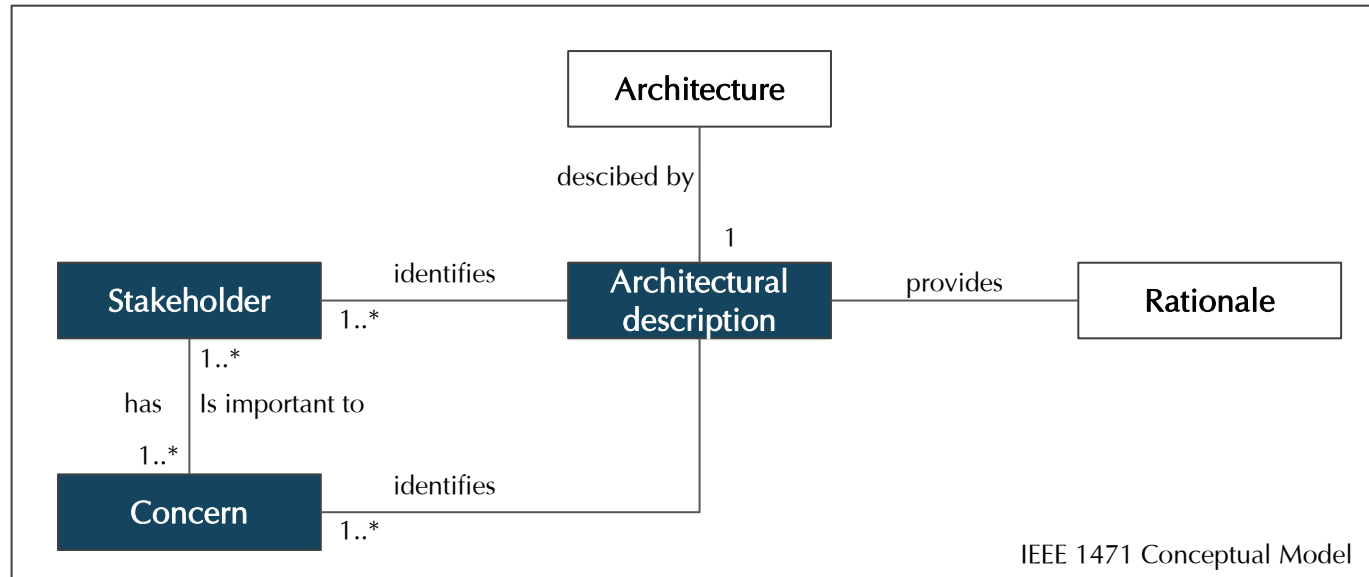
- ✓ 시스템은 [하나의] 아키텍처를 가진다.
- ✓ 시스템은 어떤 환경 속에서 미션을 수행한다. 이 환경은 언제나 제약조건을 가지고 있다.
- ✓ 시스템을 둘러싸고 있는 많은 이해관계자가 존재한다.
- ✓ **시스템은 아키텍처 [하나]를 가진다.**



[IEEE 1471 Close-up : System has an Architecture]

