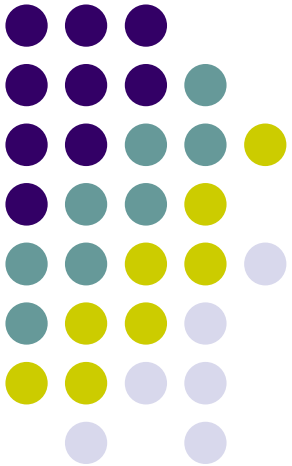
NHẬP MÔN CÔNG NGHỆ PHẦN MỀM

PHẦN II – TIẾN TRÌNH PHẦN MỀM

Bộ môn Công nghệ phần mềm,
Khoa CNTT&TT, Đại học Cần Thơ

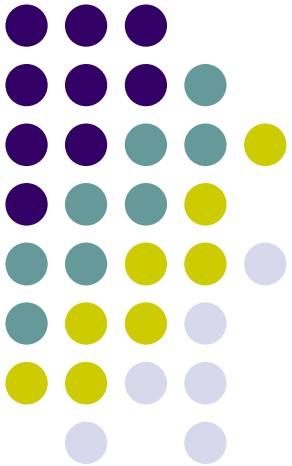


Nội dung

- Phân tích và Đặc tả
- Thiết kế
- Lập trình
- Kiểm thử
- Triển khai hệ thống và Bảo trì

TIẾN TRÌNH PHẦN MỀM

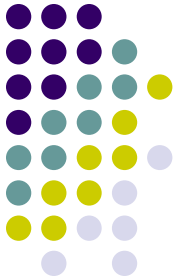
PHẦN II.1 – PHÂN TÍCH & ĐẶC TẢ YÊU CẦU



Nội dung

- Quy trình xác định các yêu cầu
- Thu thập các yêu cầu
- Phân loại các yêu cầu
- Các đặc trưng của yêu cầu
- Các ký hiệu mô hình hóa
- Các ngôn ngữ đặc tả và yêu cầu
- Lập bản mẫu cho các yêu cầu
- Tài liệu yêu cầu
- Kiểm tra và xác nhận

Quy trình xác định các yêu cầu



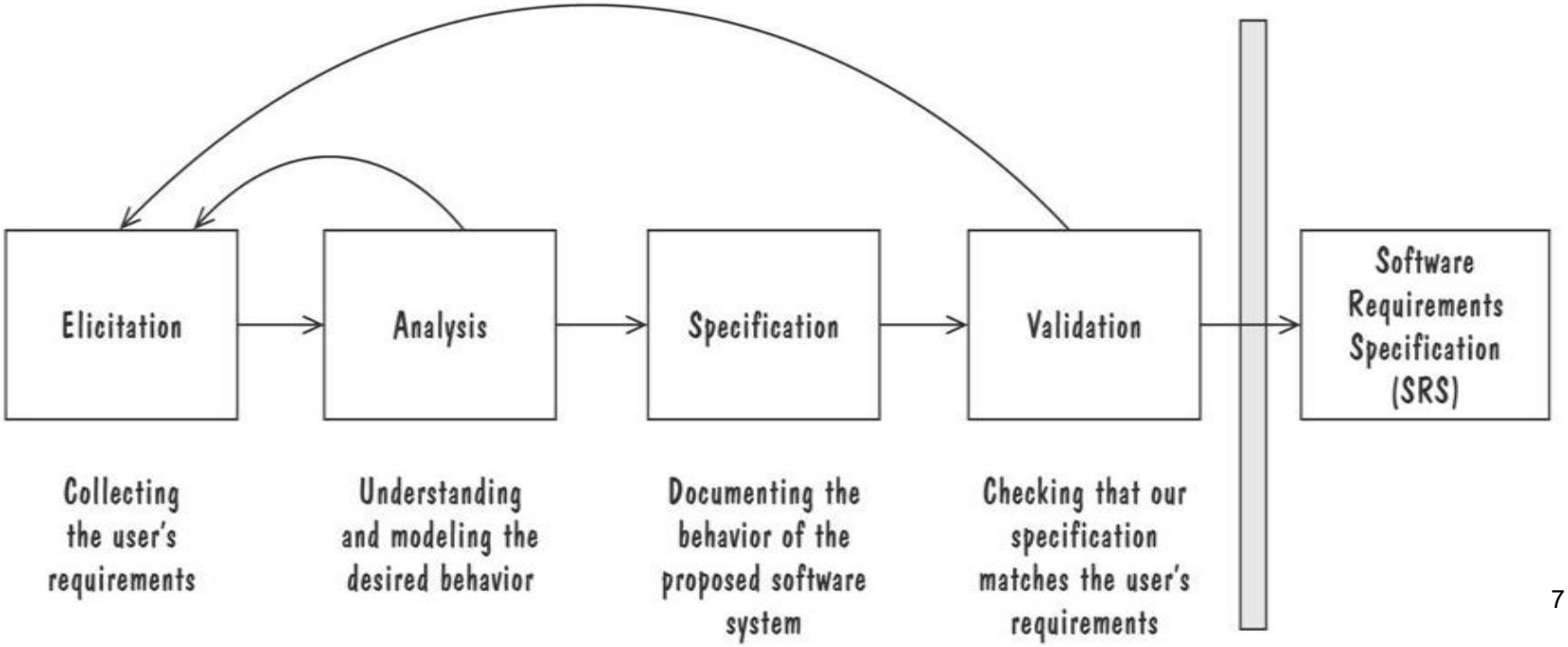
- Yêu cầu (Requirement)
  - Một yêu cầu là sự diễn đạt hành vi mong muốn.
  - Một yêu cầu đề cập đến:
    - Các đối tượng hay thực thể
    - Trạng thái của đối tượng hay thực thể
    - Các chức năng được thực hiện để thay đổi trạng thái hay các đặc trưng của đối tượng
  - Các yêu cầu tập trung vào nhu cầu của khách hàng chứ không phải tập trung vào giải pháp hay sự thực hiện.

Quy trình xác định các yêu cầu

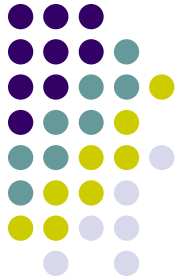
- Những nguyên nhân hàng đầu làm cho dự án thất bại:
  - Yêu cầu không hoàn chỉnh
  - Thiếu sự tham gia của người sử dụng
  - Mong muốn không thực tế
  - Thiếu sự hỗ trợ về quản lý
  - Thay đổi các yêu cầu và các đặc tả
  - Thiếu việc lập kế hoạch
  - Hệ thống không được cần nữa
- Một phần nào đó trong quy trình xác định các yêu cầu liên quan đến hầu hết các nguyên nhân này.
- Lỗi về yêu cầu có thể gây tốn kém nếu không được phát hiện sớm.

Quy trình xác định các yêu cầu

- Quy trình xác định yêu cầu
  - Được thực hiện bởi nhà phân tích yêu cầu hay nhà phân tích hệ thống.
  - Có kết quả cuối cùng là đặc tả các yêu cầu phần mềm.



Thu thập các yêu cầu

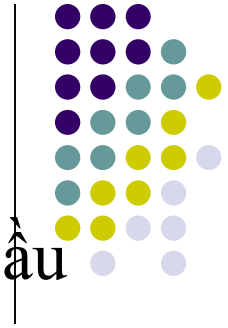


- Có khoảng cách thông tin (information gap) giữa khách hàng và nhà phân tích.

$\Rightarrow$

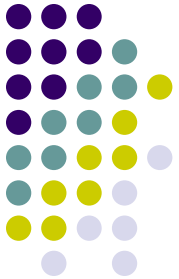
  - Việc thảo luận với các thành viên tham gia vào hoạt động thu thập yêu cầu là quan trọng.
  - Việc thảo luận đưa đến sự đồng ý giữa các bên về các yêu cầu.

Thu thập các yêu cầu



- Các thành viên tham gia vào hoạt động thu thập yêu cầu
  - **Clients**: trả tiền cho phần mềm được phát triển.
  - **Customers**: mua phần mềm sau khi nó được phát triển.
  - **Người dùng**: sử dụng hệ thống.
  - **Chuyên gia về lĩnh vực**: biết rõ vấn đề mà phần mềm phải tin học hóa.
  - **Nhà nghiên cứu thị trường**: thực hiện các cuộc khảo sát để xác định các xu hướng tương lai và những khách hàng tiềm năng.
  - **Luật sư và kiểm toán viên**: biết rõ các yêu cầu của luật pháp, chính phủ hay sự an toàn.
  - **Kỹ sư phần mềm và các chuyên gia công nghệ khác**: đảm bảo phần mềm là khả thi về kinh tế và công nghệ.

