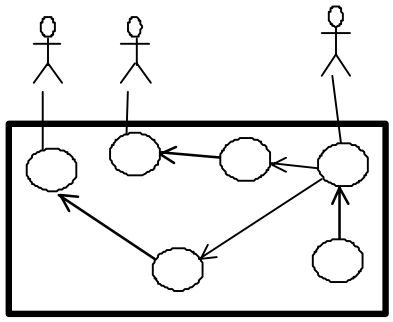
Lecture 10:
Yêu cầu phi chức năng
Non-Functional Requirements (NFRs)

- **Tiền khái niệm:**
  - Các dạng ký pháp mô hình hóa mà chúng ta đã biết
- **Yêu cầu phi chức năng (NFRs) là gì ?**
  - Các hệ số chất lượng, tiêu chuẩn thiết kế; các độ đo
  - Ví dụ về NFRs
- **Tiếp cận hướng sản phẩm (Product-oriented) với NFRs**
  - Tạo ra sự đặc tả các hệ số chất lượng
  - Ví dụ: Sự tin cậy
- **Tiếp cận hướng tiến trình (Process-oriented) với NFRs**
  - Phân tích mục tiêu linh động (softgoal) cho các thỏa hiệp trong thiết kế

Các dạng biểu đồ trong UML…



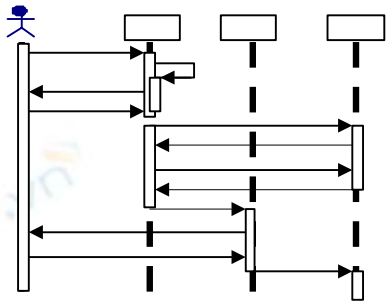
Use Cases

Khía cạnh từ người dùng

Liệt kê trực quan các

chức năng tổng quan của

các yêu cầu chính



Biểu đồ trình tự

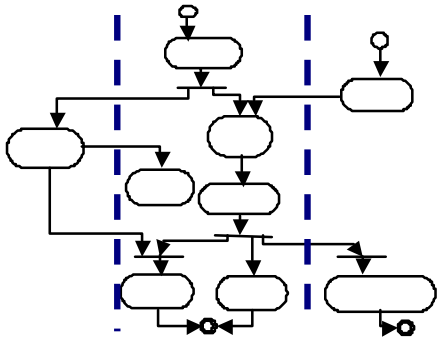
Kịch bản cụ thể

giao tiếp giữa những

người dùng và hệ thống

Trình tự của việc trao

đổi các thông báo



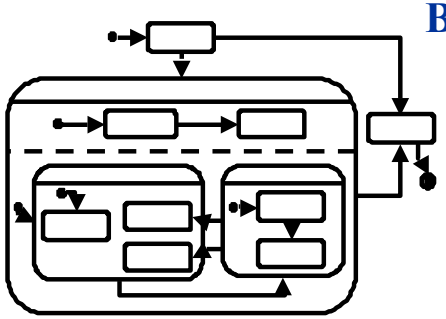
Biểu đồ hoạt động

Tiến trình hoạt động

đồng thời và

đồng bộ phụ thuộc

giữa các công việc



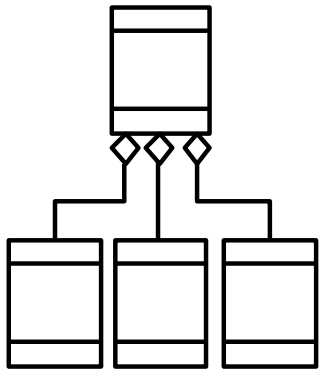
Biểu đồ chuyển trạng thái

Các đáp ứng theo sự

kiện của một đối tượng,

mô hình hóa hành vi

của một lớp giao diện.



Biểu đồ lớp

Cấu trúc thông tin

quan hệ giữa các

lớp, giao diện, hợp tác.

Thể hiện mặt tĩnh

của hệ thống.

…và những biểu đồ không thuộc - UML:

- Mô hình mục tiêu (Goal Models)
  - Nắm bắt các mục tiêu chiến lược của các đối tác
  - Tốt cho việc khảo sát các câu hỏi 'how' và 'why' với các đối tác
  - Tốt để phân tích các thỏa hiệp trade-offs, đặc biệt trên các chọn lựa thiết kế
- Mô hình Cây bắt lỗi (Fault Tree Models) - như một ví dụ trong kỹ thuật phân tích rủi ro
  - Nắm bắt các lỗi tiềm ẩn của một hệ thống và nguồn gốc nguyên nhân
  - Tốt cho phân tích rủi ro, đặc biệt trong những ứng dụng với tiêu chuẩn an toàn
- Mô hình chiến lược phụ thuộc (Strategic Dependency Models (i*))
  - Nắm bắt quan hệ giữa các tác nhân trong một tổ chức
  - Hữu ích cho quan hệ giữa mục tiêu đối tác với tổ chức thiết lập chúng
  - Tốt cho việc thấu hiểu tổ chức sẽ thay đổi như thế nào
- Mô hình quan hệ - thực thể (Enntity-Relationship Models)
  - Nắm bắt quan hệ về cấu trúc thông tin được lưu trữ
  - Tốt cho việc hiểu các ràng buộc và những giả thiết về phạm vi chủ thể
  - Lập nền tảng tốt cho thiết kế CSDL
- Các kiểu bảng lớp (Class Tables), bảng sự kiện (Event Tables) và bảng điều kiện (Condition Tables (SCR))
  - Nắm bắt hành vi động của một hệ thống phản ứng trong thực tế
  - Tốt cho việc biểu diễn chức năng kết hợp từ inputs đến outputs
  - Tốt cho việc tạo các mô hình hành vi chính xác, như suy diễn tự động

Yêu cầu phi chức năng là gì?

