ĐẠI HỌC BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

Nhóm chuyên môn Nhập môn Công nghệ phần mềm

NHẬP MÔN CÔNG NGHỆ PHẦN MỀM

Tổng quan về mô hình hóa trong phát triển phần mềm

NỘI DUNG

1. Thế nào là mô hình hóa?
2. Tại sao cần mô hình hóa?
3. Ngôn ngữ mô hình hóa

   Unified Modelling Language (UML)

MỤC TIÊU

Sau bài học này, người học có thể:

1. Nắm được khái niệm mô hình hóa hệ thống
2. Hiểu được vai trò và mức độ quan trọng của mô hình hóa trong phát triển phần mềm
3. Hiểu được vai trò của ngôn ngữ mô hình hóa UML trong phát triển phần mềm

NỘI DUNG TIẾP THEO

1. Thế nào là mô hình hoá (modelling)?

1.1. Hệ thống phức tạp (complex system)

1.2. Thế nào là mô hình

2. Tại sao cần mô hình hóa?

3. Ngôn ngữ mô hình hóa Unified Modelling Language (UML)

1. THẾ NÀO LÀ MÔ HÌNH HÓA (Modelling)?

1.1. Hệ thống phức tạp (Complex System)

- Làm thế nào để có thể mô hình các hệ thống phức tạp trong thực tế?
  - **Ví dụ:** mô hình hoá 1 máy bay chiến đấu

Hình 1.1. Ví dụ về một hệ thống phức tạp

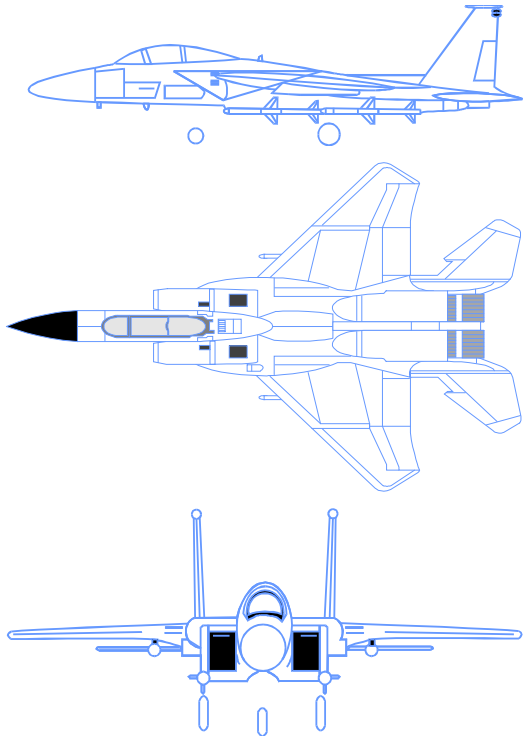
  - Nếu chỉ đơn giản quan sát một vật thể như vậy làm sao để biết cách tạo ra chúng, cấu trúc bên trong, cách thức vận hành?

Nguồn: https://in3dmau.com/mo-hinh-may-bay-su30-in3d/

1. THẾ NÀO LÀ MÔ HÌNH HÓA (Modelling)?

1.2. Thế nào là mô hình

Một mô hình là một sự đơn giản hoá thế giới thực



Hình 1.2. Cấu trúc hệ thống dưới 3 góc nhìn

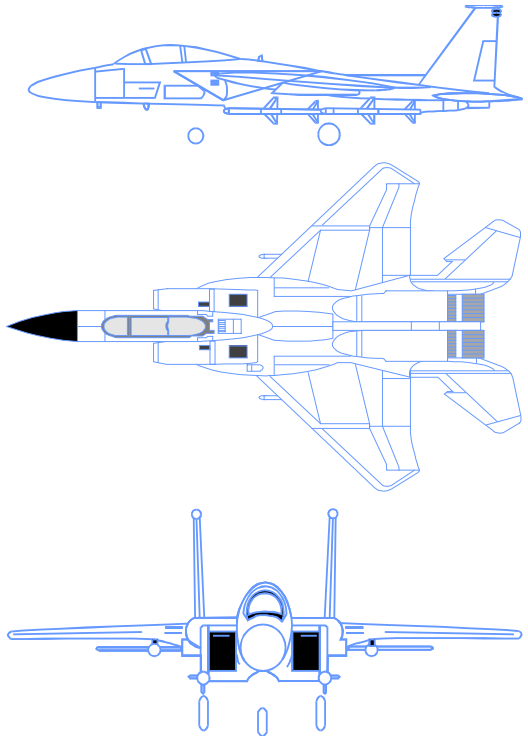
Hình 1.1. Ví dụ về một hệ thống phức tạp

