Software Engineering Q/A Sheet (#9)

date: 11/13 number: 2018312280 name: 이상수

Questions from Prof.

l. Describe the definition and benefits of the design pattern.

Design pattern은 해결 해야 할 문제와 해법을 추상적으로 기술해서 재사용하는 방식이다.

이렇게 디자인 패턴으로 만들어 두면 비슷한 문제를 만났을 때 이전에 쓴 패턴을 활용해서 쉽게 문제를 해결할 수 있다.

1. Explain the benefits of software reuse.

Reuse를 통해서 기반부터 다 만들 필요가 없고 시간비용이 줄어들고, 검증된 software를 reuse하기에 에러에 대한 수정 비용도 줄어든다.

1. Explain the activities of the configuration management with its needs.

결과물 관리라고 할 수 있다. Code, document등 관리해야할 결과물들을 관리하는 방법으로,

대표적으로 3가지가 있다.

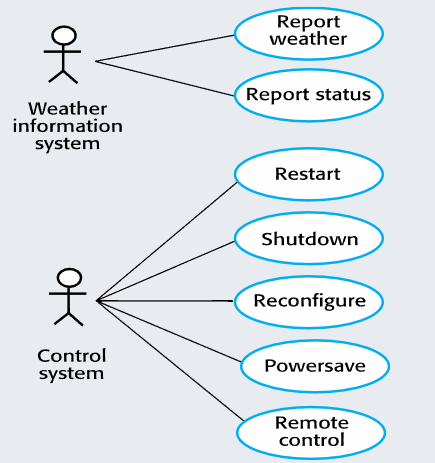
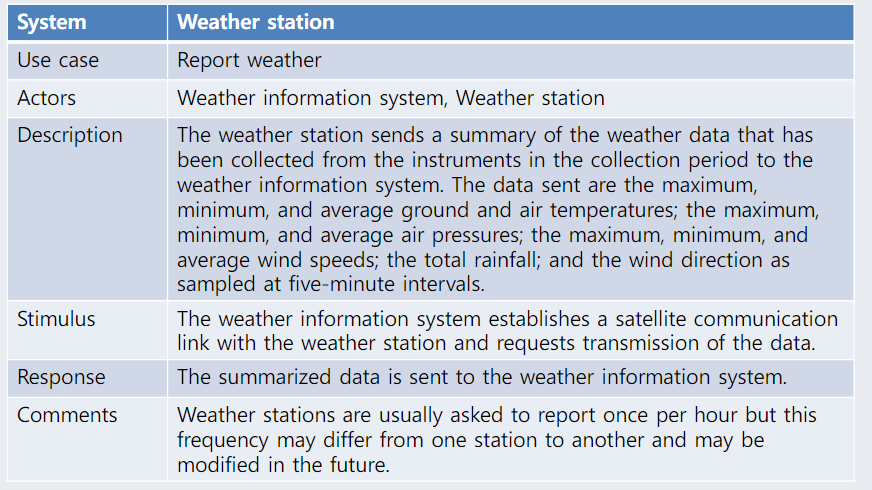
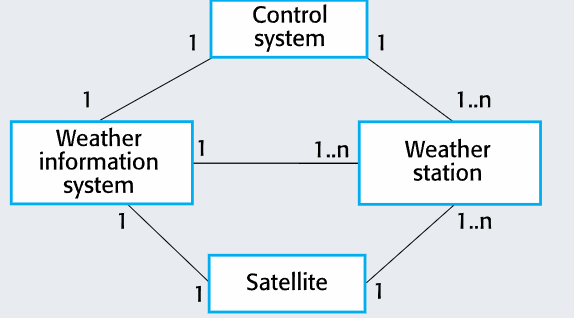
1. 버전관리 : 각 컴포넌트의 버전을 서로 업데이트 할 수 있게 관리해준다.
2. System integration : 각 시스템의 버전이 어떤 컴포넌트의 버전을 쓰는지 정하게 도움을 준다.
3. Problem tracking : 사용자들이 버그와 문제들을 개발자들에게 알릴 수 있게 해주고 어떤 개발자가 수정 중인지 다른 개발자들이 볼 수 있게 해준다.
4. Explain the value of open source from the point of view of an open-source developer and an open source user.

오픈소스developer입장에선 더 많은 소프트웨어들이 오픈소스를 활용해 더 싸고 빠르게 만들어질 것이다. 다른사람들이 이 오픈소스를 보고 bug fix등의 도움을 줄 수 있다. 자신의 명성도 넓힐 수 있다.

