

Design Model (Component Diagram / Deployment Diagram) Document

1 조

201502023 김민기

201502093 이익수

201502101 임승민

Document Revision History

REV#	DATE	AFFECTED SECTION	AUTHOR
1	2019/11/25	Introduction, Package Diagram	김민기
2	2019/11/27	Package Diagram	이익수
3	2019/11/29	Component Diagram	임승민
4	2019/11/30	Component Discription	김민기
5	2019/12/2	Deployment Diagram	임승민
6	2019/12/3	Component Discription	이익수

Table of Contents

1. INTRODUCTION	5
1.1. OBJECTIVE	5
2. CLASS DIAGRAM	6
2.1. PACKAGE DIAGRAM	6
2.2. BUSINESS PACKAGE.....	7
2.3. UI PACKAGE.....	오류! 책갈피가 정의되어 있지 않습니다.
3. COMPONENT DIAGRAM.....	8
4. COMPONENT DESCRIPTION	9
5. DEPLOYMENT DIAGRAM.....	12
[참고]	13

List of Figure

FIGURE 1 PACKAGE DIAGRAM.....	6
FIGURE 2 DESIGN CLASS DIAGRAM (BUSINESS PACKAGE).....	7
FIGURE 3 UI CLASS DIAGRAM (UI PACKAGE).....	오류! 책갈피가 정의되어 있지 않습니다.
FIGURE 4 COMPONENT DIAGRAM.....	8
FIGURE 5 DEPLOYMENT DIAGRAM.....	12

1. Introduction

1.1. Objective

이 문서는 시간표 작성 프로그램 설계 모델 (컴포넌트 다이어그램 / 배치 다이어그램)을 기술하고 있다.

컴포넌트 다이어그램은 시스템을 구성하는 컴포넌트들의 종속관계를 표현한다. 각 컴포넌트는 설계 클래스들을 포함하고 있으며, 컴포넌트가 제공하는 인터페이스 (provided interface)와 요구하는 인터페이스(required interface)를 표현한다.

배치 다이어그램은 컴포넌트 다이어그램의 각 컴포넌트들이 물리적으로 어떻게 배치되는지를 표현한다. 컴포넌트들이 배치되는 구체적인 컴퓨팅 환경과 각 컴포넌트들을 연결하기 위한 프로토콜 등을 표현한다.

시스템 전체의 구성 요소를 파악하기 위하여 패키지와 각 패키지 내의 설계 클래스들을 표현하고 있는 패키지 다이어그램과 설계 모델(클래스 다이어그램)을 포함한다.