Nguồn: https://in3dmau.com/mo-hinh-may-bay-su30-in3d/

1. THẾ NÀO LÀ MÔ HÌNH HÓA (Modelling)?

1.2. Thế nào là mô hình

Không thể hiểu rõ về hệ thống với chỉ một góc nhìn



Hình 1.1. Ví dụ về một hệ thống phức tạp

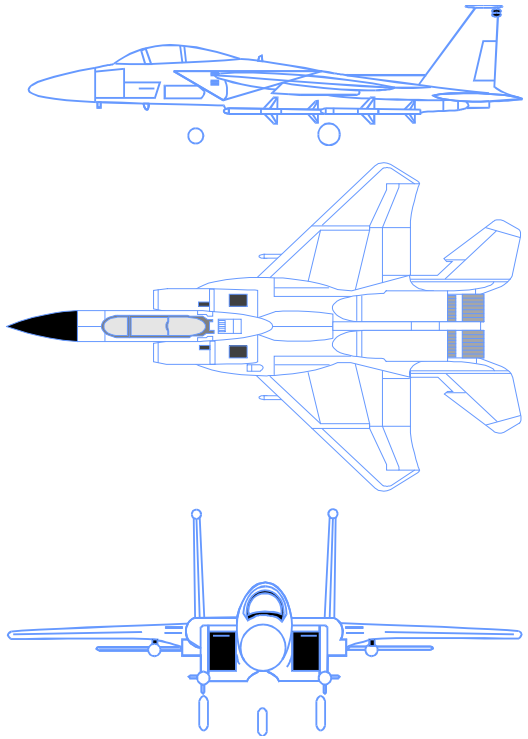
Hình 1.2. Cấu trúc hệ thống dưới 3 góc nhìn

Nguồn: https://in3dmau.com/mo-hinh-may-bay-su30-in3d/

1. THẾ NÀO LÀ MÔ HÌNH HÓA (Modelling)?

1.2. Thế nào là mô hình

Xây dựng các mô hình để hiểu rõ hơn
hệ thống mà chúng ta muốn xây dựng



Hình 1.1. Ví dụ về một hệ thống phức tạp

Hình 1.2. Cấu trúc hệ thống dưới 3 góc nhìn

Nguồn: https://in3dmau.com/mo-hinh-may-bay-su30-in3d/

NỘI DUNG TIẾP THEO

1. Thế nào là mô hình hóa (modelling)?

2. Tại sao cần mô hình hóa?

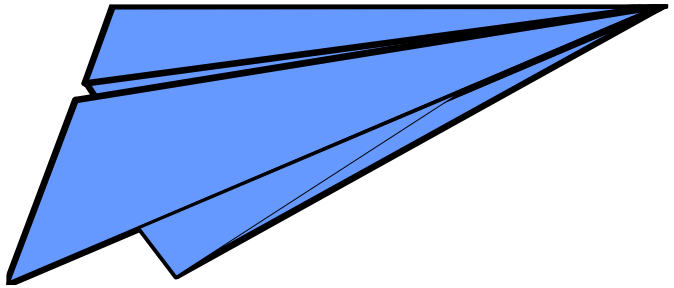
- 2.1. Thảo luận
- 2.2. Mục đích của mô hình hóa
- 2.3. Mô hình được coi như một bản vẽ thiết kế (blueprint)
- 2.4. Những nguyên lý khi mô hình hóa

3. Ngôn ngữ mô hình hóa Unified Modelling Language (UML)

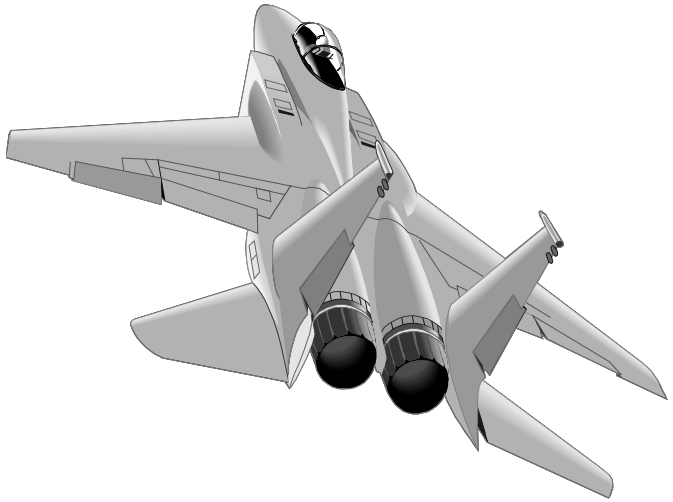
2. TẠI SAO CẦN MÔ HÌNH HÓA?

