



# Khoa CNTT Kinh tế và TMĐT Bộ môn Công nghệ thông tin

Bài giảng học phần  
Phân tích và Thiết kế Hệ thống thông tin



## Mục đích

- Cung cấp cho sinh viên những khái niệm cơ bản, nguyên lý chung trong phân tích thiết kế hệ thống.
- Cung cấp kiến thức phân tích thiết kế hệ thống theo cách tiếp cận hướng đối tượng và sử dụng ngôn ngữ UML  
→ Sinh viên có thể áp dụng trong một số bài toán đời sống như quản lý kinh doanh, dịch vụ,...



## Mục tiêu học phần

- ❑ Explain the systems analyst's role information systems development.
- ❑ Describe the basic systems development life cycle and its phases.
- ❑ Explain how organizations identify IS development projects.
- ❑ Explain the importance of linking the information system to business needs.
- ❑ Be able to create a system request.
- ❑ Describe technical, economic, and organizational feasibility assessment.
- ❑ Be able to perform a feasibility analysis.



## Cấu trúc và nội dung

- **Cấu trúc:** 3 tín chỉ (36,9)
- **Nội dung:**
  - Chương 1. Tổng quan về phân tích & thiết kế HT
  - Chương 2: Ngôn ngữ mô hình hóa và công cụ PTTK
  - Chương 3. Phân tích thiết kế hệ thống theo hướng đối tượng
  - Chương 4: Thiết kế hệ thống theo hướng đối tượng
- **Đánh giá:** thi hết học phần + bài tập lớn



## Tài liệu tham khảo

- Phân tích thiết kế các hệ thống thông tin hiện đại. Nguyễn Văn Vy, Thống Kê, 2002.
- Phân tích và thiết kế hướng đối tượng, Đặng Văn Đức, NXB Giáo Dục, 2002.
- System Analysis and Design - Complete Introductory Tutorial for Software Engineering. <http://www.freetutes.com/systemanalysis>
- Phân tích và thiết kế Hệ thống thông tin với UML. Đặng Văn Đức, NXB Giáo dục



## Chương 1. Tổng quan về PTTKHT

- 1.1. Phương pháp luận về phân tích thiết kế hệ thống
  - 1.1.1. Một số khái niệm cơ bản
  - 1.1.2. Mô hình hóa hệ thống
  - 1.1.3. Quy trình phân tích thiết kế hệ thống
- 1.2. Các hướng tiếp cận trong phân tích thiết kế
  - 1.2.1. Tiếp cận hướng chức năng
  - 1.2.2. Tiếp cận hướng đối tượng
  - 1.2.3. Đánh giá các hướng tiếp cận



## 1.1 Phương pháp luận về PTTKHT

- 1.1.1. Một số khái niệm cơ bản
- 1.1.2. Mô hình hóa hệ thống
- 1.1.3. Quy trình phân tích thiết kế hệ thống



## 1.1.1 Một số khái niệm cơ bản

- **Khái niệm**
  - **Hệ thống**: là tập hợp gồm nhiều *thành phần/đối tượng* có tổ chức và tương tác với nhau nhằm thực hiện các *mục tiêu* chung.
  - **Ví dụ**: hệ thống điều khiển giao thông, hệ thống mạng máy tính
  - **HT mở**: là HT trong đó tồn tại một số thành phần có tương tác với môi trường bên ngoài

