BỘ MÔN CÔNG NGHỆ PHẦN MỀM
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

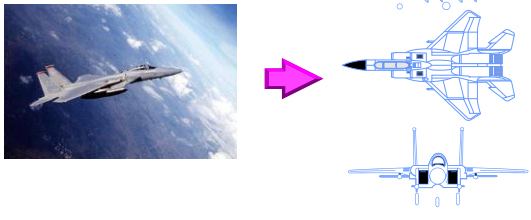
Bài 09. Tổng quan về UML và PTTK HĐT

Nội dung

1. Tổng quan về UML
2. Phân tích thiết kế hướng đối tượng
3. Công cụ phát triển OOAD

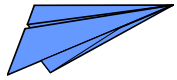
1.1. Mô hình hóa là gì?

- Giúp đơn giản hóa thế giới thực bằng các mô hình
- Giúp hiểu rõ hơn về hệ thống dưới một góc nhìn nào đó



Sự quan trọng của mô hình hóa

Mức độ quan trọng thấp → Mức độ quan trọng cao hơn



Máy bay giấy

Máy bay phản lực

Đội dự án thường không mô hình hóa

- Rất nhiều đội dự án tiến hành xây dựng ứng dụng theo hướng tiếp cận của việc gấp máy bay giấy.
  - Bắt đầu lập trình ngay khi có được yêu cầu.
  - Mất rất nhiều thời gian và tạo ra rất nhiều mã nguồn.
  - Không có bất kỳ một kiến trúc nào.
  - Phải chịu khổ với những lỗi phát sinh.
- **Mô hình hóa là một con đường dẫn đến thành công của dự án.**

1.2. UML là gì?

- Ngôn ngữ mô hình hóa thống nhất UML (Unified Modeling Language)
- UML là ngôn ngữ để:
  - trực quan hóa (visualizing)
  - xác định rõ (đặc tả - Specifying)
  - xây dựng (constructing)
  - tài liệu hóa (documenting)

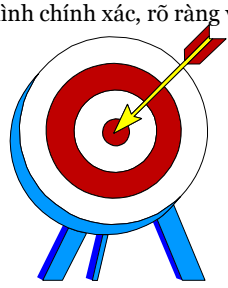
  các cấu phần (artifact) của một hệ thống phần mềm

UML là ngôn ngữ trực quan

- UML là ngôn ngữ thống nhất trực quan giúp công việc được xử lý nhất quán, giảm thiểu lỗi xảy ra
  - Có những thứ mà nếu không mô hình hóa thì không hoặc khó có thể hiểu được
  - Mô hình trợ giúp hiệu quả trong việc liên lạc, trao đổi
    - Trong tổ chức
    - Bên ngoài tổ chức

UML là ngôn ngữ để đặc tả

- UML xây dựng các mô hình chính xác, rõ ràng và đầy đủ.

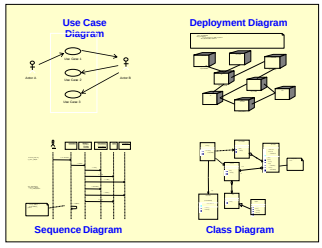


UML là ngôn ngữ để xây dựng HT

- Các mô hình UML có thể kết nối trực tiếp với rất nhiều ngôn ngữ lập trình.
  - Ánh xạ sang Java, C++, Visual Basic…
  - Các bảng trong RDBMS hoặc kho lưu trữ trong OODBMS
  - Cho phép các kỹ nghệ xuôi (chuyển UML thành mã nguồn)
  - Cho phép kỹ nghệ ngược (xây dựng mô hình hệ thống từ mã nguồn)

UML là ngôn ngữ để tài liệu hóa

- UML giúp tài liệu hóa về kiến trúc, yêu cầu, kiểm thử, lập kế hoạch dự án, và quản lý việc bàn giao phần mềm
- Các biểu đồ khác nhau, các ghi chú, ràng buộc được đặc tả trong tài liệu



