

## 13주차 3차시 UML의 다이어그램

### 【학습목표】

1. UML의 다이어그램의 종류와 그의 따른 이점과 작성 원칙을 설명할 수 있다.
2. UML의 다이어그램의 특징을 확인하여 실제로 사용할 수 있다.

### 학습내용1 : UML의 다이어그램

#### 1. 유즈케이스 다이어그램(use case diagram)

- 1) 자콥슨(jacobson)이 제안한 내용을 UML에서 그대로 채택한 방법 임.
  - 객체를 찾아내는 방법이 어렵기 때문에 사용자의 요구사항에서 자연스럽게 객체를 찾아내기 위하여 정형화된 다이어그램 형태를 이용하여 의사소통을 목적으로 작성하는 다이어그램임.
- 2) 유즈케이스 다이어그램의 특징을 요약하면 다음과 같음.
  - 프로세스 맵보다 이해 / 작성이 용이함.
  - 클래스 다이어그램이나 시퀀스 다이어그램과 상호보완적 기능으로 효과가 제고됨
  - 정형화된 다이어그램 형식으로 표현하여 의사소통이 용이함.
  - 구조적분석 설계의 자료흐름도 중에서 최상위도(context diagram)와 유사한 기능을 함.
- 3) 유즈케이스 다이어그램 작성절차는 다음과 같음.
  - 해당 시스템과 관련되는 각각의「액터(actor : 자료흐름을 주도하는 객체)」를 정의함.
  - 각 액터에 관련된 「유즈케이스」를 정의함.
  - 여기서 유즈케이스는 시스템 과 액터의 상호작용을 위하여 시스템이 제공하는 일련의 기능(function) 혹은 거래(transaction)를 의미함.
  - 각 유즈케이스에 해당하는 사건(event)은 추후 설명함.

#### 2. 시퀀스 다이어그램(sequence diagram)

- 1) 객체 사이에 「메시지(message)를 처리하는 순서(order)에 중점을 두고」작성하는 것임.
  - 이는 시스템의 동적모델을 명확하게 표현하기 때문에 객체 사이의「메시지 교환 내용」을 파악하는데 유용함.
- 2) 시퀀스 다이어그램의 이점·작성 원칙은 다음과 같음.
  - 개발자의 관점에서는 사용자가 요구하는 정확한 시스템이나 시스템의 동적인 흐름을 명확히 파악 가능함.
  - 사용자 관점에서는 요구한 시스템의 전체적 구성형태나 시스템의 흐름을 확실히 파악 가능함.
  - 유즈케이스에서 추출된 객체 사이의 상호작용을 나타내고 있기 때문에 다이어그램에서 다시 새로운 객체 추출이 가능 함.
  - 실시간 시스템 모델화에 필요한 병렬 프로세스의 표현이나비동기 메시지를 지원 함.
  - 각 유즈케이스는 하나의 시퀀스 다이어그램으로 작성하는 것이 일반적인 관행임.

- 그러므로 그렇지 못할 경우 유즈케이스를 몇 개로 분할하여 시퀀스 다이어그램을 작성토록 조치해야 함.

### 3. 협력 다이어그램(collaboration diagram)

- 1) 상호 관련이 있는 객체 사이에 관련성을 분석하기 위하여 객체들 사이에 정적인 연결관계를 표현하는 도표임.
  - 메시지가 처리되는 과정을 파악하기 용이함.
  - 객체 사이에 존재하는 결합도(coupling)의 파악이 용이함.
- 2) 시퀀스 다이어그램과 협력 다이어그램은 어느 하나만 작성하면 다른 하나는 작성하기 용이함.
  - 시퀀스 다이어그램은 메시지 처리 순서를 기초로 작성하고, 협력 다이어그램은 객체 사이의 관련성 분석을 기초로 작성하기 때문임.

