

- UNIFIED
 - 객체지향기법들의 통일
- MODELING
 - 요구사항 분석 & 객체 설계 => 모델화
- LANGUAGE
 - ■圖式表記法

https://www.uml.org/







- The UML is a language for
 - visualizing
 - specifying
 - constructing
 - documenting

the artifacts of a software-intensive system

- UML can also be applied outside the domain of software development.
- 00개념 기반, 다양한 의미-단순한 표현 가능, 학습 수월
- 업계표준
- OMG에서 제정(www.omg.org)

A picture is (sometimes) worth 10000 words!

흡연과 건강

적어도 현재까지 의학적으로 알려진 바람직한 건강행위들을 잘 지키면 앞으로 인류는 현재보다 더 건강하게 살 수 있으며 평균수명도 지금보다 30세 정도는 더 연장시킬 수 있을 것으로 믿고 있다. 여러 바람직한 건강행위 중에서도 적절한 식이, 적절한 운동, 기호품의 조절 그리고 정신건강의 유지 등이 가장 핵심적으로 지적되고 있다.

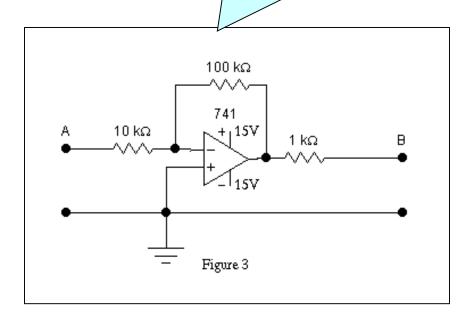
기호품의 조절에서 가장 중요한 것이 흡연과 음주다. 흡연행위는 기호행위중 가장 널리그리고 보편적으로 애용되어 왔으나 흡연이 건강에 해롭다는 사실은 분명하게 밝혀졌다. 담배는 마약이다. 최신의학계에서는 니코틴이 약물중독으로 취급되고 있다. 담배가 모든질병과 관계가 있다는 것은 구태여 말하지 않아도 누구나 다 알고 있다. 담배갑에는 '금연하면 건강해지고 장수할 수 있습니다'라는 글귀와 '흡연은 폐암등 각종 질병의원인이 되며, 특히 임산부와 청소년의 건강에 해롭습니다'라는 경고문이 실려있다. 즉,기호로서의 흡연행위가 각종 악성종양을 비롯한 많은 여러 만성질환의 이환과 사망에직간접으로 크게 기여하여 흡연자들의 평균수명을 적어도 5~8년 단축시키고 있다는놀라운 사실들이 알려져 있다. 뿐만 아니라 각종 성인병의 발생과 사망은 많은 요인들이복합적으로 작용하여 일어나고 있으나 그중에서도 단독적으로 흡연이 가장 많은 해를주는 요인이라는 것도 알려졌다. 그러므로, 금연을 해야....

A picture is (sometimes) worth 10000 words!

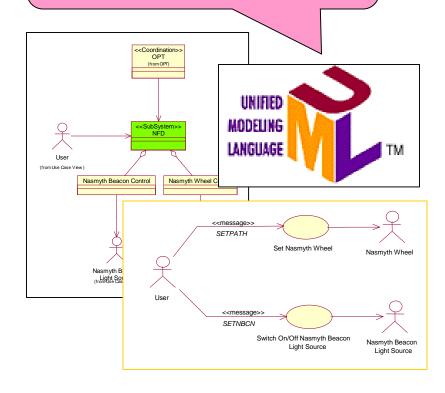




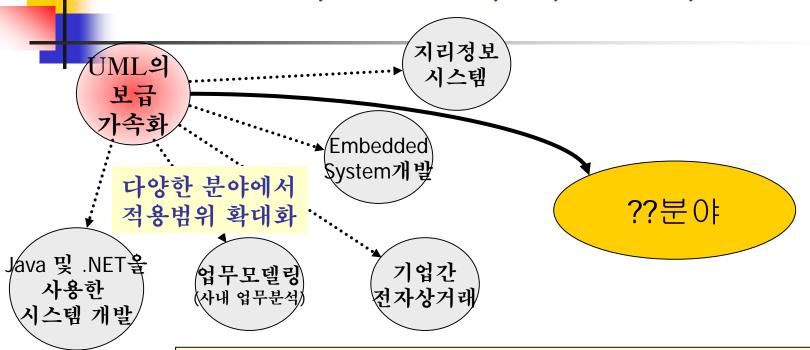
Every engineer understands an electronic diagram



