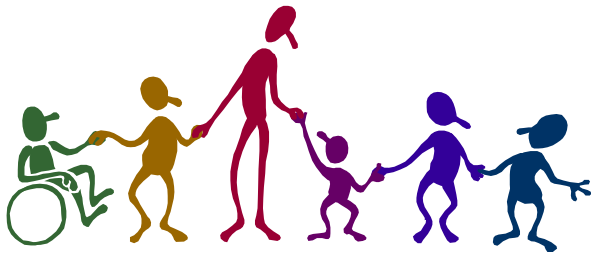
8장. 컴포넌트 다이어그램



조 은 숙
서일대학교
소프트웨어공학과
escho@seoil.ac.kr

목차

- 컴포넌트 정의
- 컴포넌트 다이어그램의 UML 표기법
- 컴포넌트와 클래스
- 컴포넌트 다이어그램의 간단한 예
- 컴포넌트 다이어그램 예제

© Eun Sook Cho

컨포넌트 정의

컴포넌트

- 시스템을 구성하는 임의의 물리적인 요소를 의미
- 물리적인 요소란 가상의 모델을 실제로 구현하여 나타내는 것을 의미
- 객체지향의 원리에 따라 업무 기능과 관련 데이터를 하나의 단위로 처리
  - **UML** 관점과 **CBD(Component Based Development)** 관점에서의 컴포넌트 의미와는 조금 다르다.

© Eun Sook Cho

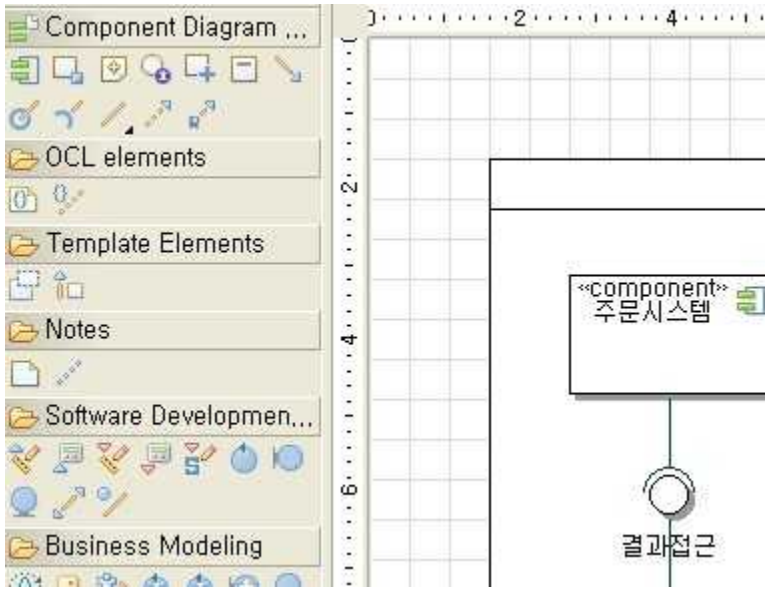
컨포넌트 정의

- 객체지향 원리에서 컴포넌트란
  - 인터페이스에 의해서 기능이 정의된, 독립적으로 개발·배포·조립이 가능한 시스템의 구성 단위로 정의
- 컴포넌트의 대표적 예
  - **J2EE** 플랫폼의 **JAR** 파일
  - 닷넷 플랫폼의 **DLL** 파일

© Eun Sook Cho

컴포넌트 정의

➔ 컴포넌트 구성요소



[그림 9-1] 컴포넌트 구성 요소

컴포넌트	논리적 요소들이 물리적으로 패키지화
인터페이스	컴포넌트가 실현하고자 하는 여러 오퍼레이션의 모임
의존관계	컴포넌트와 컴포넌트 간의 관계
지원관계	컴포넌트와 인터페이스 간의 관계

© Eun Sook Cho

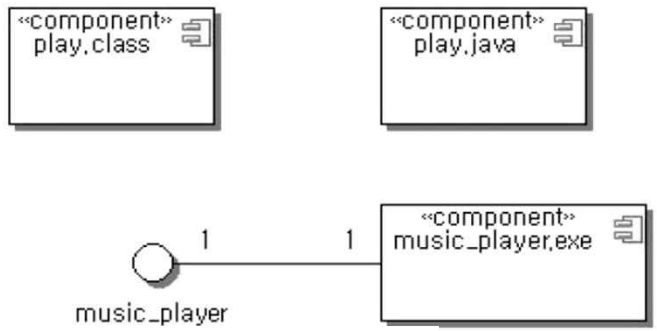
컨포넌트 정의

➲ UML 관점

- ❑ 음악 플레이어를 구성하는 클래스 파일 **(play.class)**와 서블릿 소스**(play.java)**, **music_player.exe** 등은 컴포넌트에 해당

➲ CBD 관점

- ❑ 인터페이스와 연결된 **music_player.exe**만이 컴포넌트에 해당
- ❑ 클래스 파일, 자바 소스 등은 컴포넌트에 해당하지 않음