Class Diagram

1.2. Class Diagram

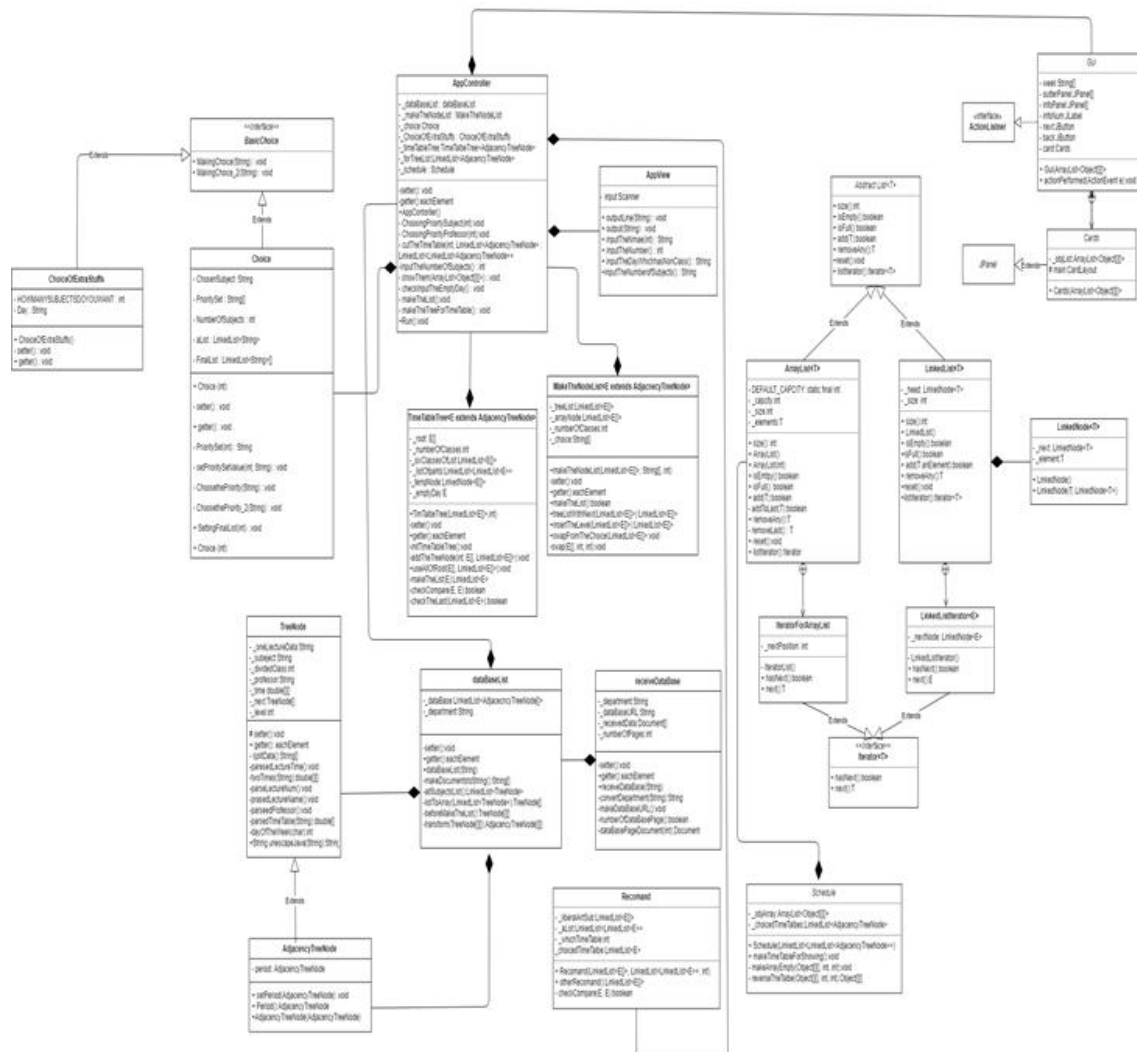


Figure 1 Class Diagram

1.3. Package Diagram

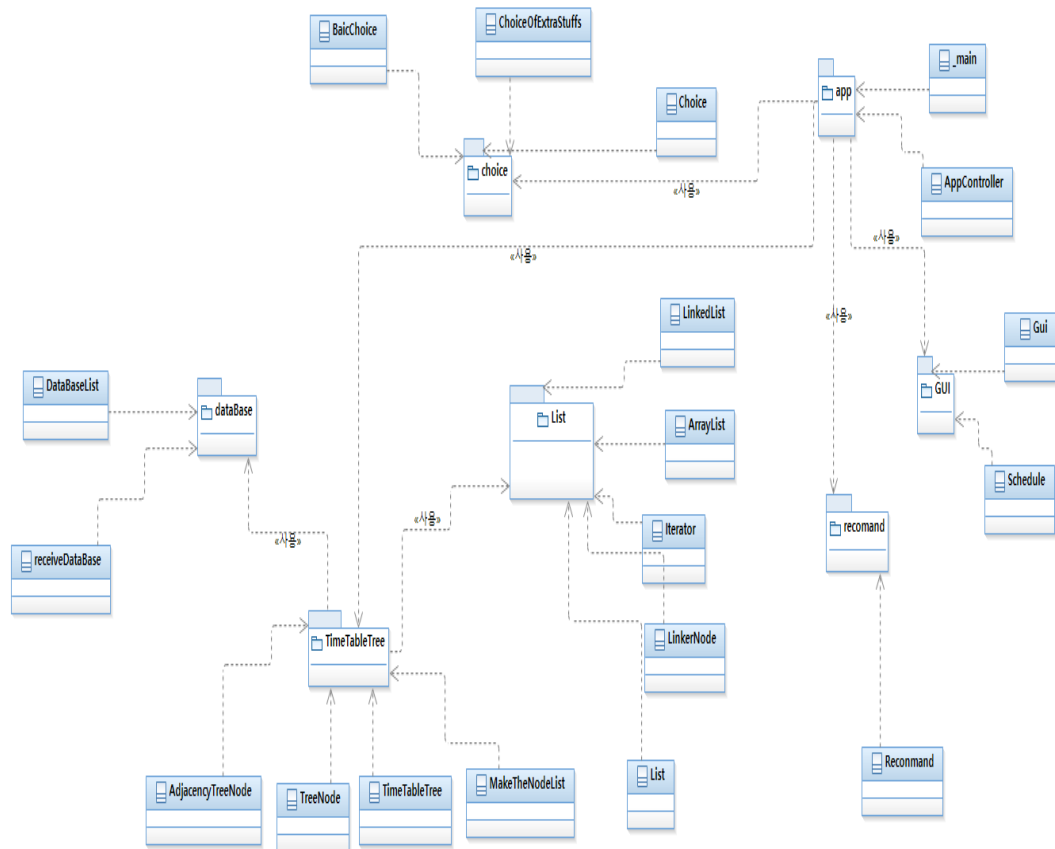


Figure 2 Package Diagram

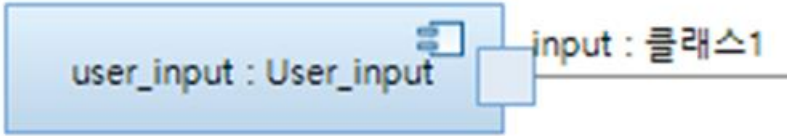
2. Component Diagram

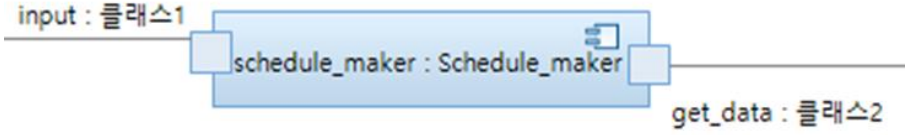


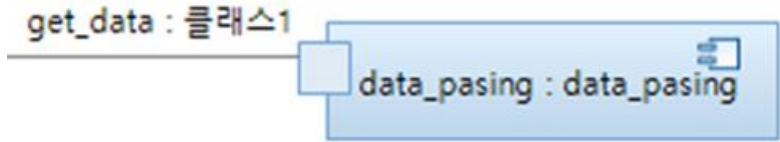
Figure 3 Component Diagram

시간표 작성 프로그램의 컴포넌트들과 인터페이스를 표시한 다이어그램이다. User_input 컴포넌트는 사용자의 입력값을 받고 이를 정리하여 schedule_maker로 넘기는 과정을 수행한다. Schedule_maker는 메인 알고리즘을 포함하고 있는 컴포넌트로 data_parsing 컴포넌트에서 데이터를 가져온 후 User_input에서 가져온 사용자의 입력정보를 토대로 데이터들을 정리하고 시간표를 작성하는 역할을 담당한다. Data_parsing 컴포넌트는 서버에서 가져온 데이터들에서 필요한 정보를 뽑아내어 정리하여 저장해놓는 역할을 하게 된다.

3. Component Description

User_input			
Component Diagram			
Component Description	사용자가 직접 접근하는 컴포넌트로서 사용자의 입력을 담당한다.		
Including Objects	AppView, BasicChoice, Choice, ChoiceOfExtraStuffs, GUI, Schedule		
Interface	Interface Type	Interface Name	Description
	Required	시간표 작성	사용자가 입력한 정보를 바탕으로 지정된 을 처리할 수 있도록 하는 기능