Every SW engineer WILL understand the UML diagrams



UML 적용분야의 증가



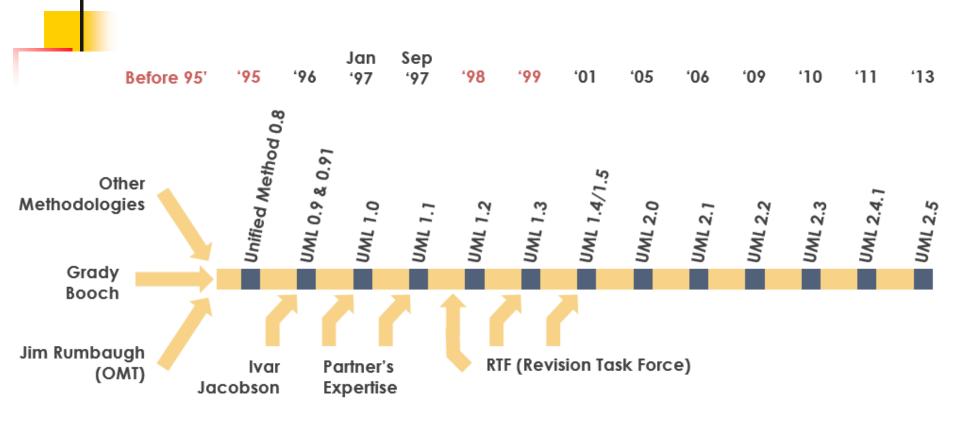
배경

- 시스템의 내부구조 및 업무가 복잡
 - → 가시화하여 전체를 파악하려는 요구 증대
- 객체지향개발의 확산
 - → SW재이용을 용이하게 해주는 분석/설계가 중요시
- 모델링언어의 "사실상의 표준"화
 - → "공용어"로서 사양을 정확하게 전달할 수 있게 됨

1.UML의 역사

<u> </u>	
version	특징
방법론 전쟁시대	50여개 이상의 OOAD방법론 亂立
UM(Unified Method) 0.8	Booch기법 + OMT기법(G. Booch, J. Rumbaugh)
UML 0.9	OOSE기법(Ivar Jacobson)과 통합
UML 0.91	
UML 1.0	주요 OO기술 전문업체 가세, OMG제안
UML 1.1	OMG 표준화
UML 1.3	
UML 1.4	
UML 1.5	
UML 2.0	컴포넌트기술 접목, 임베디드SW개발에 적용가능
UML2.1	
UML2.2	UML profile제공 → 특정용도에 맞게 자유롭게 확장가능
UML2.3	
UML 2.4.1 ~ 2.4.x	
UML2.5.1	최신버젼
	방법론 전쟁시대 UM(Unified Method) 0.8 UML 0.9 UML 0.91 UML 1.0 UML 1.1 UML 1.3 UML 1.4 UML 1.5 UML 2.0 UML 2.1 UML 2.1 UML 2.2 UML 2.3 UML 2.4.1 ~ 2.4.x

OMG UML의 진화



Before 95' - Fragmentation > 95' - Unification > 98' - Standardization > 99' - Industrialization



Documents Associated With Unified Modeling Language™ (UML®) Version 2.5

Normative Documents

OMG document number	Explanation	Format	URL
formal/15-03-01	v2.5	PDF	http://www.omg.org/spec/UML/2.5/PDF

Normative Machine Consumable Files

OMG document number	Explanation	Format	URL
ptc/13-08-17	Primitive Types	xmi	http://www.omg.org/spec/UML/20131001/PrimitiveTypes.xmi
ptc/13-08-17	UML 2.5	xmi	http://www.omg.org/spec/UML/20131001/UML.xmi
ptc/13-08-17	UML 2.5 Standard Profile	xmi	http://www.omg.org/spec/UML/20131001/StandardProfile.xmi
ptc/13-08-17	UML 2.5 Diagram Interchange	xmi	http://www.omg.org/spec/UML/20131001/UMLDI.xmi

RTF Report: http://www.omg.org/cgi-bin/doc?ptc/13-09-02



OMG UML 1.5 Specification

- UML Summary
- UML Semantics
- UML Notation Guide
- UML Standard Profiles
- UML XML Metadata Interchange DTD
- Object Constraint Language
- ...

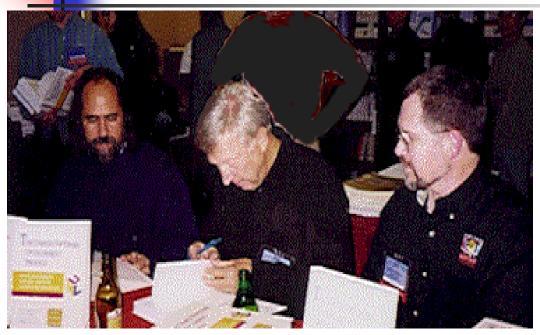




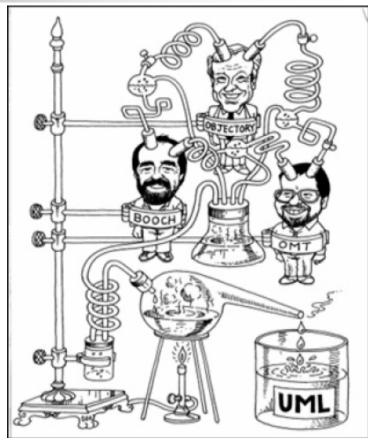
OMG UML 2.0 Specification

- Superstructure
 - UML의 구성요소, 의미, 표기법 등을 정의
- Infrastructure
 - UML자체의 기본적인 구조를 정의
- Object Constraint Language
 - UML모델내의 제약/룰 등을 기술하기 위한 언어사양
- Diagram Interchange
 - UML모델을 XML형식으로 변환하기 위한 사양(XMI)을 도형 정보도 취급할 수 있도록 확장

The Three Amigos at the UML World



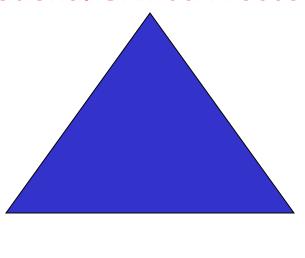
From left to right: Grady Booch, Ivar Jacobson, Jim Rumbaugh



객체지향 방법론

개발공정(Process)

