Software Architecture: Definition

Seonah Lee Gyeongsang National University

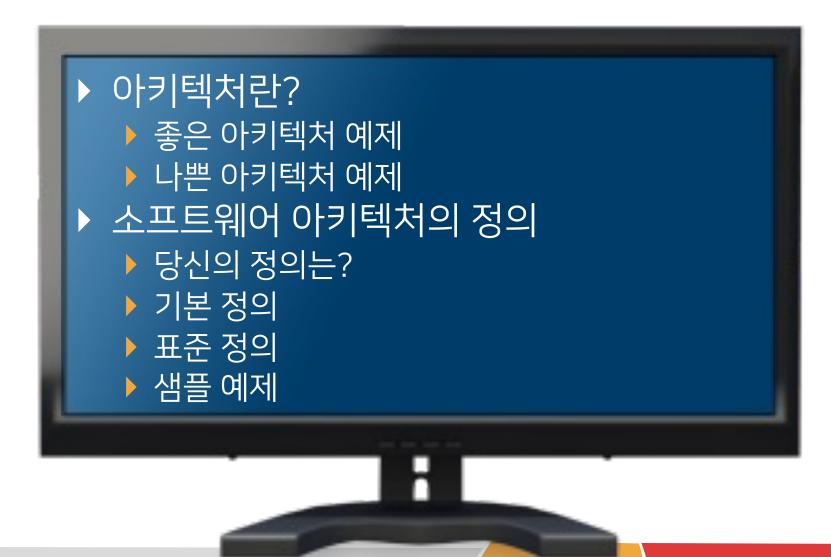
000

목 차

000

- ▶ 소프트웨어 아키텍처 정의
 - ▶ 표준에 기반한 정의
- 소프트웨어 아키텍처 역할
- ▶ 소프트웨어 아키텍처 뷰 관점
- ▶ 아키텍처 뷰 예제
 - ▶ 소프트웨어 아키텍처 뷰 예제 1
 - ▶ 소프트웨어 아키텍처 뷰 예제 2
- ▶ 정리 및 실질적인 의미 논의

소프트웨어 아키텍처의 정의





Architecture?



- ▶ 아키텍처 구축이란?
 - Support the whole building to stand
 - Abstract entity that exists without building construction material such as metal, mud or bricks
 - Difficult and dangerous to change architecture





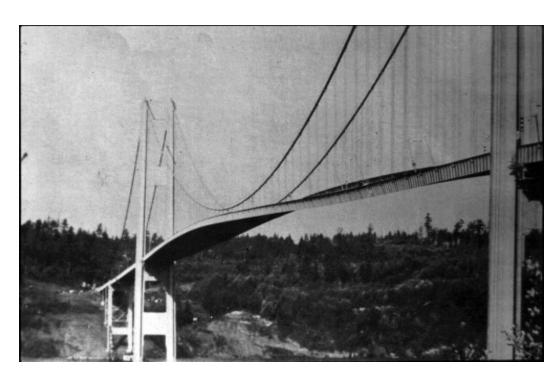
Good Architecture: Example





- ▶ Sukgoolam (석굴암)
 - Built in AD 751
 - Lasted for more than 1200 years under heavy pressure from the above and the sides.

Bad Architecture: Example





Opened: July 1, 1940

Collapsed: Nov. 7, 1940



The old Tacoma Narrows Bridge twisted and vibrated violently under 64 km/h winds on the day of the collapse

0

소프트웨어 아키텍처에 대한 당신의 정의는**?**



- ▶ 골격, 기본, 틀.
- ▶ **J2EE**를 사용한다.
- ▶ 클라이언트 프로그램은 데이타베이스에 바로 접속할 수 없다.
- ▶ 각각의 서버는 다수의 사용자 요청을 동시에 처리해야 한다.

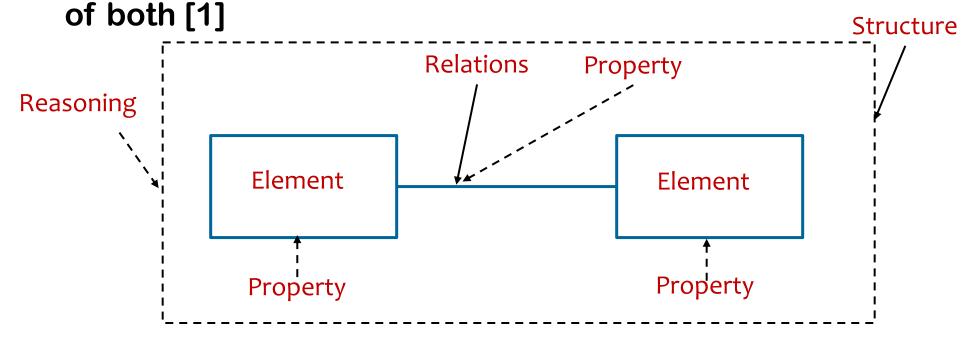
▶ 사용자 요청 처리 정보를 기업의 데이터 베이스에 넣어 서로 공유한다.





The software architecture of a system

the set of structures needed to reason about the system, which comprise software elements, relations among them, and properties





000

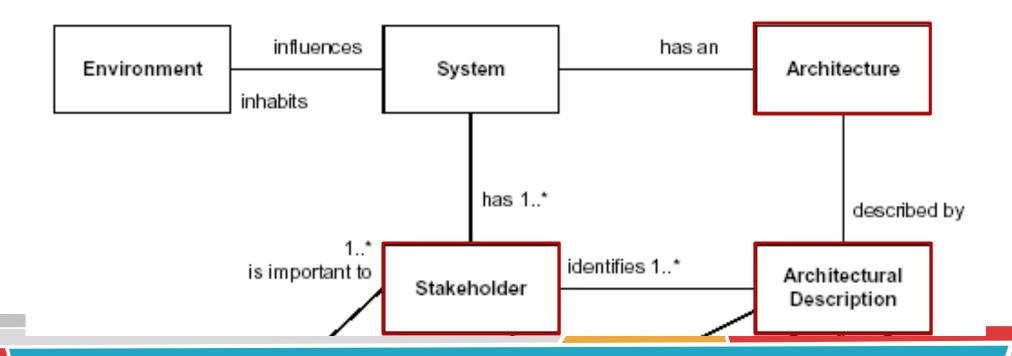
- 소프트웨어 아키텍처는 소프트웨어 시스템들의 큰 규모의 구조와 실행에 관한 연구이다 [Shaw 89].
- 소프트웨어 "아키텍처란 다음과 같은 것에 관한 중요한 결정들의 집합이다. [Jacobson 99][Kruchten 01]
 - ▶ 1)시스템의 구조를 나타내는 구조적 구성요소와 그들을 결합시키는 인터페이스,
 - ▶ 2)그들의 협동을 통해 나타나는 구조적 구성요소와 행위적 구성 요소들의 결합을 점진적으로 서브시스템에 맵핑을 하는 것,
 - ▶ 3)구조를 이끌어 나가는 아키텍처 스타일의 선택"