[그림 9-2] 음악 플레이어를 구성하는 파일의 예

© Eun Sook Cho

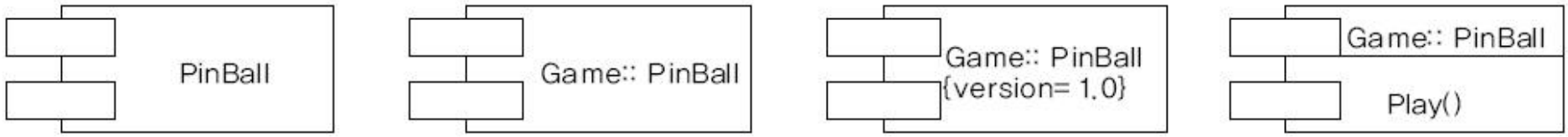
UML 표기법

컴포넌트 다이어그램

- 시스템을 구성하는 물리적인 컴포넌트와 그들 사이의 의존관계를 나타내는 다이어그램
- 컴포넌트, 인터페이스, 의존관계로 표현

컴포넌트

- 컴포넌트는 탭이 달린 직사각형으로 표현
- 모든 컴포넌트는 반드시 이름을 가지고 있어야 한다.
- 컴포넌트가 패키지에 포함되어 있다면 컴포넌트의 이름 앞에 패키지 이름을 붙일 수 있으며,
- 클래스처럼 컴포넌트에 꼬리표 값을 달아주거나 컴포넌트 내부의 오퍼레이션을 보여줄 수도 있다.



[그림 9-3] 컴포넌트의 표현 방법

UML 표기법

인터페이스

- 컴포넌트 인터페이스는 **2**가지 방식으로 표현
  - 컴포넌트와 인터페이스, 그리고 이를 연결하는 화살표 모양의 점선(의존관계)으로 나타낼 수 있다. **([그림 9-4] (a))**
  - **[그림 9-4] (b)**

(a) 인터페이스의 실체화

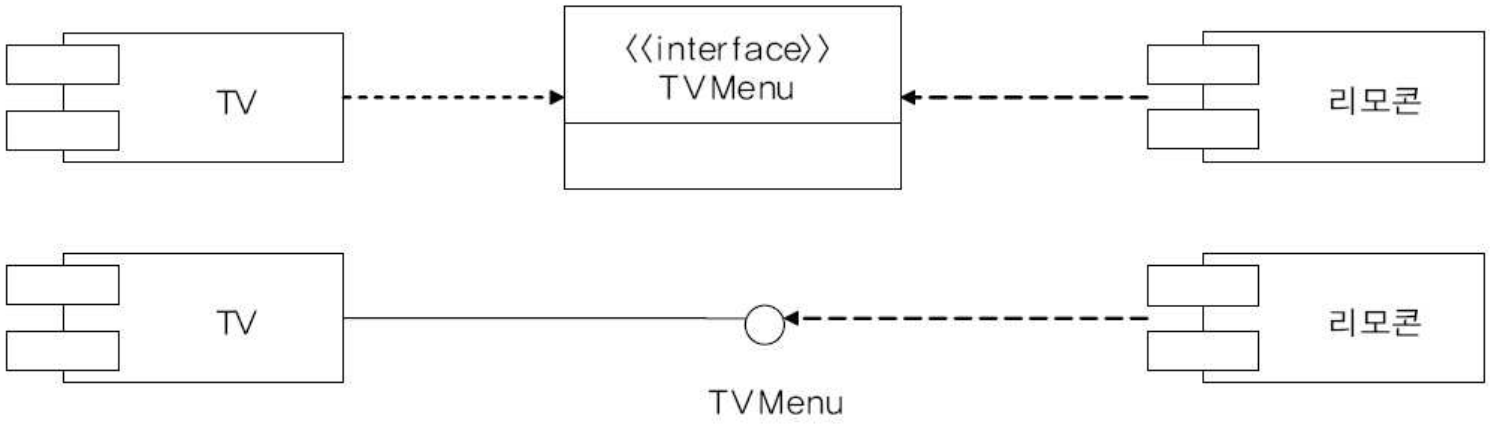
(b) 인터페이스 실체화의 간단한 표현

[그림 9-4] 인터페이스의 표현 방법

UML 표기법

➲ 인터페이스

- 인터페이스를 실체화한다는 의미
  - 실제로 동작하는 컴포넌트에 인터페이스를 적용한다는 것
- 컴포넌트 다이어그램은 실체화 관계뿐만 아니라 의존 관계도 표현
  - 의존관계는 컴포넌트와 필수의 인터페이스 사이에 설정



