

# Phân tích và Thiết kế Hệ thống (*IT3120*)

**Nguyễn Nhật Quang**

*quang.nguyennhat@hust.edu.vn*

---

Trường Đại học Bách Khoa Hà Nội  
Viện Công nghệ thông tin và truyền thông  
Năm học 2020-2021

# Nội dung học phần:

- Giới thiệu về Phân tích và thiết kế hệ thống thông tin hướng đối tượng
- **Giới thiệu về Ngôn ngữ mô hình hóa UML**
- Giới thiệu về Quy trình phát triển phần mềm
- Phân tích môi trường và nhu cầu
- Phân tích chức năng
- Phân tích cấu trúc
- Phân tích hành vi
- Phân tích ứng xử
- Thiết kế kiến trúc tổng thể của hệ thống
- Thiết kế chi tiết lớp
- Thiết kế giao diện sử dụng
- Thiết kế dữ liệu

# Giới thiệu về ngôn ngữ MHH UML

- Lịch sử phát triển của ngôn ngữ MHH UML
- Các góc nhìn của UML
- Các biểu đồ được sử dụng trong UML
- Các công cụ MHH UML miễn phí

# Lịch sử phát triển của UML (1)

- Ngôn ngữ UML (Unified Modeling Language) là một hệ thống ký pháp mô hình hóa hướng đối tượng
- 1975-1990:
  - Có nhiều ngôn ngữ MHH hướng đối tượng được phát triển
- 1990-1994:
  - Hơn 50 phương pháp phát triển hướng đối tượng, trong đó có 3 phương pháp nổi tiếng:
    - OOD - Object Oriented Design (Grady Booch)
    - OOSE - Object Oriented Software Engineering (Ivar Jacobson)
    - OMT - Object Modeling Technique (Jim Rumbaugh)

# Lịch sử phát triển của UML (2)

- 10/1994: Rumbaugh và Booch tiến hành dự án UML ở Rational, xây dựng một phương pháp hợp nhất trên cơ sở hai phương pháp Booch 93 và OMT-2
- 1995: Jacobson gia nhập dự án
- 10/1995: Phác thảo UML, phiên bản 0
- 6/1996: Phiên bản UML 0.9
- 1/1997: IBM và SoftTeam kết hợp với các thành viên => Phiên bản 1.1
- 14/11/1997: UML 1.1 được OMG (Object Management Group) công nhận là chuẩn
- 6/1998: UML 1.2
- 10/1998: UML 1.3
- 5/2001: UML 1.4
- 6/2003: UML 2.0

