

Lecture 6:

Mô hình hóa yêu cầu

□ Làm rõ các khái niệm

- ⇒ Mô hình hóa là gì ?
- ⇒ Các yêu cầu; Hệ thống; Tư duy hệ thống (Systems Thinking)

□ Vai trò của Mô hình hóa trong RE

- ⇒ Tầm quan trọng của mô hình hóa
- ⇒ Hạn chế của mô hình hóa

□ Tổng quan về các ngôn ngữ mô hình hóa

□ Nguyên tắc mô hình hóa

- ⇒ Trừu tượng hóa (Abstraction)
- ⇒ Phân tách (Decomposition)
- ⇒ Quy chiếu (Projection)
- ⇒ Mô-đun hóa (Modularity)

Khái niệm : Các định nghĩa

Application Domain

Machine Domain



□ Một vài điểm khác biệt

- ⇒ Domain Properties: những điều luôn luôn đúng trong lĩnh vực ứng dụng
- ⇒ Requirements: những điều chúng ta mong là đúng trong lĩnh vực ứng dụng
- ⇒ Specification: mô tả các hành vi chương trình cần thực hiện để đáp ứng với các yêu cầu

□ Hai tiêu chí cho kiểm tra (verification)

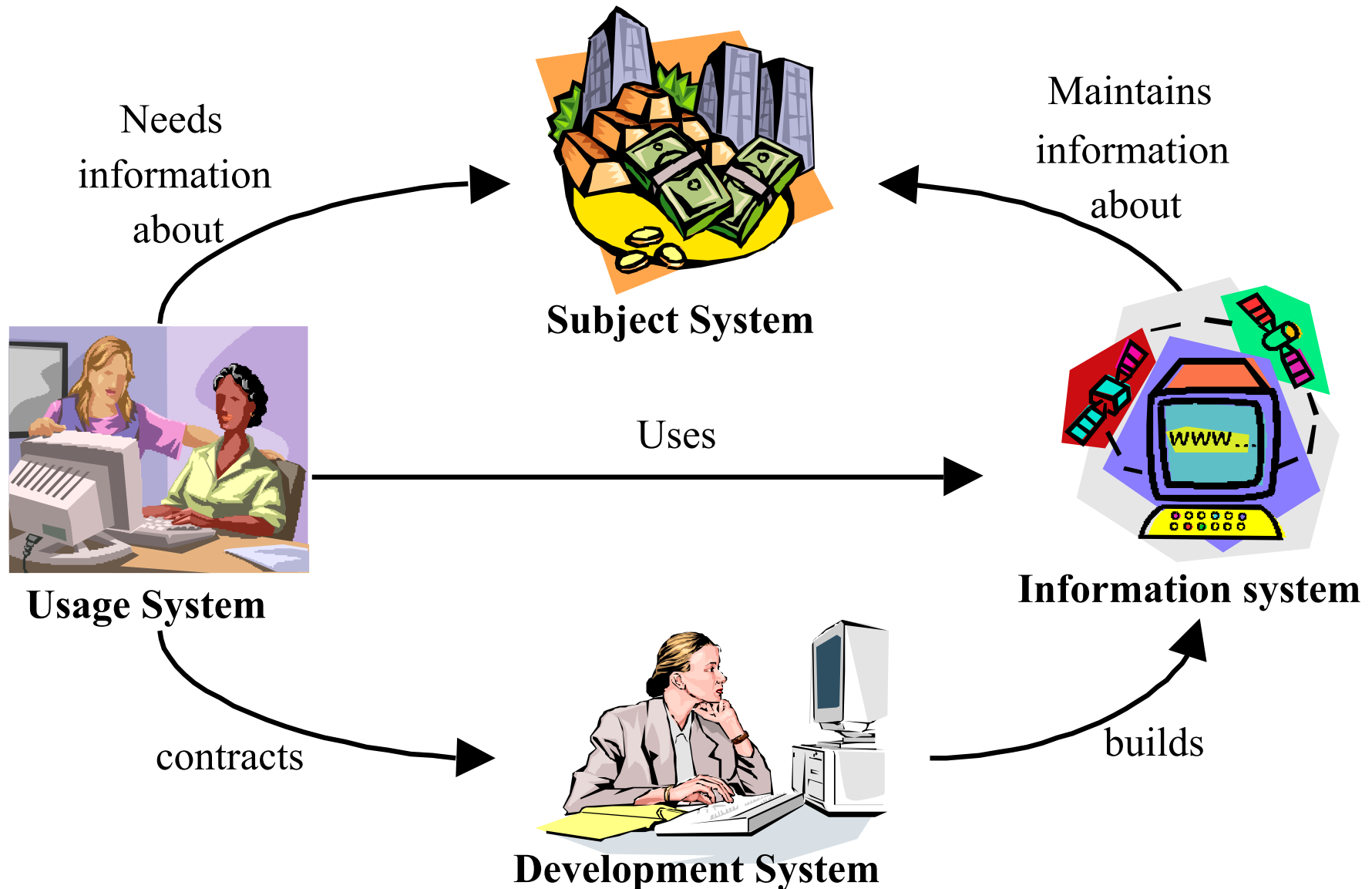
- ⇒ Chương trình (Program) thực hiện trên một máy tính (Computer) cụ thể đáp ứng với đặc tả (Specification)
- ⇒ Đặc tả (Specification) được cho trong thuộc tính của lĩnh vực (Domain properties) thỏa mãn các yêu cầu (Requirements)