2.1. Thảo luận

- Tầm quan trọng của mô hình hóa còn phụ thuộc vào mức độ phức tạp của các hệ thống chúng ta cần xây dựng.
- **Ví dụ 1:** Làm thế nào để xây dựng 1 cái máy bay giấy? (Hình 1.3)
  - Làm thế nào nếu máy bay giấy không thể bay?
- **Ví dụ 2:** Làm thế nào để xây dựng 1 máy bay chiến đấu? (Hình 1.4)



Hình 1.3. Ví dụ về một hệ thống đơn giản



Hình 1.4. Ví dụ về một hệ thống phức tạp

Nguồn: https://in3dmau.com/mo-hinh-may-bay-su30-in3d/

2. TẠI SAO CẦN MÔ HÌNH HÓA?

2.2. Mục đích của mô hình hóa

- Mô hình hóa giúp đạt được 4 mục đích:
  - Cho phép **trực quan hóa** một hệ thống cần phải xây dựng
  - Cho phép đặc tả **cấu trúc** và **hành vi** của hệ thống
  - Đưa ra một **khuôn mẫu** (**template**) hướng dẫn chúng ta xây dựng một hệ thống.
  - Tài liệu hóa lại **những quyết định** mà chúng ta đưa ra trong quá trình phát triển.

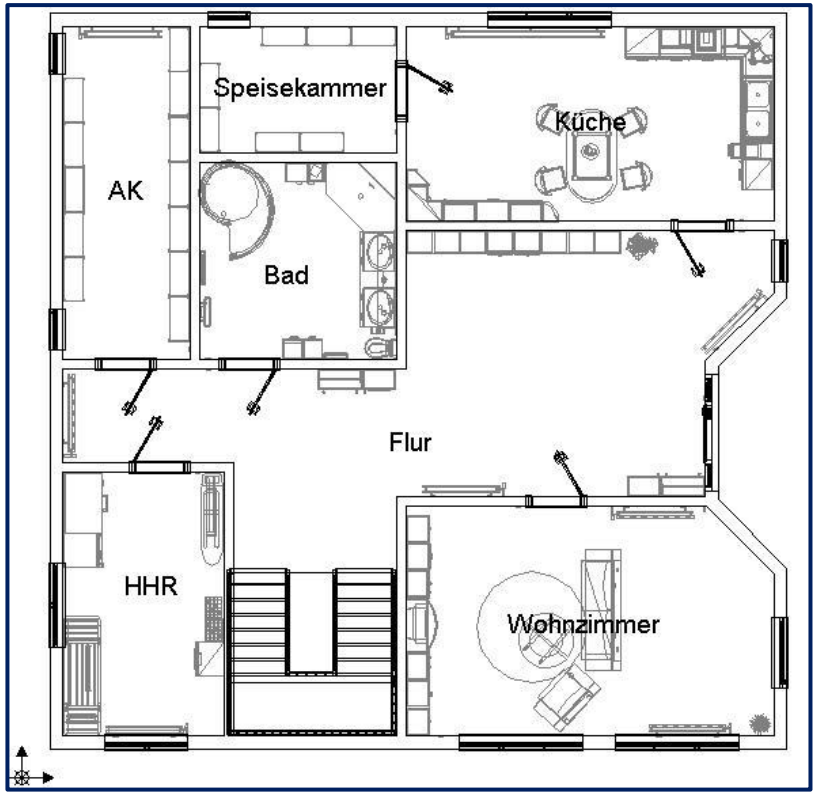
2. TẠI SAO CẦN MÔ HÌNH HÓA?

2.3 Mô hình được coi như một bản vẽ thiết kế (blueprint)

1. Requirements

- Shall fit on given piece of land.
- Each room shall have a door.
- Furniture shall fit into living room.
- Bathroom shall have a window.
- Cost shall be in budget.

2. Design



Nguồn: http://wikimedia.org
(CC nc-sa 3.0, Ottoklages)