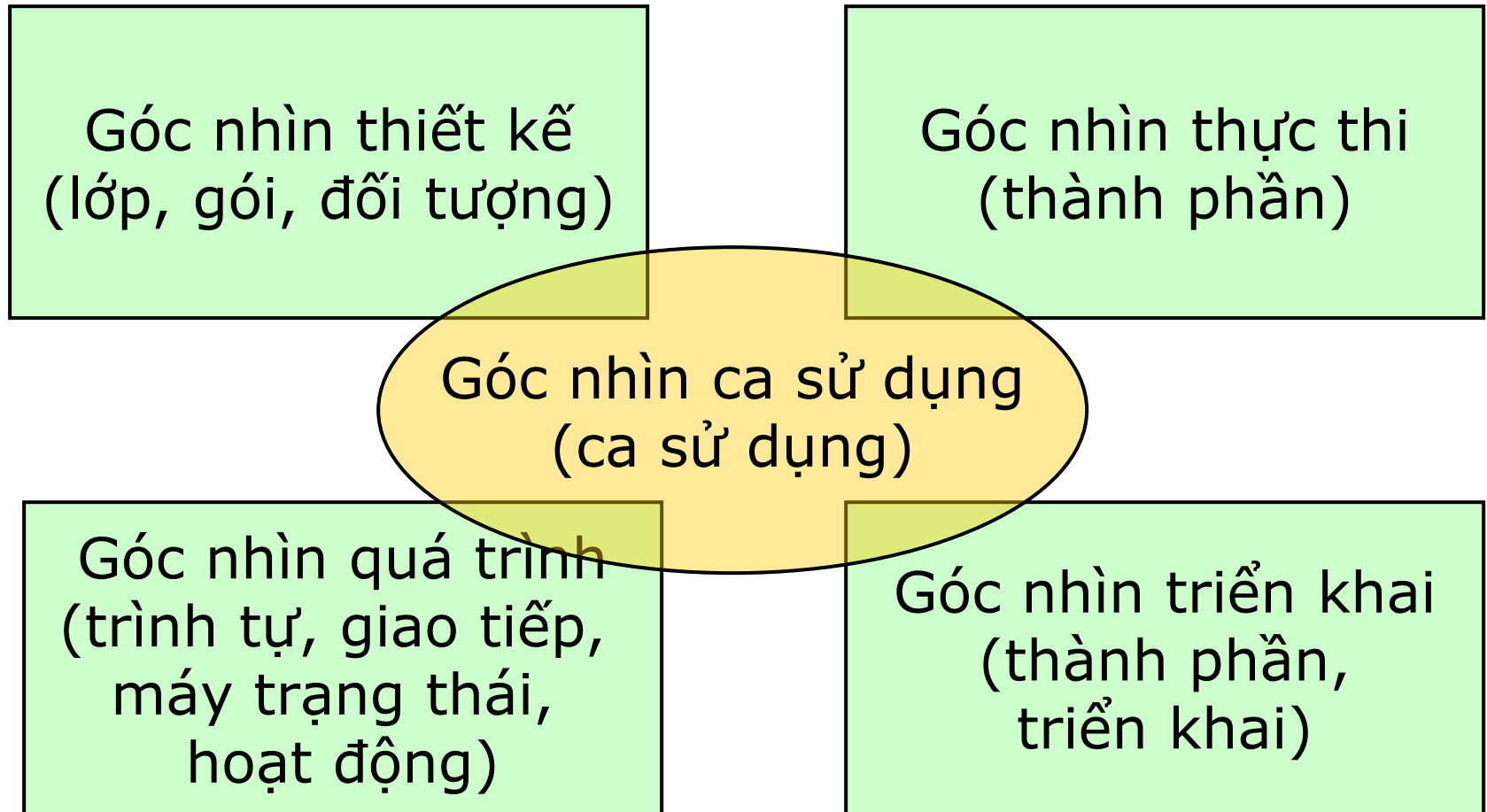
# UML - Một ngôn ngữ MHH

- UML là ngôn ngữ dùng để:
  - Mô hình hóa trực quan (Visualizing)
  - Đặc tả (Specifying)
  - Xây dựng (Constructing)
  - Làm tài liệu (Documenting)
- Có thể sử dụng trong bất kỳ tiến trình phát triển hệ thống
- Xuyên suốt vòng đời phát triển hệ thống
- Được sử dụng bởi các công nghệ cài đặt khác nhau

# Các góc nhìn của UML (1)

- UML cung cấp các mô hình để diễn tả hệ thống
- Mỗi mô hình chỉ có thể diễn tả hệ thống theo một góc nhìn (view) nhất định
- UML cung cấp 5 góc nhìn đối với hệ thống
- Mỗi góc nhìn thực hiện bởi một số biểu đồ (mô hình)
- Có thể có biểu đồ thuộc vào nhiều ( $>1$ ) góc nhìn khác nhau

# Các góc nhìn của UML (2)





# Các góc nhìn của UML (3)

- **Góc nhìn ca sử dụng (Use case view)**
  - Là góc nhìn từ ngoài nhìn vào hệ thống
  - Là cách nhìn của *người dùng cuối, người phân tích, người kiểm thử*
  - Không phản ánh tổ chức bên trong, mà chỉ làm rõ các chức năng chính/quan trọng mà hệ thống phải đáp ứng cho người dùng
  - **Sắc thái tĩnh: Biểu đồ ca sử dụng (Use case diagram)**
  - **Sắc thái động: Biểu đồ giao tiếp (Communication diagram), Biểu đồ máy trạng thái (State diagram), và Biểu đồ hoạt động (Activity diagram)**

# Các góc nhìn của UML (4)

## ■ Góc nhìn thiết kế (Design view)

- ❑ Còn được gọi là góc nhìn logic (Logical view)
- ❑ Là góc nhìn vào bên trong (cấu trúc) hệ thống, cho thấy các nhiệm vụ của hệ thống
- ❑ Là cách nhìn của *người thiết kế hệ thống*
- ❑ Sắc thái tĩnh: Biểu đồ lớp (Class diagram), Biểu đồ đối tượng (Object diagram)
- ❑ Sắc thái động: Biểu đồ giao tiếp (Communication diagram), Biểu đồ máy trạng thái (State diagram), Biểu đồ hoạt động (Activity diagram)

# Các góc nhìn của UML (5)

## ■ Góc nhìn quá trình (Process view)

- Còn được gọi là góc nhìn song hành
- Phản ánh các quá trình điều khiển, các quá trình thực hiện, cho thấy sự hoạt động đồng bộ của hệ thống
- Được thể hiện (sử dụng) với các biểu đồ như trong Góc nhìn thiết kế, tập trung vào các lớp chủ động
  - Lớp chủ động: Lớp biểu diễn cho các quá trình điều khiển và quá trình thực hiện