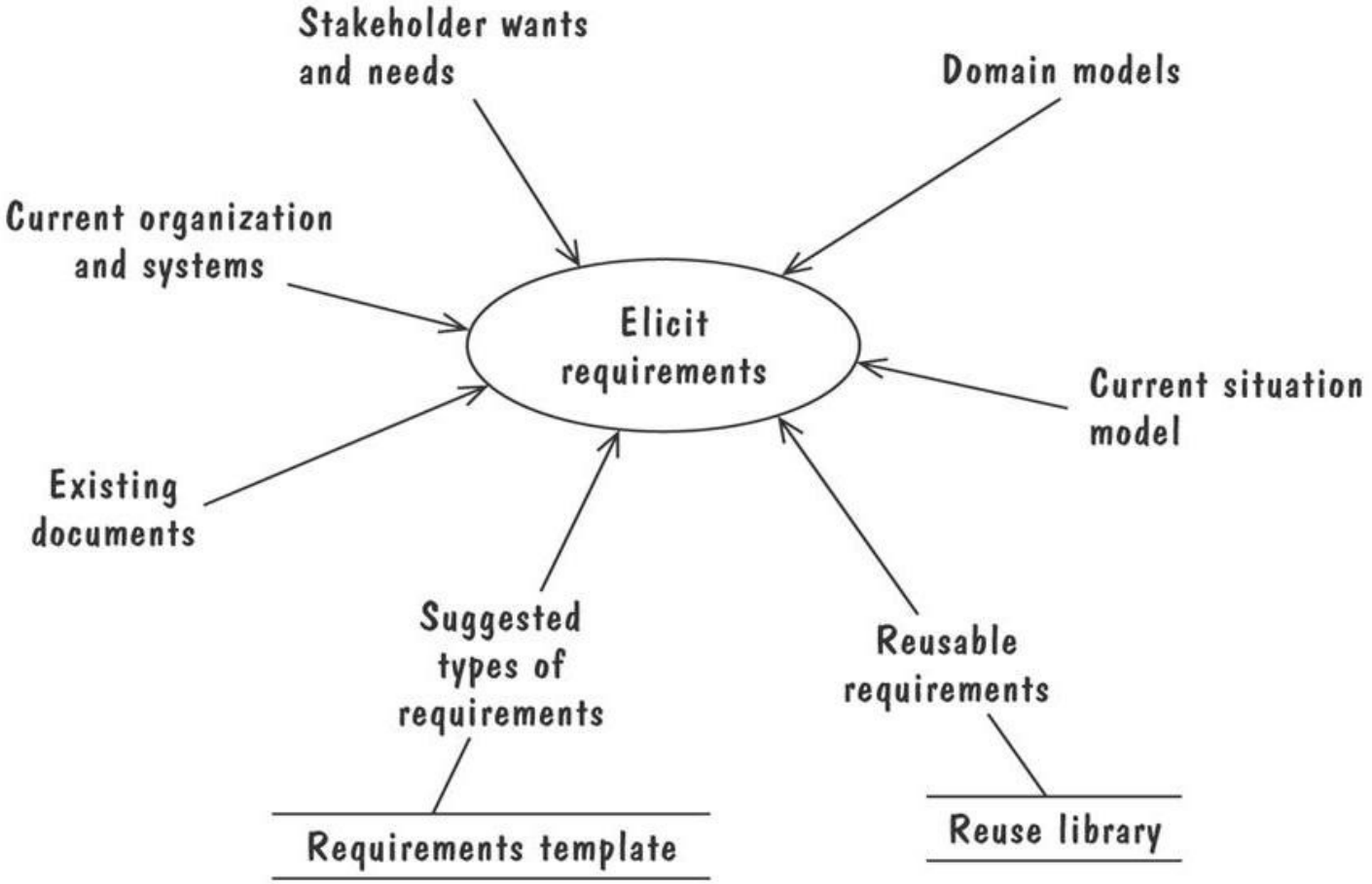
Thu thập các yêu cầu



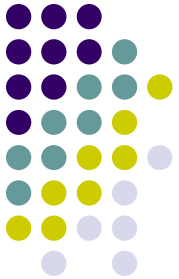
- Những cách thu thập yêu cầu
  - Phỏng vấn những cá nhân tham gia trong hệ thống.
  - Phỏng vấn những nhóm người tham gia vào hệ thống.
  - Xem lại các tài liệu có sẵn.
  - Quan sát hệ thống hiện hành (nếu hệ thống tồn tại).
  - Theo người dùng để học về nghiệp vụ của họ một cách chi tiết hơn.
  - Sử dụng các chiến lược xác định vấn đề như thiết kế ứng dụng chung (Joint Application Design).
  - Vận dụng trí tuệ tập thể (brainstorming) của người dùng hiện tại và tiềm năng để có được các yêu cầu.

Thu thập các yêu cầu

- Các nguồn yêu cầu

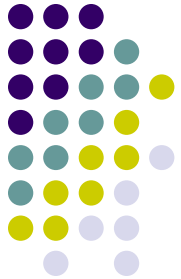


Phân loại các yêu cầu



- Phân loại các yêu cầu
- Giải quyết những xung đột
- Các loại tài liệu yêu cầu

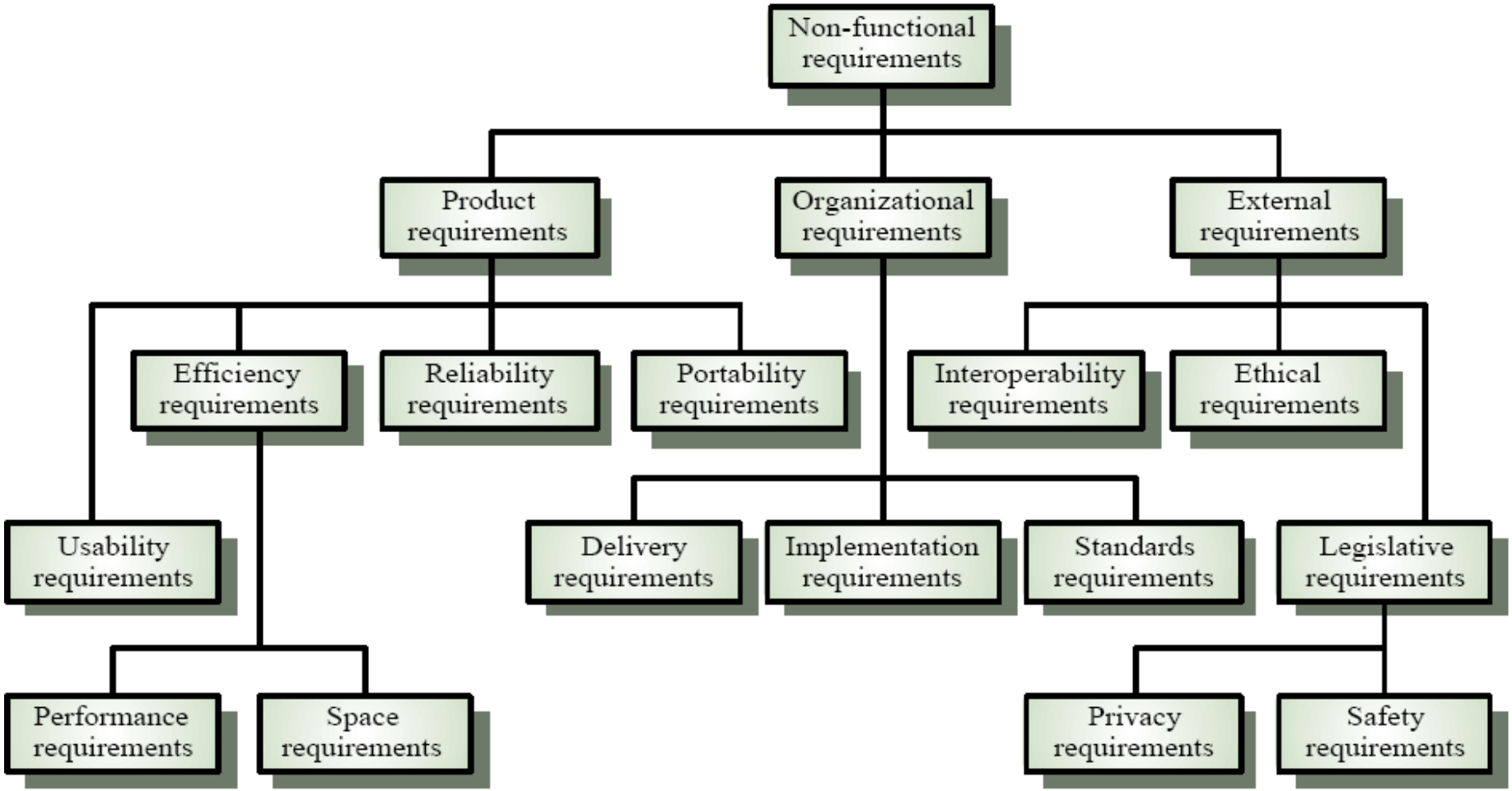
Phân loại các yêu cầu



- **Yêu cầu chức năng (functional requirement)** mô tả chức năng và dịch vụ mà hệ thống phải cung cấp.
- **Yêu cầu phi chức năng (non-functional requirement)** mô tả một đặc trưng nào đó về chất lượng nào mà phần mềm phải có.
- **Ràng buộc thiết kế** như chọn nền hay các thành phần của giao diện.
- **Ràng buộc quy trình** như sự hạn chế về các kỹ thuật và các tài nguyên được sử dụng để xây dựng hệ thống.