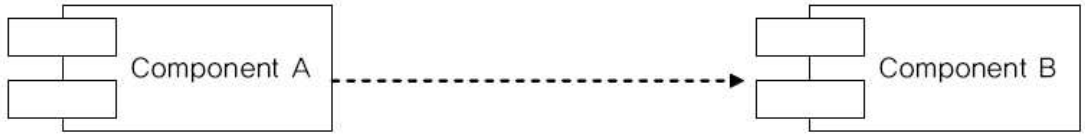
[그림 9-5] 컴포넌트와 인터페이스의 실체화와 의존관계 표현 방법

© Eun Sook Cho

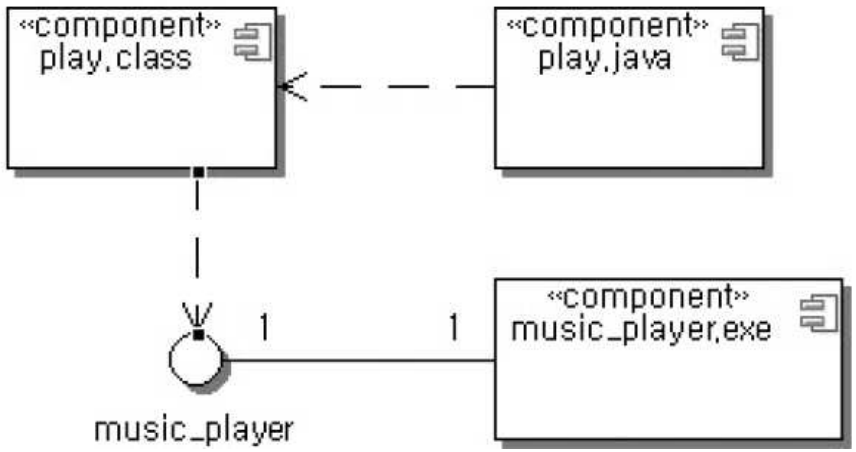
UML 표기법

➔ 의존관계

- 컴포넌트 사이의 의존관계는
  - 한 컴포넌트에 변경이 발생한 경우, 그 변경의 범위를 추적해서 파악하고 싶을 때 매우 유용



- 컴포넌트 사이의 의존관계 예



- play.class에서 play.java로의 의존관계는
  - ✓ 클래스 파일이 play.java의 소스 파일을 컴파일 함으로써 얻어지는 것
- play.java 컴포넌트에서 music_player 인터페이스로의 의존관계는
  - ✓ music_player.exe 컴포넌트가 인터페이스를 통하여 이용되는 것을 의미

© Eun Sook Cho

컴포넌트와 클래스

컴포넌트와 클래스의 공통점

- 둘 다 이름이 있고,
- 정해진 인터페이스를 실현할 수 있으며,
- 의존성과 일반화 및 연관관계에 참여할 수 있고 중첩이 가능
- 인스턴스를 가질 수 있으며,
- 교류에 참여할 수 있다