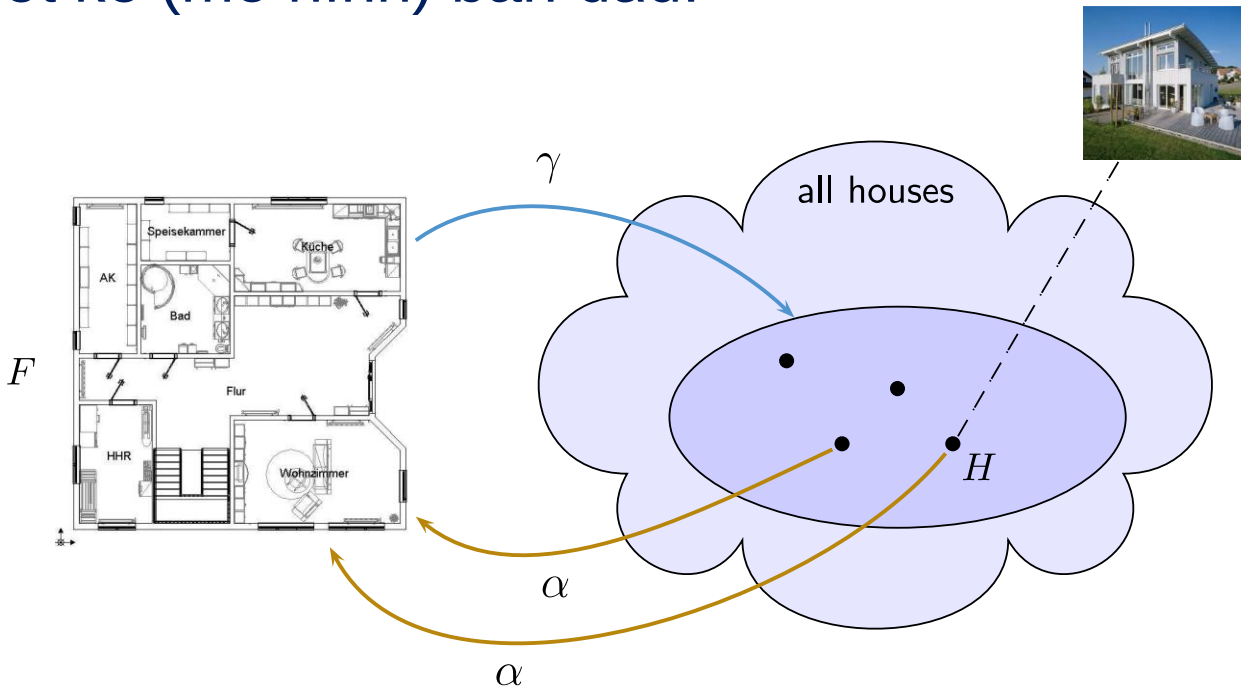
3. System

Nguồn: http://wikimedia.org
(CC nc-sa 3.0, Bobthebuilder82)

2. TẠI SAO CẦN MÔ HÌNH HÓA?

2.3 Mô hình được coi như một bản vẽ thiết kế (blueprint)

- Tuỳ thuộc vào công cụ, kỹ thuật và nền tảng mà chúng ta sử dụng, các phiên bản khác nhau của hệ thống có thể được tạo ra, tuy nhiên tất cả sẽ đều tuân thủ theo bản thiết kế (mô hình) ban đầu!



Hình 2.1. Ánh xạ từ bản vẽ thiết kế sang hệ thống được xây dựng

2. TẠI SAO CẦN MÔ HÌNH HÓA?

2.4. Những nguyên lý khi mô hình hóa

- Có rất nhiều mô hình khác nhau, mỗi mô hình cung cấp một góc nhìn riêng về hệ thống.
- Việc lựa chọn mô hình nào phụ thuộc một cách mật thiết vào việc chúng ta giải quyết yêu cầu đặt ra như thế nào?
- Môt mô hình duy nhất không thể biểu diễn được cả một hệ thống.
- Mọi mô hình đều có thể được biểu diễn ở các mức độ chi tiết khác nhau.
- Những mô hình tốt giúp kết nối hệ thống với thực tế.

NỘI DUNG TIẾP THEO

1. Thế nào là mô hình hóa (modelling)?
2. Tại sao cần mô hình hóa?

3. Ngôn ngữ mô hình hoá Unified Modelling Language (UML)

- 3.1. Tại sao sử dụng UML
- 3.2. Nguồn gốc ra đời UML
- 3.3. Lịch sử UML
- 3.4. UML là gì?
- 3.5. Một mô hình duy nhất không thể đủ để mô hình hóa
- 3.6. Các thành phần của UML

