QA Sheet 7

2017310695 이주민

1. Describe the purpose of system modeling of existing and new systems, respectively.

- Purpose of system modeling of existing systems : 이미 존재하고 있는 시스템이 어떤 역할을 하는지를 명세화 해주고, 이 시스템의 장점과 단점에 대해 논의할 수 있게 해줌. 이러한 정보들은 결국 새로운 시스템에 대한 요구사항을 도출할 수 있도록 해줌.
- Purpose of system modeling of new systems : 요구 공학 과정에서, 제시된 요구사항들에는 어떠한 내용들이 있는지를 다른 이해 당사자들에게 설명해줄 때 사용됨. 엔지니어들은 이 모델을 이용하여 디자인을 구상하고, 문서화 과정을 진행함.

2. Explain the four perspectives of system modeling and connect the appropriate diagrams in UML to each perspective.

① External perspective : 시스템의 배경이나 개발하려고 하는 시스템의 환경을 모델링하는 관점으로, 사용되는 UML 다이어그램으로는 Activity diagram이 있음

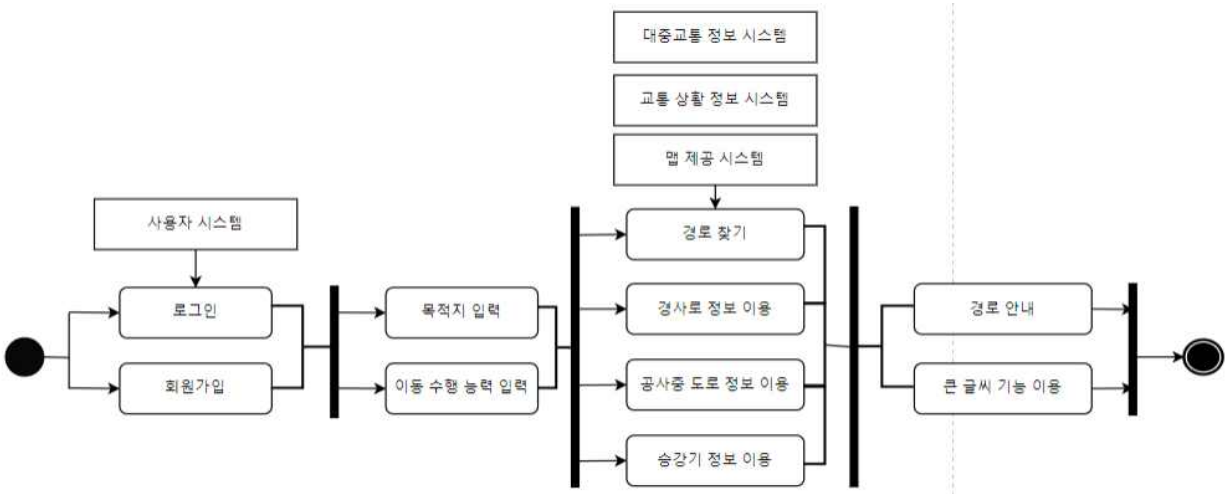
② Interaction perspective : 시스템과 개발 환경 사이 혹은 시스템의 컴포넌트들 사이의 상호작용을 모델링하는 관점으로, 사용되는 UML 다이어그램으로는 Use case diagram, Sequence diagram이 있음

③ Structural perspective : 시스템의 구성을 모델링하거나 시스템에서 사용되는 데이터의 구조를 모델링하기 위해 사용되는 관점으로, 사용되는 UML 다이어그램으로는 Class diagram이 있음

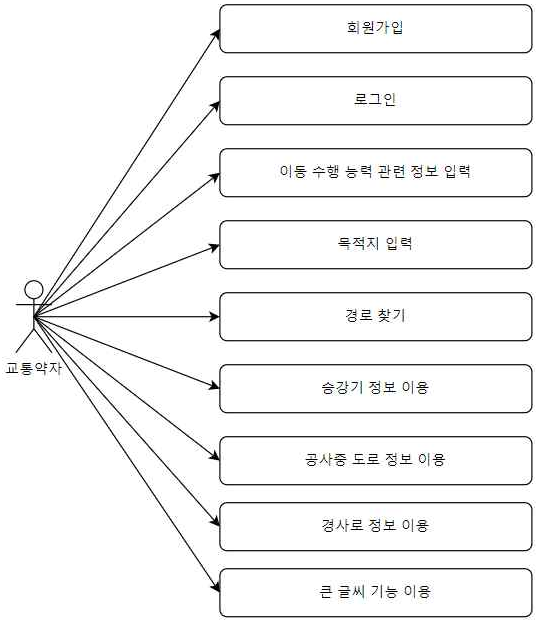
④ Behavioral perspective : Dynamic behavior를 모델링하거나, 시스템이 외부로부터 주어지는 event에 대해 어떻게 반응하는지를 모델링하는 관점으로 사용되는 UML 다이어그램으로는 State diagram이 있음