© Eun Sook Cho

컴포넌트와 클래스

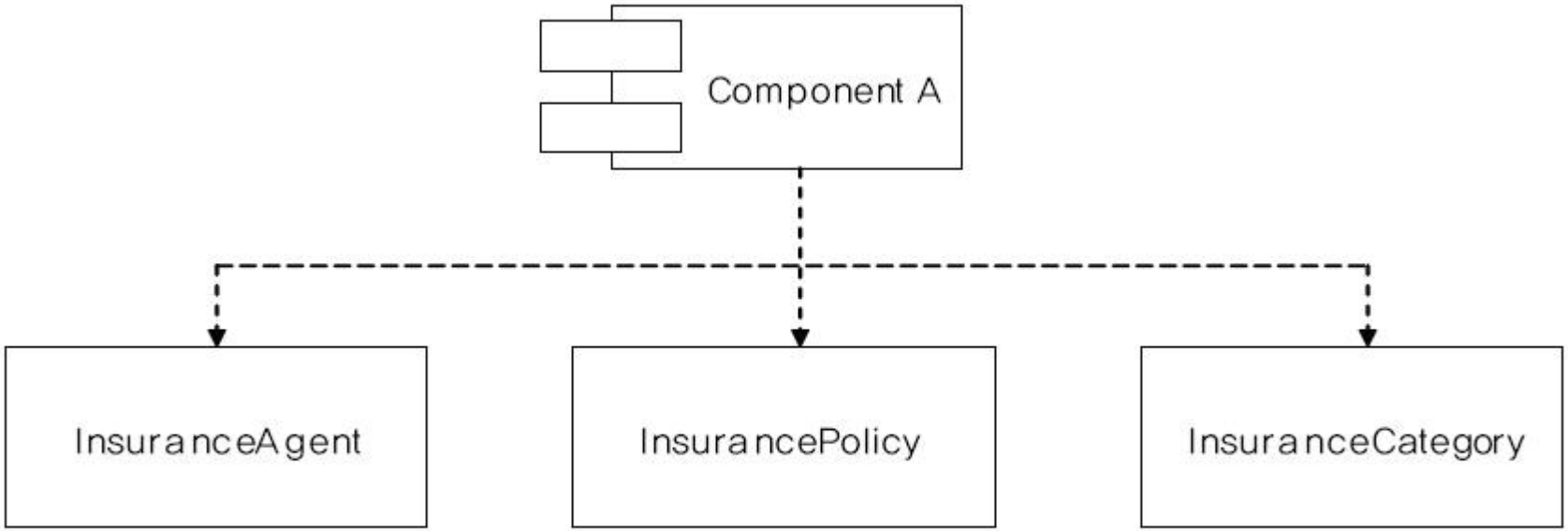
컴포넌트와 클래스의 차이점

- 클래스는 논리적으로 추상화한 것이며, 컴포넌트는 물리적인 것
  - 컴포넌트는 노드(또는 프로세서)에 존재할 수 있지만 클래스는 그렇지 않다.
  - 모델링을 하는 것이 노드에 직접 존재한다면 컴포넌트를 사용하고, 그렇지 않으면 클래스 사용
- 컴포넌트는 서로 다른 논리적 요소들을 물리적으로 패키지화한 것
  - 컴포넌트는 클래스나 통신과 같은 서로 다른 논리 요소들을 물리적으로 구현한 것 (**[그림 9-8]**)
- 클래스는 속성과 오퍼레이션을 직접 가질 수 있지만, 컴포넌트는 자신의인터페이스를 통해서만 접근할 수 있는 오퍼레이션들만 갖는다.
  - 컴포넌트와 클래스는 둘다 인터페이스를 실현할 수 있으나, 컴포넌트가 갖는 서비스들은 항상 자신의 인터페이스를 통해서만 접근이 가능

© Eun Sook Cho

컨포넌트와 클래스

➔ 컴포넌트와 클래스의 관계



[그림 9-8] 컴포넌트와 클래스의 관계

© Eun Sook Cho

컴포넌트 다이어그램의 예

- **[그림 9-9]**
  - **<<executable>>** 스테레오타입을 사용하여 실행 컴포넌트임을 나타낸다.
  - **item_search.exe** 컴포넌트
    - 부품을 검색하는 컴포넌트

[그림 9-9] item_search.exe 컴포넌트

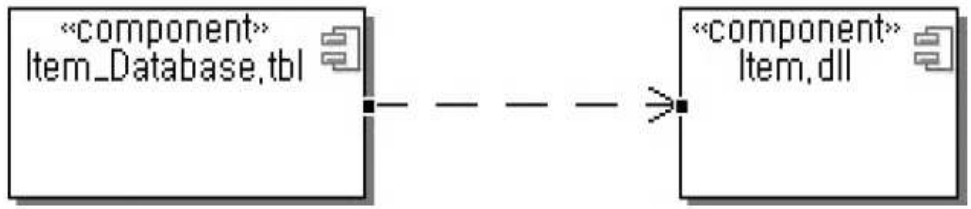
- **[그림 9-10]**
  - **<<table>>** 스테레오타입을 사용하여 실행 코드가 참조하는 데이터베이스 컴포넌트임을 나타낸다.
  - **item_database.tbl** 컴포넌트는
    - 데이터베이스를 관리하기 위한 컴포넌트

[그림 9-10] item_database.tbl 컴포넌트

© Eun Sook Cho

컴포넌트 다이어그램의 예

- **[그림 9-11]**
  - **item_database.tbl** 컴포넌트가 부품의 종류를 확인하기 위해 **item.dll** 컴포넌트를 추가한 것
  - **item.dll** 컴포넌트는
    - **<<library>>** 스테레오타입을 사용하여 실행 중에 실행 코드가 참조하는 라이브러리의 집합임을 나타낸다.
  - **item_database.tbl** 컴포넌트는 **item.dll** 컴포넌트를 이용한다"고 해석