□ Hai tiêu chí cho kiểm chứng (validation)

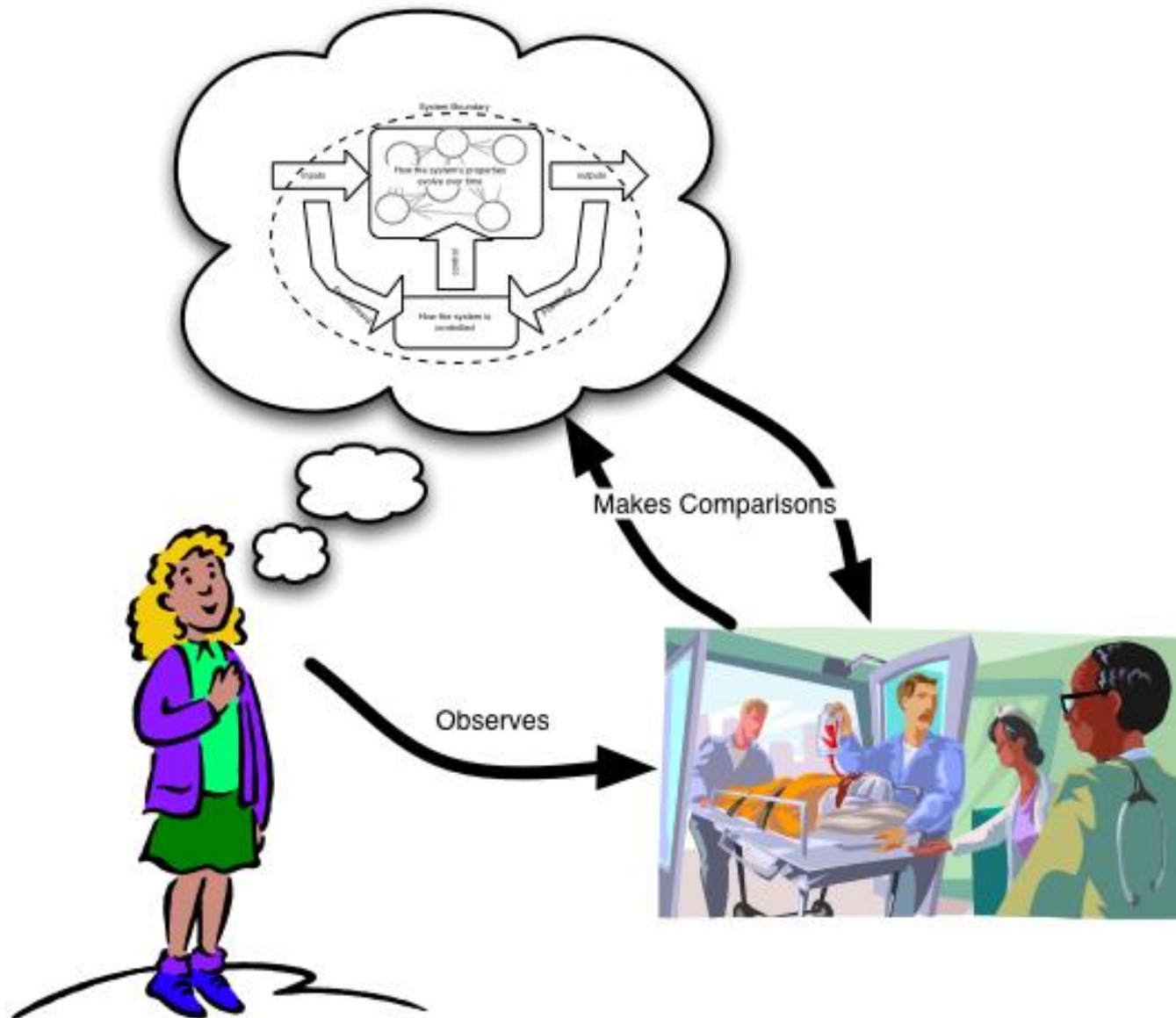
- ⇒ Chúng ta đã xem xét (và hiểu) tất cả các yêu cầu (Requirements) quan trọng?
- ⇒ Chúng ta đã xem xét (và hiểu) tất cả các thuộc tính lĩnh vực (Domain properties) liên quan?

Khái niệm : Từ hệ thống đến mô hình

Source: Adapted from Loucopoulos & Karakostas, 1995, p73



Khái niệm : Tư duy hệ thống



Mô hình hóa

□ Mô hình hóa có thể hướng dẫn suy luận

- ⇒ Nó có thể giúp bạn chỉ ra câu hỏi gì để hỏi
- ⇒ Nó có thể giúp làm nổi rõ các yêu cầu ẩn chứa
 - i.e. giúp bạn hỏi những câu chính xác?

□ Mô hình hóa có thể cung cấp sự đo lường cho quy trình:

- ⇒ Việc hoàn thiện của mô hình -> hoàn thiện của suy luận (?)
 - i.e. chúng ta có thể hoàn thiện tất cả các thành phần của mô hình, được không?