# Các góc nhìn của UML (6)

## ■ Góc nhìn thực thi (Implementation view)

- ❑ Còn được gọi là góc nhìn thành phần (Component view)
- ❑ Là góc nhìn đối với dạng phát hành của phần mềm
- ❑ Cho thấy các thành phần và tập tin tương đối độc lập, có thể lắp ráp để hệ thống chạy được
- ❑ Sắc thái tĩnh: Biểu đồ thành phần (Component diagram)
- ❑ Sắc thái động: Biểu đồ giao tiếp (Communication diagram), Biểu đồ máy trạng thái (State diagram), Biểu đồ hoạt động (Activity diagram)

# Các góc nhìn của UML (7)

## ■ Góc nhìn triển khai (Deployment view)

- ❑ Là góc nhìn về kiến trúc phần cứng và nền tảng hạ tầng mà trên đó hệ thống được triển khai
- ❑ Chỉ rõ sự phân bố, sắp đặt các thành phần của hệ thống trên các đơn vị phần cứng và nền tảng hạ tầng
- ❑ Sắc thái tĩnh: Biểu đồ triển khai (Deployment diagram)
- ❑ Sắc thái động: Biểu đồ giao tiếp (Communication diagram), Biểu đồ máy trạng thái (State diagram), Biểu đồ hoạt động (Activity diagram)

# Các góc nhìn của UML (8)

- Mỗi vai trò trong tiến trình phát triển hệ thống (vd: phân tích, thiết kế, tích hợp, kiểm định, người dùng cuối,...) thường chỉ quan tâm tới một góc nhìn nào đó của hệ thống
- 5 góc nhìn có sự liên hệ và bổ trợ lẫn nhau
- Góc nhìn ca sử dụng (Use case view) có ảnh hưởng (liên quan) đến 4 góc nhìn còn lại

# Các biểu đồ của UML 2.0

## ■ Các biểu đồ về cấu trúc:

- ⇒ *Biểu đồ lớp (Class diagram)*
- ⇒ *Biểu đồ đối tượng (Object diagram)*
- ⇒ *Biểu đồ triển khai (Deployment diagram)*
- ⇒ *Biểu đồ gói (Package diagram)*
- ⇒ *Biểu đồ thành phần (Component diagram)*
- ⇒ *Biểu đồ cấu trúc đa hợp (Composite structure diagram)*

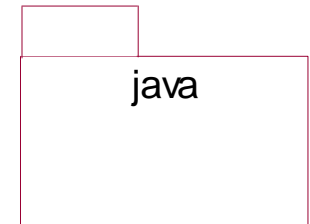
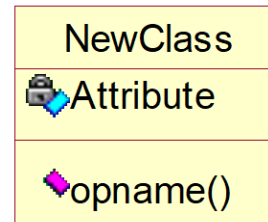
## ■ Các biểu đồ về hành vi:

- ⇒ *Biểu đồ ca sử dụng (Use case diagram)*
- ⇒ *Biểu đồ hoạt động (Activity diagram)*
- ⇒ *Biểu đồ trình tự (Sequence diagram)*
- ⇒ *Biểu đồ giao tiếp (Communication diagram),*
- ⇒ *Biểu đồ máy trạng thái (State diagram)*
- ⇒ *Biểu đồ thời gian (Timing diagram)*
- ⇒ *Biểu đồ tổng quan tương tác (Interaction overview diagram)*

# Các phần tử của biểu đồ

## ■ Các nút (node)

- Các yếu tố của mô hình (biểu đồ)
- Có dạng đồ họa 2 chiều
- Vd: Lớp, Gói,...



## ■ Các đường (path)

- Các yếu tố của mô hình (biểu đồ)
- Có dạng đồ họa tuyến tính
- Vd: Liên kết, Phụ thuộc, Khái quát





# Bổ sung ý nghĩa cho các biểu đồ

- Đặc tả (specification)
  - Một phát biểu dạng văn bản (textual statement) về cú pháp và ngữ nghĩa
- Tô điểm (adornment)
  - Các vai trò, các cơ số, hạn định (giới hạn), đường viền đậm nét,...
- Khuôn dập (stereotype)
  - Chuỗi ký tự, được đặt trong ngoặc kép
- Tính chất (property) và giá trị gán nhãn (tagged value)
  - Đưa thêm thông tin cho các phần tử mô hình
  - Vd: {nhãn=giá trị}, {nhãn\_boolean}
- Ràng buộc (constraint)
  - Thêm các điều kiện/hạn chế đối với một yếu tố của mô hình