오픈소스user입장에선 오픈 소스를 활용해 자신이 개발하고자 하는 내용에 도움을 받아 싸고 빠르게 개발할 수 있다. Code의 안전성이 올라간다. 이를 변형하여 새로운 open-source code를 개발할 수 있다.

## Discuss the whole design process learned in this chapter with your team members. **Define the context and modes of use of the system**

* 설계되는 소프트웨어와 외부 환경 사이의 관계를 이해하는 것은 필요한 시스템 기능을 제공하는 방법과 시스템 환경과 통신하기 위해 시스템을 구조화하는 방법을 결정하는 데 필수적이다.
* 또한 context를 이해하면 시스템의 boundaries를 설정할 수 있다. 시스템 경계를 설정하면 설계 중인 시스템에 구현된 기능과 관련된 다른 시스템에 있는 기능을 결정하는 데 도움이 된다.
* 시스템 컨텍스트 모델(system context model)은 개발 중인 시스템의 환경에서 다른 시스템을 보여주는 구조적(structural) 모델이다. (개발하고 있는 시스템 말고도 딴 시스템들까지 구조적으로 보여주는 것)
* 상호 작용 모델(interaction model)은 시스템이 사용되는 동안 환경과 어떻게 상호 작용하는지를 보여주는 dynamic 모델이다.
* weather station의 system context model
  + weather station의 예
  + 땅이 넓은 나라의 정부는 수 많은 기상 관측소를 외딴 지역에 배치한다.
  + 기상 관측소는 기온, 압력, 일조량, 강우량, 풍속, 풍향을 측정하는 장비들로부터 데이터를 수집한다.



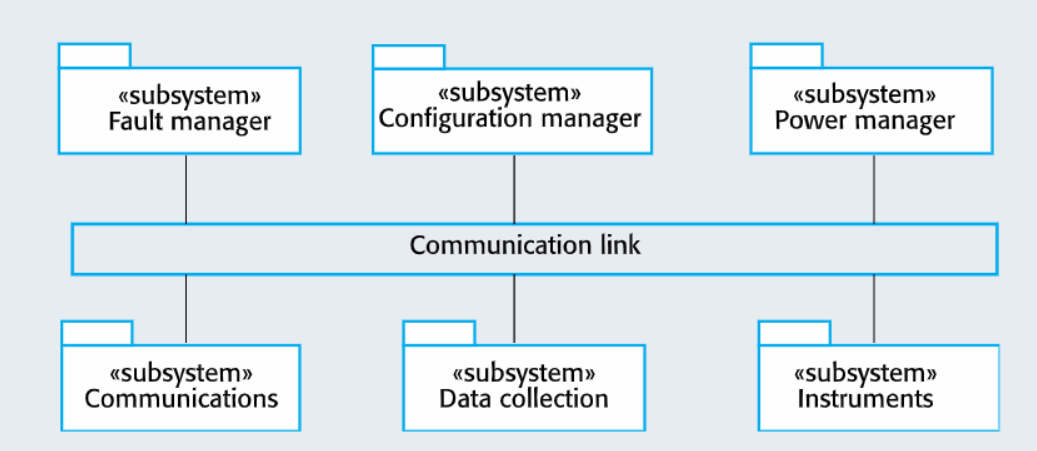
Design the system architecture

- 시스템 구조를 설계합니다.

- 주요 컴포넌트와 그들 간의 상호작용을 식별하고 architectural pattern을 사용해서 컴포넌트를 구조화합니다.

- architectural pattern 예시 - layered 모델, client-server 모델

- 예시 :

- weather station의 high-level arichitecture : 공통의 communication link를 사용해서 메세지를 broadcasting하는 방식으로 소통하는 독립적인 subsystem들로 구성되어있습니다.

Identify the principal system object Classes

- 주요 시스템의 Object Class들을 식별하여 정리한다.

- 왕도는 없고, System Designer의 기술, 경험, 그리고 Domain 지식을 바탕으로 만들어진다.

- Iterative Process로, 한번에 정리되지 않고 지속적으로 수정된다.

- Approaches to Identification

- System에 대한 자연어적 설명을 바탕으로 \*\*문법적 분석\*\*을 사용하라. (명사에 주목!)

- Object, Attributes : 명사 / Operations, Services : 동사

- Tangible Things (눈으로 볼 수 있는 / 만질 수 있는 객체)를 대상으로 식별하라.