**1.1.1 Các khái niệm cơ bản**

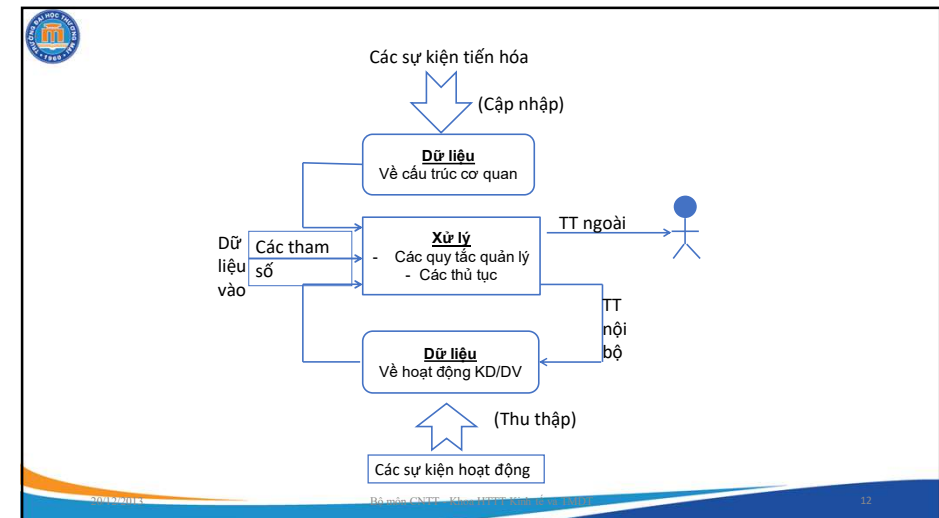
- **Khái niệm**
  - **HT quản lý:** là các phương tiện, biện pháp để theo dõi, kiểm tra và định hướng hoạt động của tổ chức nhằm đạt được mục tiêu đã đề ra
  - **Thành phần**
    - Hệ thống quyết định: xác định mục tiêu mà tổ chức phải vươn tới, tác động lên HT tác vụ để thực hiện mục tiêu đó
    - Hệ thống tác vụ: thực hiện các hoạt động của tổ chức theo chiến lược mà HT quyết định đề ra
    - Hệ thống thông tin: phân tích và cung cấp TT về tình hình của HT tác vụ và chuyển các chỉ thị của HT quyết định cho HT tác vụ
  - **Chú ý:** ranh giới phân chia các thành phần


**1.1.1 Các khái niệm cơ bản**

Mối quan hệ các thành phần trong HT quản lý

**1.1.1 Các khái niệm cơ bản**

- **Vai trò của HTTT:**
  - Thu thập TT
  - Xử lý TT
  - Truyền thông tin
- **Các thành phần của HTTT**
  - Con người và thiết bị
  - Dữ liệu: Là các thông tin được lưu và duy trì nhằm phản ánh thực trạng hiện thời hay quá khứ của DN
  - Các xử lý: Là những quá trình biến đổi thông tin, nhằm:
    - Sinh ra các thông tin theo thể thức quy định
    - Trợ giúp ra các quyết định






## 1.1.1 Các khái niệm cơ bản

- **Phân loại các HTTT:**
  - HT xử lý dữ liệu (DPS – Data Processing System)
  - HTTT quản lý (MIS – Management Information System)
  - HT hỗ trợ quyết định (DSS – Decision Support System)
  - Hệ chuyên gia (ES – Expert System)


13



# Người phân tích hệ thống

Nguyên tắc và kỹ năng


14



## Nguyên tắc phân tích thiết kế Systems Analyst Role

- Key role in developing information systems
  - Analyzing the business situation
  - Identifying opportunities for improvements
  - Designing an information system to implement the improvements


15



## Nguyên tắc phân tích thiết kế Systems Analyst Roles

- Interaction with an array of people
  - Technical specialists (DBAs, network admins, programmers)
  - Business people (users, managers, steering committee)
  - Others (vendors, consultants)
- Variety of specialized roles
  - People-oriented: change management analyst, project management
  - Business-oriented: requirements analyst, business analyst
  - Technically-oriented: infrastructure analyst
  - Generalist: systems analyst


16



## What Do System Analysts Like About Their Work?

- Challenge (Thử thách)
- Technology (Công nghệ)
- Variety (Đa dạng)
- Constant Change (Thường xuyên thay đổi)
- Problem Solving (Giải quyết vấn đề)


17



## What Do System Analysts Dislike About Their Work?

- Management's lack of communication/recognition
- End-user mistakes and demands
- Stress/pressure/burnout
- Ever-changing business technology
- Unrealistic deadlines