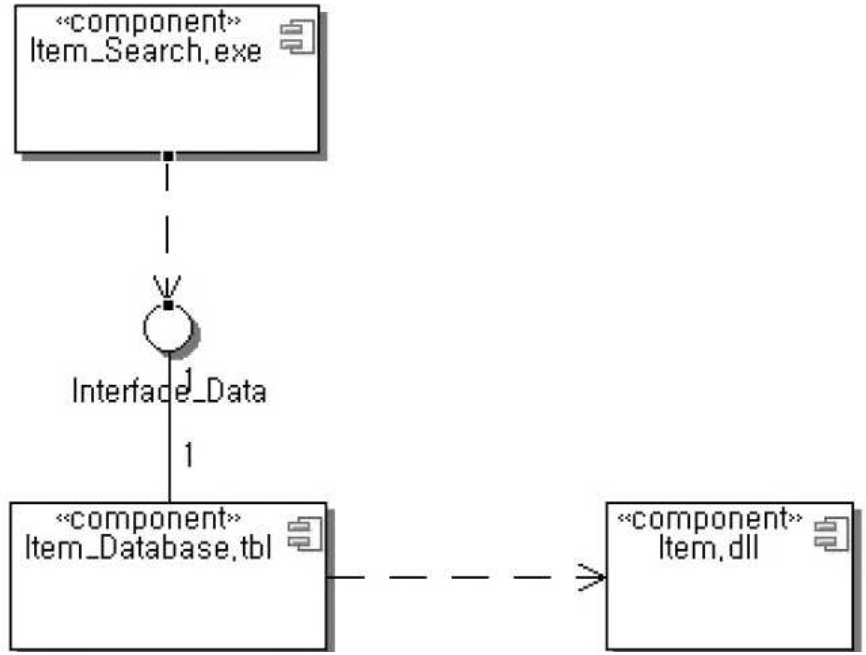


[그림 9-11] item.dll 컴포넌트

© Eun Sook Cho

컴포넌트 다이어그램의 예

- **[그림 9-12]**
  - **item_search.exe** 컴포넌트가 **item_database.tbl** 컴포넌트에 접근하기위해 **item_database.tbl** 컴포넌트에 **Interface_data** 인터페이스를 추가
  - **item_search.exe** 컴포넌트와 **Interface_data** 인터페이스 사이에 **item_search.exe**컴포넌트에서 **Interface_data** 인터페이스 쪽으로 화살표를 추가
  - "**item_search.exe** 컴포넌트는 반드시 **Interface_data** 인터페이스를 거쳐 **item_database.tbl** 컴포넌트를 이용한다"고 해석

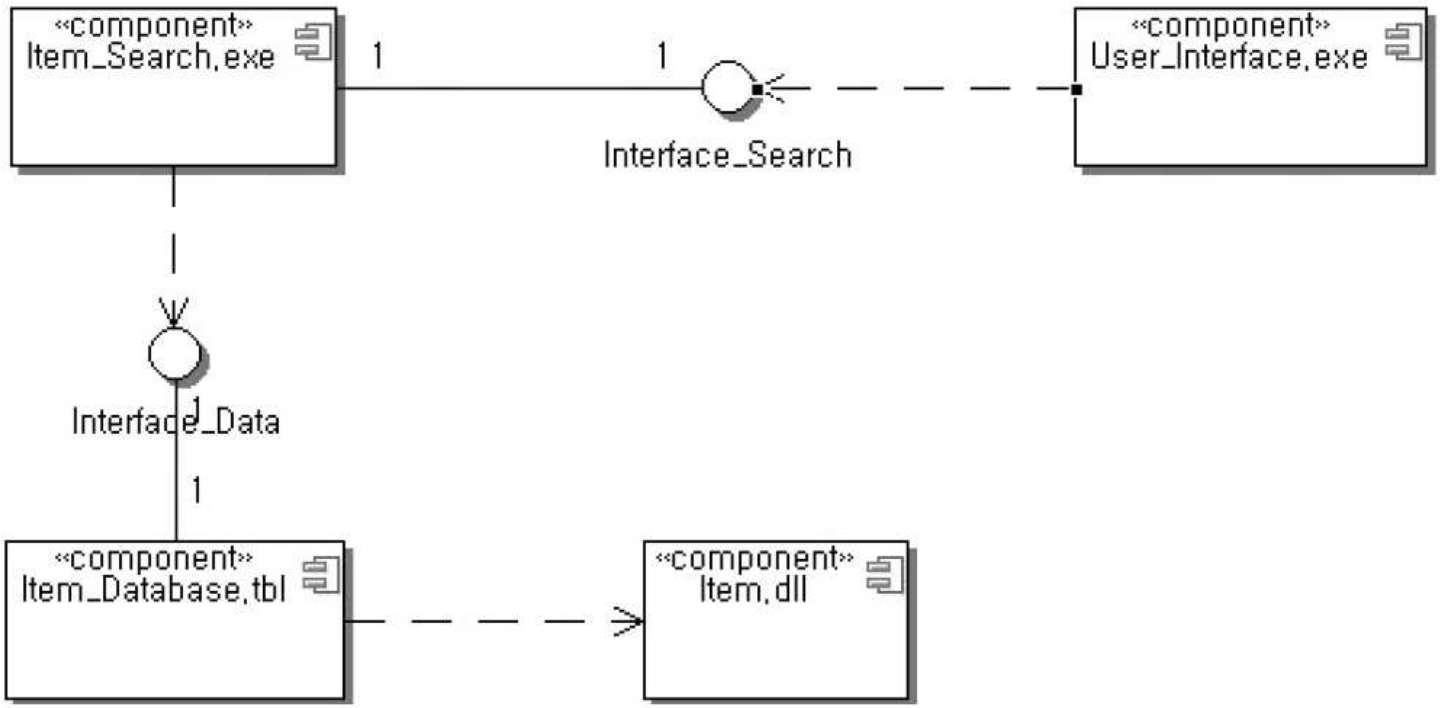


[그림 9-12] Interface_data 인터페이스의 추가

© Eun Sook Cho

컴포넌트 다이어그램의 예

- **[그림 9-13]**
  - 사용자 인터페이스를 처리하는 애플리케이션인 **user_interface.exe** 컴포넌트를 추가한 그림
  - 사용자 인터페이스에 접근하기 위해 **Interface_search** 인터페이스를 제공