Rational Unified Process

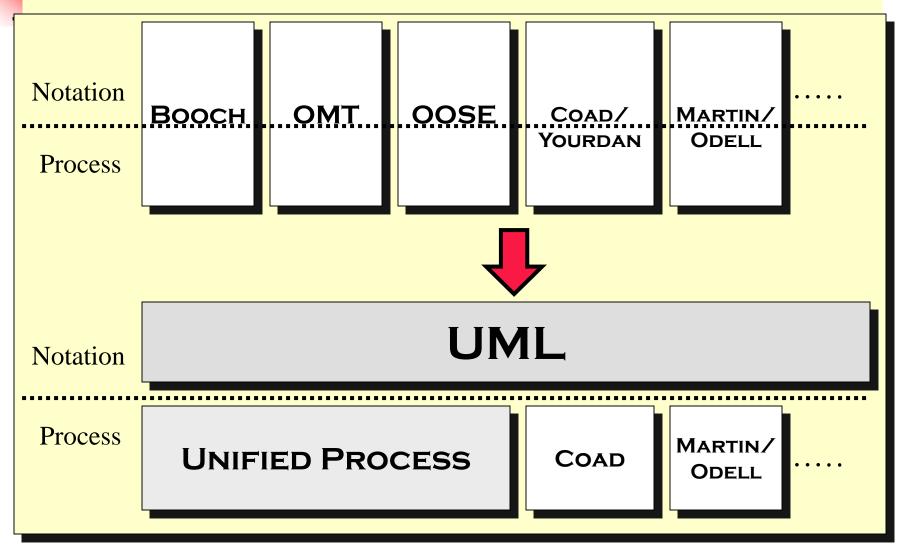


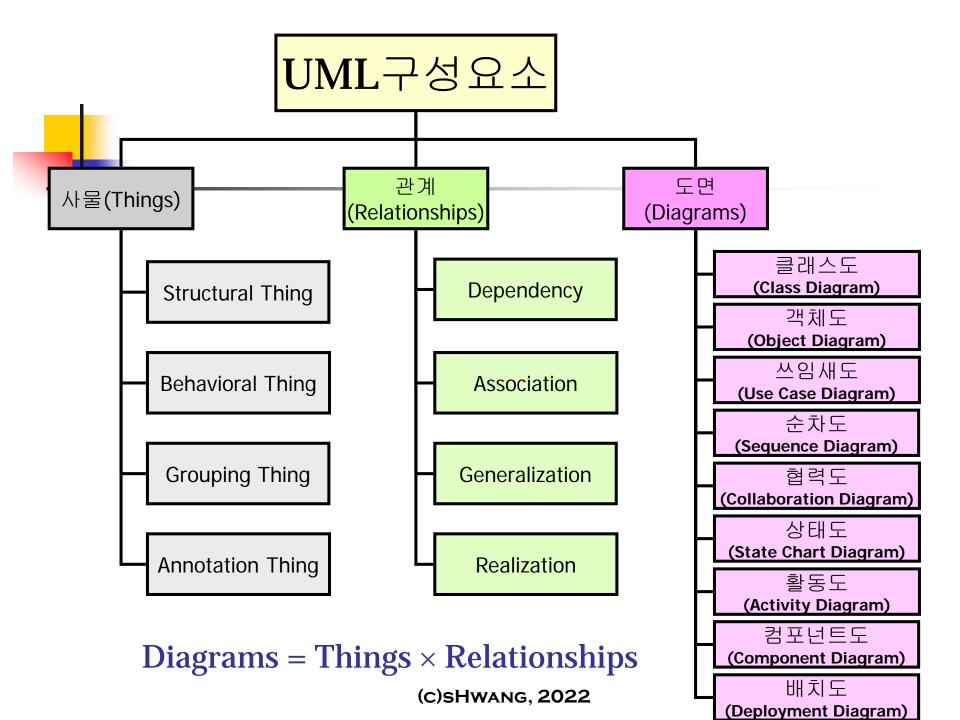
표기법(Notation) UML

<u>개발도구(Tool)</u> Rational ROSE,

. . .

UML: a de facto standard notation





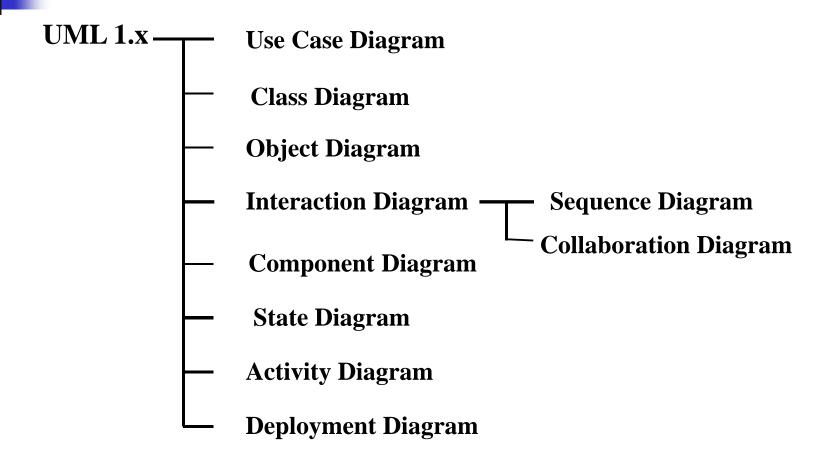
Things in the UML

- Structural things
 - mostly static parts of a model
 - class, interface, collaboration, use case, active class, component, node
- Behavioral things
 - dynamic parts of UML models
 - interaction, state machine
- Grouping things
 - organizational parts of UML
 - package, subsystem
- Annotational things
 - explanatory parts of UML
 - note