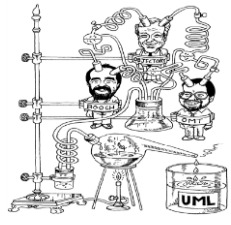
1.3. Lịch sử phát triển của UML

- Vào 1994, có hơn 50 phương pháp mô hình hóa hướng đối tượng:
  - Fusion, Shlaer-Mellor, ROOM, Class-Relation,Wirfs-Brock, Coad-Yourdon, MOSES, Syntropy, BOOM, OOSD, OSA, BON, Catalysis, COMMA, HOOD, Ooram, DOORS ...
- "Meta-models" tương đồng với nhau
- Các ký pháp đồ họa khác nhau
- Quy trình khác nhau hoặc không rõ ràng

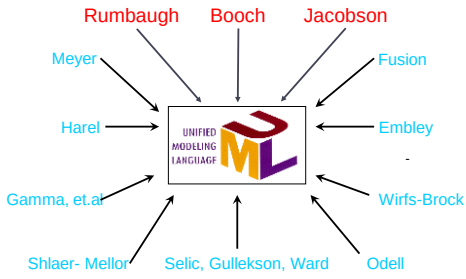
→ Cần chuẩn hóa và thống nhất các phương pháp

1.3. Lịch sử phát triển của UML (2)

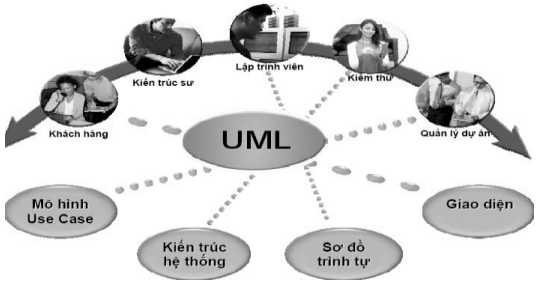
- UML được 3 chuyên gia hướng đối tượng hợp nhất các kỹ thuật của họ vào năm 1994:
  - Booch91 (Grady Booch): Conception, Architecture
  - OOSE (Ivar Jacobson): Use cases
  - OMT (Jim Rumbaugh): Analysis
- Thiết lập một phương thức thống nhất để xây dựng và "vẽ" ra các yêu cầu và thiết kế hướng đối tượng trong quá trình PTTK phần mềm → UML được công nhận là chuẩn chung vào năm 1997.



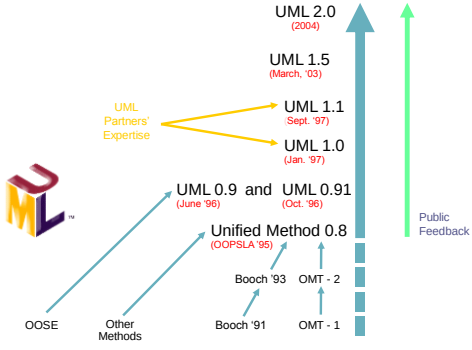
UML là một ngôn ngữ hợp nhất



UML là một ngôn ngữ thống nhất

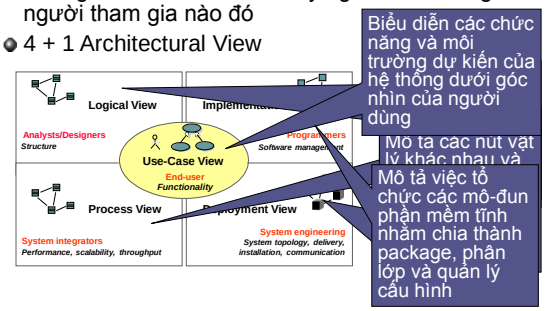


1.3. Lịch sử phát triển của UML (3)



1.4. Các khung nhìn của UML

- Khung nhìn của mô hình có ý nghĩa với những người tham gia nào đó
- 4 + 1 Architectural View