Phân loại các yêu cầu

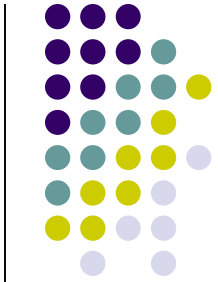
- Các loại yêu cầu phi chức năng



Phân loại các yêu cầu

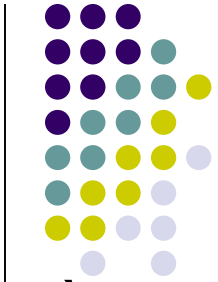
- Giải quyết sự xung đột giữa các yêu cầu
  - Giải quyết sự xung đột bằng cách sắp thứ tự ưu tiên cho các yêu cầu.
  - Ba hạng mục ưu tiên:
    - *Cần thiết*: phải được đáp ứng một cách hoàn toàn.
    - *Mong muốn*: mong được đáp ứng cao nhưng không nhất thiết.
    - *Tùy chọn*: có thể được đáp ứng nhưng cũng có thể bị loại trừ.

Các đặc trưng của yêu cầu



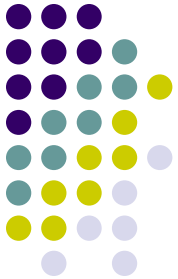
- Chính xác (Correct)
- Nhất quán (Consistent)
- Không mơ hồ (Unambigious)
- Hoàn chỉnh (Complete)
- Khả thi (Feasible)
- Có liên quan (Relevant)
- Có thể kiểm thử (Testable)
- Có thể theo vết (Traceable)

Các ký hiệu mô hình hóa



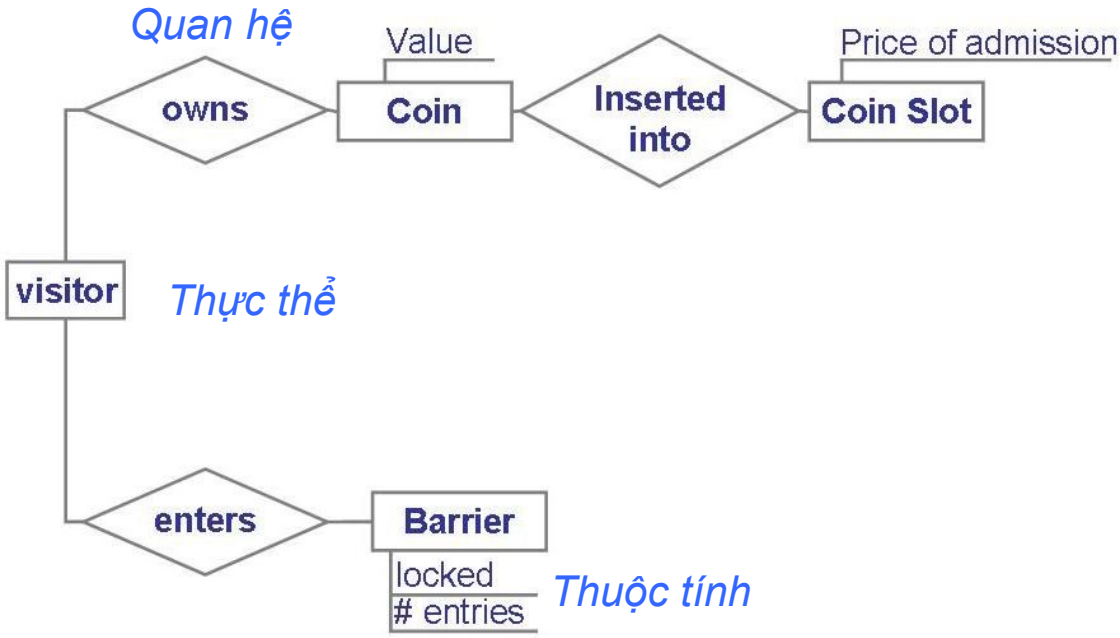
- Việc có các ký hiệu chuẩn để mô hình hóa, lập tài liệu và giao tiếp với các quyết định là quan trọng.
- Việc mô hình hóa giúp ta hiểu thấu đáo các yêu cầu (hoạt động còn mơ hồ hay chưa biết, sự xung đột giữa các kết xuất hay sự không nhất quan trong các yêu cầu).
- Một số mô thức (paradigm) ký hiệu cơ bản:
  - Lưu đồ thực thể quan hệ (Entity Relationship Diagram - ERD)
  - Dò theo sự kiện (Event Traces)
  - Máy trạng thái (State Machines)
  - Lưu đồ dòng dữ liệu (Data Flow Diagram)
  - Hàm và quan hệ
  - Logic
  - Đặc tả đại số

Mô thức ký hiệu - Lưu đồ thực thể quan hệ



- Là lưu đồ ký hiệu dạng đồ thị
- Thường được dùng để biểu diễn các mô hình mức quan niệm
- Các thành phần cốt lõi trong lưu đồ
  - *Thực thể*: được vẽ như một *hình chữ nhật*, biểu diễn cho tập các đối tượng trong thế giới thực mà chúng có chung các đặc điểm và cách hoạt động
  - *Quan hệ*: được vẽ như một *cạnh* nối hai thực thể, với hình thoi ở chính giữa cạnh xác định loại quan hệ
  - *Thuộc tính*: một diễn giải trong thực thể - mô tả dữ liệu hay các đặc tính được kết hợp với thực thể

Mô thức ký hiệu - Lưu đồ thực thể quan hệ

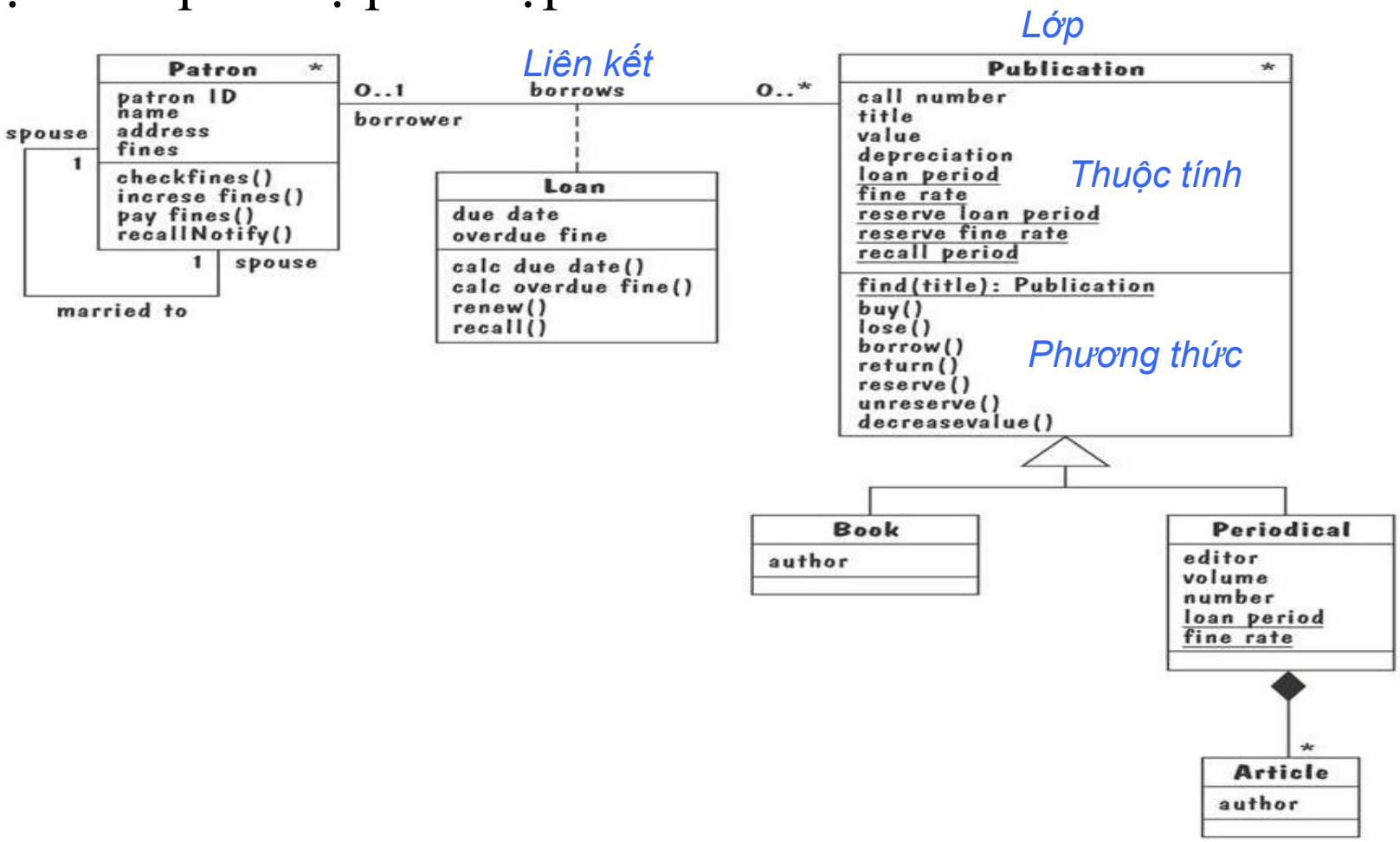


- Ví dụ: sơ đồ thực thể - quan hệ của bài toán cửa quay

- Lưu đồ thực thể quan hệ là phổ biến vì
  - Nó cung cấp một cái nhìn tổng quan về vấn đề phải được xác định.
  - Tính tổng quan là tương đối ổn định khi có thay đổi trong các yêu cầu.
- Lưu đồ thực thể quan hệ có vẻ phù hợp để mô hình hóa vấn đề sớm trong quy trình xác định các yêu cầu.