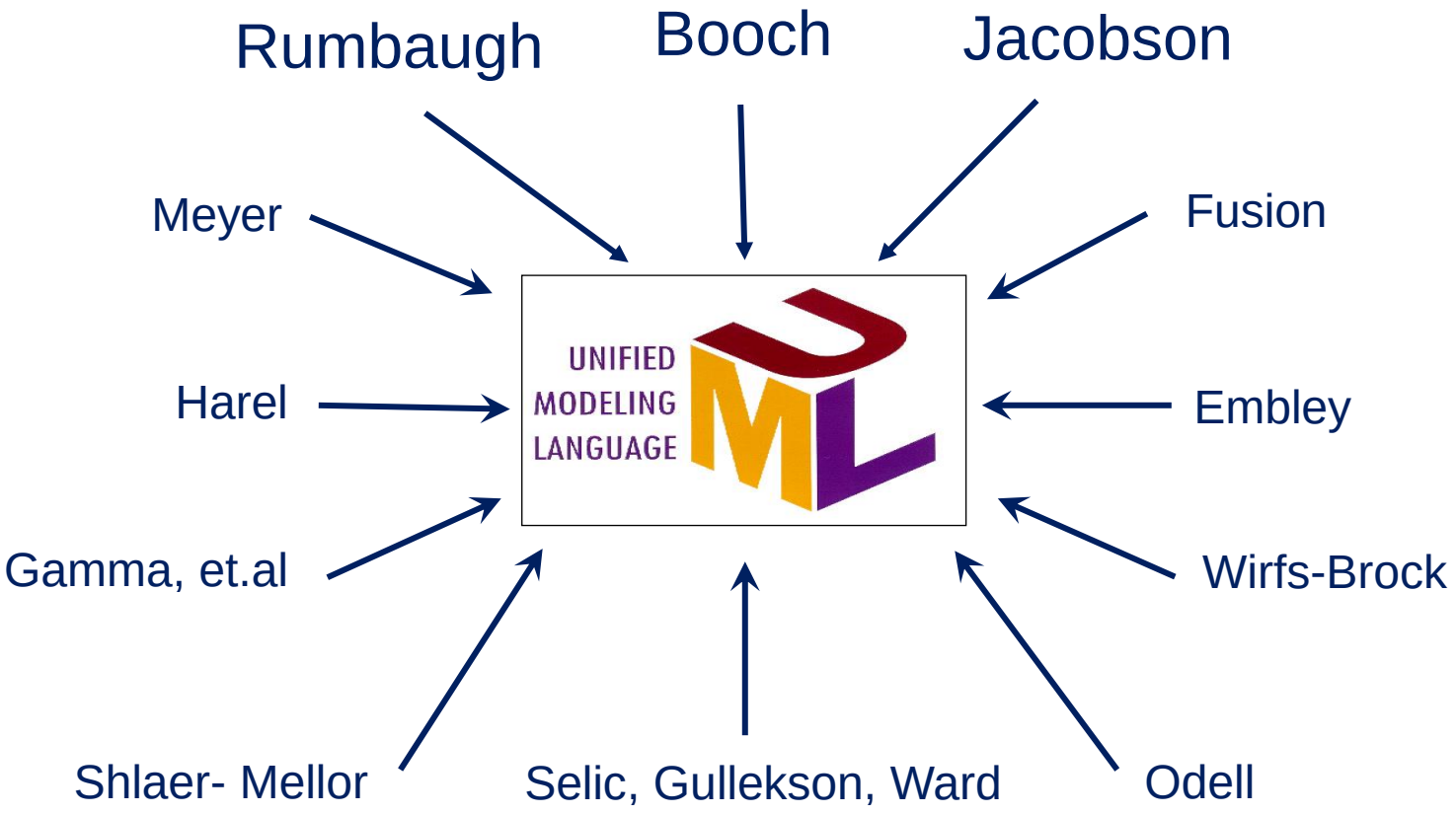
3.1. Tại sao lại sử dụng UML

- 1980s: phân tích và thiết kế hướng cấu trúc
- 1990s: phân tích và thiết kế hướng đối tượng
- Giữa những năm 1990: nhiều hơn 50 phương pháp hướng đối tượng với rất nhiều định dạng khác nhau ra đời.
  - Fusion, Shlaer-Mellor, ROOM, Class-Relation, Wirfs-Brock, MOSES, Syntropy, BOOM, OOSD, OSA, BON, Catalysis, COMMA, HOOD, Ooram, DOORS…
- Một ngôn ngữ mô hình hóa chung nhất do đó được mong đợi ra đời!