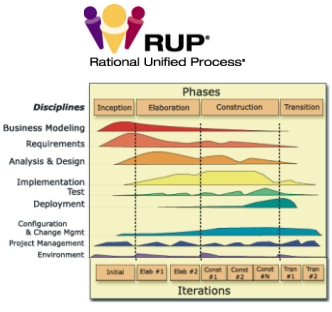


Các biểu đồ UML

- Biểu đồ use case (Use Case Diagram)
- Biểu đồ hoạt động (Activity Diagram)
- Biểu đồ tương tác (Interaction Diagrams)
  - Biểu đồ trình tự (Sequence Diagram)
  - Biểu đồ giao tiếp/cộng tác (Communication/Collaboration Diagram)
- Biểu đồ trạng thái (Statechart Diagram)
- Biểu đồ cấu trúc tĩnh (Static Structure Diagrams)
  - Biểu đồ lớp (Class Diagram)
  - Biểu đồ đối tượng (Object Diagram)
- Biểu đồ thực thi (Implementation Diagrams)
  - Biểu đồ thành phần (Component Diagram)
  - Biểu đồ triển khai (Deployment Diagram)

Quy trình và UML

- UML là ký pháp chứ không phải là phương pháp
  - UML có thể áp dụng cho tất cả các pha của quy trình phát triển phần mềm
  - "Rational Unified Process" - quy trình phát triển cho UML



Nội dung

1. Tổng quan về UML
2. Phân tích thiết kế hướng đối tượng
3. Công cụ phát triển OOAD

2.1. Tầm quan trọng của OOAD

- Nhiều người phát triển dự án
  - Cho rằng phần mềm chủ yếu được xây dựng bằng cách gõ "code" từ bàn phím
  - Không dành đủ thời gian cho quá trình phân tích và thiết kế phần mềm
- → Họ phải "cày bừa" để hoàn thành chương trình vì
  - Không hiểu hoặc hiểu sai yêu cầu
  - Giao tiếp với các thành viên không tốt
  - Không tích hợp được với module của đồng nghiệp…
- → Họ nhận ra rằng "Phân tích" và "Thiết kế" cần được coi trọng hơn, nhưng đã quá muộn

2.1. Tầm quan trọng của OOAD (2)

- Cần thiết lập một cơ chế hiệu quả để nắm bắt yêu cầu, phân tích thiết kế
- Cơ chế này phải như là một "ngôn ngữ thống nhất" giúp cho quá trình hợp tác hiệu quả giữa các thành viên trong nhóm phát triển phần mềm.
- → OOAD

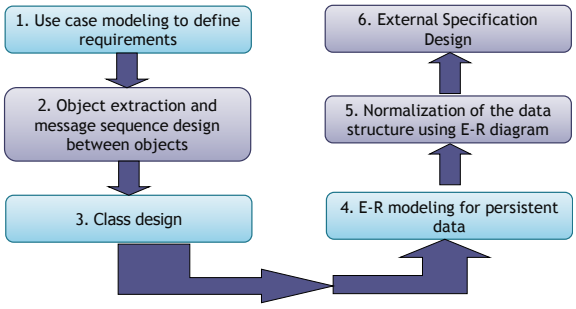
2.2. Mục đích của OOAD

- Chuyển các yêu cầu của bài toán thành một bản thiết kế của hệ thống sẽ được xây dựng
- Tập trung vào quá trình phân tích các YÊU CẦU của hệ thống và thiết kế các MÔ HÌNH cho hệ thống đó trước giai đoạn lập trình
- Được thực hiện nhằm đảm bảo mục đích và yêu cầu của hệ thống được ghi lại một cách hợp lý trước khi hệ thống được xây dựng
- Cung cấp cho người dùng, khách hàng, kỹ sư phân tích, thiết kế nhiều cái nhìn khác nhau về cùng một hệ thống