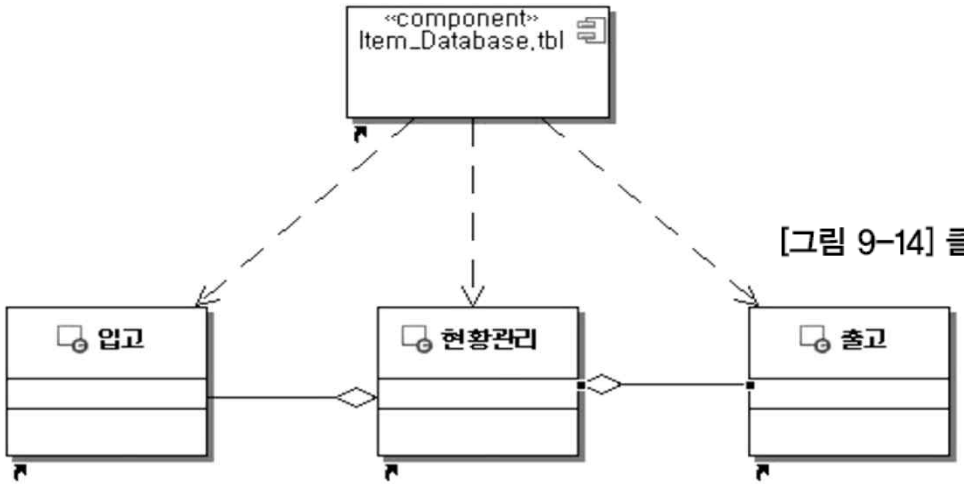
[그림 9-13] 사용자 인터페이스와 공유 인터페이스

© Eun Sook Cho

컴포넌트 다이어그램의 예

클래스를 컴포넌트로 매핑

- 클래스로부터 컴포넌트를 구성할 때 클래스들 중에 서로 연관되어 분류가 가능한 것들은 컴포넌트에 소속될 수 있다.
- 컴포넌트에 포함되는 클래스들의 인터페이스는 컴포넌트의 인터페이스가 된다.
- **[그림 9-14]**
  - **item_database.tbl** 데이터베이스 테이블 컴포넌트를 정의한 것
  - 입고 클래스, 출고 클래스, 현황관리 클래스로 구성



[그림 9-14] 클래스를 기반으로 생성된 컴포넌트

© Eun Sook Cho

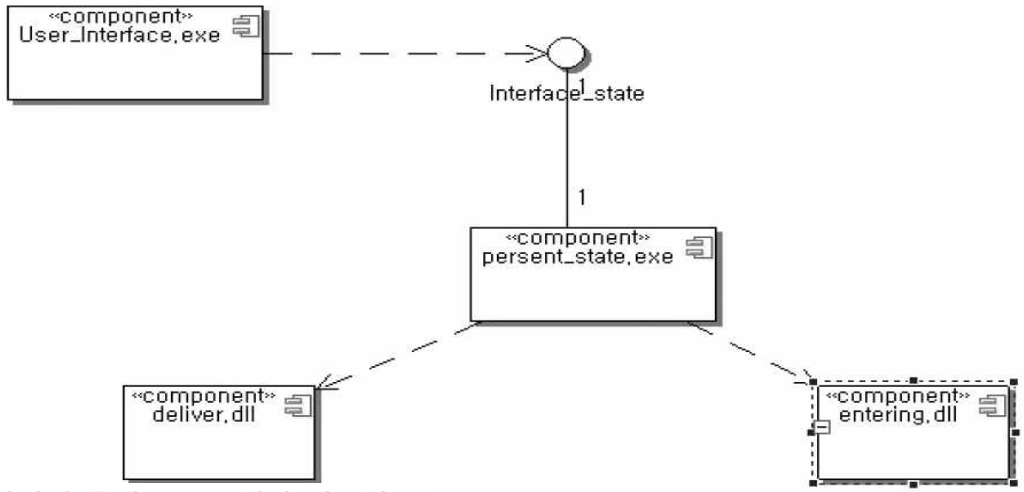
컴포넌트 다이어그램의 예

- **[그림 9-15]**
  - 애플리케이션에서 중요한 역할을 하는 것은 클래스의 전체 또는 일부를 구현하는 것
  - **<<executable>>** 컴포넌트를 생성하기 위해서는 여러 클래스를 묶어서 하나의 실행 파일로 컴파일

[그림 9-15] 클래스로부터 생성한 item_database.exe 컴포넌트

컴포넌트 다이어그램의 예

- **[그림 9-16]**
  - 실행 파일, 라이브러리, 테이블, 문서처럼 단일 클래스로 구성되는 컴포넌트도 있다.
  - **present_state.exe** 컴포넌트는 여러 개의 클래스를 하나의 컴포넌트로 컴파일하지 않고 각각의 클래스를 개별적으로 컴파일한 라이브러리 컴포넌트를 참조
  - 사용자 인터페이스는 하나의 **DLL** 컴포넌트로 구성