Architecture: Standard Definition 1

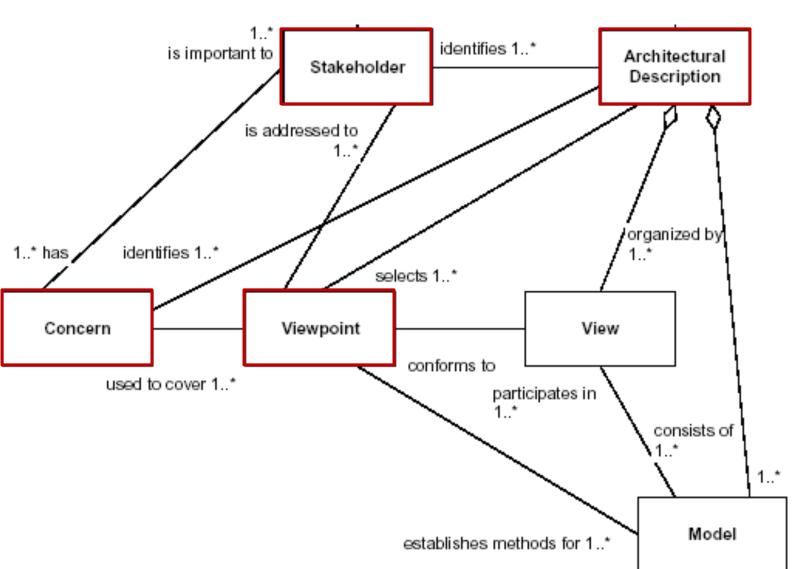


- ▶ IEEE 1471 아키텍처 정의
 - ▶ "시스템의 근본적인 조직형태로써, 그것은 구성컴포넌트들과, 그들 서로와 환경에 대한 관계 그리고 그 설계와 진화를 관장하는 원칙들에 담겨있다"



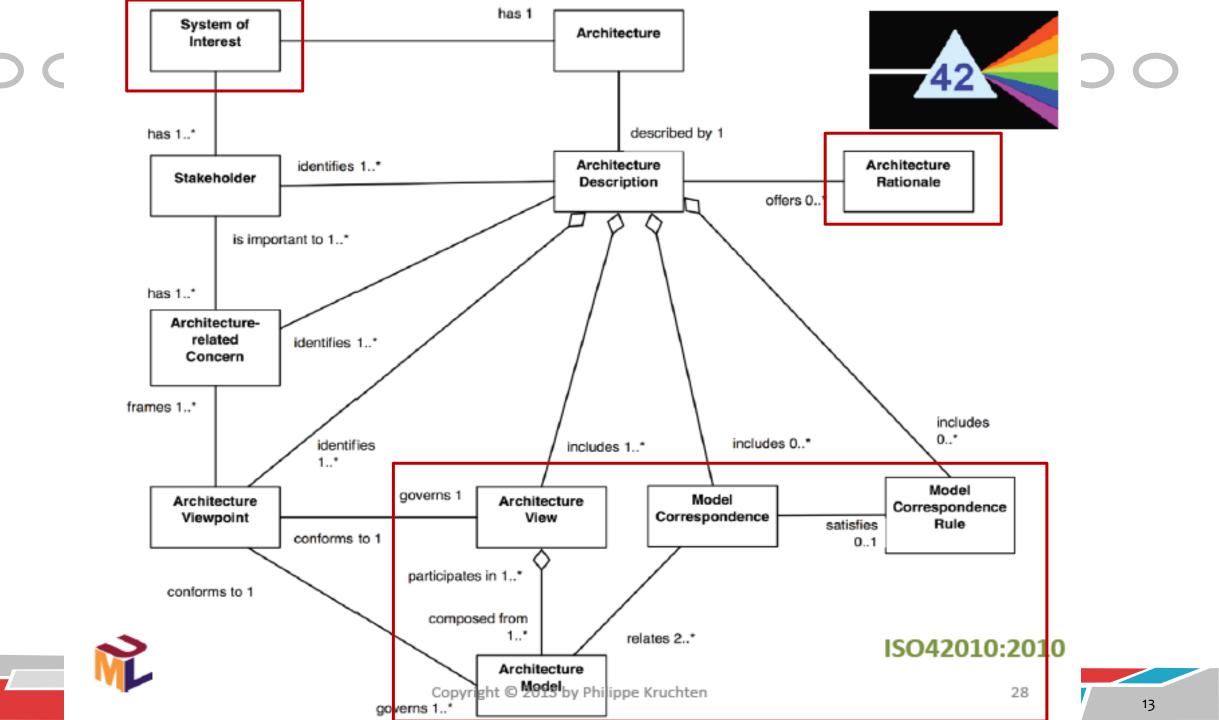


Architecture: Standard Definition 1



Architecture: Standard Definition 2

- ▶ ISO/IEC 42010, Systems and software engineering Architecture description, 2011
 - ▶ 시스템의 근본적인 개념 혹은 특성으로, 환경에서의 시스템에 대해, 시스템의 요소 및 그 관계에 대해, 또한 그 설계와 진화의 원칙에 대한 것이다.
 - The fundamental concepts or properties of a system in its environment embodied in its elements, their relationships, and in the principles of its design and evolution
 - ▶ Environment: 시스템에 영향을 주는 모든 것
 - ▶ Principles: 시스템의 설계 및 진화의 원칙





