

럼바우가 개발한 OMT와 UML의 차이점은

럼바우가 개발한 **OMT(Object Modeling Technique)**와 UML(Unified Modeling Language)의 주요 차이점은 다음과 같습니다:

1. 정의와 개발 배경

특징	OMT	UML
풀 네임	Object Modeling Technique	Unified Modeling Language
개발 시기	1991년, James Rumbaugh와 그의 팀에 의해 개발됨	1997년, James Rumbaugh, Grady Booch, Ivar Jacobson의 협력으로 Object Management Group(OMG)에서 표준화됨
목적	객체 지향 시스템 분석 및 설계에 초점	객체 지향 소프트웨어 설계 및 시스템 모델링을 위한 표준화된 언어
역할	UML의 전신으로, UML의 여러 요소에 영향을 줌	다양한 객체 지향 모델링 기법을 통합하여 표준화된 모델링 언어로 발전

2. 모델링 구성

특징	OMT	UML
모델링 구성 요소	세 가지 주요 모델: 객체 모델(Object Model), 동적 모델(Dynamic Model), 기능 모델(Functional Model)	다양한 다이어그램(클래스, 유스케이스, 시퀀스, 상태 등)을 포함하여 더 폭넓은 표현 가능
다이어그램 종류	객체 다이어그램, 상태 다이어그램, 데이터 흐름도(DFD)	클래스 다이어그램, 유스케이스 다이어그램, 시퀀스 다이어그램 등 13가지 이상의 표준화된 다이어그램
표현 방식	비교적 간단하고 특정 목적에 맞춰져 있음	더 복잡하고 다양한 시스템 설계 및 분석 요구를 충족하도록 확장

3. 표준화와 적용 범위

특징	OMT	UML
표준화 여부	독립적인 방법론으로 표준화되지 않음	OMG(Object Management Group)에 의해 공식 표준으로 채택
적용 범위	소프트웨어 설계와 분석에 국한됨	소프트웨어 설계뿐만 아니라 비즈니스 프로세스 모델링 등 다양한 도메인에 적용 가능

4. 주요 차별점

- OMT는 객체 지향 설계를 위해 세 가지 주요 모델(객체, 동적, 기능)을 제안하며, 비교적 간단한 구조를 가지고 있습니다^{[1] [2]}.
- UML은 OMT를 포함하여 Booch Method와 OOSE(Object-Oriented Software Engineering)를 통합한 결과물로, 더 많은 다이어그램과 기능을 제공하며 복잡한 시스템의 설계 및 문서화를 지원합니다^{[3] [4]}.

요약

OMT는 UML의 전신으로서 객체 지향 설계의 기본 개념을 제공했으며, UML은 이를 확장하고 표준화하여 더 폭넓은 시스템 모델링 요구를 충족시킨 언어입니다. UML은 OMT보다 더 많은 다이어그램과 기능을 제공하며, 다양한 도메인에서 활용될 수 있는 강력한 도구로 발전했습니다.

✻

1. https://en.wikipedia.org/wiki/Object-modeling_technique
2. [https://cio-wiki.org/wiki/Object_Modeling_Technique_\(OMT\)](https://cio-wiki.org/wiki/Object_Modeling_Technique_(OMT))
3. <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-uml/>
4. <https://www.differencebetween.info/difference-between-uml-and-omt>