# Mô hình hóa với UML

- Mô hình hóa hệ thống theo nhiều góc nhìn
  - Có thể sử dụng 5 góc nhìn đối với hệ thống
  - Tùy vào hệ thống nhỏ/lớn, đơn giản/phức tạp => Quyết định mô tả hệ thống theo những góc nhìn phù hợp
- Mô hình hóa hệ thống theo nhiều mức độ trừu tượng hóa khác nhau (different abstraction levels)
  - Tùy thuộc vào giai đoạn (của tiến trình phát triển hệ thống) và nhu cầu sử dụng
  - Có thể ở mức khái quát (logical/overview level) hoặc ở mức chi tiết (detailed level)

# Các công cụ MHH UML miễn phí (1)

- **ArgoUML** (<http://argouml.tigris.org/>)
  - Từ mô hình sinh mã nguồn (diagram-to-codes generation) cho các ngôn ngữ: C++, C#, Java, PHP4, PHP5, Ruby
  - Từ mã nguồn sinh ngược lại mô hình (reverse engineering) cho các ngôn ngữ: Java
- **BOUML** (<http://www.bouml.fr/>)
  - Từ mô hình sinh mã nguồn cho các ngôn ngữ: C++, Java, PHP, IDL, Python, MySQL
  - Từ mã nguồn sinh ngược lại mô hình cho các ngôn ngữ: C++, Java, PHP, MySQL
- **NClass** (<http://nclass.sourceforge.net/>)
  - Từ mô hình sinh mã nguồn cho các ngôn ngữ: C#, Java
  - Từ mã nguồn sinh ngược lại mô hình cho các ngôn ngữ: C#, Java
- **Umbrello UML Modeller** (<https://umbrello.kde.org/>)
  - Từ mô hình sinh mã nguồn cho các ngôn ngữ: C++, Java, Perl, PHP, Python
  - Từ mã nguồn sinh ngược lại mô hình cho các ngôn ngữ: C++, IDL, Pascal/Delphi, Ada, Python, Java

# Các công cụ MHH UML miễn phí (2)

- **WhiteStarUML** (<https://sourceforge.net/projects/whitestaruml/>)
  - Từ mô hình sinh mã nguồn cho các ngôn ngữ: Java, C#, C++, SQL
  - Từ mã nguồn sinh ngược lại mô hình cho các ngôn ngữ: Java, C#, C++, SQL
- **Open ModelSphere** (<http://www.modelsphere.com/org/>)
  - Từ mô hình sinh mã nguồn cho các ngôn ngữ: Java, SQL
  - Từ mã nguồn sinh ngược lại mô hình cho các ngôn ngữ: Java
- **Modelio** (<https://www.modelio.org/>)
  - Từ mô hình sinh mã nguồn cho các ngôn ngữ: Java
  - Từ mã nguồn sinh ngược lại mô hình cho các ngôn ngữ: Java
- **Dia** (<https://wiki.gnome.org/Apps/Dia/>)
  - Từ mô hình sinh mã nguồn cho các ngôn ngữ: Python, C++, JavaScript, Pascal, Java, PHP
- **Papyrus** (<http://www.eclipse.org/papyrus/>)
  - Từ mô hình sinh mã nguồn cho các ngôn ngữ: C/C++, Java
- **Acceleo** (<https://www.eclipse.org/acceleo/>)
  - Từ mô hình sinh mã nguồn cho các ngôn ngữ: JEE, .Net, Php, ...

# Các công cụ MHH UML miễn phí (3)

- **UML Designer** (<http://www.uml designer.org/>)
- **yEd** (<http://www.yworks.com/>)
- **UMLet** (<http://www.umlet.com/>)
- **Violet** (<http://alex dp.free.fr/violetuml editor/page.php>)
- **PlantUML** (<http://plantuml.com/>)
- **Astah** (<http://astah.net/editions/community>)
- **Visual Paradigm** (<https://www.visual-paradigm.com/solution/freeuml tool/>)
- **MetaUML** (<https://github.com/ogheorghies/MetaUML/wiki>)
- **TinyUML** (<https://sourceforge.net/projects/tinyuml/>)
- **Eclipse UML2 Tools** (<https://www.eclipse.org/>)
- **NetBeans IDE** (<https://netbeans.org/>)
- **Oracle JDeveloper** (<http://www.oracle.com/technetwork/developer-tools/jdev/overview/index.html>)

# Các công cụ thiết kế GUI miễn phí

- **Pencil** (<https://pencil.evolus.vn/>)
- **Wireframe** (<https://wireframe.cc/>)
- **InVision** (<https://www.invisionapp.com/studio>)
- **Adobe XD** (<https://www.adobe.com/in/products/xd.html>)
- **Lunacy** (<https://icons8.com/lunacy>)
- **Fluid UI** (<https://www.fluidui.com/>)
- **Sketch UX Kit** (<http://www.mikolajdobrucki.com/sketchuxkit/>)