3. IEEE 1471 Close-up – Architectural Description

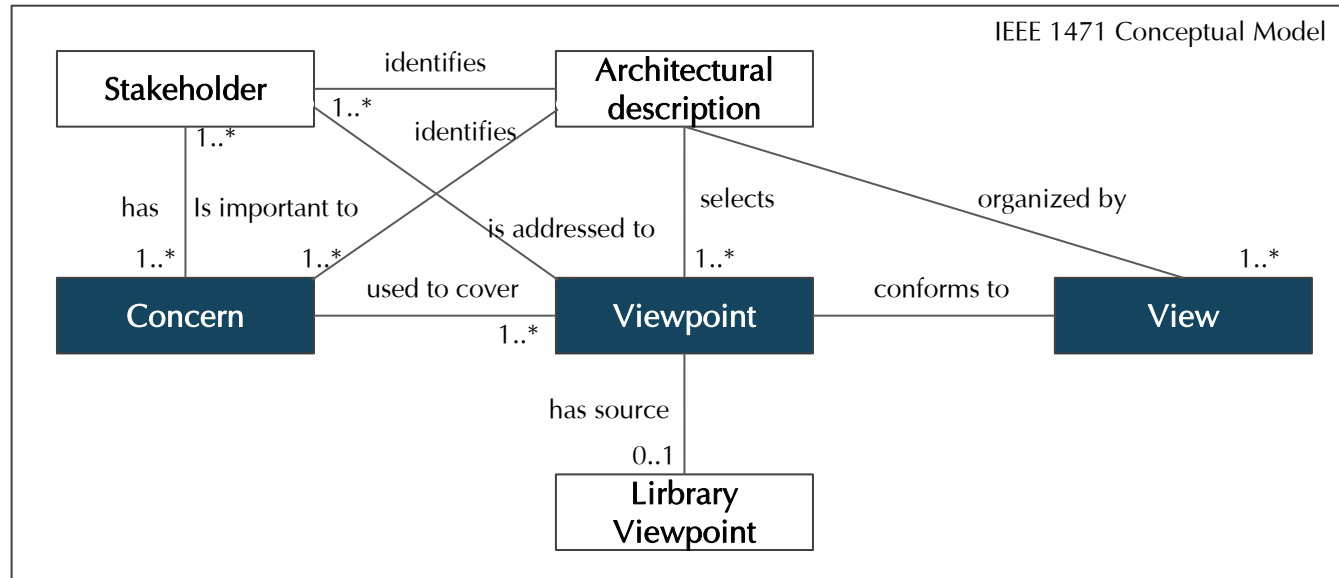
- ✓ 아키텍처는 아키텍트의 머릿속에 존재하는 구상(conception)으로 아키텍처 서술로 표현합니다.
- ✓ 아키텍처 서술은 논리적인 근거 또는 타당성(rationale)을 제공해야 합니다.
- ✓ 아키텍처 서술에서 이해관계자와 그들의 관심사(concern)를 식별하고 명세합니다.
- ✓ 이해관계자는 여러 관심사를 가지고 있으며, 관심사는 아키텍처 설계를 통해서 해결하여야 합니다.



[IEEE 1471 Close-up : AD identifies many Concerns]

3. IEEE 1471 Close-up – Viewpoint and concerns.

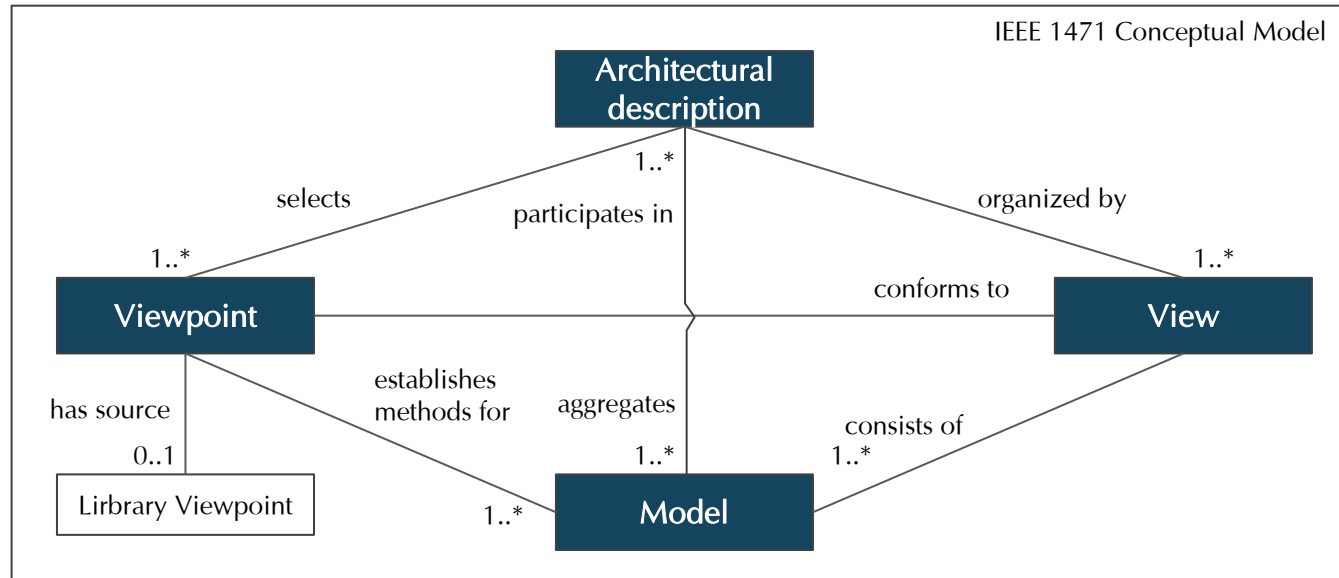
- ✓ AD는 여러 개의 Viewpoint를 가지고, Viewpoint는 관심사를 표현하는데 사용합니다.
- ✓ Viewpoint는 유사한 관심사의 묶음 단위로 설정됩니다.
- ✓ View는 Viewpoint 명세를 준수합니다.
- ✓ 조직에서 미리 정의한 표준 Library Viewpoint로부터 프로젝트를 위한 Viewpoint를 가져올 수 있습니다.