3.2. Nguồn gốc ra đời UML

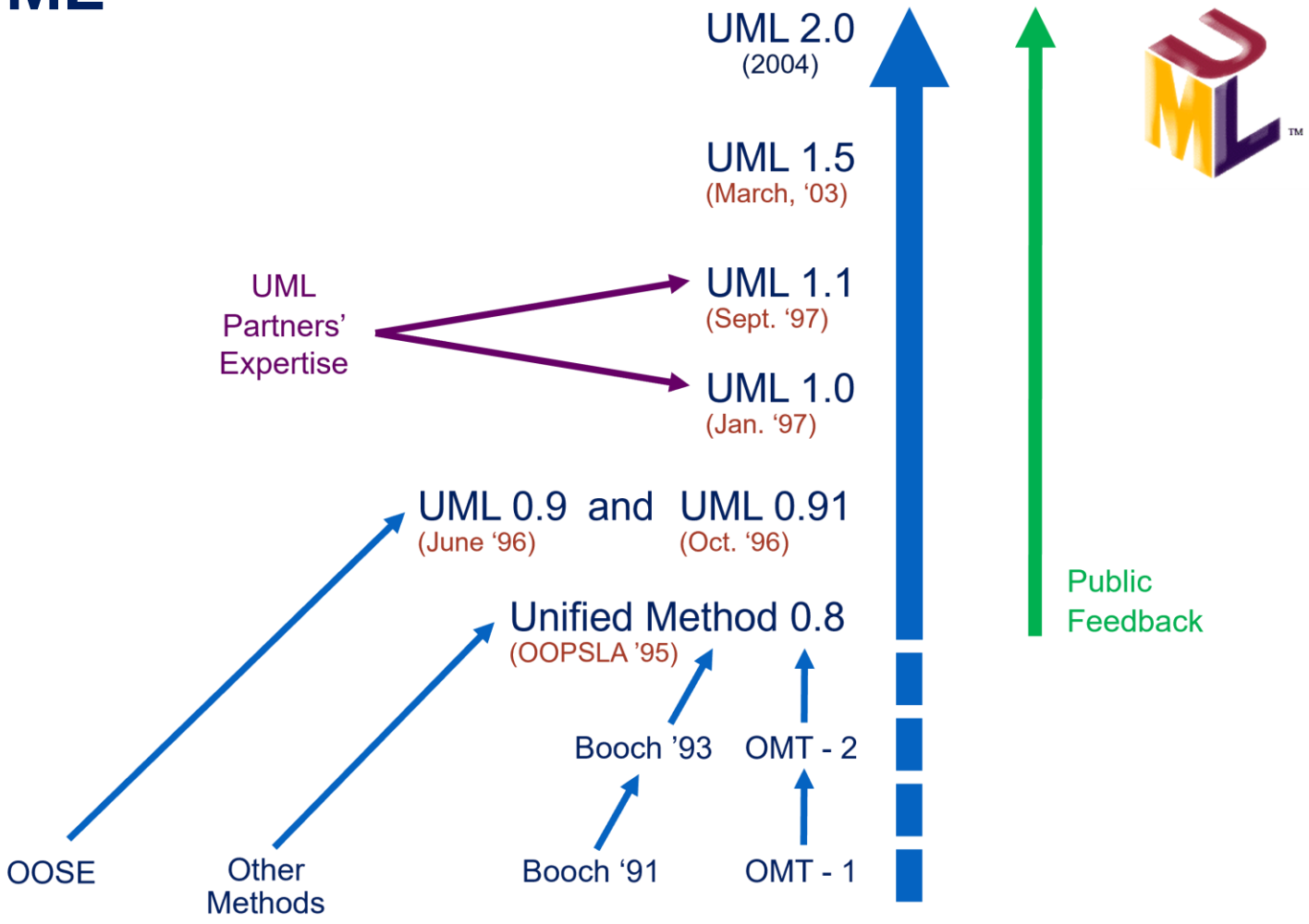
- 1980s: phân tích và thiết kế hướng cấu trúc
- 1990s: phân tích và thiết kế hướng đối tượng
- Giữa những năm 1990: nhiều hơn 50 phương pháp hướng đối tượng với rất nhiều định dạng khác nhau ra đời
  - Fusion, Shlaer-Mellor, ROOM, Class-Relation, Wirfs-Brock, MOSES, Syntropy, BOOM, OOSD, OSA, BON, Catalysis, COMMA, HOOD, Ooram, DOORS…
- Một ngôn ngữ mô hình hóa chung nhất do đó được mong đợi ra đời!

3.2. Nguồn gốc ra đời UML



Hình 3.1. Sự hợp nhất của các ngôn ngữ mô hình hoá

3.3. Lịch sử UML



Hình 3.2. Lịch sử phát triển UML

3.4. UML là gì?

- UML là ngôn ngữ cho
  - Trực quan hoá
  - Đặc tả
  - Xây dựng
  - Tài liệu hóa

  mọi thành phần của một hệ thống phần mềm

3.4. UML là gì?

- Trực quan hóa
  - Việc giao tiếp bằng các mô hình mang tính khái niệm (khái quát) thường dễ mắc lỗi trừ khi tất cả mọi người tham gia đều có chung một hệ quy chiếu về khái niệm.
  - Có những thành phần của hệ thống phần mềm mà chúng ta không thể nắm rõ được cho đến khi chúng ta xây dựng mô hình cho chúng.
  - Mô hình trong UML được biểu diễn một cách trực quan và mang ngữ nghĩa rõ ràng giúp việc giao tiếp dễ dàng giữa đội phát triển và khách hàng hoặc giữa các thành viên trong đội phát triển.