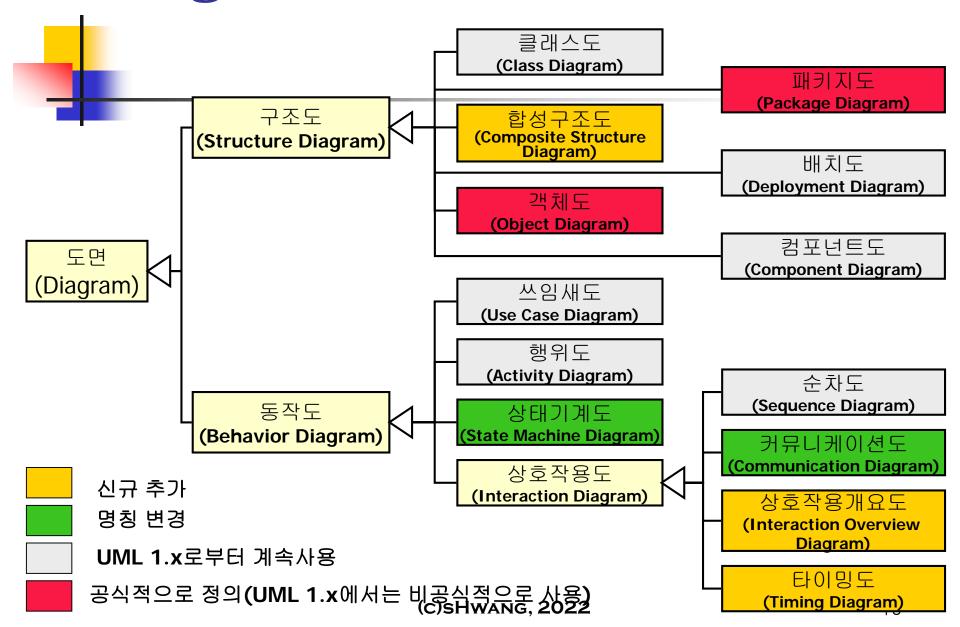
Relationships in the UML

- Dependencies
 - State that a change in specification of one thing (the supplier) may affect another thing that uses it (the client), but not necessary the reverse
 - Generalizations
 - A relationship between a general thing (called the superclass or parent)
 and a more specific kind of that thing (called the subclass or child)
 - also known as the "is-a-kind-of" relationship
 - A child inherits the properties of its parents
 - Associations
 - A structural relationship that specifies that objects of one thing are connected to objects of another
 - Aggregation & Composition
 - Realizations
 - Relations between similar elements of different degrees of detail.
 - Relations between an interface and its implementation.

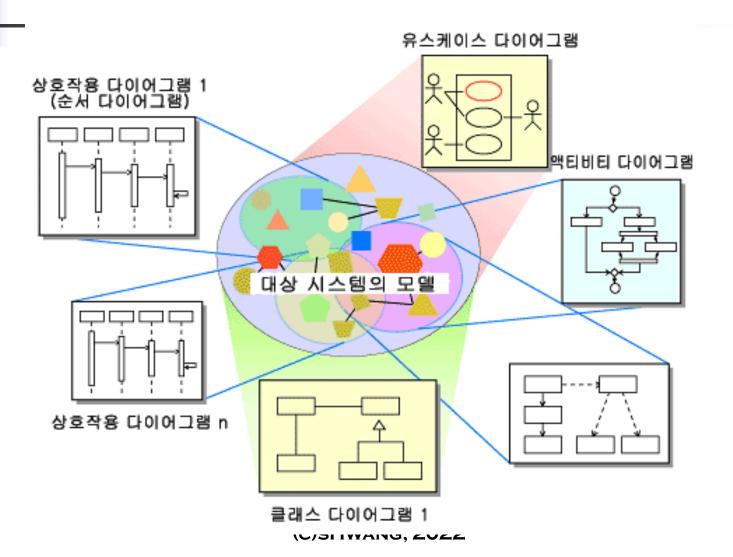




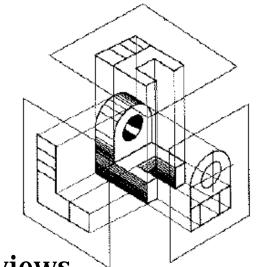
Diagrams in UML2.x



Why so many diagrams?



Why so many diagrams?



- System models are organized into multiple views
 - Different abstraction levels
- Different aspects (e.g., workflow, domain concepts, deployment)
- Each view conforms to some viewpoint that prescribes some appropriate modeling notation
- Each viewpoint is relevant to some stakeholder

UML Diagram의 분류(UML 1.x)

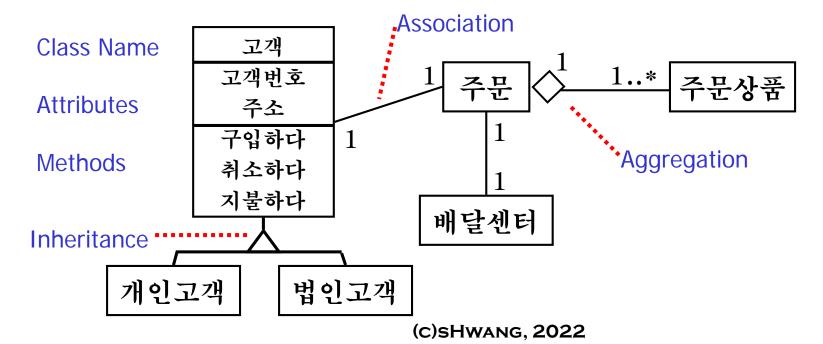
분류	도면	표현대상
静的子조를 표현	Class Diagram Object Diagram	Class Object
動的구조를 표현	Use Case Diagram Statechart Diagram Sequence Diagram Activity Diagram Collaboration Diagram	Class Object Class/Object Object
구현時의 구조 를 표현	Component Diagram Deployment Diagram	