[IEEE 1471 Close-up : Viewpoint is used to cover the Concerns]

3. IEEE 1471 Close-up – View and Model (1/2)

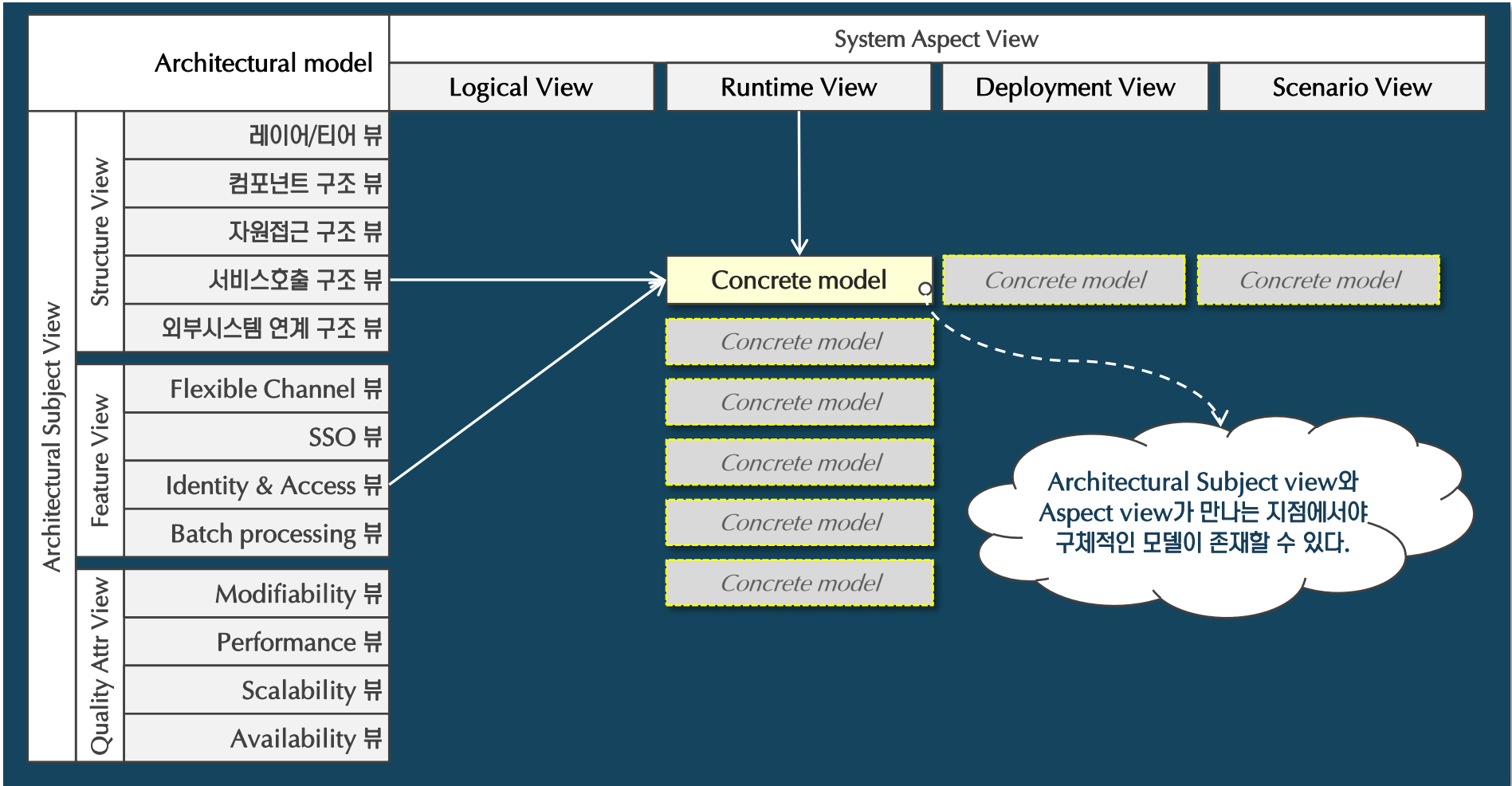
- ✓ AD에는 여러 Model이 필요합니다.
- ✓ Model과 View는 n:n 관계를 가집니다.
- ✓ View와 Model 관계 이해는 IEEE 1471 개념 모델을 이해하는 데 매우 중요합니다.
- ✓ 여기서 Model 은 어떤 Model을 의미할까요 ?