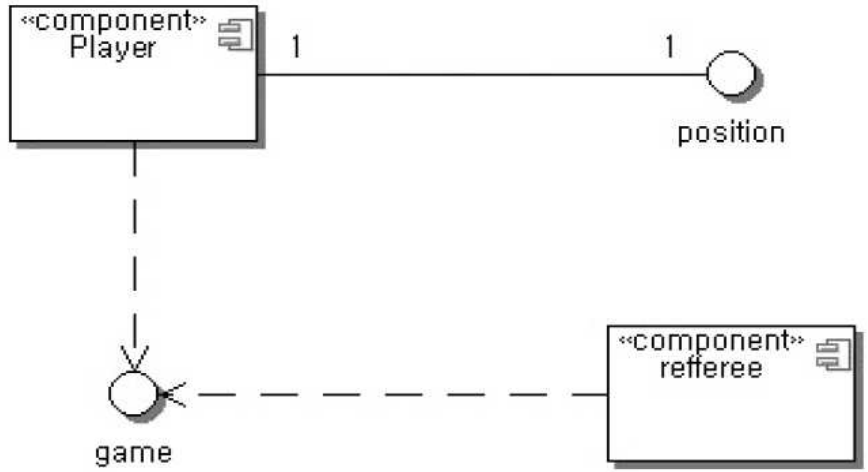
[그림 9-16] 하나의 클래스로 구성된 컴포넌트

© Eun Sook Cho

컴포넌트 다이어그램 예

➔ 축구 경기

- 축구 경기는 선수**(Player)**와 심판**(Referee)**으로 구성
- 각 선수는 고유의 포지션**(position)**을 가져 경기에 임한다.