□ Mô hình hóa có thể giúp phơi bày các vấn đề

- ⇒ Sự mâu thuẫn trong các mô hình có thể dẫn đến nhiều thứ đáng quan tâm...
 - e.g. các yêu cầu xung đột hoặc không thể thực hiện
 - e.g. nhầm lẫn các thuật ngữ, phạm vi, etc
 - e.g. bất đồng giữa các đối tác

□ Mô hình hóa có thể giúp kiểm tra sự thấu hiểu của bạn

- ⇒ Lý giải trên các mô hình để hiểu kết quả của nó
 - Nó có đạt được những đặc tính mà chúng ta mong muốn?
- ⇒ Xây dựng hình ảnh bằng các mô hình giúp quan sát/kiểm chứng các yêu cầu

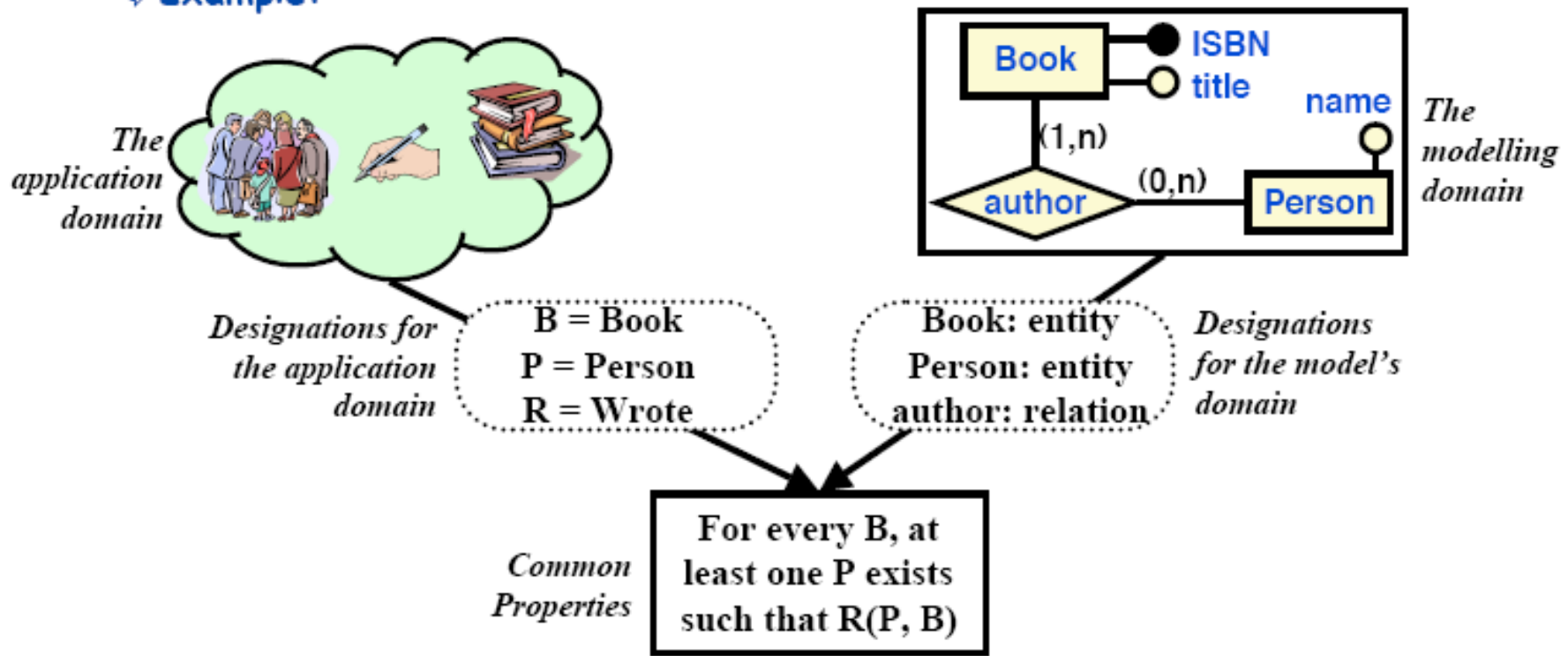
RE gồm nhiều bước mô hình hóa

Source: Adapted from Jackson, 1995, p120-122

□ Mô hình thì tốt hơn chỉ là sự mô tả

- ↳ Nó có các hiện tượng của nó và có quan hệ chủ thể giữa các hiện tượng này.
 - Mô hình sẽ hữu ích khi các hiện tượng của mô hình phù hợp một cách có hệ thống với các hiện tượng trong lĩnh vực mà nó cần được mô hình hóa

↳ Example:

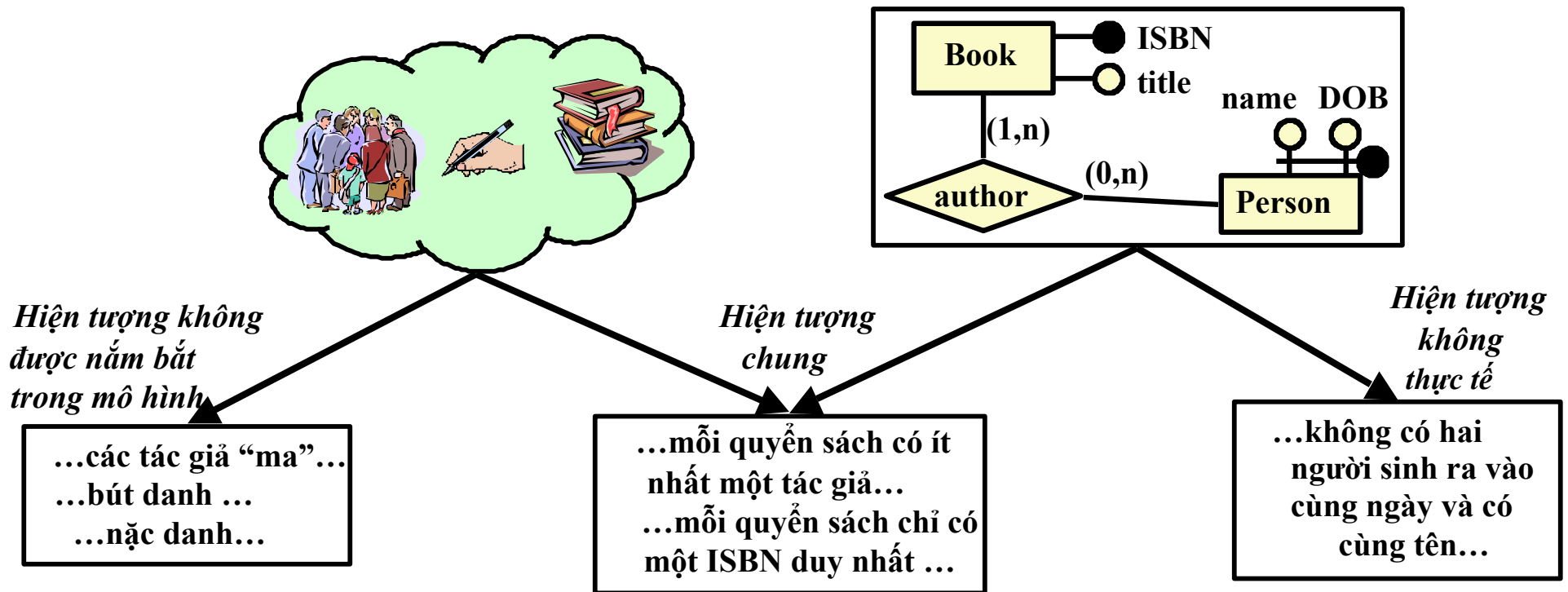


“Đó chỉ là mô hình”

Source: Adapted from Jackson, 1995, p124-5

□ Rất thường thấy rằng:

- ↪ Hiện tượng trong mô hình thì không hiện diện trong lĩnh vực ứng dụng
- ↪ Hiện tượng trong lĩnh vực ứng dụng thì không có trong mô hình



□ Một mô hình không khi nào là hoàn hảo

- ↪ “Nếu bản đồ và địa hình không giống nhau, hãy tin vào địa hình”
- ↪ Tìm kiếm sự hoàn hảo của một mô hình thì không là việc tốt cho thời gian của bạn...

Chọn ký pháp cho việc mô hình hóa

Source: Adapted from Loucopoulos & Karakostas, 1995, p72-73

□ Ngôn ngữ tự nhiên

- ⇒ Cực kỳ diễn cảm và linh hoạt
 - hữu ích cho suy diễn, và lập các mô hình ký hiệu dễ đọc
- ⇒ Khó để nắm bắt được các quan hệ mẫu chốt

□ Ký pháp bán hình thức

- ⇒ Nắm được cấu trúc và một số ngữ nghĩa
- ⇒ Có thể thực hiện (một số) hoạt động, kiểm tra tính nhất quán, ảnh động, etc.
 - E.g. lược đồ, bảng, cấu trúc tiếng Anh, etc.
- ⇒ Gần như là trực quan – cho phép chuyển thông tin một cách nhanh chóng đến các dạng đối tác khác nhau

← UML phù hợp ở đây

□ Ký pháp hình thức

- ⇒ Ngữ nghĩa chính xác, có thể suy luận rộng
 - các mô hình dựa trên cơ sở toán (e.g. lý thuyết tập hợp, FSMs (finite-state machine), etc)
- ⇒ Các mô hình rất chi tiết (có thể chi tiết hơn cả cái chúng ta cần)
 - RE hình thức thì không chấp nhận việc mô hình hóa, điều này thì khác với hầu hết các dạng khoa học máy tính khác

Mục tiêu của kỹ pháp mô hình hóa

Source: Adapted from Loucopoulos & Karakostas, 1995, p77

□ Cài đặt độc lập

↳ Không mô hình sự hiển thị dữ kiện, cách tổ chức bên trong, etc.

□ Tính trừu tượng

↳ Đưa ra các khía cạnh thiết yếu
➤ e.g. những thứ không buộc phải thay đổi thường xuyên

□ Tính hình thức

↳ Cú pháp không mơ hồ
↳ Ngữ nghĩa biểu cảm

□ Tính kiến trúc

↳ Có thể thiết kế từng phần của mô hình để kiểm soát được độ phức tạp và kích thước của nó
↳ Thiết kế cần có sự giao tiếp dễ dàng

□ Dễ phân tích

↳ Cho phép phân tích dữ liệu mơ hồ, chưa đầy đủ và không nhất quán

□ Dễ lẫn vết

↳ Cho phép các phần tử tham chiếu
↳ Cho phép liên kết với thiết kế cài đặt, etc.