3.4. UML là gì?

- Đặc tả
  - Các mô hình UML mang ngữ nghĩa chính xác, tường minh và đầy đủ thông tin

3.4. UML là gì?

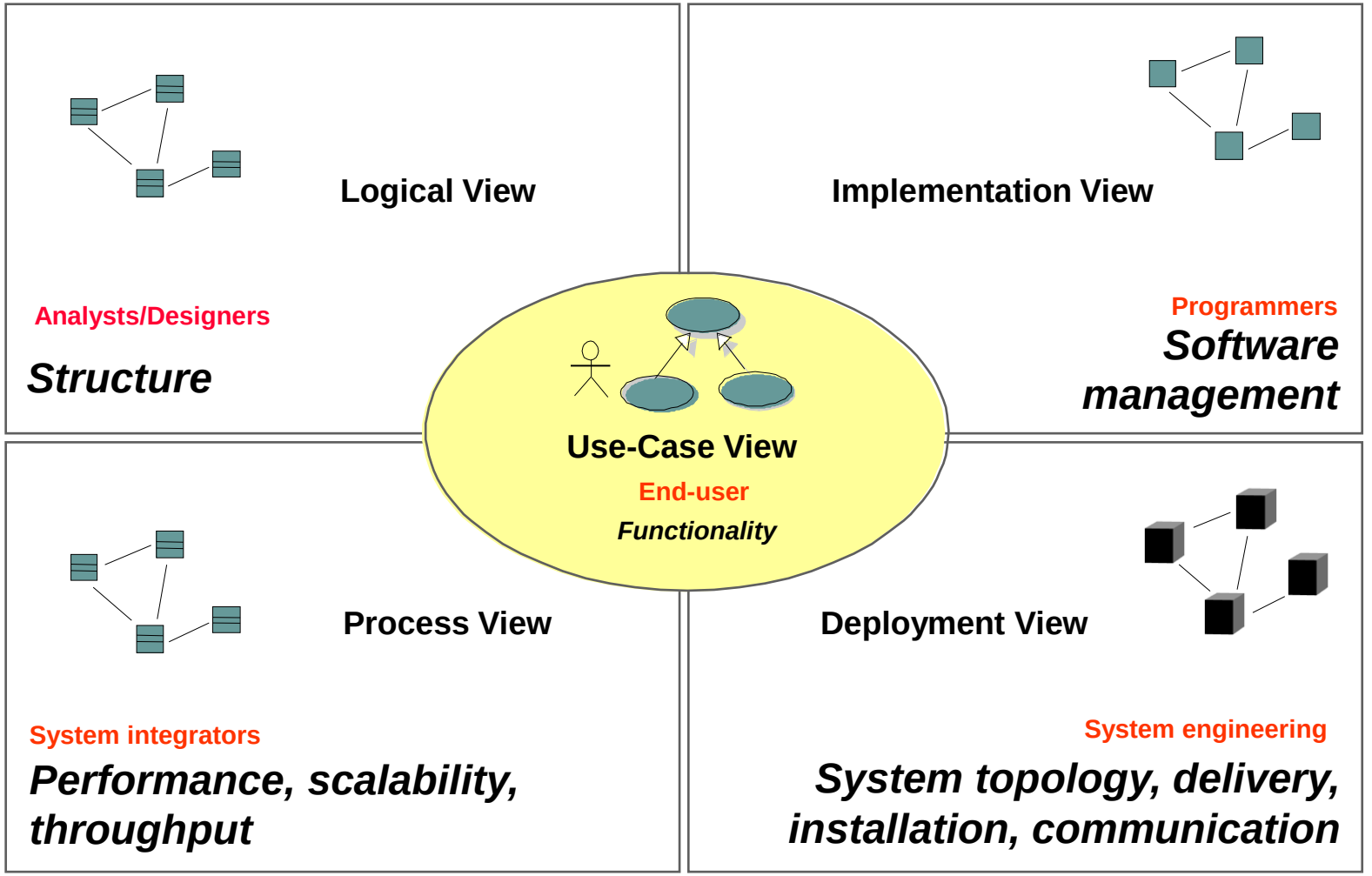
- Xây dựng
  - Các mô hình UML có thể kết nối trực tiếp với nhiều loại ngôn ngữ lập trình khác nhau.
    - Java, C++, Visual Basic
    - Bảng trong các hệ quản trị dữ liệu quan hệ hoặc kho lưu trữ lâu dài trong các hệ quản trị dữ liệu hướng đối tượng.

3.4. UML là gì?

- Tài liệu hóa: ngôn ngữ UML hướng tới xây dựng tài liệu cho một hệ thống phần mềm, từ kiến trúc tới yêu cầu, kiểm thử, lập kế hoạch project và quản lý phiên bản.