Schedule_maker			
Component Diagram			
Component Description	<p>시간표를 작성해주는 컴포넌트로 여기서 User_input 컴포넌트에서 입력 받은 정보와 data_parsing 에서 가져온 강의 데이터를 사용하여 사용자가 선호도가 높을 것으로 예상되는 시간표들을 작성하여 사용자에게 보여지게 된다</p>		
Including Objects	<p>Recommand, AdjacencyTreeNode, MakeTheNodeList, TimeTableTree, TreeNode, ArrayList, LinkedList, LinkedNode,List, ApplicationController, _main</p>		
Interface	Interface Type	Interface Name	Description
	Provided	시간표 작성	시간표작성을 위한 기능 및 입력정보 호출
	Interface Type	Interface Name	Description
	Required	데이터 파싱	시간표 작성을 하기 위해 데이터 정보를 해당 컴포넌트한테서 받음

data_parsing			
Component Diagram	 <pre> classDiagram class "data_parsing : data_parsing" class "get_data : 클래스1" "data_parsing : data_parsing" -- "get_data : 클래스1" : provided </pre>		
Component Description	서버에서 가져온 데이터들을 필요한 정보에 맞게 수정하고 정리하여 저장해놓는 역할을 담당한다.		
Including Objects	dataBaseList, receiveDataBase		
Interface	Interface Type	Interface Name	Description
	Provided	데이터 파싱	스케줄러가 실행하는 기능을 처리할 수 있도록 데이터를 가공하여 이를 제공하는 역할 수행