Mô thức ký hiệu - Lưu đồ thực thể quan hệ

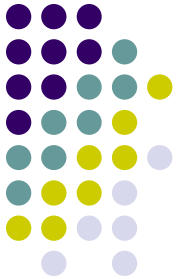
- Ví dụ: sơ đồ lớp UML (UML Class Diagram) là một lưu đồ thực thể quan hệ phức tạp



Mô thức ký hiệu - Lưu đồ thực thể quan hệ

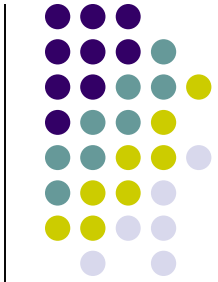
- Sơ đồ lớp UML là lưu đồ thực thể quan hệ phức tạp
  - Các thuộc tính và các phương thức được kết hợp với *lớp* hơn là với các thể hiện của lớp.
  - *Thuộc tính* phạm vi lớp, được biểu diễn như một thuộc tính được gạch chân, là một giá trị dữ liệu mà nó được chia sẻ bởi tất cả các thể hiện của lớp.
  - *Phương thức* phạm vi lớp, được biểu diễn như một phương thức được gạch chân, là một phương thức được thực hiện bởi lớp trừu tượng hơn là bởi các thể hiện của lớp.
  - *Liên kết* (association), được biểu diễn bởi một đường nối giữa hai lớp, chỉ ra mối quan hệ giữa các thực thể của các lớp.

Mô thức ký hiệu - Lưu đồ thực thể quan hệ



- Các loại liên kết
  - Liên kết kết tập (aggregation association)
  - Liên kết cấu thành (composition association)
  - Liên kết tổng quát hóa
  - …

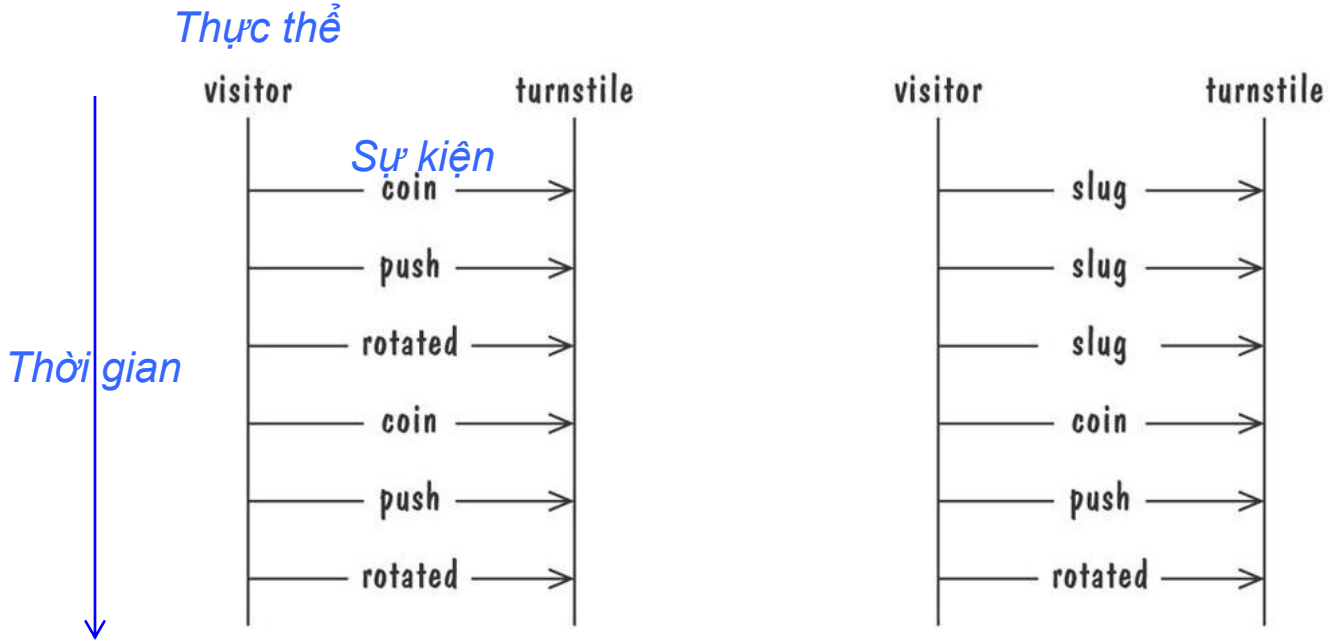
Mô thức ký hiệu - Dò theo sự kiện



- Mô hình thực thể quan hệ không nói gì về cách mà các thực thể hành xử.
- Dò theo sự kiện là sự mô tả ở dạng đồ thị của một chuỗi các sự kiện mà chúng được trao đổi giữa các thực thể của thế giới thực.
  - *Trục tung*: đường thời gian của một *thực thể* riêng biệt; tên của thực thể xuất hiện trên đỉnh của trục.
  - *Trục hoành*: một *sự kiện* hay *sự tương tác* giữa hai thực thể.
  - Thời gian tiến triển theo hướng từ trên xuống.
- Mỗi đồ thị mô tả một đơn dò theo sự kiện, biểu diễn cho một trong một vài cách hoạt động có thể có.

Mô thức ký hiệu - Dò theo sự kiện

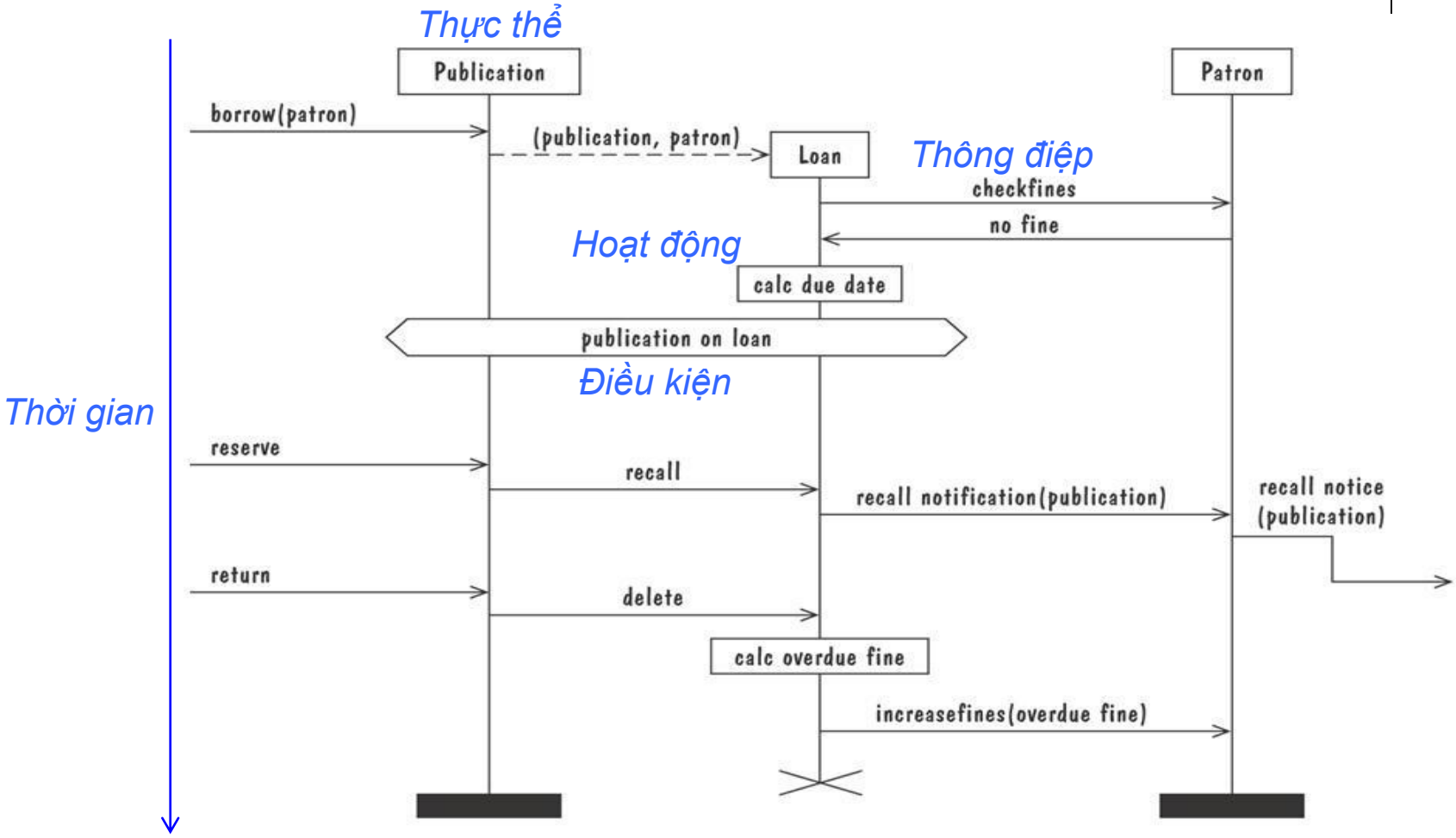
- Ví dụ: 2 biểu diễn đồ thị của bài toán cửa quay