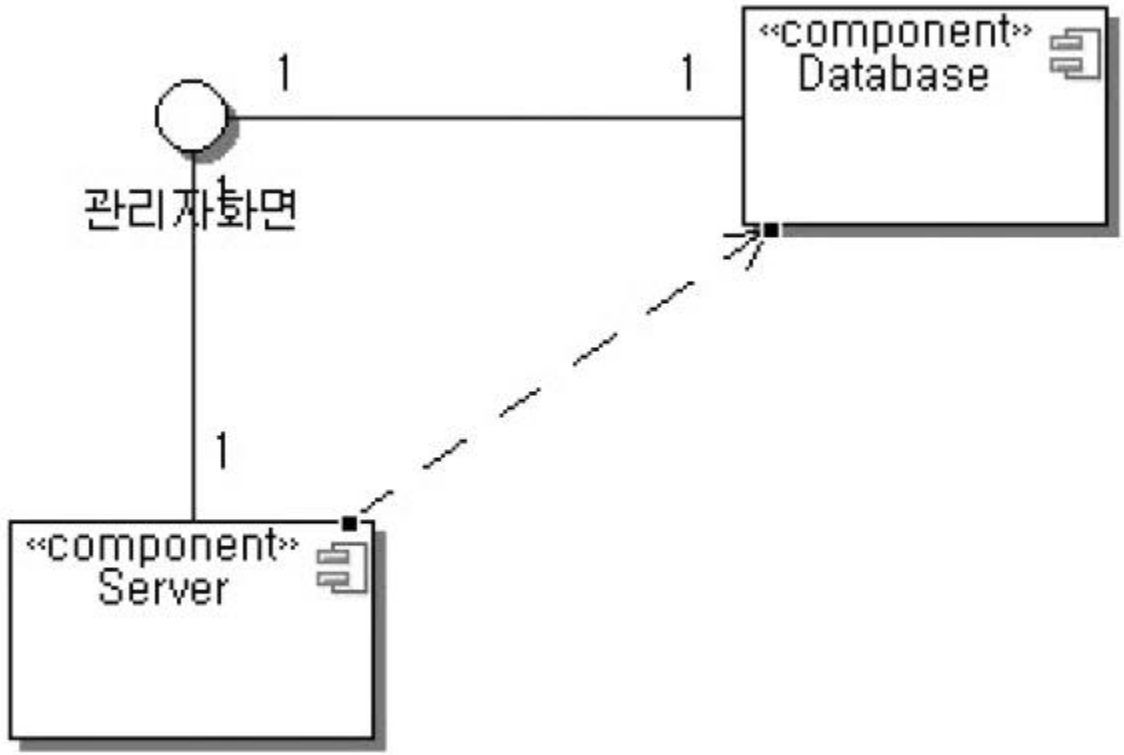
[그림 9-17] 축구 경기 컴포넌트 다이어그램

© Eun Sook Cho

컴포넌트 다이어그램 예

서버와 데이터베이스

- 서버는 데이터베이스를 바탕으로 되어 있으며,
- 관리자를 통해 서버와 데이터베이스를 관리



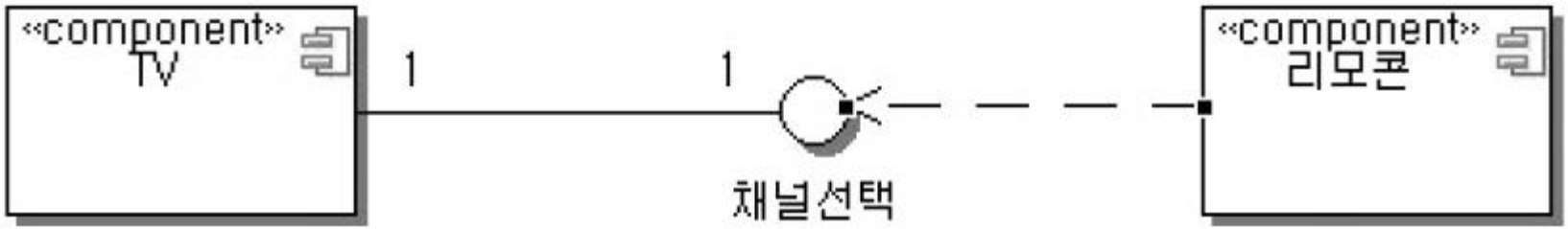
[그림 9-18] 서버와 데이터베이스 컴포넌트 다이어그램

© Eun Sook Cho

컴포넌트 다이어그램 예

➲ TV 채널 선택

- TV는 다양한 채널이 있으며 리모콘을 통해 채널신호를 입력



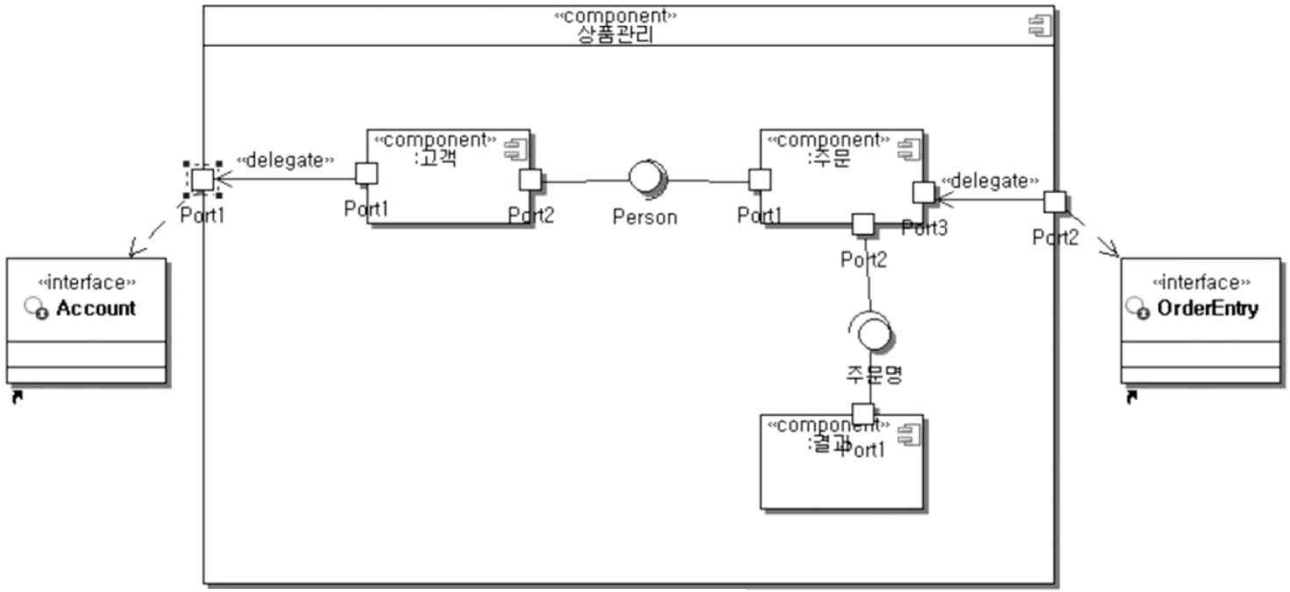
[그림 9-19] TV 채널 선택 컴포넌트 다이어그램

© Eun Sook Cho

컴포넌트 다이어그램 예

➔ 상점 관리

- 고객이 상점에 입장하면 상품을 선택하고 선택한 상품을 주문
- 주문한 상품은 주문명을 확인한 후 그 결과를 보여준다.



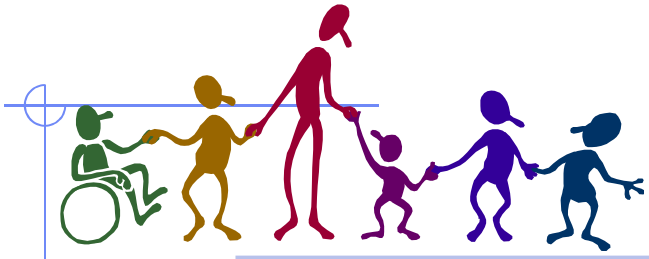
[그림 9-20] 상점관리 컴포넌트 다이어그램

© Eun Sook Cho

요약

- 컴포넌트 다이어그램
- 컴포넌트 구성요소
- 컴포넌트
- 인터페이스
- 컴포넌트 다이어그램 작성 단계

© Eun Sook Cho



Q & A