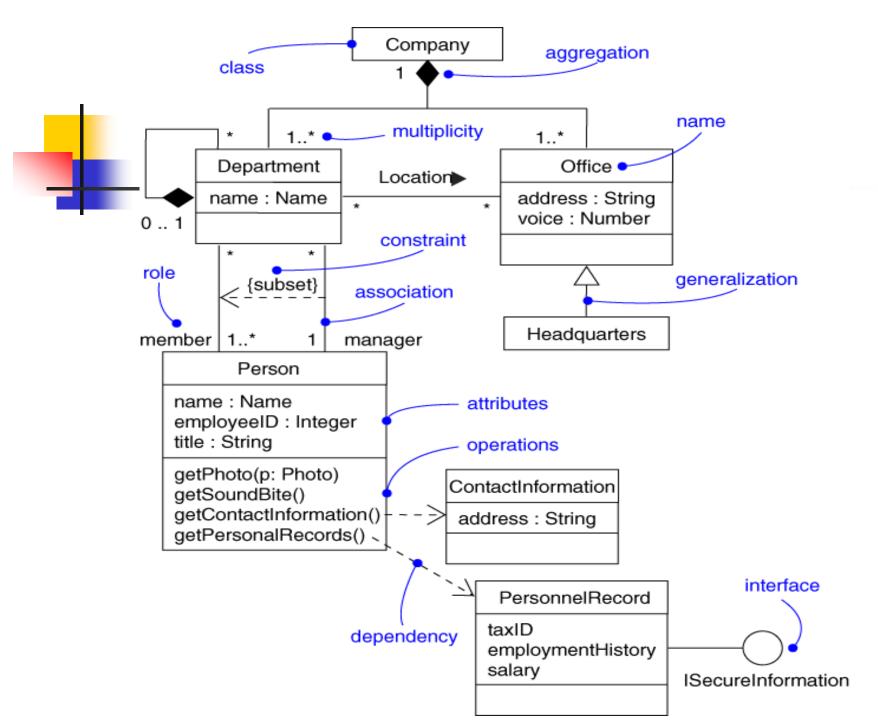
UML 2.x의 Diagrams

Diagram	주요 특징
Object Dgm	어떤 시점에서 시스템 내의 객체들간의 정적인 관계를 표현
Class Dgm	클래스들간의 관계를 토대로, 시스템의 정적인 구조를 표현
Composite Structure Dgm	클래스와 컴포넌트 등의 내부구조를 표현
Component Dgm	컴포넌트(소프트웨어부품)들간의 정적인 구조를 표현
Package Dgm	패키지들간의 의존관계 및 내포관계 등을 표현
Depolyment Dgm	시스템의 물리적인 구성 및 배치를 표현
Activity Dgm	업무의 흐름 및 처리실행순서 등을 표현
UseCase Dgm	유저의 관점에서 바라본 시스템의 기능을 표현
Sequence Dgm	객체들사이의 상호수수작용(메시지송수신)을 시간흐름에 따라서 표현
Communication Dgm	객체들사이의 관계를 토대로 상호수수작용을 표현
StateMachine Dgm	시스템을 구성하는 객체들의 상태천이 및 이에 대응하는 동작을 표현
Timing Dgm	객체가 시간축에 따라서 어떻게 상태천이를 수행하는지를 표현
Interaction Overview Dgm	Interaction Dgm들(Seq. D & Com. D)이 서로 어떤 관계를 가지고 있는 지를 표현

(1)Class Diagram(클래스圖)

- □ 문제영역內 클래스 및 클래스들間의 靜的인 構造를 표현
- □ UML의 중심이 되는 圖面
- □ 각 객체들이 보유한 속성 및 동작을 倂記
- □ ER Diagram의 발전된 형태