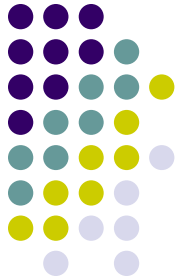
Mô thức ký hiệu - Dò theo sự kiện

- Ví dụ: **sơ đồ tuần tự thông điệp** là ký hiệu dò theo sự kiện được cải tiến, với các phương tiện cho phép tạo ra hay hủy bỏ các thực thể, xác định các hoạt động và định thời, và tạo ra các theo vết
  - *Đường dọc* biểu diễn cho một *thực thể* tham gia
  - *Thông điệp* được vẽ bằng một *mũi tên* hướng từ thực thể gửi sang thực thể nhận.
  - *Hoạt động* được vẽ bằng *hình chữ nhật* được gán nhãn và được đặt trên đường thực thi của thực thể
  - *Điều kiện* là các trạng thái quan trọng trong sự tiến hóa của thực thể, được biểu diễn bằng *hình lục giác* có gán nhãn

Mô thức ký hiệu - Dò theo sự kiện



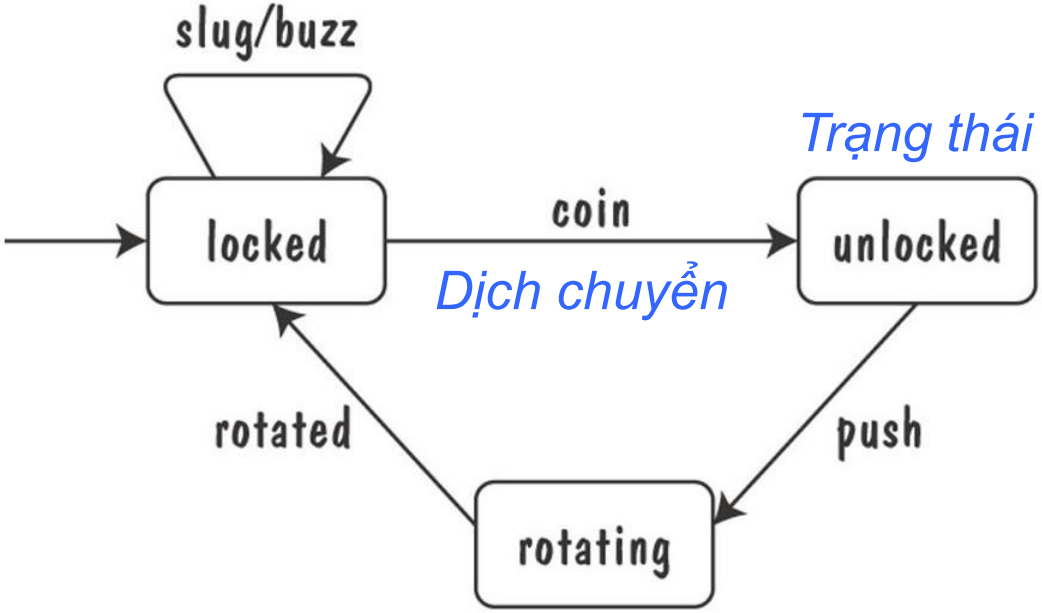
Mô thức ký hiệu - Máy trạng thái



- Là sự mô tả ở dạng đồ thị của tất cả các cuộc đối thoại giữa hệ thống và môi trường của nó.
- Có
  - Nút (*trạng thái*) biểu diễn một tập ổn định các điều kiện mà nó tồn tại giữa các lần xuất hiện của sự kiện.
  - Cạnh (*dịch chuyển*) biểu diễn cho sự thay đổi về hành vi hay điều kiện do sự xuất hiện của một sự kiện.
- Hữu ích cả cho việc xác định hành vi động và cho việc mô tả cách mà hành vi nên thay đổi để đáp ứng được lịch sử của các sự kiện.

Mô thức ký hiệu - Máy trạng thái

- Đường đi: bắt đầu từ trạng thái bắt đầu của máy và đi theo các dịch chuyển từ trạng thái này sang trạng thái khác.
- Máy trạng thái hữu hạn: với mỗi trạng thái hay sự kiện, có một đáp ứng duy nhất.
- Ví dụ: sơ đồ trạng thái máy cho bài toán cửa quay

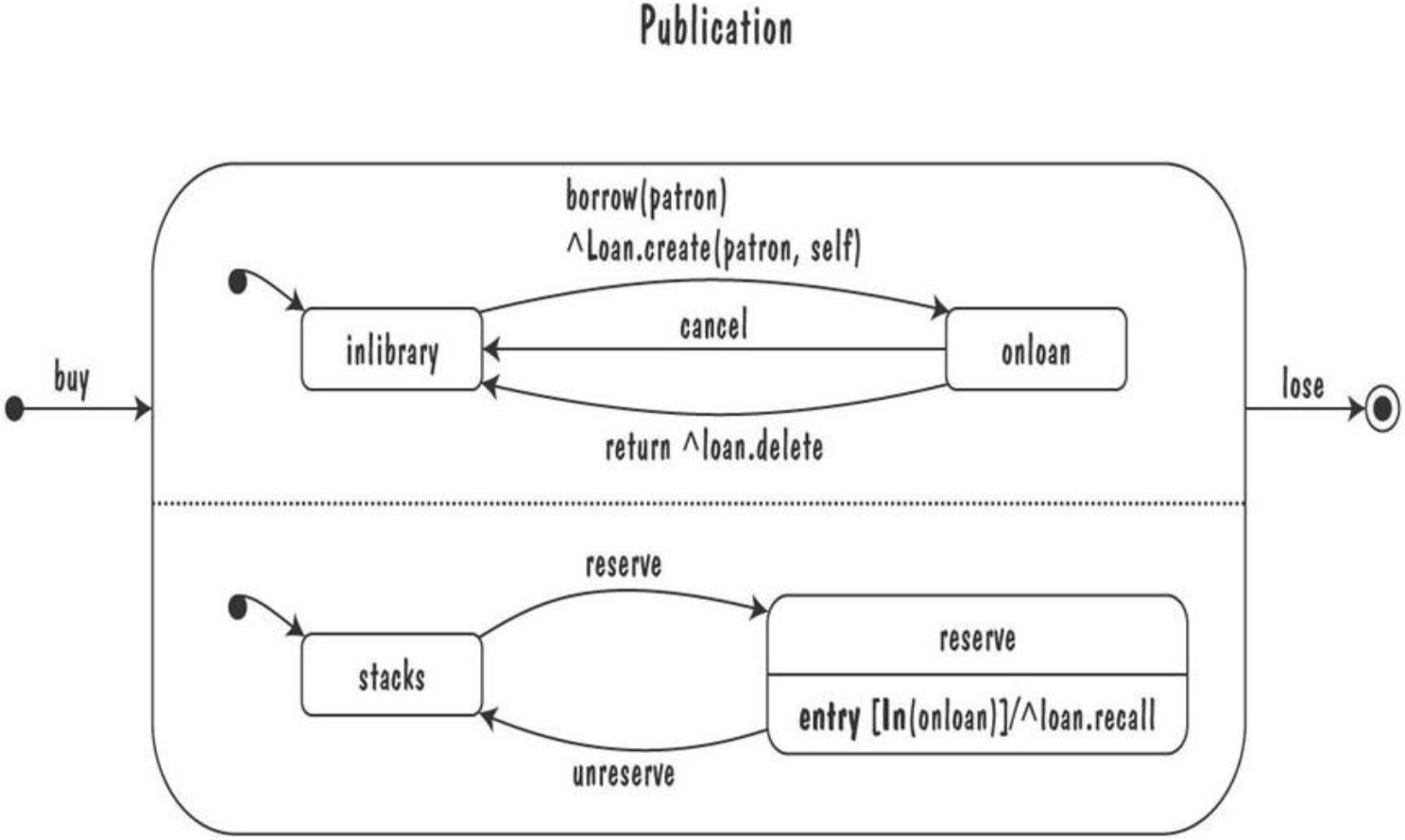


Mô thức ký hiệu - Máy trạng thái

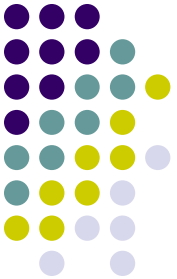
- Ví dụ: sơ đồ trạng thái UML
  - Mô tả hành vi động của các đối tượng trong một lớp UML.
  - Có cú pháp phong phú, bao gồm sự phân cấp trạng thái, sự đồng thời và giao tiếp giữa các máy.