Architecture: Definition



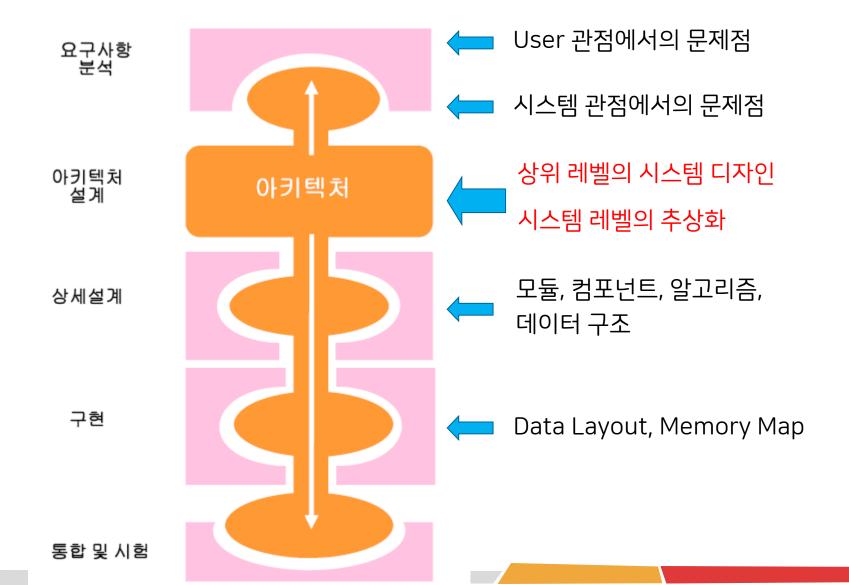
- Philippe Kruchten, 2013
 - architecture = { elements, form, rationale } *
 Perry & Wolf 1992
 - A skeleton, not the skin
 - More than structure
 - Embodies or addresses many "ilities"
 - Executable, therefore verifiable

소프트웨어 아키텍처의 역할

▶ 문제에서 솔루션으로의 전환 ▶ 품질의 요구사항 반영 ▶ 위험 감소 ▶ 변경의 손쉽게 포용







000

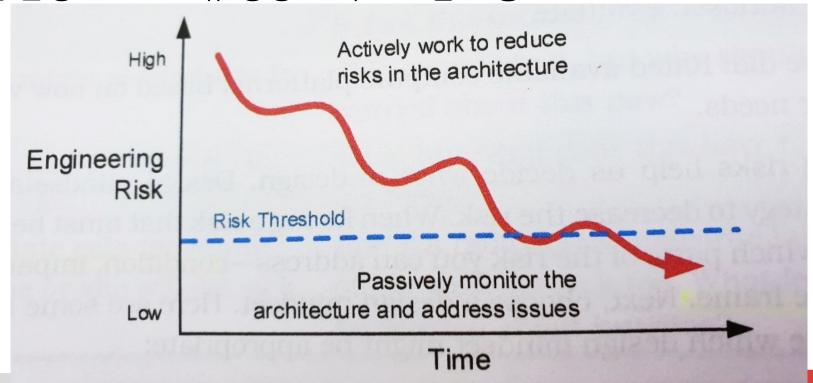
- Software architecture determines quality of the system
 - ▶ 소프트웨어 아키텍처는 소프트웨어 시스템의 성능과 품질을 결정
 - ▶ 두 개의 시스템이 서로 다른 구조를 가지고 있는 것은 품질 속성이 다르기 때문임
 - ▶ 포탈의 communication channel
 - ▶ 우주 왕복선의 communication channel

Usability Portability Reusability

Performance Maintainability

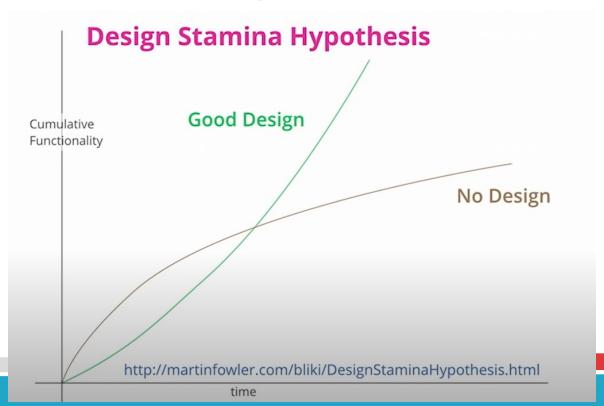


- Software architecture helps you avoid costly mistakes
 - 아키텍처에서 적극적으로 위험을 감소의 노력을 기울이면, 시간이 갈수록 설계는 안정화되고 오류 정정 노력에 초점 가능



000

- Software architecture enables agility
 - ▶ 소프트웨어는 변경에 유연하게 대응할 수 있어야 함
 - ▶ 소프트웨어 아키텍처는 변경을 가능하게 하는 구조를 제공