□ Tính khả thi

↳ có thể cho mô hình hoạt động để so sánh nó với thực tế

□ Tối thiểu hóa

↳ Không dư thừa các khái niệm trong lược đồ mô hình hóa
➤ i.e. không chọn lựa hiển thị các vấn đề nào đó không liên quan

Khảo sát các kỹ thuật mô hình hóa

□ Mô hình hóa nghiệp vụ

- ⇒ Mục đích & Mục tiêu
- ⇒ Kiến trúc tổ chức
- ⇒ Công việc & các phụ thuộc
- ⇒ Tác nhân, vai trò, dự định

Mô hình hóa tổ chức:

i*, SSM, ISAC

Mô hình hóa mục tiêu:

KAOS, CREWS

□ Mô hình hóa thông tin & hành vi

- ⇒ Cấu trúc thông tin
- ⇒ Quan điểm hành vi
 - Kịch bản và tình huống
 - Mô hình máy trạng thái
 - Dòng thông tin
- ⇒ Các yêu cầu về thời gian/trình tự

Mô hình hóa thông tin:

E-R, Class Diagrams

Phân tích cấu trúc:

SADT, SSADM, JSD

Phân tích hướng đối tượng:

OOA, OOSE, OMT, UML

Các phương pháp hình thức:

SCR, RSML, Z, Larch, VDM

□ Mô hình hóa chất lượng hệ thống

- ⇒ Những gì 'có thể':
 - Có thể sử dụng, đáng tin cậy, có thể phát triển, an toàn, bảo mật, khả thi, tương tác,...

Thỏa thuận chất lượng:

QFD, win-win, AHP,

Đặc tả NFRs:

Timed Petri nets (mức độ thực thi)

Task models (tính dễ sử dụng)

Probabilistic MTTF (độ tin cậy)

Unified Modelling Language (UML)

□ Phương pháp hướng đối tượng thế hệ thứ ba

- ⇒ **Booch, Rumbaugh & Jacobson là những tác giả đầu tiên**
 - Vẫn còn đang tiến hóa
 - Nỗ lực chuẩn hóa sự tiến triển trên các dạng hướng đối tượng khác nhau
- ⇒ **Hoàn toàn là ký pháp**
 - Không có phương pháp mô hình nào liên quan tới nó!
 - Được dự định là một thiết kế ký pháp (một số đặc tính không phù hợp với RE)
- ⇒ **Đã trở thành một công nghệ chuẩn**
 - Nhưng được làm chủ bởi IBM/Rational (đã bán nhiều công cụ và dịch vụ UML)

□ Có một khung mô hình (meta-model) chuẩn

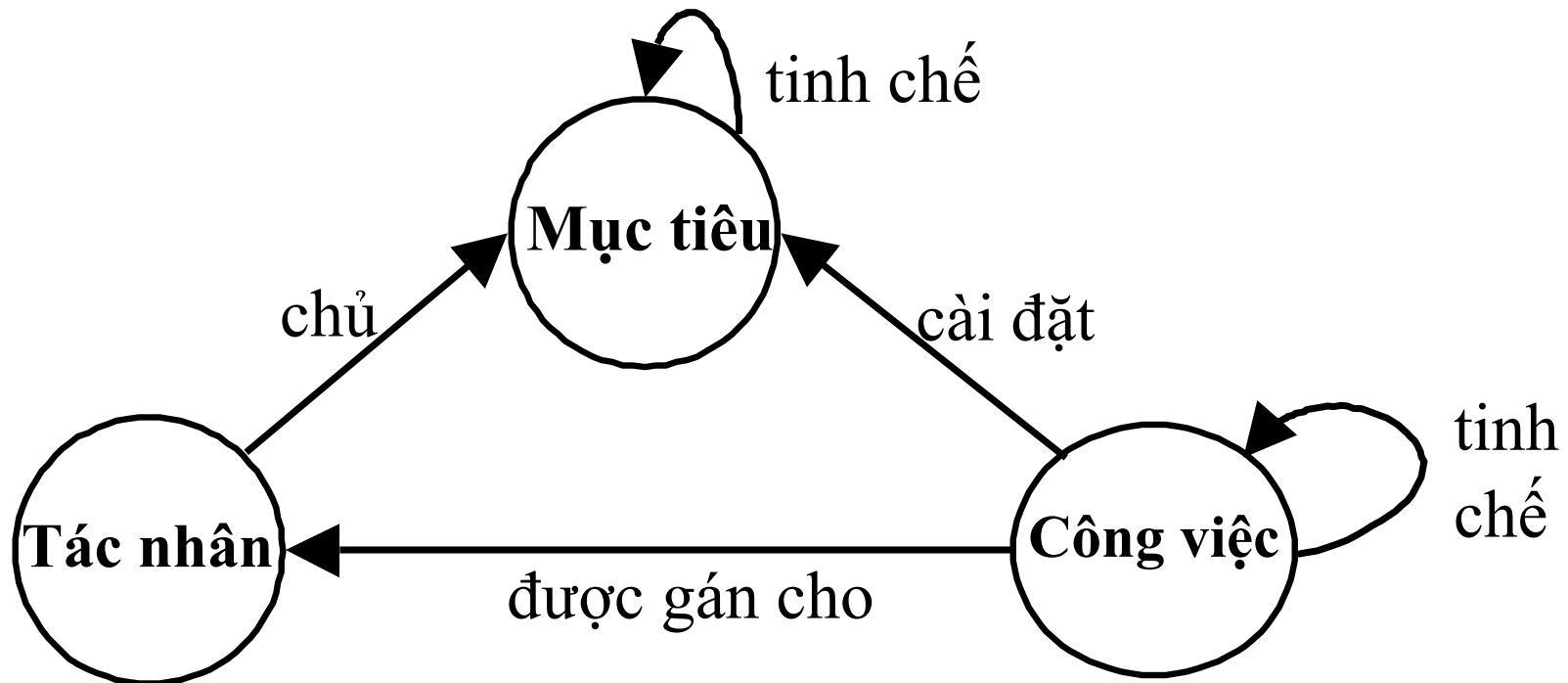
- ⇒ **Use case diagrams**
- ⇒ **Class diagrams**
- ⇒ **Message sequence charts**
- ⇒ **Activity diagrams**
- ⇒ **State Diagrams**
- ⇒ **Module Diagrams**
- ⇒ **Platform diagrams**