Mô thức ký hiệu - Máy trạng thái

Sơ đồ trạng thái của UML cho lớp `Publication`

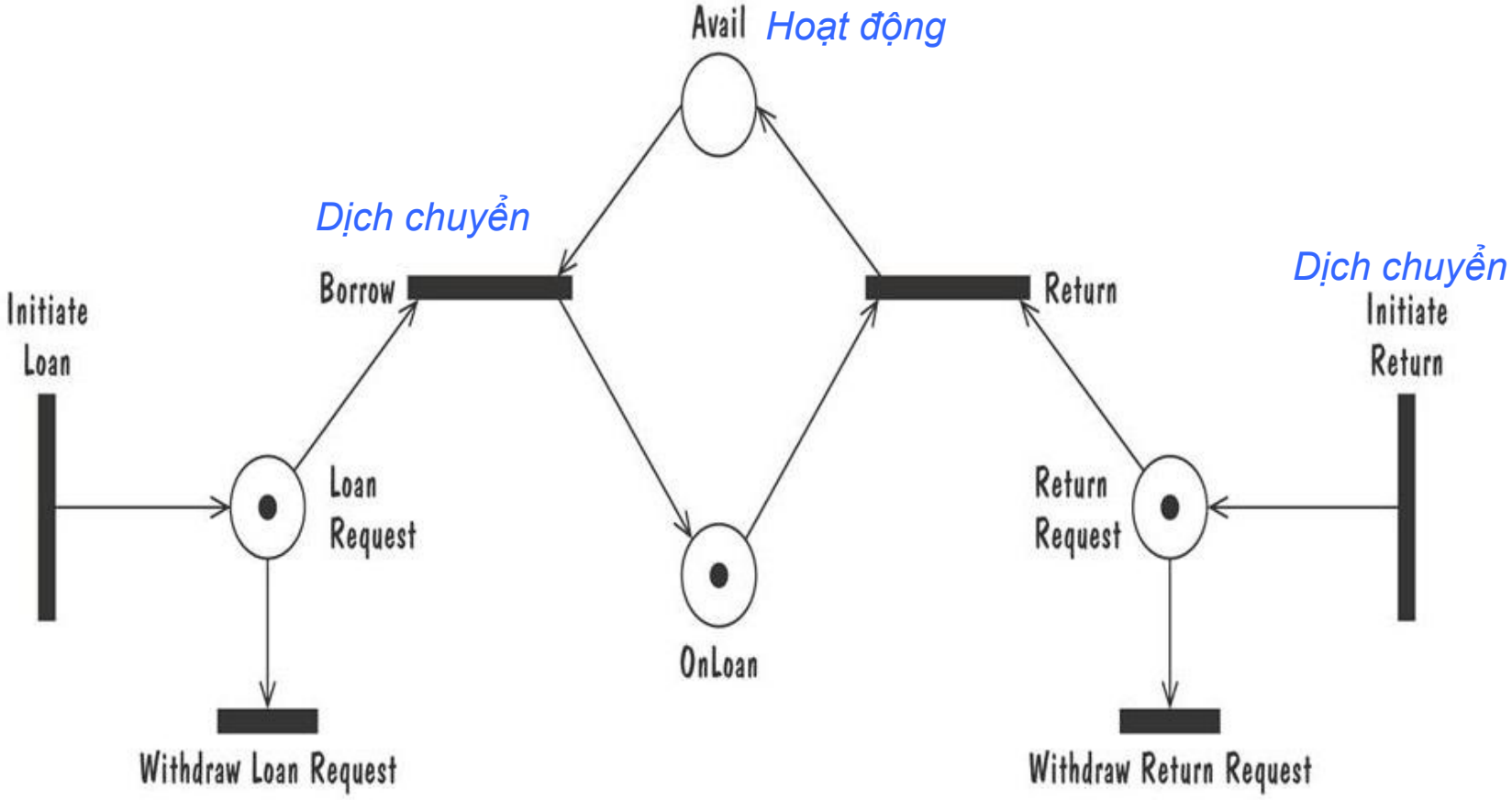


Mô thức ký hiệu - Máy trạng thái

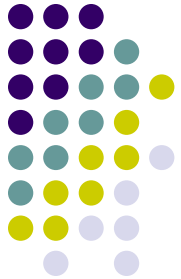


- Ví dụ: mạng Petri
  - Ký pháp dịch chuyển trạng thái được sử dụng để mô hình hóa các hoạt động đồng thời và sự tương tác của chúng.
    - *Vòng tròn* biểu diễn cho các *hoạt động* hay các *điều kiện*.
    - *Thanh ngang/ dọc* biểu diễn cho các *dịch chuyển*.
    - *Cung* nối kết một dịch chuyển với các hoạt động và điều kiện vào và ra của nó.

Mô thức ký hiệu - Máy trạng thái

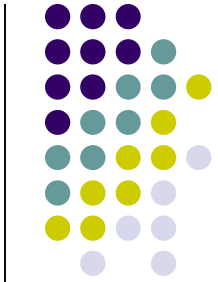


Mô thức ký hiệu - Lưu đồ dòng dữ liệu



- Lược đồ thực thể quan hệ phân rã vấn đề theo các thực thể.
- Dò theo sự kiện phân rã vấn đề theo kịch bản.
- Máy trạng thái phân rã vấn đề theo trạng thái điều khiển.
- Cả ba chỉ mô tả các hành vi mức thấp => rất khó nhìn thấy chức năng mức cao của mô hình.

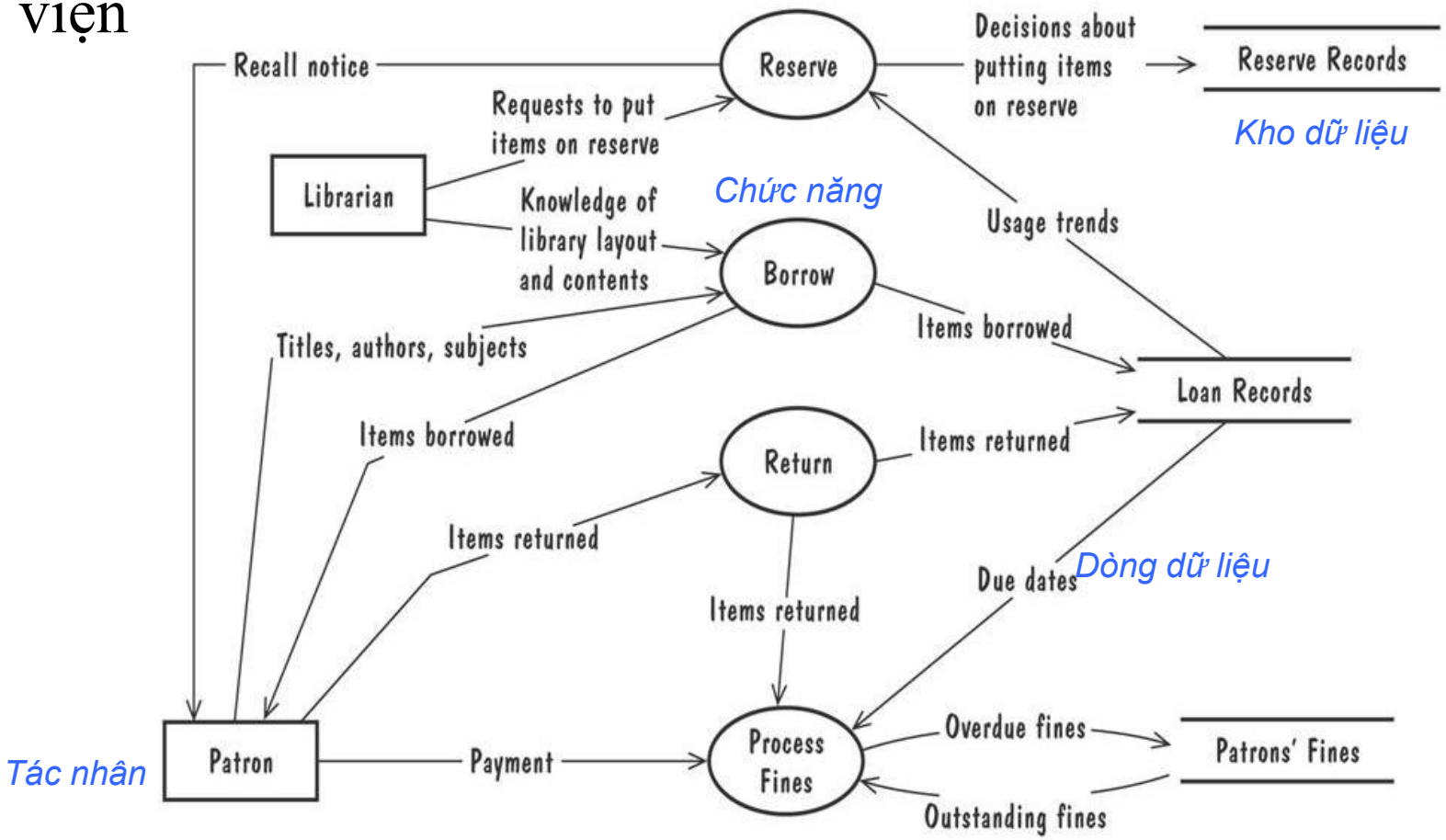
Mô thức ký hiệu - Lưu đồ dòng dữ liệu



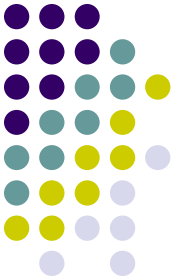
- Lưu đồ dòng dữ liệu mô hình hóa chức năng và dòng dữ liệu từ chức năng này sang chức năng khác.
  - *Hình elip* biểu diễn cho *quy trình* hay *chức năng*: biến đổi dữ liệu
  - *Mũi tên* biểu diễn cho *dòng dữ liệu* (vào hay ra của chức năng).
  - *Hai đường song song* biểu diễn cho *kho dữ liệu*: lưu giữ các dữ liệu.
  - *Hình chữ nhật* biểu diễn cho các *tác nhân*: các thực thể cung cấp dữ liệu vào và nhận kết quả kết xuất.

