

04 도메인의 격리

모델의 각 요소(객체)를 하나하나 구분하지 않고 **하나의 시스템**으로 바라 볼 수 있도록 시스템에서 도메인과 관련이 적은 기능으로부터 도메인 객체를 분리

Domain-Driven Design, Modeling, Design

Layered Architecture

Pattern-Oriented Software Architecture Volume 1: A System of Patterns. pp. 31 ~ 51 참고

계층화의 핵심 원칙

계층화의 핵심 원칙은

한 계층의 모든 요소는 **오직 같은 계층에 존재하는 다른 요소** 혹은 계층상 “아래”에 위치한 요소에만 의존한다는 것이다.

위로 거슬러 올라가는 의사소통은 반드시 간접적인 메커니즘을 거쳐야 한다.

계층화의 가치

계층화의 가치는 각 계층에서 컴퓨터 프로그램의 **특정 측면만을 전문적으로 다룬다**는 데 있다. ...

거듭 말하지만 경험과 관례를 바탕으로 널리 받아들여지는 계층화가 어느 정도 정해졌다. ... 대다수의 성공적인 아키텍처에서는 아래의 네 가지 개념적 계층으로 나뉜다.

계층	설명
User Interface (or Presentation Layer)	사용자에게 정보를 보여주고 사용자의 명령을 해석하는 일을 책임진다. 간혹 사람이 아닌 다른 컴퓨터 시스템이 외부 행위자가 되기도 한다.
Application Layer	소프트웨어가 수행할 작업을 정의하고 표현력 있는 도메인 객체가 문제를 해결하게 한다. 이 계층에서 책임지는 작업은 업무상 중요 하거나 다른 시스템의 응용 계층과 상호작용하는 데 필요한 것이다. 이 계층은 얇게 유지된다. 여기에는 업무 규칙이나 지식이 포함되지 않으며 , 오직 작업을 조정하고 아래에 위치한 계층에 포함된 도메인 객체의 협력자에게 작업을 위임한다. 응용 계층에서는 업무 상황을 반영하는 상태는 없지만 사용자나 프로그램의 작업에 대한 진행상황을 반영하는 상태를 가질 수는 있다.
Domain Layer (or Model Layer)	업무 개념과 업무 상황에 관한 정보, 업무 규칙 을 표현하는 일을 책임진다. 이 계층에서는 업무 상황을 반영하는 상태를 제어하고 사용하며 , 그와 같은 상태 저장과 관련된 기술적인 세부사항은 인프라스트럭처에 위임 한다. 이 계층은 업무용 소프트웨어의 핵심 이다.
Infrastructure Layer	상위 계층을 지원하는 일반화된 기술적 기능 을 제공한다. 이러한 기능에는 애플리케이션에 대한 메시지 전송, 도메인 영속화, UI에 위젯을 그리는 것 등이 있다. 또한 인프라스트럭처 계층은 아키텍처 프레임워크를 통해 네 가지 계층에 대한 상호작용 패턴을 지원 할 수도 있다.



Model-Driven Design을 가능케 하는 것은 결정적으로 **도메인 계층을 분리**하는 데 있다.

계층 간 관계 설정

아키텍처 프레임워크

도메인 계층은 모델이 살아가는 곳

The Smart UI "Anti-Pattern"

지능형 UI “안티 패턴”

다른 종류의 격리

From:
<http://wiki.ledyx.xyz/> - Ledyx WIKI

Permanent link:
http://wiki.ledyx.xyz/doku.php?id=doamin-driven_design:04_isolating_the_domain&rev=1691623661

Last update: 2023/08/09 23:27