3.5. Một mô hình duy nhất không thể đủ để mô hình hóa

- Mọi hệ thống phần mềm đều cần tiếp cận bằng một tập hợp các mô hình độc lập
- 4+1 View



Hình 3.1. 4+1 View trong mô hình hoá phần mềm

3.6. Các thành phần của UML

- Diễn tả 2 khía cạnh của hệ thống:
  - **Cấu trúc tĩnh** (static structure) biểu diễn các thành phần của phần mềm (component, package, class) và các quan hệ giữa các thành phần đó.
  - **Hành vi động** (dynamic behavior) biểu diễn cách các thành phần tương tác với nhau để thực hiện các thay đổi về trạng thái của hệ thống theo thời gian.

Hình 3.2. Ví dụ về sự khác nhau giữa view động và view tĩnh

Nguồn: https://totalfootballanalysis.com/article/how-to-create-a-match-plan-tactical-analysis-tactics

3.6. Các thành phần của UML

- Use-case diagram: biểu đồ ca sử dụng
- Class diagram: biểu đồ class
- Object diagram: biểu đồ đối tượng
- State machine diagram: biểu đồ máy hữu hạn trạng thái
- Activity diagram: biểu đồ hoạt động
- Interaction diagram: biểu đồ tương tác
- Deployment diagram: biểu đồ triển khai

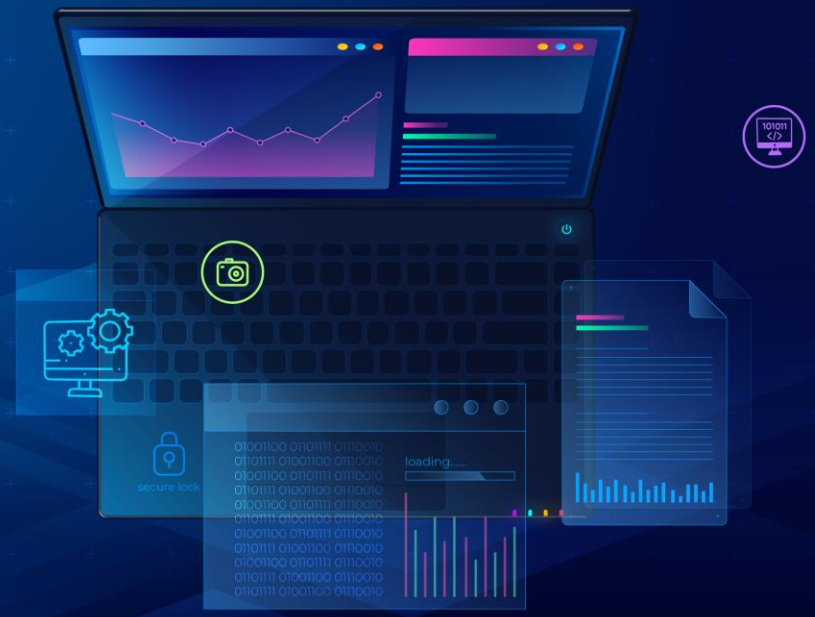
TỔNG KẾT VÀ GỢI MỞ

1. Bài học đã cung cấp cho người học một số **khái niệm cơ bản** về mô hình hoá trong công nghệ phần mềm
2. Tiếp sau bài này, **người học có thể tự tìm hiểu thêm** về tài liệu đặc tả các mô hình thông dung trong UML phiên bản 2.0 trở đi để nắm rõ về mục đích và các đặc điểm của từng loại mô hình cũng như vai trò và vị trí của chúng trong quá trình phân tích và xây dựng phần mềm

NHẬP MÔN CÔNG NGHỆ PHẦN MỀM

Tổng quan về mô hình hoá trong phát triển phần mềm

Biên soạn:
TS. Bùi Thị Mai Anh
Trình bày:
TS. Bùi Thị Mai Anh



NHẬP MÔN CÔNG NGHỆ PHẦN MỀM

Bài học tiếp theo:

Quy trình xây dựng và đặc tả yêu cầu phần mềm

Tài liệu tham khảo

[1] R. Pressman, Software Engineering: A Practitioner's Approach. 8th Ed., McGraw-Hill, 2016.

[2] I. Sommerville, Software Engineering. 10th Ed., AddisonWesley, 2017.

[3] Pankaj Jalote, An Integrated Approach to Software Engineering, 3rd Ed., Springer.

[4] Shari Lawrence Pleeger, Joanne M.Atlee, Software Engineering theory and practice. 4th Ed., Pearson, 2009

[5] ***UML 2 Toolkit.*** Hans-Erik Eriksson and Magnus Penker. Wiley Publishing Inc.

URL: http://www.ges.dc.ufscar.br/posgraduacao/UML_2_Toolkit.pdf

[7] ***Astah Manual.*** URL: http://astah.net/tutorial/astah%20professional%20referencemanual.pdf