Mô thức ký hiệu - Lưu đồ dòng dữ liệu

- Lưu đồ dòng dữ liệu mức cao biểu diễn cho bài toán thư viện

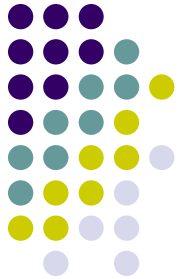


Mô thức ký hiệu - Lưu đồ dòng dữ liệu



- Thuận lợi:
  - Cung cấp một mô hình trực quan về chức năng mức cao của hệ thống được đề nghị và những phụ thuộc dữ liệu giữa các quy trình khác nhau.
- Khó khăn :
  - Có thể làm tăng tính mơ hồ đối với người phát triển phần mềm, người chưa quen với vấn đề đang được mô hình hóa.

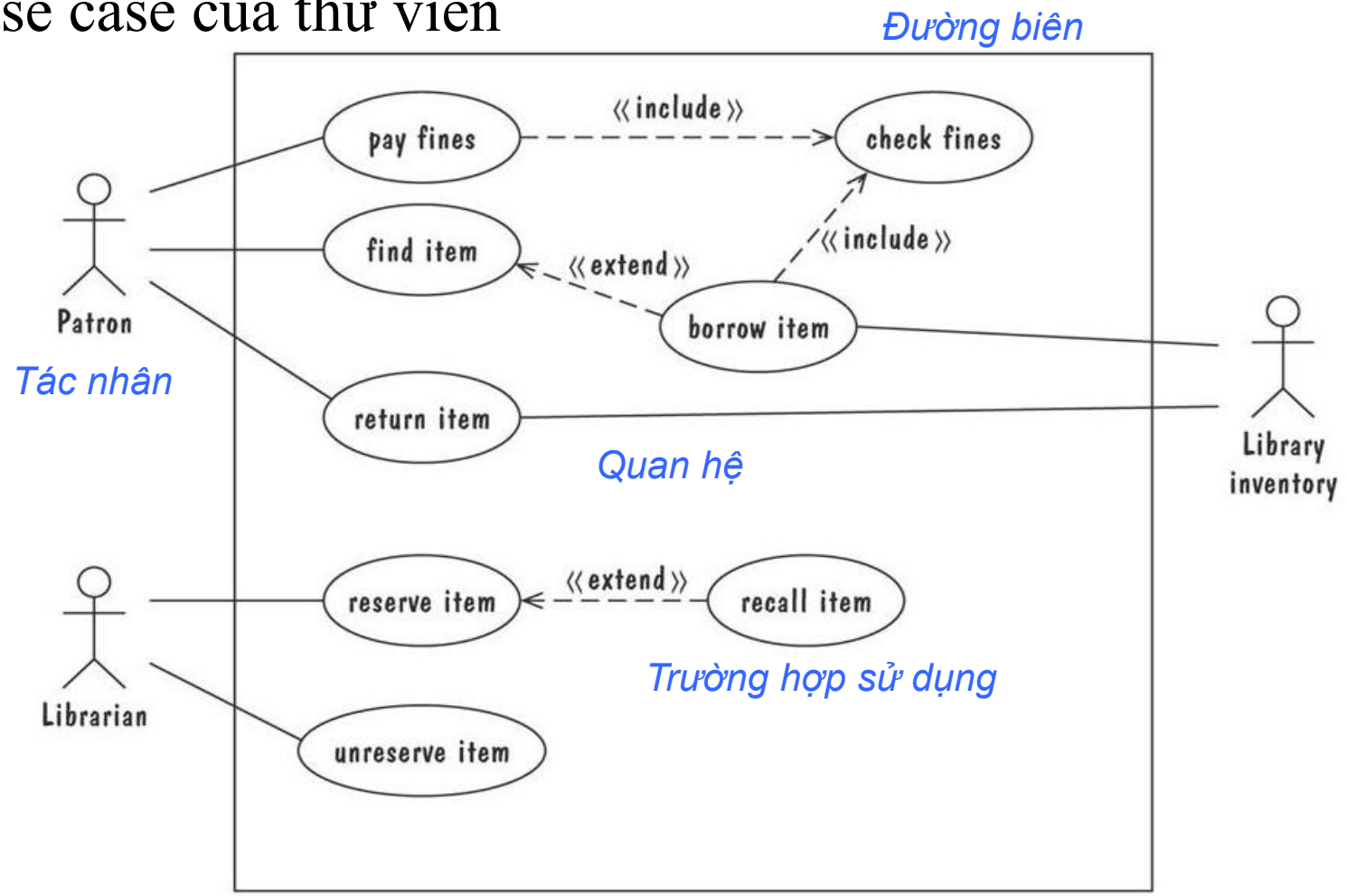
Mô thức ký hiệu - Lưu đồ dòng dữ liệu



- Ví dụ: sơ đồ trường hợp sử dụng UML
  - Các thành phần
    - *Đường biên* của hệ thống (được ký hiệu bởi *hình chữ nhật*)
    - *Tác nhân* (được ký hiệu bởi *hình người* hay << >>)
    - *Trường hợp sử dụng* – use case - (được ký hiệu bằng *hình oval*). Một use case biểu diễn một chức năng được yêu cầu nào đó và biến thể của nó
    - *Quan hệ* giữa tác nhân và use case hay giữa các use case (được biểu diễn bằng các *đường liền hay đường nét đứt*)

Mô thức ký hiệu - Lưu đồ dòng dữ liệu

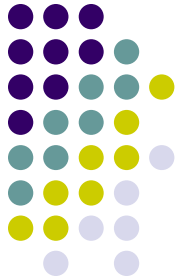
Một số use case của thư viên



Mô thức ký hiệu – Hàm và quan hệ

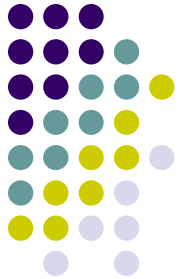
- **Phương thức hay cách tiếp cận hình thức:** các kỹ thuật thiết kế và đặc tả dựa vào toán học.
- Phương thức hình thức mô hình hóa các yêu cầu hay hoạt động phần mềm như một tập các hàm hay quan hệ toán học, chúng ánh xạ từ đầu vào của hệ thống tới đầu ra của hệ thống.
  - *Hàm*: xác định trạng thái của sự thực thi của hệ thống, và các kết xuất.
  - *Quan hệ*: được sử dụng khi 1 giá trị nhập ánh xạ tới nhiều hơn 1 giá trị xuất.
- Ví dụ: bảng quyết định

Mô thức ký hiệu – Logic, Đặc tả đại số



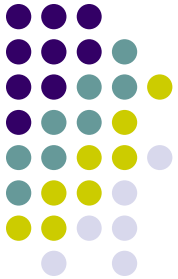
- **Logic**
  - Một logic bao gồm một ngôn ngữ diễn tả các thuộc tính và một tập các quy tắc suy luận ra các thuộc tính kết quả, mới từ những thuộc tính đã có.
  - Ví dụ: ngôn ngữ đặc tả yêu cầu hình thức Z
- **Đặc tả đại số**
  - Xác định hành vi của các phép toán bằng cách xác d9inh6 sự tương tác giữa các cặp phép toán thay vì mô hình hóa các phép toán riêng lẻ.
  - Khó có thể định nghĩa được tập các tiên đề mà nó là hoàn chỉnh, nhất quán và phản ánh hành vi mong muốn.
  - Ví dụ: SDL Data

Các ngôn ngữ đặc tả và yêu cầu



- Ngôn ngữ mô hình hóa hợp nhất (Unified Modeling Language - UML)
- Ngôn ngữ mô tả và đặc tả (Specification and Description Language – SDL)

Các ngôn ngữ đặc tả và yêu cầu

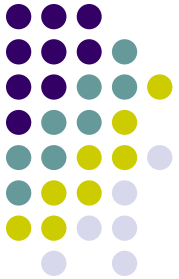


- UML (Unified Modeling Language)
  - Kết hợp nhiều sơ đồ ký hiệu.
  - Các sơ đồ UML được sử dụng trong suốt quá trình định nghĩa và đặc tả các yêu cầu.
    - Sơ đồ trường hợp sử dụng (Lưu đồ dòng dữ liệu mức cao)
    - Sơ đồ lớp (Lưu đồ thực thể quan hệ)
    - Sơ đồ tuần tự (Dò theo sự kiện)
    - Sơ đồ cộng tác (Dò theo sự kiện)
    - Sơ đồ trạng thái (Máy trạng thái)

Các ngôn ngữ đặc tả và yêu cầu

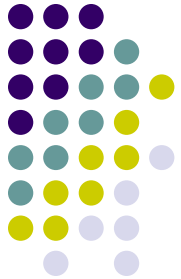
- Ngôn ngữ mô tả và đặc tả (SDL)
  - Xác định hành vi của các quy trình phân tán, đồng thời và thời gian thực mà chúng giao tiếp với nhau thông qua các hàng đợi thông điệp không giới hạn.
  - Bao gồm:
    - Sơ đồ hệ thống SDL (Lưu đồ dòng dữ liệu)
    - Sơ đồ khối SDL (Lưu đồ dòng dữ liệu)
    - Sơ đồ quy trình SDL (Máy trạng thái)
    - Kiểu dữ liệu SDL (Đặc tả đại số)
  - Thường được đi kèm bởi một tập sơ đồ tuần tự của thông điệp.