Meta-Modelling

□ Có thể so sánh lược đồ mô hình hóa dùng meta-models:

- ⇒ Mỗi lược đồ sẽ nắm bắt các hiện tượng gì ?
- ⇒ Cách thức soạn thảo các mô hình cần dựa theo hướng dẫn nào?
- ⇒ Cần thực hiện phân tích gì trên các mô hình?

□ Ví dụ về meta-model:



Nguyên tắc mô hình hóa

□ Dễ dàng sửa đổi và tái sử dụng

- ⇒ **Những nhà phân tích có kinh nghiệm thường sử dụng lại kinh nghiệm trước đây của họ**
 - họ sử dụng lại các thành phần (của mô hình mà họ đã xây dựng trước đó)
 - họ sử dụng lại cấu trúc (của mô hình mà họ đã xây dựng trước đó)
- ⇒ **Những nhà phân tích thông minh có thể hoạch định cho tương lai**
 - họ tạo ra các thành phần có thể sử dụng lại trong mô hình của họ
 - họ cấu trúc mô hình của họ để chúng dễ dàng sửa đổi

□ Các ý niệm hữu ích:

- ⇒ **Trừu tượng hóa (Abstraction) :** Tháo bỏ các chi tiết để tập trung vào những thứ quan trọng
- ⇒ **Phân tách (Partitioning) :** Phân chia vấn đề thành các phần độc lập, để khảo sát riêng biệt
- ⇒ **Quy chiếu (Projection) :** Phân chia các khía cạnh (views) khác nhau và mô tả chúng một cách riêng biệt
- ⇒ **Mô-đun hóa (Modularization) :** Chọn lựa các cấu trúc ổn định theo thời gian để dễ định vị sự thay đổi
- ⇒ **Mẫu (Patterns) :** Cấu trúc của một mô hình đã có xuất hiện trong nhiều ứng dụng khác nhau

Nguyên tắc 1: Phân tách

□ Sự phân tách

⇒ **Nắm rõ được sự tập hợp (aggregation)/phần của quan hệ (relationship)**

□ Ví dụ:

⇒ **Mục tiêu là khai thác một con tàu vũ trụ**

⇒ **Phân tách vấn đề thành các vấn đề con:**

- Các chỉ dẫn và cách điều khiển;
- Quản lý dữ liệu;
- Chỉ huy và kiểm soát;
- Kiểm soát môi trường;
- Theo dõi các thiết bị đo đạc;
- etc

⇒ **Chú ý: đây không phải thiết kế, chỉ là sự phân tích vấn đề**

- Thiết kế thực sự phải có đủ mọi thành phần, không có liên quan với các vấn đề con này

⇒ **Tuy nhiên, cách chọn lựa của phân tách vấn đề sẽ có thể được phản ánh trong thiết kế**

Nguyên tắc 2: Trừu tượng hóa

Source: Adapted from Davis, 1990, p48 and Loucopoulos & Karakostas, 1995, p78

□ Trừu tượng hóa

- ⇒ Là cách tìm kiếm sự tương tự giữa các khái niệm bằng việc lờ đi một số các chi tiết
- ⇒ Tập trung vào mối quan hệ tổng quan/cụ thể giữa các hiện tượng
 - Phân loại vào thành nhóm các thực thể khi chúng có vai trò tương tự như thành phần của một nhóm độc lập
 - Quan hệ kế thừa biểu diễn sự tương tự giữa các lớp khác nhau trong một mối quan hệ kết hợp '(is_a)'

□ Ví dụ:

- ⇒ Yêu cầu là : kiểm soát lỗi trên tàu vũ trụ
- ⇒ Phải nhóm các lỗi khác nhau vào thành lớp LỖI

Dựa vào vị trí:

- ⇒ lỗi thiết bị đo,
- ⇒ lỗi truyền thông,
- ⇒ lỗi xử lý,
- ⇒ etc



Dựa vào triệu chứng:

- ⇒ không có đáp ứng từ thiết bị;
- ⇒ đáp ứng không chính xác;
- ⇒ tự báo lỗi;
- ⇒ etc...

Nguyên tắc 3: Quy chiếu

Source: Adapted from Davis, 1990, p48-51

□ Quy chiếu:

- ⇒ Phân chia các lĩnh vực của mô hình thành nhiều khía cạnh (viewpoints)
 - tương tự các phép chiếu được dùng bởi kiến trúc sư trong xây dựng

□ Ví dụ:

- ⇒ Cần lập các mô hình về yêu cầu cho tàu vũ trụ
- ⇒ Phân chia mô hình :
 - độ an toàn
 - khả năng chỉ huy
 - khả năng chịu lỗi
 - đúng thời gian và trình tự
 - Etc...