### 4. 클래스 다이어그램(class diagram)

- 1) 클래스 다이어그램의 작성은 다음과 같이 함.
  - 작성초기에는 분석단계에서 요구사항에 나타난 「엔티티(entity)」를 「클래스(class)」로 전환함.
  - 작성초기에는 객체, 클래스가 함께 나타남.
    - 이때 객체(object) 중에서 완전히 추상화가 이루어지지 않은 것은 그 특성에 따라서 새로운 클래스(class)로 정의됨.
  - 분석단계에서는 사용자 입장에서 본 문제영역에 대한 속성과, 연산 및 관련성 등을 정의한 다음에 설계단계를 거치면서 「시스템에서 실제로 구현될 클래스」로 변환됨.
- 2) 클래스 다이어그램에서 「클래스」의 표시는 「사각형」으로 함.
  - 작성초기에는 클래스 사이의 관계를 「연관관계」로 나타내다가, 단계적으로 상세화 시키면서 상속·집합·관련성·포함 등으로 표현함.
  - 관련이 있는 양 객체 사이에 대응관계를 나타내기 위하여 그들의 대응 내용을 관계의 양쪽 끝에 표기하고, 그 관계가 어떤 역할(role)인가를 표기함.
- 3) 클래스 다이어그램의 이점은 다음과 같음.
  - 사용자가 원하는 시스템의 구조를 용이하게 정의 가능함.
  - 분석단계에서 사용자 인터페이스의 프로토타입(prototype) 작성이 용이함.
  - 입출력 화면이 하나의 객체로 표시되기 때문에 시스템 구조화가 용이함

### 5. 상태 다이어그램(state diagram)

- 1) 「복잡한 객체」나 「객체 내부에 프로세스의 동적행위」를 모델화 하기 위해서 작성함.
- 2) 객체가 생명주기 동안에 어떤 일을 하는지 파악하려고 할 경우에 작성하면 효과적임.
- 3) 시스템의 흐름을 객체단위로 상세하게 나타내는 장점이 있지만, 작성하기 어렵다는 단점으로 특별한 경우가 아니면 작성하지 않음.

## 6. 패키지 다이어그램(package diagram)

- 1) 시스템 규모가 거대하고 복잡한 경우 모든 객체를 하나의 집단으로 통합하여 구현하는 것은 불가능함.
- 2) 이 경우 수많은 객체들을 관련 정도 · 특성을 중심으로 집단화시켜서 패키지(package)로 만들어 구현함이 바람직함.
- 3) 패키징을 위하여 작성되는 것이 패키지 다이어그램(package diagram)임.
- 4) 패키지 다이어그램의 이점은 다음과 같음.
  - 클래스 사이의 관계를 쉽게 파악이 가능함.
  - 객체들을 모듈로 그룹핑(grouping)하는 도구의 역할을 하기 때문에 시스템의 구현이 용이함.
  - 대규모 프로젝트 수행에 특히 효과적임.
  - 패키지는 다수의 서브패키지·클래스를 가져도 무방함.
  - 각 패키지는 구현과정을 거치면서 모듈 · 컴포넌트(component)가 됨.

## 7. 활동 다이어그램(activity diagram)

- 1) 하나의 유즈케이스를 대상으로 하여 시스템의 흐름을 쉽게 파악하기 위하여 활동 (activity)을 중심으로 나타낸 도표임.
- 2) 구조적 기법에서 자료흐름도(DFD)와 유사한 기능을 함.

## 8. 배치 다이어그램(deployment diagram)

- 1) 기존의 시스템 구조도는 각 프로세스의 배치방법, 서버와 클라이언트 사이의 연결상태나 통신방법을 나타내고 있지만 표준화를 위한 '기준'이나 '표준형식'이 다르다는 이유로 「의사소통」에 문제가 야기됨.
- 2) 위와 같은 문제를 해결하기 위하여 시스템 구조도를 표준화시켜서 작성한 것이「배치 다이어그램」임.
- 3) 배치 다이어그램은 분산처리 형태, 서버의 클러스팅, 데이터베이스 등을 효과적으로 나타냄

## 【학습정리】

1. UML의 다이어그램을 전반적으로 알아본다.
2. 유즈케이스 다이어그램을 파악한다.
3. 시퀀스 다이어그램을 이해한다.