어떤 diagram을 어떤 목적으로 쓰는지 알기

3. Draw a process diagram for your term project.



4. Draw use case diagrams for your term project.



5. Learn and explain more about component, package and deployment diagram.

① Component diagram : 실제 구현 모듈인 컴포넌트 간의 관계나 컴포넌트 간의 인터페이스를 표현하며, 이는 구현 단계에서 사용되는 다이어그램임

② Package diagram : use case나 class 등의 모델 요소들을 그룹화한 패키지들의 관계를 표현함

③ Deployment diagram : 결과물, 프로세스, 컴포넌트 등 물리적 요소들의 위치를 표현하며, 노드와 의사소통(통신) 경로로 표현함. 이는 구현 단계에서 사용되는 다이어그램임

6. Explain why generalization is effective for complexity management.

: 만약 시스템에 변화가 생겼을 경우에, 일반화를 사용하여 모델링을 한 경우에는 이 변화에 대응하기가 훨씬 쉬워짐. 객체 지향 프로그래밍 언어에서는 클래스 상속을 이용하여 일반화를 하는데 상위 클래스와 연관된 멤버나 연산은 하위 클래스와도 연관이 있게 된다(계승). 하위 클래스들은 필요한 경우에 특정한 멤버나 연산을 더 추가할 수 있다.

7. Compare date-centric and event-centric models for purposes and modeling techniques.

- Data-centric model : 시스템에 input data를 넣어서 해당 output을 도출하는 과정에 포함된 액션들의 시퀀스를 나타냄. 이 model은 end-to-end processing에 사용될 수 있으므로 요구사항 분석 단계를 위해 만들어짐.
- Event-centric model : 시스템이 외부나 내부 이벤트들에 대해 어떻게 반응하는지를 보여줌. 이 model은 시스템이 유한한 상태를 가지고 있다고 가정하고, 어떠한 이벤트가 하나의 상태에서 다른 상태로의 전이를 일으킨다는 것을 전제로 함. 해당 model은 control system이나 embedded system에 사용하기 위해 만들어짐.

형태는 똑같을 수도 있지만, system에서 어떻게 사용되냐에 따라서 data/event로 나뉠 수 있음

8. Explain the advantages and disadvantages of model-driven engineering.
- Advantage
① 시스템을 더욱 높은 차원에서 추상화하여 생각할 수 있음
② 코드를 자동적으로 생성할 수 있으므로, 시스템이 새로운 플랫폼에 적응하는데 비용이 적게 듦
- Disadvantage
① 추상적이라 실제로 구현하기에는 적합하지 않다.
② Translator를 생성하는 것이 코드를 생성하는 것보다 비용도 더 많이 들고 어려울 수 있다.
modeling은 code를 생성하기 위해서 함. coding하는 사람은 자연어로 된 문서만으로만 작성하기에는 부족함이 있음. working하는 process의 동작에 대해 집중하고 싶으나 그 외에 신경써야 할 부분이 많음 하지만, MDE에서는 abstract을 사용하여 detail한 부분(language/platfrom에 대한 부분)은 가리고, 한 가지 view point만 가지고 한가지에만 집중함. 오직 PIM에만 집중하고 싶음!

9. Compare CIM, PIM, and PSM in MDA.
- CIM : 개발하려고 하는 시스템의 domain에 대한 모델
- PIM : 시스템의 동작을 모델링하는 것으로, 주로 정적인 시스템의 구조나 외부나 내부의 이벤트에 대해 어떻게 반응하는지를 나타내는 UML 모델을 이용하여 표현함
- PSM : PIM을 특정 플랫폼에 dependent한 특성으로 변환시킨 것.
CIM과 PSM은 전문 기업에서 이미 given된 것으로 가정.

10. Explain the slide about MDA transformation flow.
- 첫 번째 translator : CIM, Domain specific guidelines, 고객의 요구사항을 반영하여 PIM 생성
- 두 번째 translator : PIM, Platform specific patterns and rules를 반영하여 PSM 생성
- 마지막 translator : PSM, Language specific patterns를 반영하여 최종적으로 실행 가능한 코드 생성

MY Question
1. 만약에 하위 클래스가 상위 클래스의 속성을 가지지 않는다면 어떡하는가?
: 하위 클래스는 상위 클래스에 대해 is-a, kind-of 사이의 관계인데 이는 상위 클래스의 속성을 하위 클래스가 모두 받으므로 부족함. 따라서, aggregation을 사용해서 part of 관계를 사용함. 또는, 아예 그 상위클래스에서 벗어나는 방법도 있음.

2. package diagram과 deployment diagram의 차이점은?
: deployment는 phsical diagram이고, package는 logical diagram임