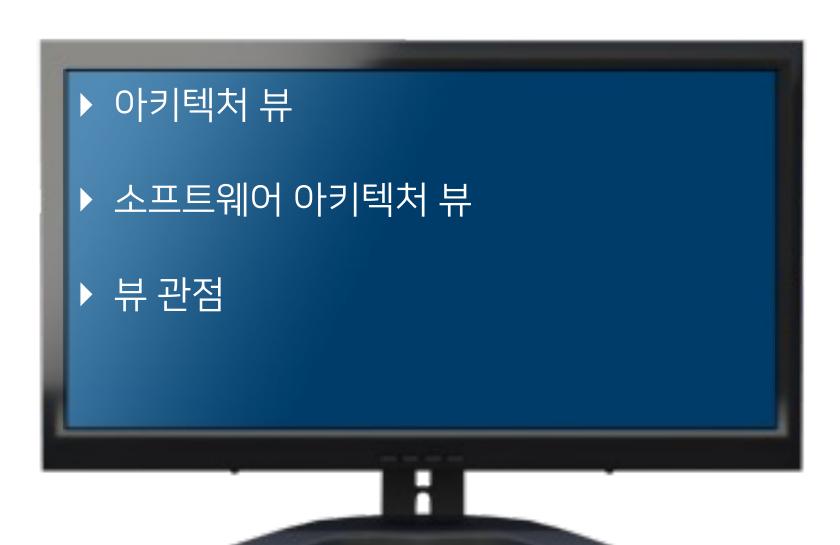


Design Stamina Hypothesis

Cumulative Functionality

소프트웨어의 기능의 고도화에 대한 그래프입니다

소프트웨어 아키텍처 뷰관점

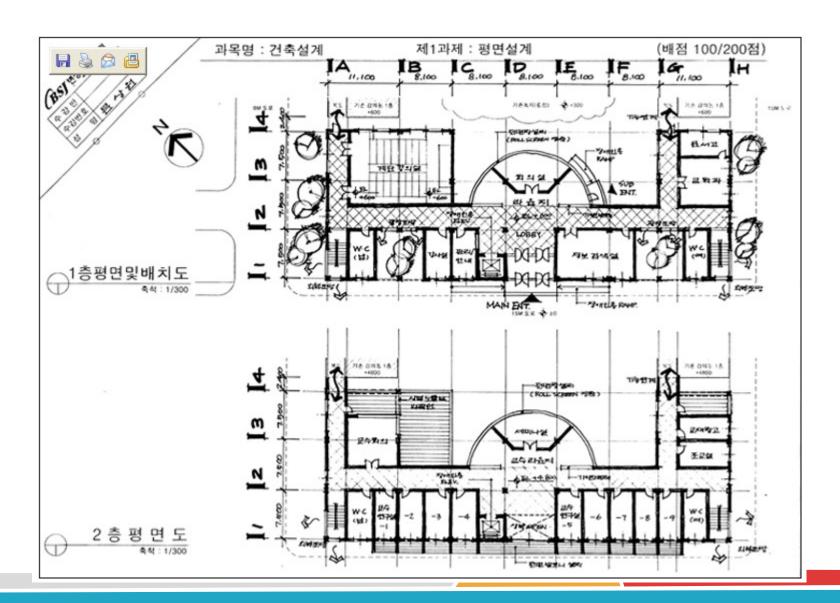




Architecture View



▶ 평면도

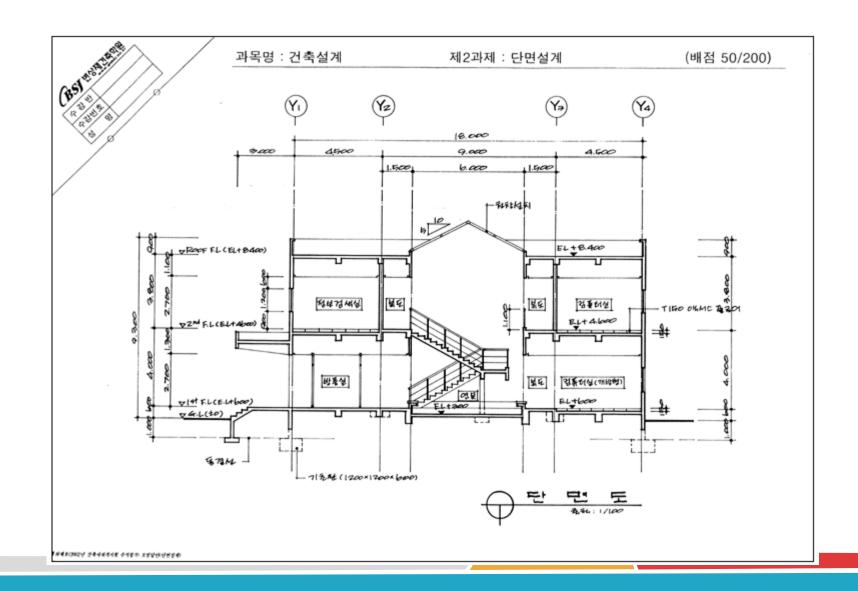




Architecture View



▶ 단면도

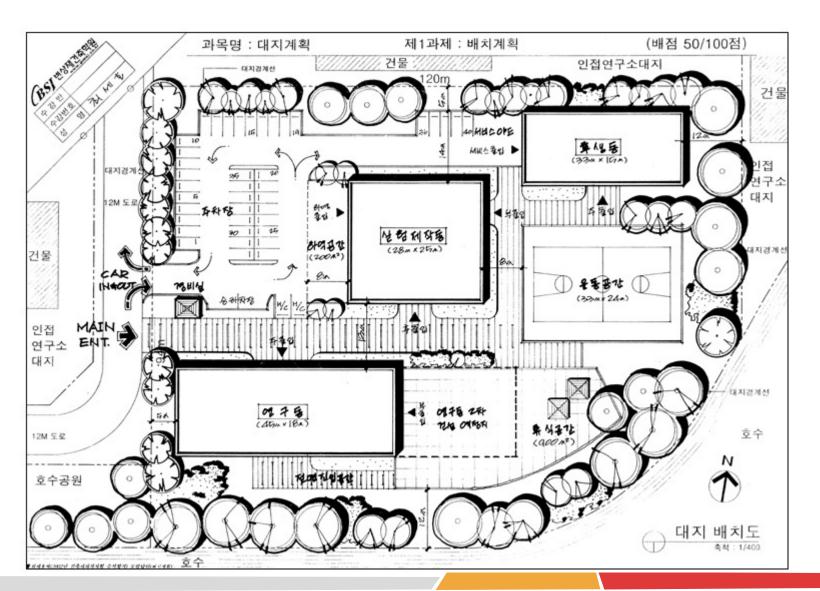




Architecture View



▶ 배치도





소프트웨어를 본적이 있는가? 소프트웨어를 만져본 적이 있는가?