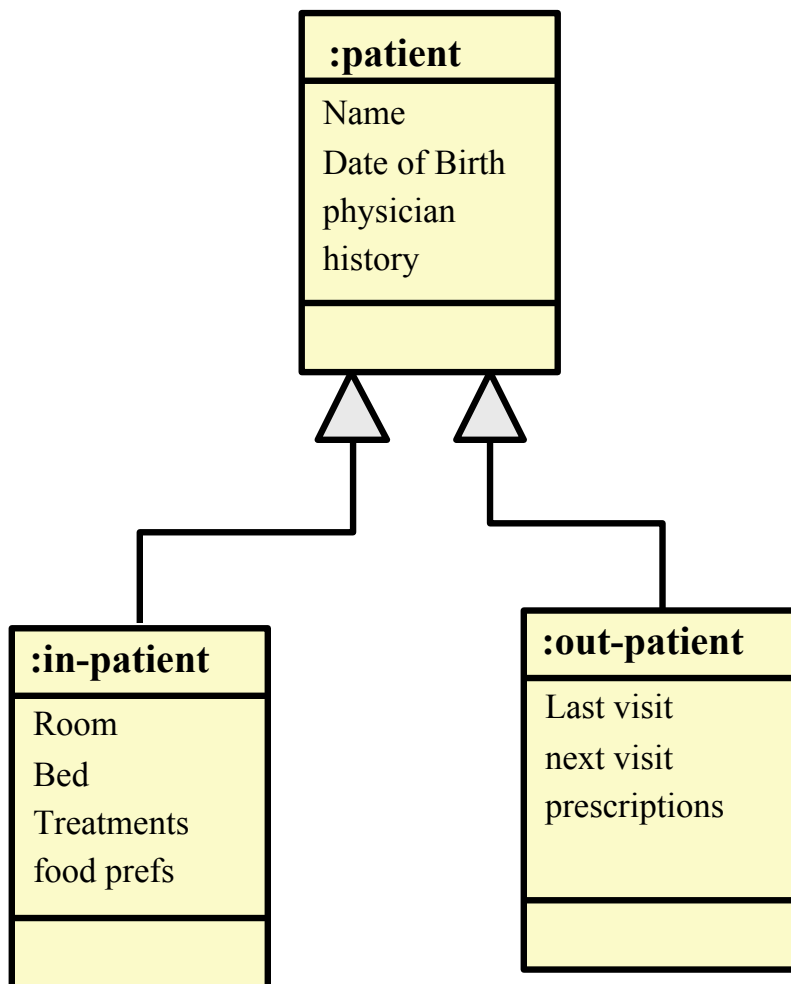
□ Chú ý:

- ⇒ Quy chiếu và Phân tách thì tương tự nhau:
 - Phân tách định nghĩa một 'phần' của quan hệ
 - Phép chiếu định nghĩa một 'khía cạnh' của quan hệ
- ⇒ Phân tách thừa nhận mỗi phần thì tương đối độc lập