Chức năng vs. Phi chức năng

- Các yêu cầu chức năng mô tả cái hệ thống sẽ làm
  - Các chức năng có thể nắm bắt trong use cases
  - Các hành vi có thể được phân tích bằng việc vẽ biểu đồ trình tự, biểu đồ trạng thái, etc.
  - … và khả năng lần vết để giải quyết những vấn đề rắc rối của một chương trình
- Các yêu cầu phi chức năng là những ràng buộc toàn thể trên hệ thống phần mềm
  - e.g. chi phí phát triển, chi phí vận hành, khả năng thực thi, độ tin cậy, khả năng bảo trì, tính khả chuyển, tính thiết thực, etc.
  - Thường được biết như chất lượng phần mềm, hoặc chỉ là "các khả năng" ("ilities")
  - Thường không được cài đặt trong một mô-đun duy nhất của chương trình

Những trở ngại của NFRs

- Khó để mô hình
- Thường ở trạng thái không hình thức, và vì thế mà:
  - Thường mâu thuẫn,
  - Khó thực hiện trong suốt quá trình phát triển
  - Khó đánh giá khách hàng nào ưu tiên để phân phối
- Khó tạo ra các tiêu chuẩn để có thể đo lường chúng
  - Chúng ta muốn ổn định chúng theo cách có thể đo lường được sẽ đáp ứng chúng như thế nào.

Ví dụ về NFRs

Yêu cầu giao diện

- Giao diện của hệ thống mới sẽ như thế nào trong môi trường của nó?
  - Giao diện người dùng "thân thiện"
  - Giao diện đối với các hệ thống khác

Yêu cầu thực thi

- Giới hạn về thời gian / không gian
  - Thời gian tải nạp, thời gian đáp ứng, kích thước dữ liệu nhập và không gian lưu trữ
  - e.g. "hệ thống phải kiểm soát 1000 giao dịch trên giây"
- Độ tin cậy
  - Tính sẵn dùng của các thành phần
  - Sự nguyên vẹn của thông tin dùng duy trì và cung cấp cho hệ thống
  - e.g. "hệ thống phải có ít hơn 1 giờ đình trệ hoạt động trong 3 thángs"
- Tính bảo mật
  - E.g. Cho phép thông tin lưu hành, hoặc phân quyền người dùng
- Khả năng chịu lỗi
  - E.g. Hệ thống sẽ cần năng lực tồn tại, chịu đựng các sự cố tự nhiên, etc

Yêu cầu vận hành

- Các ràng buộc vật lý (kích thước, trọng lượng),
- Mức kỹ năng & khả năng nhân sự
- Dễ bảo trì
- Các điều kiện về môi trường
- etc

Yêu cầu chu trình sống

- "Future-proofing"
  - Khả năng bảo trì
  - Khả năng mở rộng
  - Tính khả chuyển
  - Thị trường mong đợi hoặc vòng đời sản phẩm
- Những giới hạn phát triển
  - E.g giới hạn thời gian phát triển,
  - Tài nguyên sẵn dùng
  - Các chuẩn về phương pháp
  - etc.

Yêu cầu kinh tế

- e.g. giới hạn nghiêm ngặt đúng thời gian và/hoặc vốn dài hạn.

Tiếp cận NFRs

Sản phẩm vs. Tiến trình?

- Tiếp cận hướng sản phẩm (Product-oriented Approaches)
  - Tập trung vào chất lượng của hệ thống (hoặc phần mềm)
  - Nắm bắt các tiểu chuẩn thiết lập của mỗi yêu cầu
  - … để mà chúng ta có thể đo lường chúng khi sản phẩm được thiết kế
- Tiếp cận hướng tiến trình (process-oriented Approaches)
  - Tập trung vào các yêu cầu phi chức năng (NFRs) nào có thể dùng trong tiến trình thiết kế
  - Phân tích tương tác giữa NFRs và các chọn lựa thiết kế
  - … để mà chúng ta có thể đưa ra các quyết định thiết kế phù hợp

Định lượng (Quantitative) vs. Định tính (Qualitative)?

- Tiếp cận định lượng
  - Tìm thang đo các thuộc tính về chất lượng
  - Tính toán mức độ cho một thiết kế đáp ứng với các mục tiêu chất lượng nào
- Tiếp cận định tính
  - Nghiên cứu các dạng quan hệ giữa các mục tiêu chất lượng
  - Lý do của các sự thỏa hiệp (trade-offs), etc.

Chất lượng phần mềm

- Nghĩ đến một đồ vật thông thường
  - e.g. Một cái ghế – bạn sẽ đo "chất lượng" của nó như thế nào?
    - Chất lượng kết cấu? (e.g. độ chắc của những mối nối,…)
    - Giá trị thẩm mỹ? (e.g. tính thanh lịch,…)
    - Đáp ứng mục tiêu? (e.g. sự thoải mái,…)
- Tất cả các độ đo chất lượng đều có quan hệ
  - Không có thang đo nào tuyệt đối
  - Đôi khi chúng ta có thể nói A tốt hơn B…
    - … nhưng thường rất khó để nói tốt hơn thế nào !
- Đối với phần mềm :
  - Chất lượng kết cấu?
    - Phần mềm thì không được chế tạo (mà là phát triển)
  - Giá trị thẩm mỹ?
    - nhưng hầu hết phần mềm thì trực quan
    - giá trị thẩm mỹ là một sự quan tâm bên lề
  - Đáp ứng mục tiêu?
    - Cần phải hiểu rõ mục tiêu