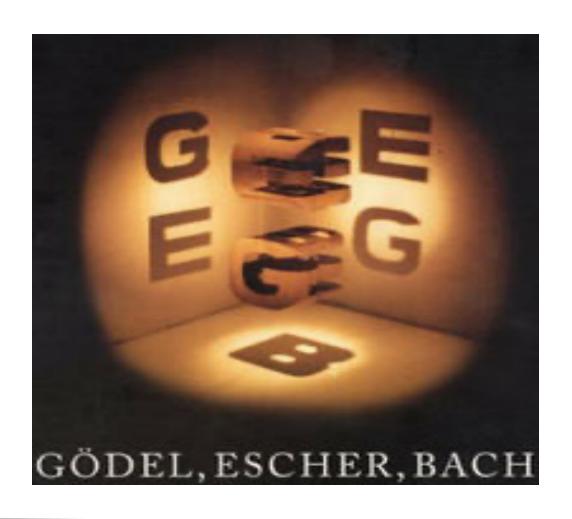
소프트웨어 이해관계자의 소프트웨어 인지능력 차이 발생





소프트웨어 가시화 필요성 증대

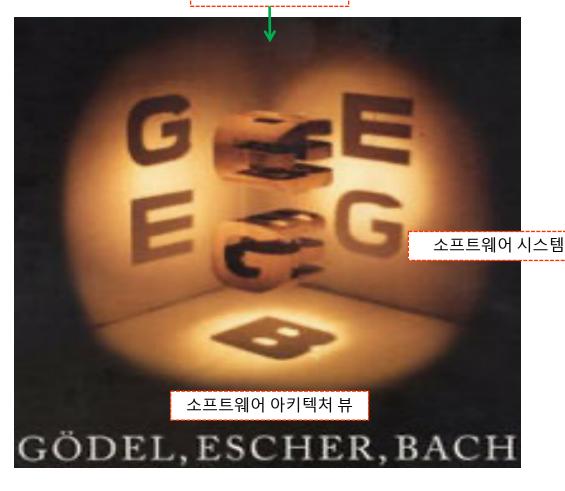




- 아키텍처는 여러 개의 뷰 관점이 필요하다
- Architecture needs multiple viewpoints



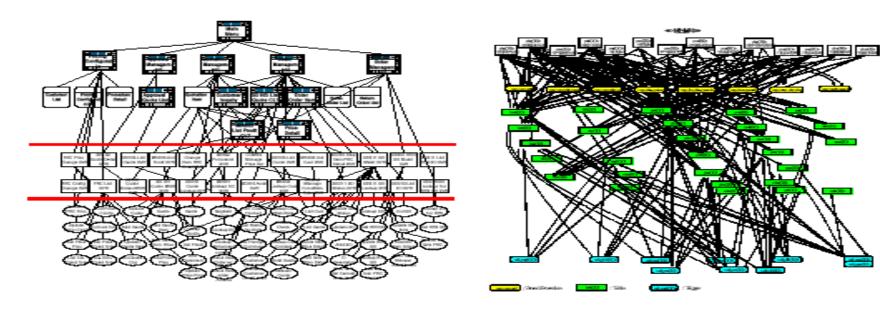
뷰 포인트



- ▶ 뷰 포인트란?
 - ▶ View를 구조화 하는 관점
- ▶ 뷰(View) 란?
 - ▶ 시스템을 이루는 SW요소들의 집합 과 SW요소들의 연관 관계를 추상적 으로 표현한 것

000

- ▶ 소프트웨어 아키텍처를 이해시키려면?
 - ▶ 추상화, 간결한 표현 필요

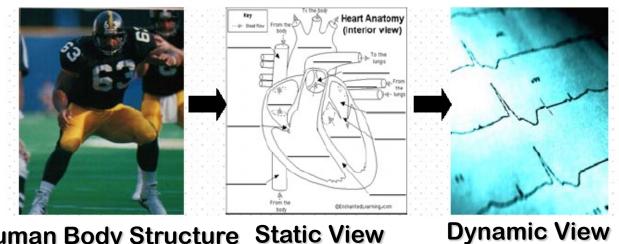


Project A+ (MVC Layered Architecture) Project A (No Architecture)

