

Q/A Sheet #7 – System Modeling

Date: 4/25 number: 2016311821 name: 한승하

Questions from Prof

1. Describe the purpose of system modeling of existing and new systems, respectively.
 - A. Existing system에서의 system modeling은 Requirement engineering 과정에서 기존에 존재하는 system 중 개발할 system에 도움이 될 만한 system을 찾아내고 modeling하여, 새로이 개발되는 system의 Requirement에 도움이 될 수 있게 하는 것을 목적으로 한다.
 - B. New system에서의 system modeling은 기존의 system에서 개선 혹은 확장된 Requirement들을 modeling하여 stakeholder들에게 이를 쉽게 이해시키기 위함과 Engineer들이 model들을 기반으로 design과정에 사용하거나, 문서를 작성하는 것에 도움이 되는 것을 목적으로 한다.
2. Explain the four perspectives of system modeling and connect the appropriate diagrams in UML to each perspective
 - A. External perspective
 - i. System이 Operation되는 배경 혹은 문맥, 상황 및 환경에 대한 관점을 의미한다. (Context and environment)
 - ii. UML diagrams중 Use case diagrams을 사용할 수 있다.
 - B. Interaction perspective
 - i. System과 환경 사이에 있어서의 interaction, 또 System의 component사이의 interaction에 대한 관점을 의미한다.
 - ii. UML diagrams중 Sequence diagrams을 사용할 수 있다.
 - C. Structural perspective
 - i. System의 구성 및 구조에 대한 관점을 의미한다.
 - ii. UML diagrams중 Activity diagrams, Class diagrams을 사용할 수 있다.
 - D. Behavioral perspective

- i. System이 외부의 events에 대해 어떻게 반응하는가, 혹은 실행시에 어떤 동작들을 하는가 (dynamic behavior)에 대한 관점을 의미한다.
- ii. UML diagrams중 State diagrams을 사용할 수 있다.

3. Draw a process diagram for your team project.

4. Draw use case diagrams for your team project.

아직 Team project의 outline이 모두 정해지지 않아서 이후 q/a sheet에서 작성하겠습니다.

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

- A. Component는 function을 구현하여 재사용 가능한, interface를 제공하는 software으로 jar혹은 dll과 같은 물리적인 파일이다.
- B. Package는 관련성 있는 UML요소들을 하나의 그룹화 시킨 것을 의미하며, Package간에는 의존관계가 존재할 수 있다.
- C. Deployment diagram은 Software의 조각들이 실제로 어떤 하드웨어에 배치되어 실행되는지를 보여주는 System의 Physical한 관점을 이야기한다.

6. Explain why generalization is effective for complexity management

- A. Generalization은 class들에 대한 hierarchy를 표현하는 방법으로, 상위 Class와 하위 Class의 Is – a, Kind – of의 관계를 표현하는데 자주 사용된다. Generalization을 사용하면 복잡한 대량의 class들을 개별적으로 관리하지 않고, 계층관계를 통해 보다 효율적으로 관리할 수 있다. 또한 generalization의 inheritance 특성을 사용하면, 대량의 class로 구성되어지는 system의 보다 효율적인 변경과 관리가 가능하다.

7. Compare data-centric and event-centric models for purpose and modeling techniques.

- A. System은 Input data에 대한 output data의 단위로의 관점으로 바라볼 수 있다. Data-driven modeling은 처리하고자 하는 대상이 data이고 입력되어지는 data에 대해 기대되어지는 output data의 생성이 목적인 System으로써의 관점을 의미한다. 따라서 input data에서 output data로의 action의 sequence로 system을 표현할 수 있

다.

- B. Event-driven modeling은 System을 외부로부터의 event와 그에 대한 response로 이루어진다는 관점으로 바라본다. 따라서 System이 Event로 인해 하나의 State에서 다른 State로의 이동하는 State간 이동을 기반으로 Graphical notation이 가능하며, 이는 주로 Embedded system에서 많이 사용된다. 하지만 이러한 State가 많아지고 복잡해지면, Graphical notation을 한눈에 알아보기 어려워지므로, 이러한 경우 Stepwise refinement로 기술해 나가는 것이 필요하다.

8. Explain the advantages and disadvantages of model-driven engineering

- A. MDE는 program보단 model에 집중한다. Model-driven engineering은 System에 대해 abstraction level을 높일 수 있으며, 자동적으로 코드가 생성될 수 있어, 다양한 language에 대해 쉽게 변경, 혹은 새로운 platform에 대한 변경에 쉽게 적용할 수 있다 라는 강점을 이야기한다.
- B. Model은 지나치게 추상적이기 때문에 구현하는데 있어 필요한 모든 요소들을 포함하지 못한다. 따라서 많은 사람의 implementation이 필요하다. 또한 새로운 language, platform으로의 적응을 위해 translator 등이 필요하기 때문에 MDA가 실질적인 cost의 이득이 없을 수도 있다. 라는 단점을 이야기한다.

9. Compare CIM, PIM, and PSM in MDA.

- A. CIM은 개발하려고 하는 system의 domain에 대한 model을 이야기한다.
- B. PIM은 특정 platform에 independent한 model로 system의 동작에 대한 model을 이야기한다 이는 system이 외부 event에 대한 response을, 또는 static system structure에 대한 modeling을 이야기한다.
- C. PSM은 PIM을 특정 platform에 대해 dependent한 platform specific model로 변환시킨 modeling을 말한다. 이는 PIM에서 platform specific한 patterns혹은 rules들을 입력으로 받아 만들어진다.

10. Explain the slide about MDA transformation flow.

- A. MDA transformation flow는 CIM으로부터 Executable code로의 변환과정을 이야기한다. 이는 CIM과 Domain specific guidelines를 사람의 개입을 통해 PIM을 생성하고, 생성된 PIM에 Platform specific pattern and rules를 input으로 추가하여 PSM을 도출한 뒤 이를 Language에 대해 변환하여 Executable code를 생성해낸다. 이 모든 과정

에는 Translator의 동작이 필요하다.

Questions of yourself

1. Data flow diagram이 UML에서 빠진 이유가 무엇인가?
 - A. UML은 기본적으로 객체지향을 목적으로 만들어졌다. 하지만 DFD는 function 중심의 관점에서 사용된 diagram이기 때문에 UML에서 빠지게 되었다 하지만 이러한 data flow에 대한 표현이 필요하기 때문에 UML에선 activity diagram을 사용하게 된다.
2. MDA의 Translator의 역할은 무엇인가?
 - A. Translator는 CIM과 사람의 implement, Domain specific guidelines를 입력으로 받아 PIM을, PIM과 Platform specific patterns and rules을 입력으로 받아 PSM을 PSM과 Language specific patterns를 입력으로 받아 Executable code를 만들어내는 유닛이다. 따라서 이 Translator를 만드는 부하가 굉장히 크다. 하지만 한번 만들어지면 여러 상황에서 재사용 될 수 있다.