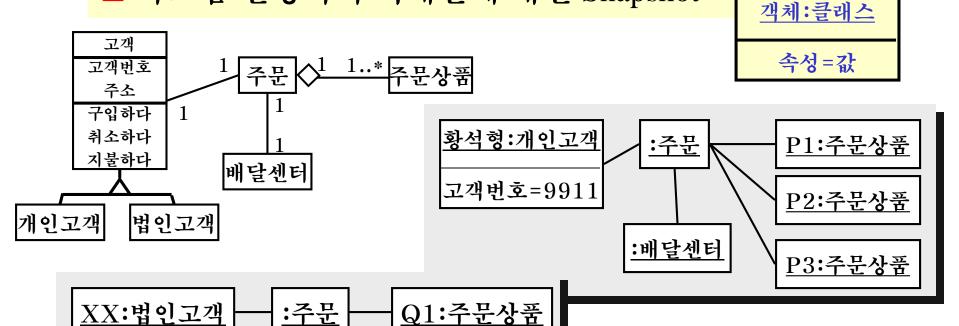




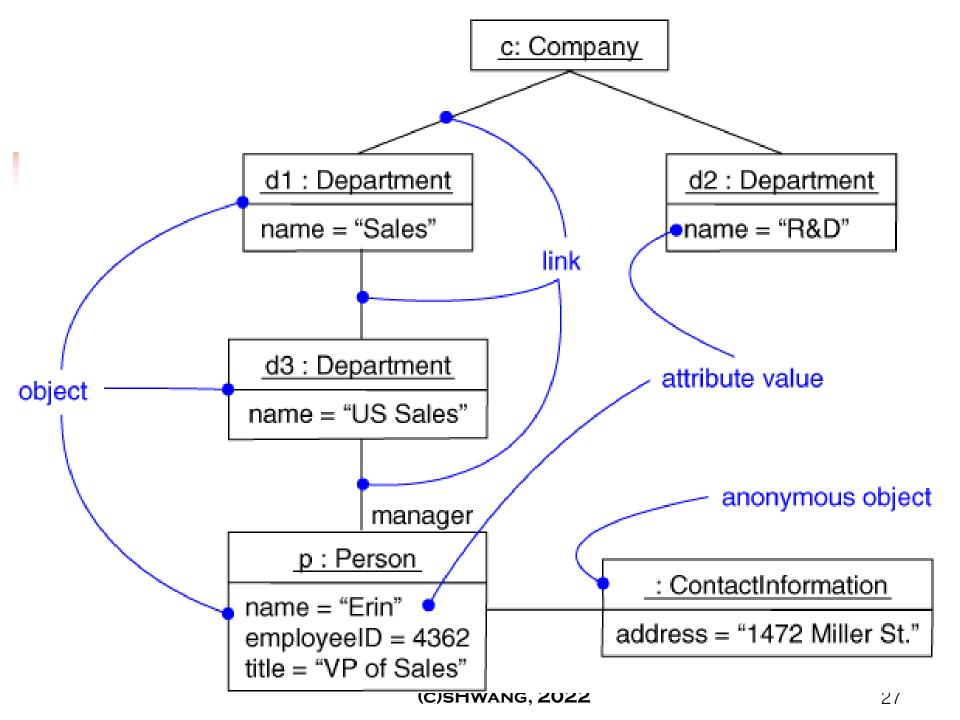
(2)Object Diagram(객체圖)

- □ Class Diagram의 클래스로부터 생성된 객체를 표현
- □ 시스템 실행시의 객체들에 대한 Snapshot

:배달센터



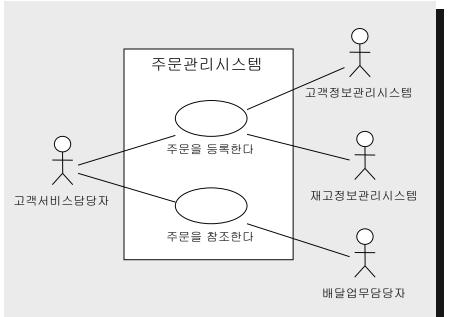
26



(3) Use Case Diagram(쓰임새圖)

- □ 시스템內에 구현해야 되는 기능 및 서비스항목 등의 요구사양 표현
- □ 개발예정의 시스템을 중심에 두고,

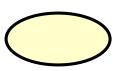
시스템과 이용자(외부시스템 포함) 들 사이의 수수작용표현



Actor : 시스템 이용자(외부시스템 포함)

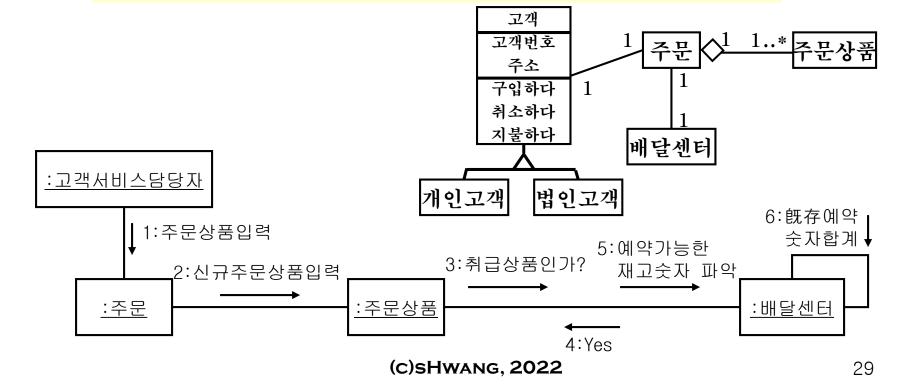


Use Case : 각 Actor들에 대한, 시스템이 제공하는 기능/서비스



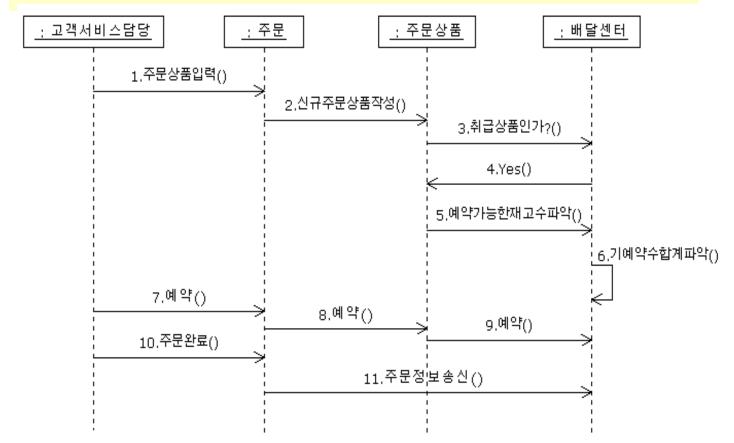


- □ 객체들 간의 상호메시지수수작용을 2차원 평면에 표현
 - □메시지송수신 경로 : 객체들간의 접속Link로 표현
 - □메시지의 흐름 : 각 Link에 번호를 붙여서 표현