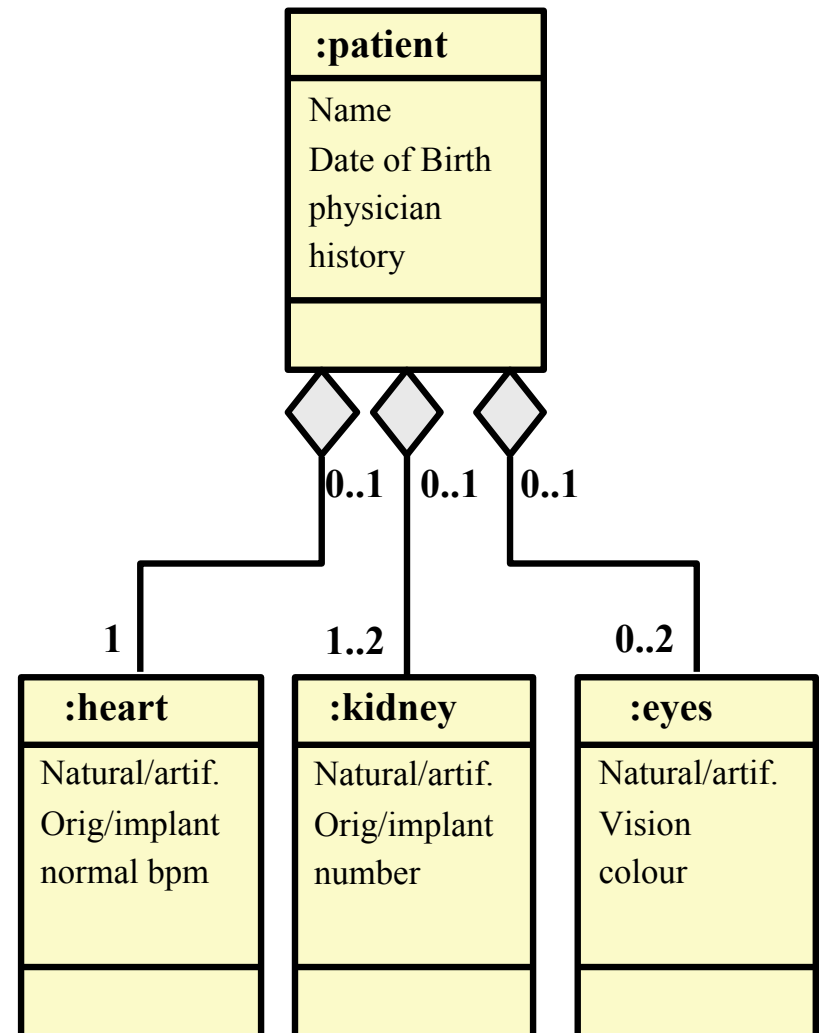
Một ví dụ tổng quan về UML

Source: Adapted from Davis, 1990, p67-68

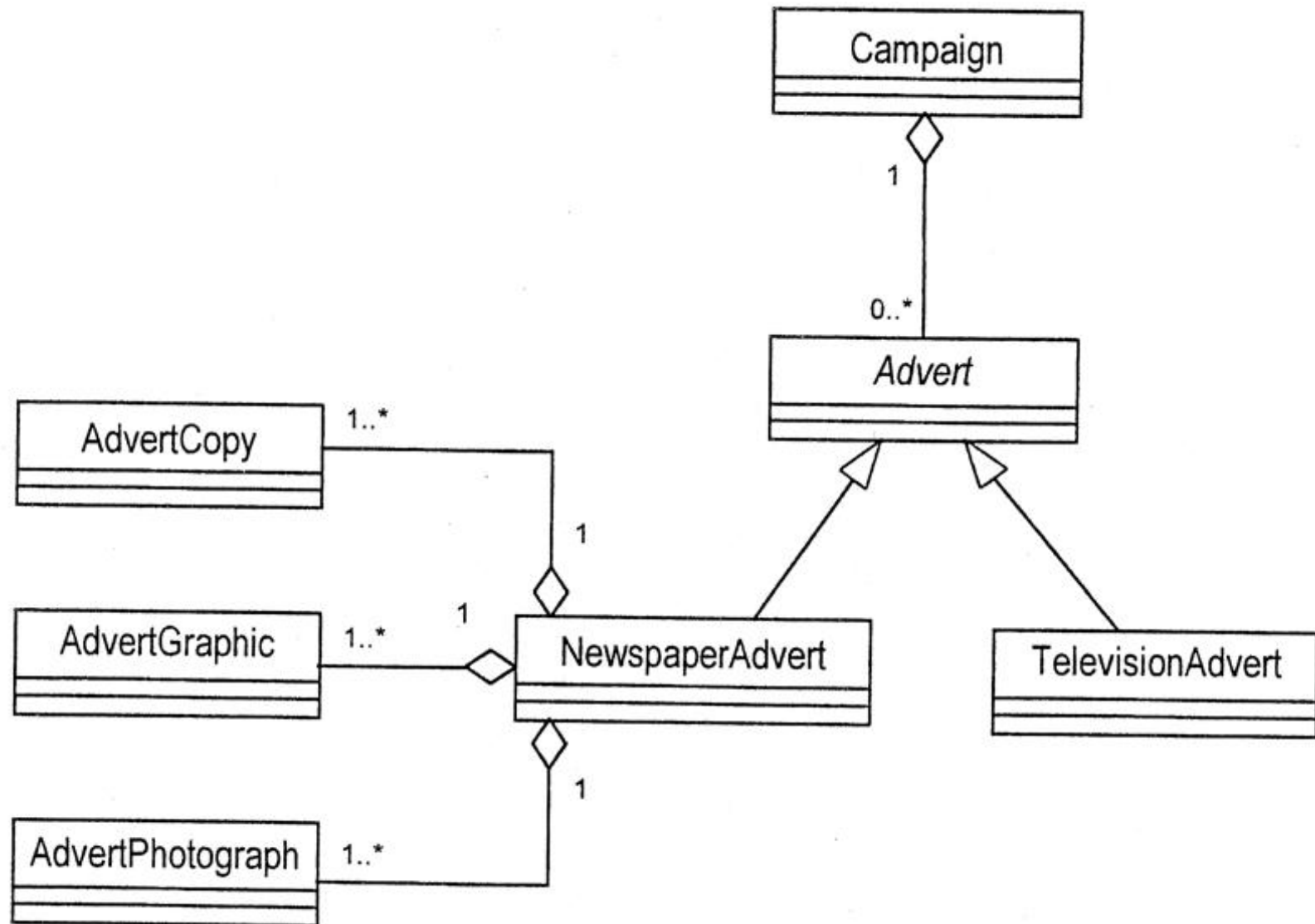
Quan hệ thừa kế
(hệ thống phân cấp trừu tượng)



Quan hệ tập hợp
(hệ thống phân cấp phân chia)



Đây là mô hình cho vấn đề gì ?



Kết luận

□ Mô hình hóa đóng vai trò trọng tâm trong RE

- ⇒ Cho phép chúng ta khảo sát vấn đề một cách hệ thống
- ⇒ Cho phép chúng ta kiểm tra sự hiểu biết của mình

□ Có nhiều lựa chọn ký pháp cho mô hình

- ⇒ Trong course này, chúng ta sẽ dùng các dạng ký pháp của UML

□ Tất cả các mô hình thường thiếu chính xác

- ⇒ Sử dụng các mô hình được hiệu chỉnh liên tục
- ⇒ ...nhưng có thể biết khi nào ngừng việc hoàn chỉnh mô hình
- ⇒ Mỗi mô hình được tạo ra cho một mục đích riêng
- ⇒ Mục đích thường không được biểu diễn trong mô hình
- ⇒ ... Vì thế nên mỗi mô hình đều cần có một sự giải thích