[IEEE 1471 Close-up : Viewpoint is used to cover the Concerns]

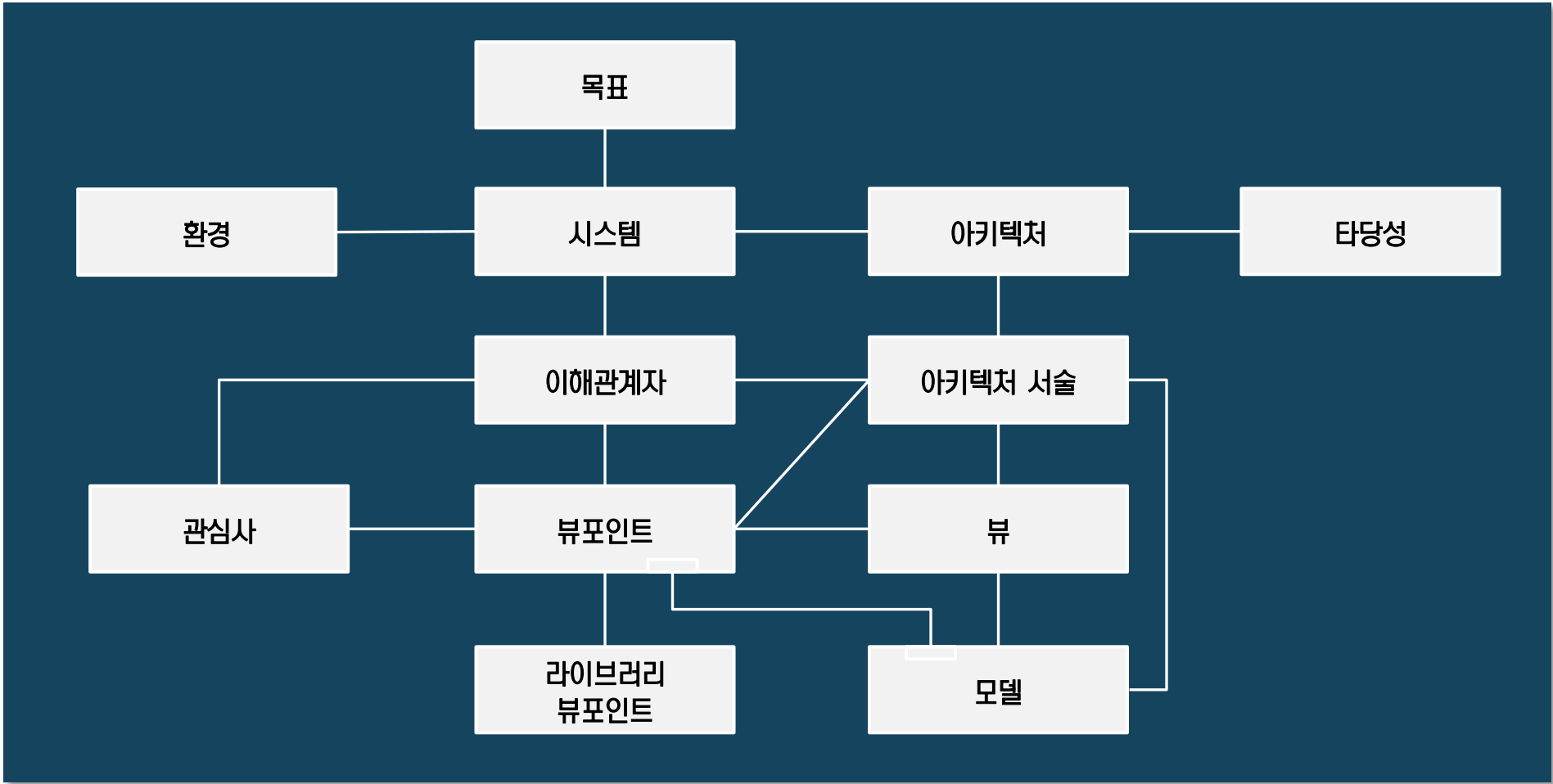
3. IEEE 1471 Close-up – View and Model (2/2)

- ✓ View와 View 가 만나는 지점에서 구체적인 Model이 존재합니다.
- ✓ 아래 도표의 의미에 대해 토론합니다.