- □ 객체들 간에 송수신되는 메시지들의 순서를 표현
- □ 객체들간의 메시지수수작용을 시간적인 관점에서 표현



Interaction Diagram(相互作用圖)

객체들간의 메시지수수작용을 표현



 : 고객서비스담당
 : 주문
 : 주문상품
 : 배달센터

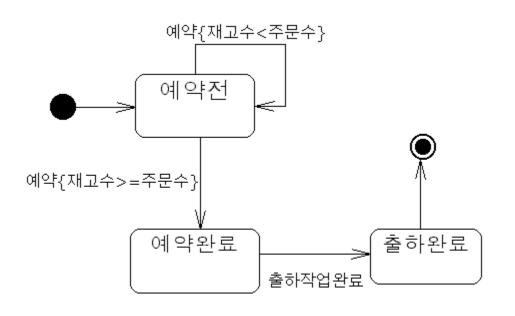
 1.주문상품입력()
 2.신규주문상품작성()

 : 시간의 흐름 重視
 3.취급상품인가?()

 5.예약가능한재고수파악()

(6) Statechart Diagram(狀態圖)

- □ 외부의 자극에 의한, 각 객체들의 내부상태변화를 표현
 - □狀態遷移圖
 - □ 객체의 全Life-Cycle(생성에서부터 소멸까지) 정의가능

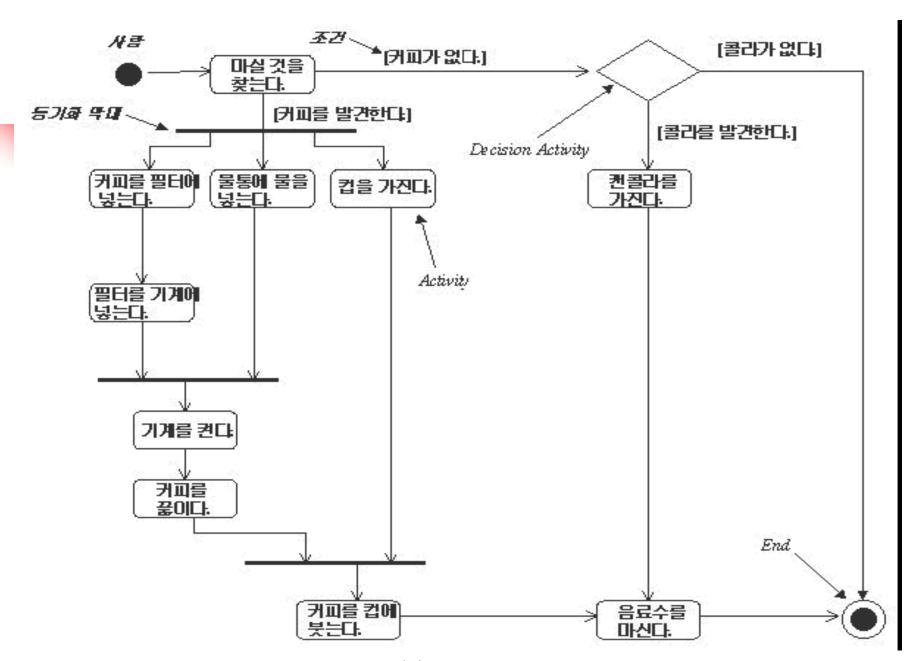


- InitialState
- FinalState

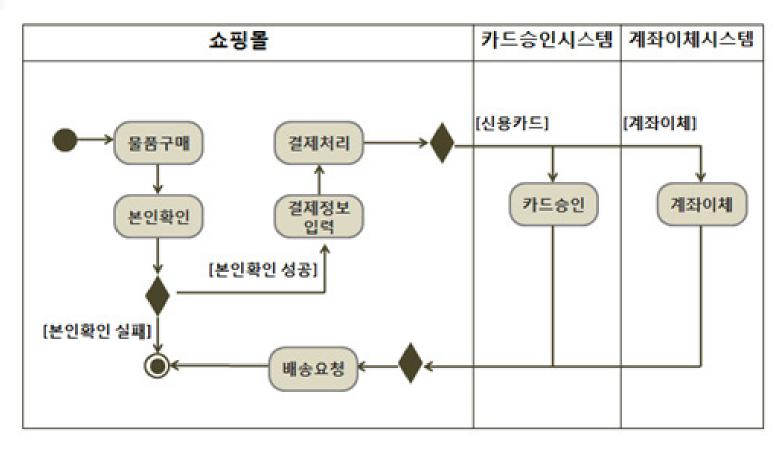
상태



- □ 시스템의 動的인 측면을 구체적으로 표현
 - □ 각 부서의 업무활동의 흐름 및 작업순서,
 - □ 어플리케이션의 각 기능에 대한 동작순서,...
 - □ Data Flow Diagram, Flow Chart 등과 유사