Lập bản mẫu cho các yêu cầu



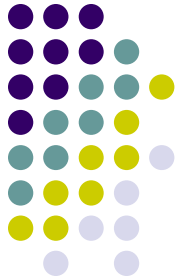
- Xây dựng bản mẫu
  - Để thu thập các chi tiết của hệ thống được đề nghị
  - Để cố gắng lấy được thông tin phản hồi từ người dùng tiềm năng về:
    - Những khía cạnh mà họ muốn cải tiến.
    - Những đặc tính là không quá hữu ích.
    - Chức năng đang thiếu.
  - Giúp xác định xem vấn đề của khách hàng có giải pháp khả thi hay không.
  - Hỗ trợ trong việc thăm dò các chọn lựa để tối ưu hóa những yêu cầu về chất lượng.

Lập bản mẫu cho các yêu cầu



- Các cách tiếp cận để làm bản mẫu
  - Cách tiếp cận được làm ra để sử dụng một lần duy nhất (throwaway prototype)
    - Được phát triển để nghiên cứu thêm về vấn đề hay giải pháp được đề nghị , và không bao giờ được xem là một phần của phần mềm được phân phối.
  - Cách tiếp cận tiến hóa (evolutionary prototype)
    - Được phát triển không chỉ giúp chúng ta trả lời các câu hỏi mà còn được kết hợp vào sản phẩm cuối cùng.
    - Bản mẫu cuối cùng phải biểu thị các yêu cầu về chất lượng của sản phẩm cuối.
  - Cả hai kỹ thuật này đôi khi được gọi là làm bản mẫu nhanh.

Lập bản mẫu cho các yêu cầu

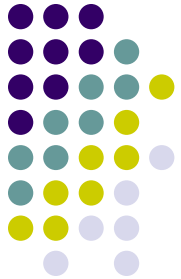


- Sự khác nhau giữa lập bản mẫu và mô hình hóa
  - Lập bản mẫu
    - Tốt khi trả lời câu hỏi về giao diện người dùng.
  - Mô hình hóa
    - Trả lời nhanh câu hỏi về những ràng buộc thứ tự mà theo đó các sự kiện nên xuất hiện, về sự đồng bộ của các hoạt động.

Tài liệu yêu cầu

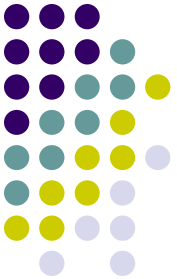
- Các loại tài liệu yêu cầu
  - **Định nghĩa các yêu cầu**: một danh sách hoàn chỉnh về những thứ mà khách hàng muốn đạt được
    - Mô tả các thực thể trong môi trường nơi hệ thống sẽ được cài đặt (các thực thể trong thế giới thực của khách hàng).
    - Mô tả các phép biến đổi hay các ràng buộc lên các thực thể đó.
  - **Đặc tả các yêu cầu**: diễn tả lại các yêu cầu như một đặc tả về cách mà hệ thống được đề nghị sẽ hoạt động
    - Chỉ tham khảo tới các thực thể mà hệ thống có thể truy xuất chúng qua giao diện của hệ thống (chỉ các thực thể trong thế giới thực mà chúng có trong hệ thống được đề nghị).

Tài liệu yêu cầu



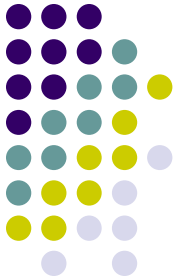
- Định nghĩa các yêu cầu - Các bước của quy trình
  - Phác thảo mục đích chung và phạm vi của hệ thống, bao gồm các lợi ích liên quan, các mục tiêu và mục đích.
  - Mô tả nền tảng và nhân tố cơ bản ẩn sau sự đề xuất một hệ thống mới.
  - Mô tả những đặc trưng cần thiết của một giải pháp có thể chấp nhận.
  - Mô tả môi trường trong đó hệ thống sẽ vận hành.
  - Mô tả phác thảo về đề xuất giải quyết vấn đề của khách hàng (nếu khách hàng có đề xuất).
  - Liệt kê các giả thiết về cách thức môi trường hoạt động.

Tài liệu yêu cầu



- Đặc tả các yêu cầu - Các bước của quy trình
  - Mô tả chi tiết tất cả các đầu vào, đầu ra, bao gồm:
    - Các nguồn của đầu vào
    - Các đích của đầu ra
    - Các miền giá trị
    - Định dạng dữ liệu cho dữ liệu vào/ra
    - Các giao thức của dữ liệu
    - Tổ chức và định dạng của cửa sổ
    - Ràng buộc thời gian
  - Diễn đạt lại chức năng được yêu cầu dưới dạng các đầu vào/ra của giao diện.
  - Đưa ra tiêu chuẩn phù hợp cho từng yêu cầu về chất lượng của khách hàng.

Tài liệu yêu cầu



- Chuẩn IEEE cho đặc tả các yêu cầu phần mềm

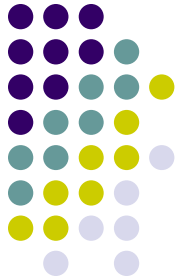
1.Introduction to the Document

- 1.1 Purpose of the Product
- 1.2 Scope of the Product
- 1.3 Acronyms, Abbreviations, Definitions
- 1.4 References
- 1.5 Outline of the rest of the SRS

2.General Description of Product

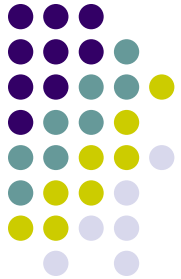
- 2.1 Context of Product
- 2.2 Product Functions
- 2.3 User Characteristics
- 2.4 Constraints
- 2.5 Assumptions and Dependencies

Tài liệu yêu cầu



3. Specific Requirements
   - 3.1 External Interface Requirements
     - 3.1.1 User Interfaces
     - 3.1.2 Hardware Interfaces
     - 3.1.3 Software Interfaces
     - 3.1.4 Communications Interfaces
   - 3.2 Functional Requirements
     - 3.2.1 Class 1
     - 3.2.2 Class 2
     - …
   - 3.3 Performance Requirements
   - 3.4 Design Constraints
   - 3.5 Quality Requirements
   - 3.6 Other Requirements
4. Appendices

Kiểm tra và xác nhận

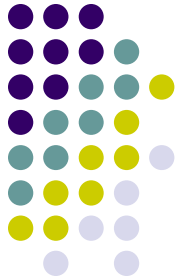


- Xác nhận (công nhận hợp lệ, validation) các yêu cầu: kiểm tra xem định nghĩa các yêu cầu có phản ánh chính xác nhu cầu của khách hàng.
- Kiểm tra (verification) các yêu cầu: kiểm tra xem một tài liệu được tạo ra có tương hợp với tài liệu khác. Tại mức yêu cầu, ta kiểm tra xem đặc tả yêu cầu có phù hợp với định nghĩa yêu cầu.

Kiểm tra và xác nhận

- Các kỹ thuật xác nhận
  - Walkthroughs
  - Reading
  - Interviews
  - Reviews
  - Checklists
  - Models to check functions and relationships
  - Scenarios
  - Prototypes
  - Simulation
  - Formal inspections

Kiểm tra và xác nhận

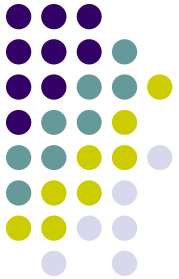


- Kiểm tra
  - Kiểm tra xem tài liệu đặc tả các yêu cầu có tương hợp với định nghĩa các yêu cầu.
  - Đảm bảo rằng nếu ta thực hiện một hệ thống mà nó đáp ứng sự đặc tả thì hệ thống sẽ đáp ứng các yêu cầu của khách hàng.
  - Đảm bảo rằng mỗi yêu cầu trong tài liệu định nghĩa là có thể theo vết trong đặc tả.

Kiểm tra và xác nhận

- Các kỹ thuật kiểm tra

Cross-referencing
Simulation
Consistency checks
Completeness checks
Check for unreachable states or transitions
Model checking
Mathematical proofs



HẾT PHẦN II.1