4. Deployment Diagram

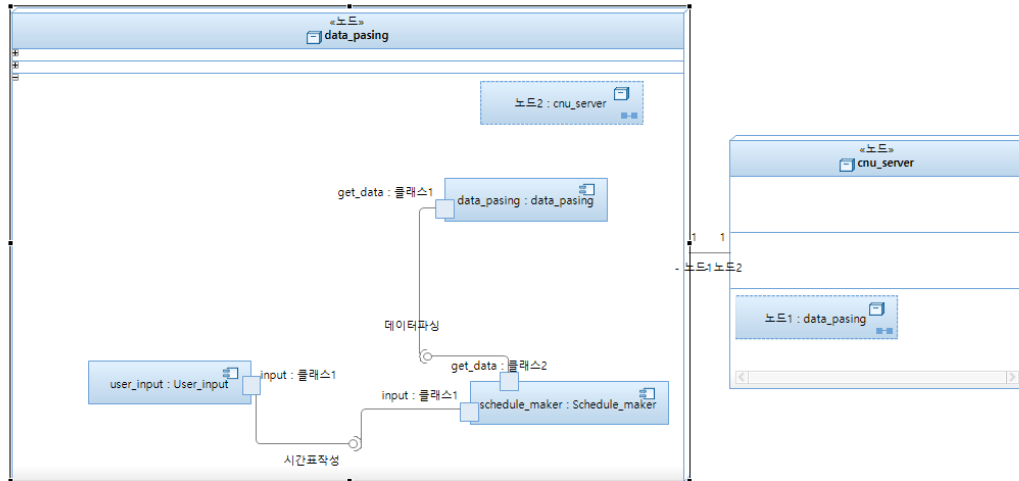


Figure 4 Deployment Diagram

시간표 작성 프로그램의 배치 다이어그램이다. 위에서 설명한 3가지 컴포넌트들이 한 노드(컴퓨터)에 포함되어 있으며 데이터를 가져올 충남대 서버도 노드로 배치되어 있는 상황을 표현하고 있다. 실제로 통신을 이용한 메커니즘 혹은 서버를 운영하여 사용되는 프로그램이 아니라 jsoup를 이용하여 HTML 문서에 저장된 데이터를 가져오므로 추가적인 통신 메커니즘은 포함되지 않는다.

[참고]

1. 컴포넌트 다이어그램

- 컴포넌트의 의미: a coherent package of software that can be independently developed and delivered as a unit, and that offers interfaces by which it can be connected with other components to compose a larger system.
- A component can include anything that a package can, including executable code, source code, designs, specifications, tests, and documentation
- Components may be single objects, or built from several objects.
- 컴포넌트 도출하기 위한 팁: 관련성이 높은 객체들을 패키지로 그룹화하여 컴포넌트로 모델링
- 인터페이스의 의미: 어떤 컴포넌트가 다른 컴포넌트에게 제공하는 서비스로서 관련된 오퍼레이션의 집합
- 인터페이스 도출하기 위한 팁: 각 컴포넌트가 제공하는 서비스를 정의하는데 초점을 둔다. 서비스에 대한 오퍼레이션을 나열하며 높은 수준의 동작을 정리한다. 즉, API(application program interface)를 정의한다.

2. 배치(Deployment) 다이어그램

- 노드의 의미: 육면체로 표현되는 모델링 요소로 물리적인 하드웨어를 나타냄. 예를 들어, 컴퓨터, 프린터, 라우터 등.
- Connection: 두 노드간의 통신 메커니즘으로 물리적인 매체(예, 광 케이블)나 소프트웨어 프로토콜(예, TCP/IP)로 표현함.
- 컴포넌트: 노드에 탑재되는 실행가능한 소프트웨어 컴포넌트