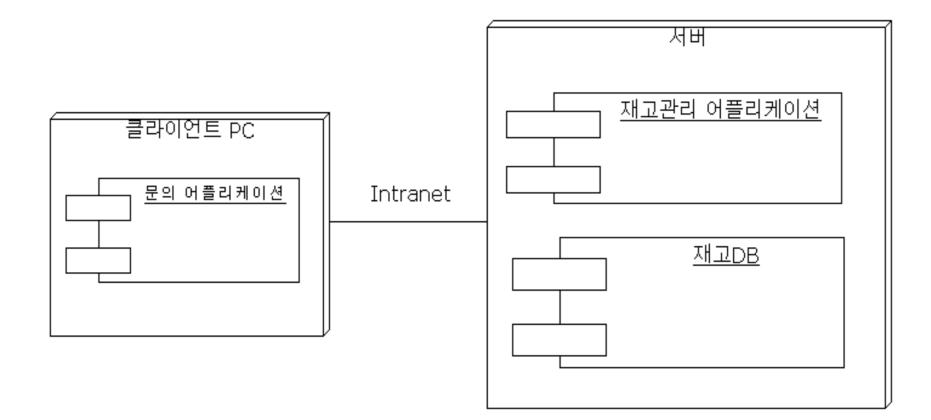
(8) Component Diagram

- □ 소프트웨어의 내부구성을 표현
 - □ 어떠한 부품들로 패키지화되는가?
 - □ 어떠한 인터페이스 제공?
 - □ 부품들사이의 의존관계



(9) Deployment Diagram(配置圖)

- □ 시스템 動作時의 구성을 표현
 - □ 각 컴포넌트에 대한, 컴퓨터 및 네트워크 상의 물리적 배치



요약: UML 1.4의 도면들

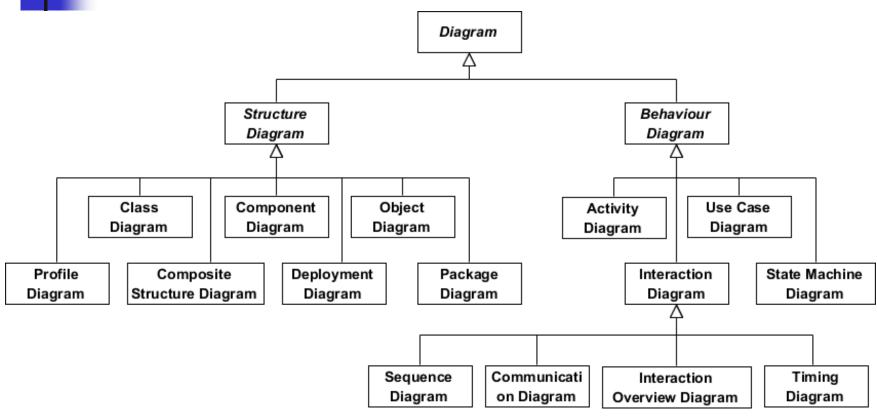
명칭	역할
Use Case Diagram	시스템과 사용자사이의 수수작용(쓰임새)을 표현
Sequence Diagram	객체들사이의 상호수수작용을 시간적 관점에서 표현
Collaboration Diagram	객체들사이의 상호수수작용을 2차원평면에 표현
Class Diagram	클래스와 클래스들 사이의 관계를 표현
Object Diagram	객체와 객체들사이의 관계를 표현
StateChart Diagram	객체의 상태변화를 표현
Activity Diagram	객체들의 행위변화를 표현
Component Diagram	소프트웨어의 내부구성(컴포넌트와 컴포넌트들 사이의 관계 등)을 표현
Deployment Diagram	시스템동작시의 구성(객체/컴포넌트의 물리적 배치상황) 을 표현

UML2.1의 Diagrams

Diagram	주요 특징
Object Dgm	어떤 시점에서 시스템 내의 객체들간의 정적인 관계를 표현
Class Dgm	클래스들간의 관계를 토대로, 시스템의 정적인 구조를 표현
Composite Structure Dgm	클래스와 컴포넌트 등의 내부구조를 표현
Component Dgm	컴포넌트(소프트웨어부품)들간의 정적인 구조를 표현
Package Dgm	패키지들간의 의존관계 및 내포관계 등을 표현
Depolyment Dgm	시스템의 물리적인 구성 및 배치를 표현
Activity Dgm	업무의 흐름 및 처리실행순서 등을 표현
UseCase Dgm	유저의 관점에서 바라본 시스템의 기능을 표현
Sequence Dgm	객체들사이의 상호수수작용(메시지송수신)을 시간흐름에 따라서 표현
Communication Dgm	객체들사이의 관계를 토대로 상호수수작용을 표현
StateMachine Dgm	시스템을 구성하는 객체들의 상태천이 및 이에 대응하는 동작을 표현
Timing Dgm	객체가 시간축에 따라서 어떻게 상태천이를 수행하는지를 표현
Interaction Overview Dgm	Interaction Dgm들(Seq. D & Com. D)이 서로 어떤 관계를 가지고 있는 지를 표현

4

UML 2.5 diagrams



 https://www.visual-paradigm.com/guide/uml-unifiedmodeling-language/what-is-uml/

3. UML의 활용범위

- 인간의 능력(아이디어)을 可視的 手段을 사용하여 용이하 게 導出시켜 표현 가능
- 기획, 설계, 제조, 유통 등, 대부분의 분야에서 사용가능
- 분석/설계 대상영역의 구조적/동적 측면을 전세계 공통적으로 모델화 가능
- UML활용예
 - 아이디어의 정리 및 스케치
 - 비즈니스 모델링
 - 시스템 개발시의 표준설계도면
 - 사용자와 개발자들 사이의 의사소통도구
 - 소프트웨어개발 이외의 영역에서도 유효
 - **...**