- Roles, Events, Interactions, Location, Organizational Units

- “행위(Behavioural)”, 행동에 대한 접근

- Scenario 기반 분석

Develop design models

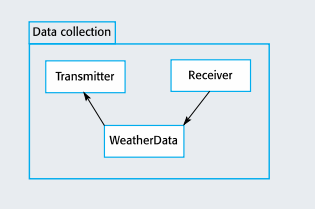
- Design models는 object와 class of obejcts를 그 관계에 따라 보여준다.

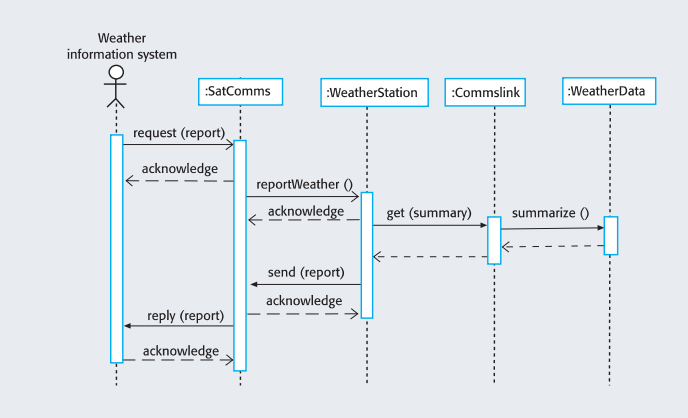
- Static models는 object classes와 그들의 관계에 대한 static한 구조를 보여준다.

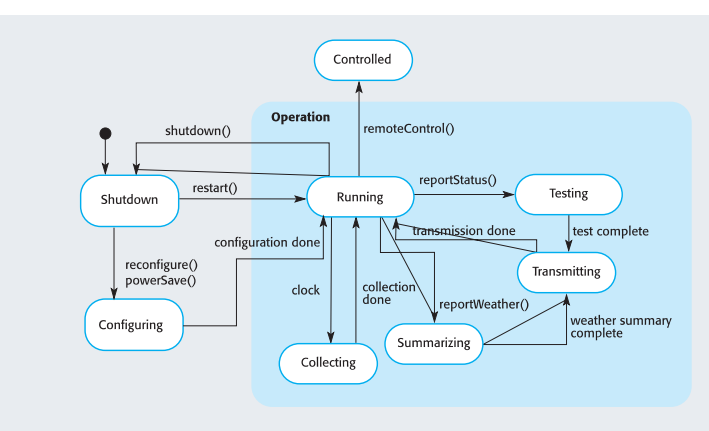
- Dynamic models는 objects들 간에 dynamic한 상호작용을 보여준다.

- 예시 :

- Subsystem models는 논리적으로 연관된 objects간에 그룹이 subsystem으로 합쳐진 모습을 보여준다.



Sequence models 는 object들 간에 상호작용 절차를 시간순으로 보여준다.

State machine models는 각 object가 특정 event에 따라 어떻게 변화하는지 보여준다.  


use-case models, aggregation models, generalization models등도 있다.

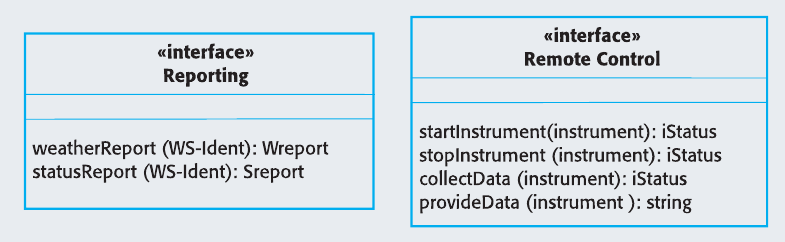
Specify object interfaces.

- 접속명세서를 작성하는 단계에서는, 각각의 컴포넌트나 오브젝트가 서로 엑세스하기 위해 필요한 인터페이스를 설정한다. 이러한 인터페이스는 parallel한 개발을 위해 요구된다.

- 인터페이스 디자인에서는 데이터의 자세한 표현은 지양해야 하며, 따라서 attributes와 같은 것은 정의되지 않도록 한다. 하지만 데이터에 접속하기 위한 operations는 꼭 포함해야 한다.

- object는 viewpoint에 따라서 여러 인터페이스를 지닐수도 있다.

- weather station의 예



Questions from you

1. 오픈소스나 디자인 패턴을 배울 방법에는 무엇이 있을 까?
2. 커뮤니티에 참여하기, 다양한 알고리즘등을 알아보고 공부하기

팀질문 :

reuse했을 때 단점:

reuse한 코드가 오히려 문제가있을 수있다.