2.3. Phương pháp OOAD

- OOAD được chia thành 2 giai đoạn
  - Phân tích hướng đối tượng (OOA)
  - Thiết kế hướng đối tượng (OOD)
- OOA là giai đoạn nhằm tạo ra các mô hình cơ bản (mô hình khái niệm) của hệ thống dựa theo những gì khách hàng yêu cầu về hệ thống của họ
- OOD sẽ bổ sung thêm các thông tin thiết kế chi tiết cho các mô hình nói trên

2.3. Phương pháp OOAD (2)



25

2.4. Công cụ UML – OOAD

- Công cụ mã nguồn mở:
  - EclipseUML
  - UmlDesigner
  - ArgoUML...
- Công cụ thương mại:
  - **Enterprise Architect**
  - IBM Rational Software Architect
  - Microsoft Visio
  - Visual Paradigm for UML
  - SmartDraw...

26

Case study: Hệ thống đăng ký học đơn giản

- Hệ thống quản lý việc đăng ký học của sinh viên trong trường đại học theo từng học kỳ của năm học.
- Người dùng phải đăng nhập để sử dụng hệ thống qua Internet. Người quản trị có thể quản lý người dùng trong hệ thống bao gồm: tìm kiếm, thêm mới, cập nhật và xóa bỏ thông tin.
- Người quản lý khóa học thông qua giao diện đồ họa của hệ thống có thể thêm, xóa và cập nhật các thông tin cơ bản của khóa học cho học kỳ tiếp theo như:
  - Mã khóa học* (có dạng XX9999 – trong đó XX là viết tắt của khoa và 9999 là 4 chữ số)

27

Case study: Hệ thống đăng ký khóa học

- Tên khóa học*
- Ngày bắt đầu, ngày kết thúc
- Mô tả
- Mục tiêu
- Giảng viên
- Các môn học tiên quyết,...
- Giảng viên có thể cập nhật thông tin các khóa học được phân công qua giao diện dành cho giảng viên trên hệ thống.
- Vào đầu mỗi học kỳ, người quản lý khóa học sẽ mở chức năng đăng ký cho sinh viên.

28

Case study: Hệ thống đăng ký khóa học

- Sau khi hết thời gian đăng ký chức năng này sẽ bị đóng lại và sinh viên không thể đăng ký học được nữa.
- Chức năng đăng ký sẽ hiển thị danh sách các khóa học cho học kỳ đó và cho sinh viên lựa chọn. Sinh viên cũng có thể hủy bỏ từ danh sách khóa học đã đăng ký.
- Mỗi khóa học có tối đa 30 và tối thiểu là 3 sinh viên đăng ký.
  - Khóa học đã đầy thì không thể đăng ký thêm
  - Khóa học có ít hơn 3 sinh viên đăng ký sẽ bị hủy bỏ

29

Case study: Hệ thống đăng ký khóa học

- Khi có một khóa học bị hủy bỏ thì hệ thống cần thông báo cho các sinh viên đã đăng ký khóa học đó
- Sinh viên không thể đăng ký một khóa học khi chưa hoàn thành các điều kiện tiên quyết của nó.
- Sau khi kết thúc thời gian đăng ký:
  - Người quản lý khóa học có thể xem tất cả các danh sách đăng ký
  - Giảng viên chỉ có thể xem danh sách đăng ký khóa học do mình phụ trách
- Người dùng thông thường có thể thực hiện tìm kiếm và xem thông tin chi tiết về các khóa học.

30

Case study: Hệ thống đăng ký khóa học

- Các yêu cầu khác:
  - Hệ thống có thể hỗ trợ tối đa 500 người dùng đồng thời, mỗi thao tác phản hồi kết quả trong không quá 10 giây
  - Hệ thống có các tài liệu và video hướng dẫn cho người mới sử dụng.