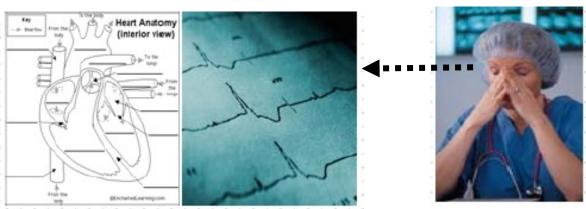


Structures 와 View의 관계





Human Body Structure Static View



심장 전문의

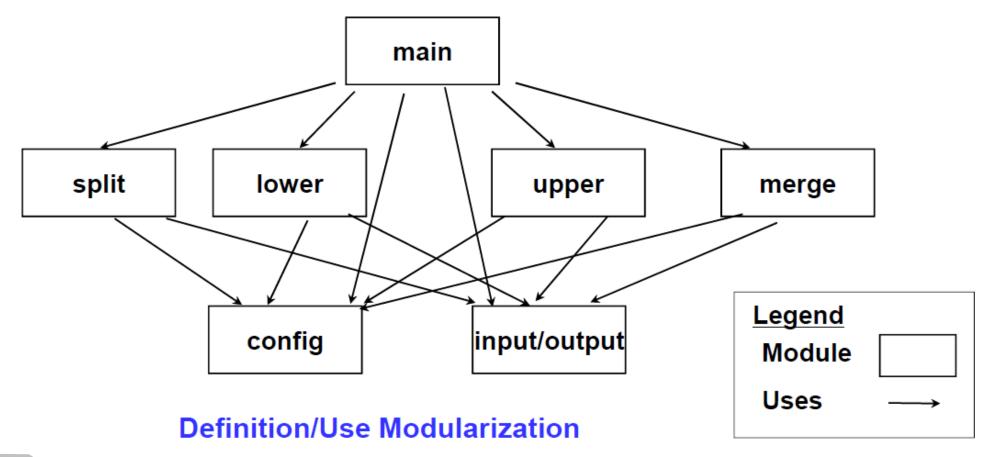
Static View

Dynamic View

외과 전문의

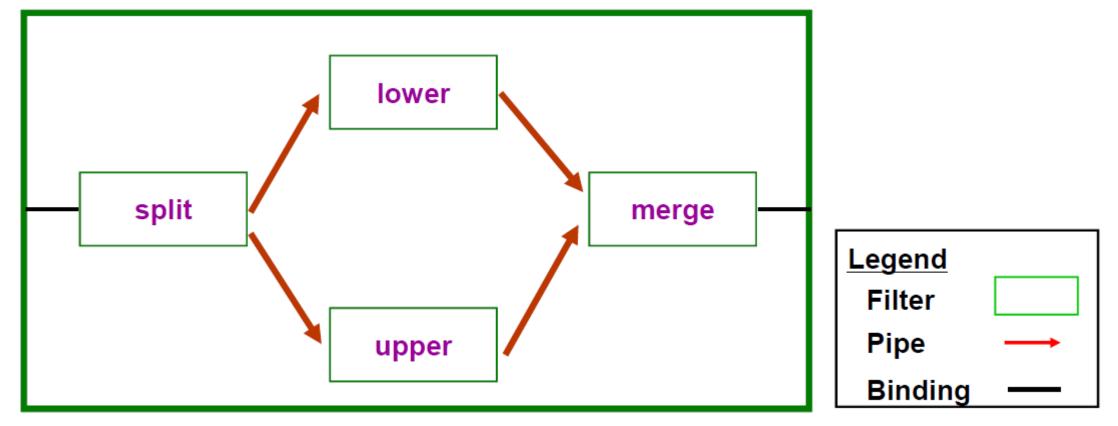
000

Module View



000

▶ Runtime View (혹은 Component and Connector View)



00

실제 예제: TestGen System

00

▶ TestGen 시스템은? Context Diagram 정적 관점 뷰 동적 관점 뷰 ▶ 구현 관점 뷰

What is the TestGen System?

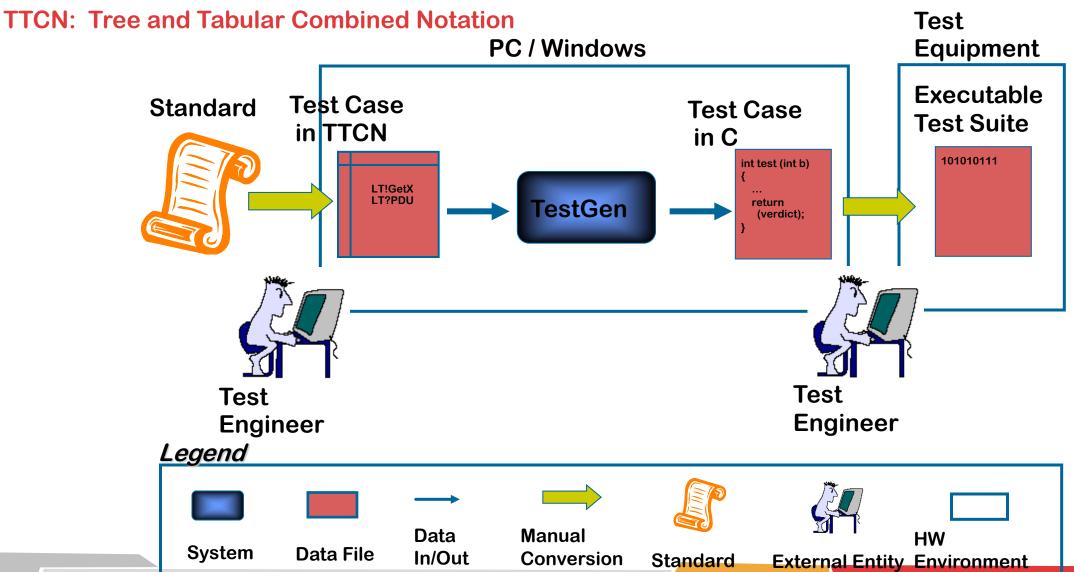
- Used to specify tests for many kinds of IT / Telecom products
 - Wireless LANs, cordless phones, broadband technologies, CORBAbased platforms
- Built for TTA (Telecommunication Technology Association)
 - **▶ IT standards organization**
 - develops, tests and certifies standards