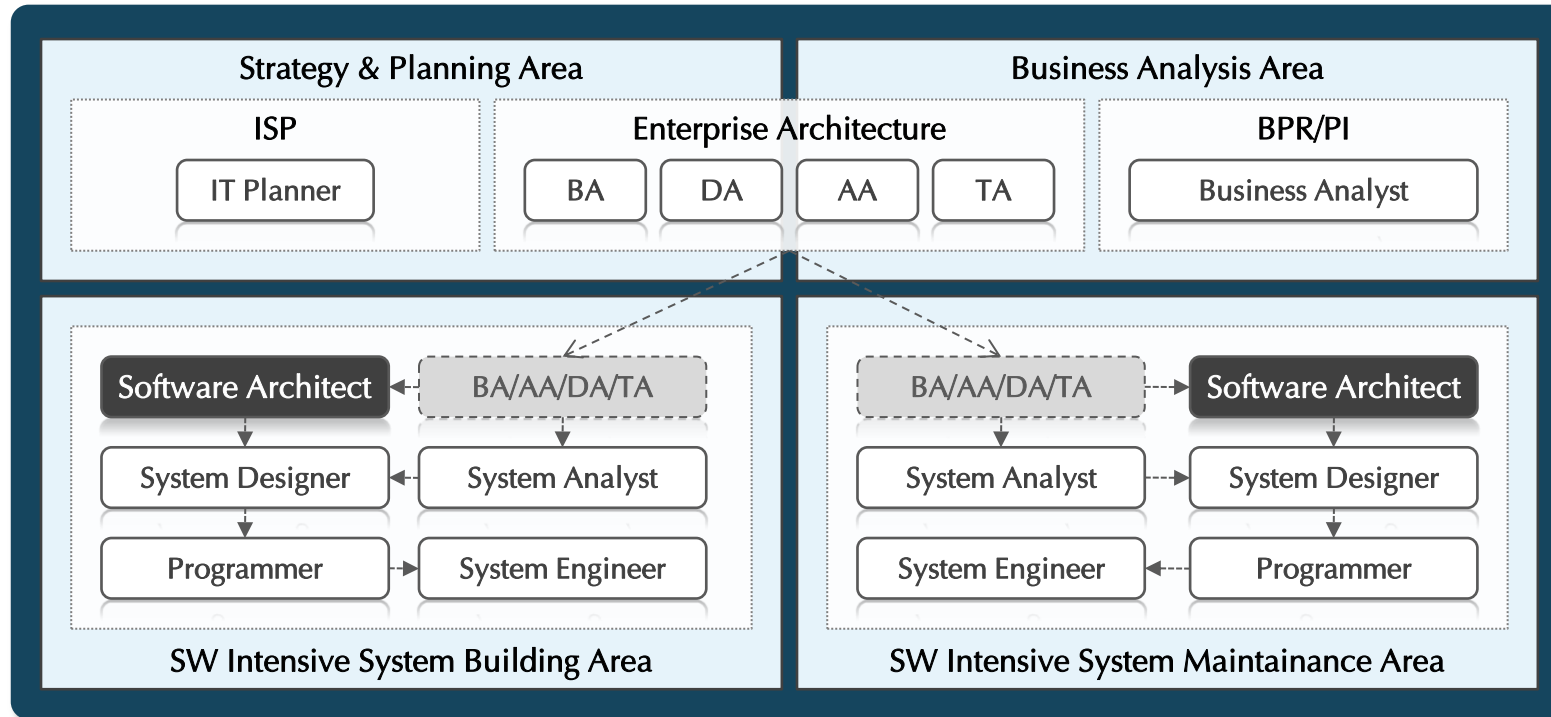
4. IEEE 1471 Review

- ✓ 아키텍처 서술을 위한 개념 모델의 의미를 되새겨 봅니다.
- ✓ 이 개념 모델을 확장하여 아키텍처 설계를 위한 구체적인 모델로의 확장에 대해 토론합니다.



5. SA Summary

- ✓ AA, DA, TA, BA의 정확한 정의에 대한 정확하게 이해합니다.
- ✓ IEEE 1471 기반으로 Software Architecture를 정확하게 이해합니다.
- ✓ 이들 간의 차이에 대해 토론합니다.
- ✓ SA = [S]oftware intensive sytem's [A]rchitecture



6. 토의

- ✓ 질의 응답
- ✓ 토론

감사합니다...

- ❖ 넥스트리컨설팅(주)
- ❖ CEO 송태국 / 대표 컨설턴트
- ❖ tsong@nextree.co.kr