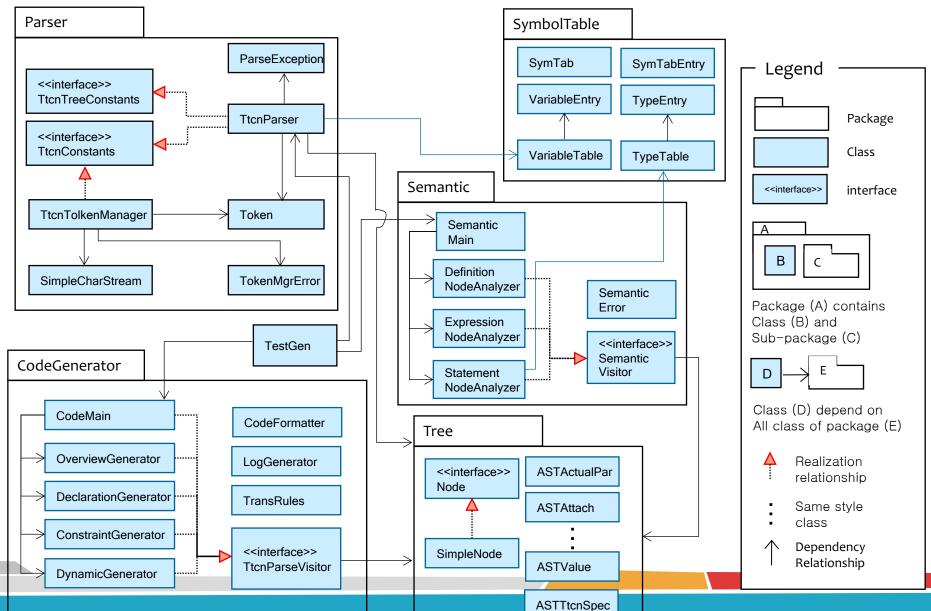
Context Diagram





Module View

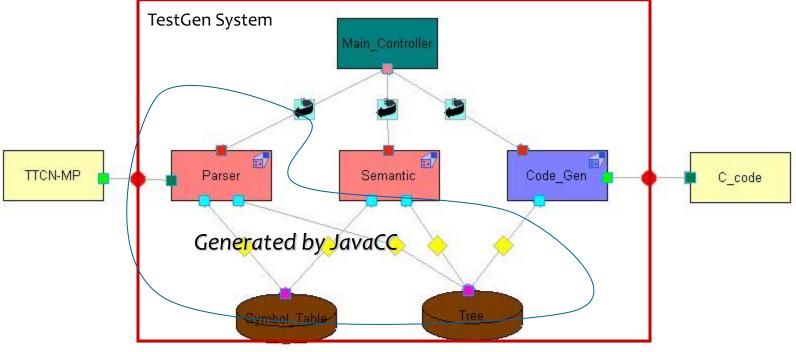


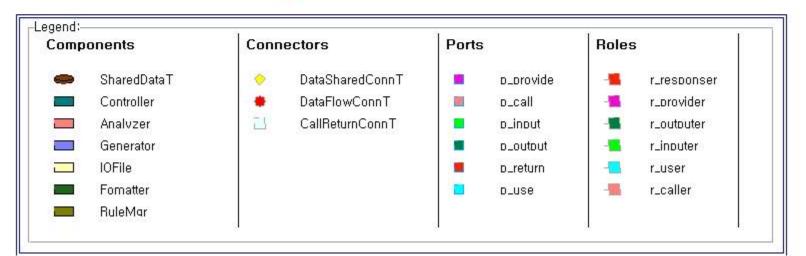




Runtime(C&C) View

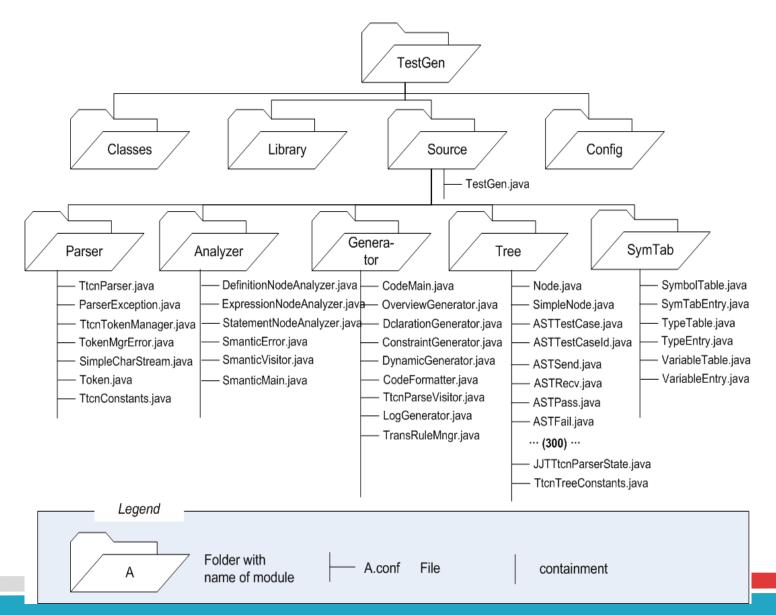






Allocation View





00

실제 예제: NavClus System

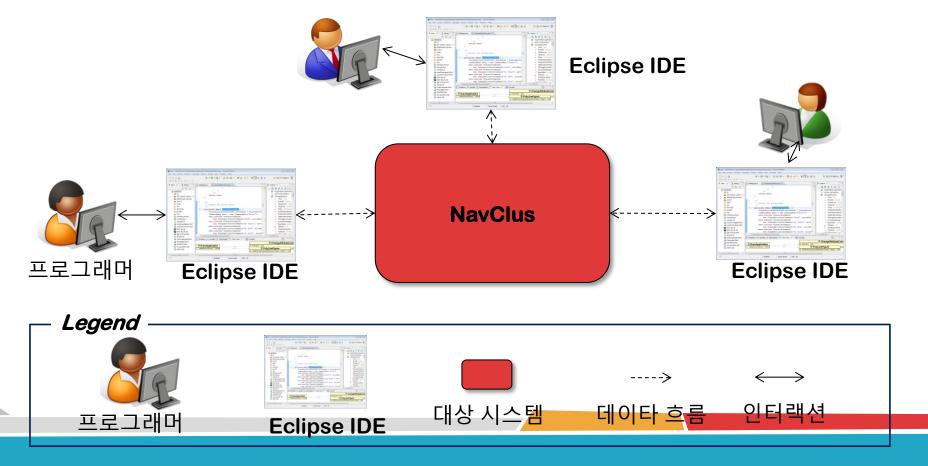
00

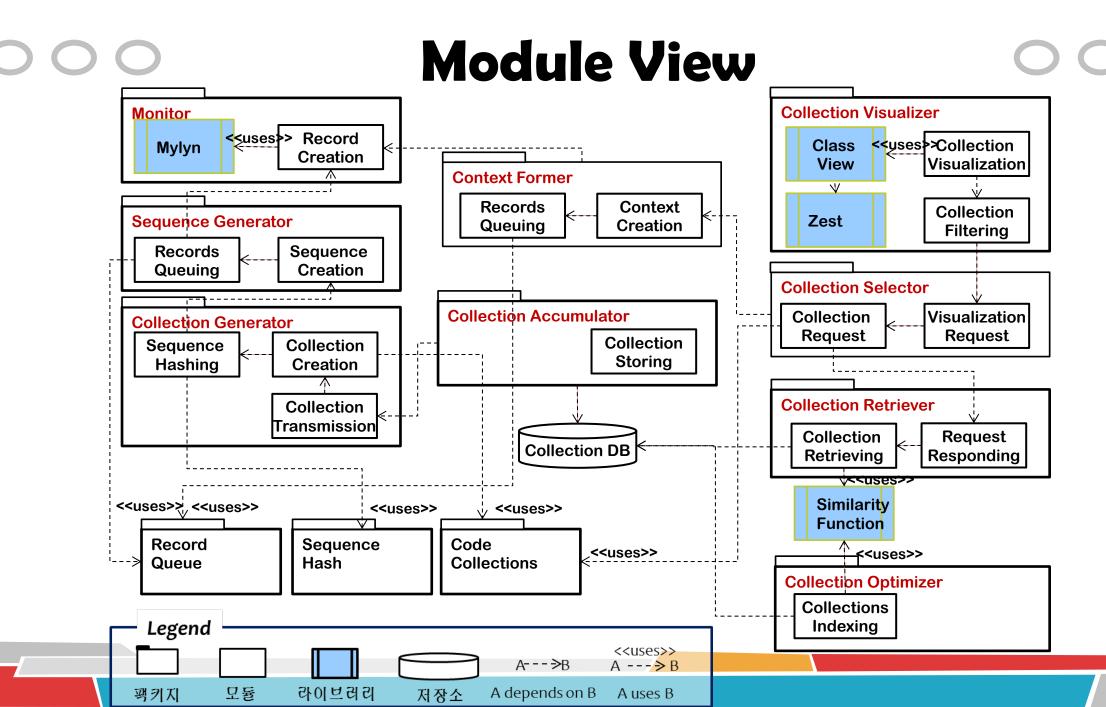
▶ NavClus 시스템은? Context Diagram 정적 관점 뷰 동적 관점 뷰 ▶ 구현 관점 뷰

O O What is the NavClus System?

000

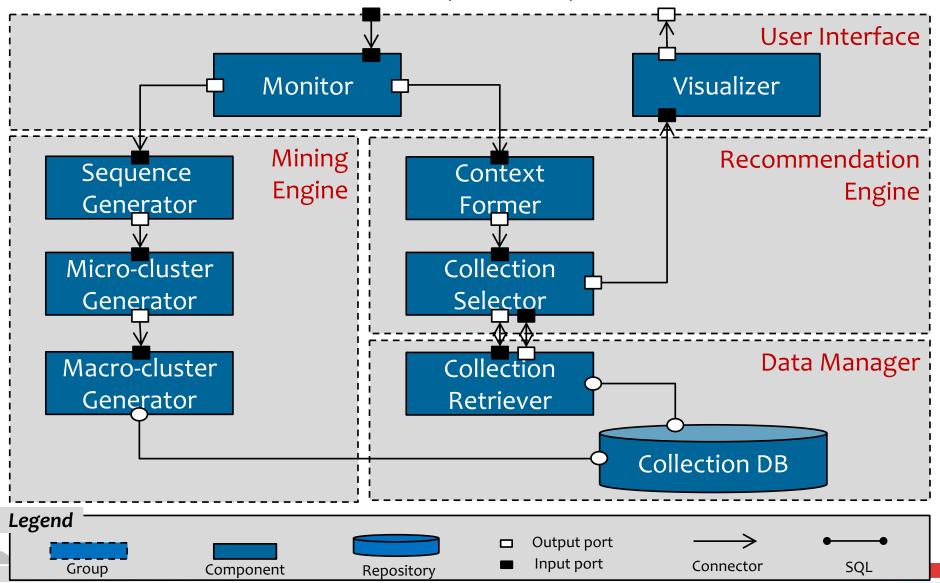
▶ Eclipse IDE를 통해 여러 명의 개발자가 자신들의 인터랙션 히스토리를 통해 작업과 관련한 코드를 공유할 수 있도록 하는 코드 추천 시스템





Runtime(C&C) View

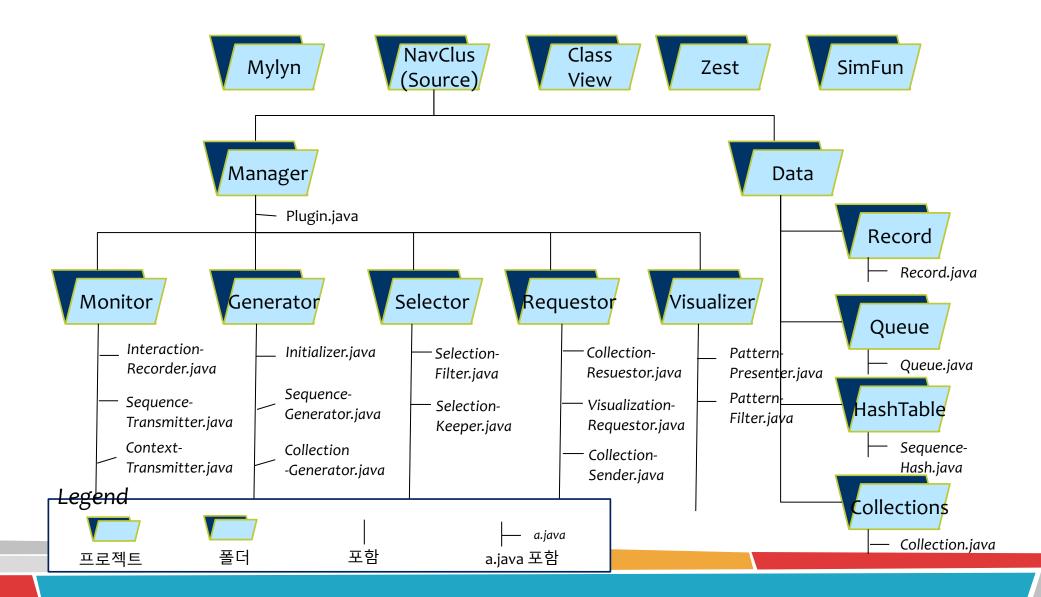






Allocation View





OOO Summary: Software Architecture? OOO

- ▶ 소프트웨어 시스템의 구성 요소간의 연결을 보임
 - Shows the overall relation between components of a software system
- ▶ 소프트웨어 시스템의 성능과 품질을 결정함
 - Determines performance and quality of the system
- ▶ 시스템이 존재하기 전, 전체 시스템을 이해하도록 함
 - Helps us understand the whole system "before" it comes into existence



Question?





Seonah Lee saleese@gmail.com