18



## Preparing for Your Career

- Working knowledge of information technology
- Computer programming experience & expertise
- General business knowledge
- Problem-solving skills
- Interpersonal communication skills
- Flexibility and adaptability
- Character and ethics
- Systems analysis & design skills


19



## 1.1.2 Mô hình hóa hệ thống

- **Nguyên lý chế ngự sự phức tạp:** Lờ đi những chi tiết không quan trọng
- **Khái niệm mô hình:** là một dạng trừu tượng hóa của một hệ thống thực. Hay mô hình là một biểu diễn của một hệ thống thực, được diễn tả:
  - Ở một mức độ trừu tượng hóa nào đó
  - Theo một quan điểm (góc nhìn) nào đó
  - Bởi một hình thức hiểu được nào đó (văn bản, bảng, đồ thị ...)
- **Khái niệm mô hình hóa:** là việc dùng mô hình để nhận thức và diễn tả một hệ thống

20




## 1.1.2 Mô hình hóa hệ thống

- **Mục đích của mô hình hóa:**
  - Để hiểu
  - Để trao đổi
  - Để hoàn chỉnh
- Hiện nay: PTTKHT sử dụng các mô hình dạng biểu đồ (diagram)
- **Mức độ mô hình hóa HT**
  - Mức logic
  - Mức vật lý

→ Mọi quá trình phát triển hệ thống luôn có hai giai đoạn phân biệt: phân tích và thiết kế


SIG/2011/13      Bộ môn CNTT - Khoa CNTT - Trường ĐHQGHN      21



## 1.1.2 Mô hình hóa hệ thống

- **Các phương diện mô tả hệ thống (bốn trục mô hình hóa)**
  - Mô tả các chức năng HT phải thực hiện
  - Mô tả các đặc điểm tính của hệ thống: các thông tin, các quan hệ
  - Mô tả cách ứng xử của HT
  - Mô tả kiến trúc của HT (các thành phần)
- **Các phương pháp mô hình hóa**
  - Các phương pháp hệ thống
  - Các phương pháp hướng chức năng/ cấu trúc
  - Phương pháp theo sự kiện
  - Các phương pháp hướng dữ liệu
  - Các phương pháp hướng đối tượng


SIG/2011/13      Bộ môn CNTT - Khoa CNTT - Trường ĐHQGHN      22



## 1.1.3. Quy trình phân tích thiết kế hệ thống

- Giai đoạn 1: Khảo sát dự án
- Giai đoạn 2: Phân tích hệ thống
- Giai đoạn 3: Thiết kế
- Giai đoạn 4: Thực hiện
- Giai đoạn 5: Kiểm thử
- Giai đoạn 6: Triển khai và bảo trì

SIG/2011/13      Bộ môn CNTT - Khoa CNTT - Trường ĐHQGHN      23



## Vòng đời phát triển hệ thống (SDLC) The Systems Development Life Cycle

The overall process of systems development

SIG/2011/13      Bộ môn CNTT - Khoa CNTT - Trường ĐHQGHN      24

## Làm thế nào để xây dựng các hệ thống? How Do Systems Get Built?

Systems Development Life Cycle (SDLC)

- Planning
- Analysis
- Design
- Implementation

25

## Pha lập kế hoạch Planning Phase

- Project Initiation
  - Prepare system request
  - Perform preliminary feasibility analysis
- Set Up the Project
  - Project Plan, including work plan & staffing plan

26

## Pha phân tích Analysis Phase

- Determine Analysis Strategy
  - Study existing system and its problems
- Collect and Analyze Requirements
  - Develop new system concept
  - Describe new system with analysis models
- Prepare and Present System Proposal
  - Summarize results of the Analysis Phase
  - Go/No Go decision made by sponsor and steering committee

27

## Pha thiết kế Design Phase

- Determine Design Strategy
  - Build / Buy / Outsource
- Design system components
  - Architecture, interface, database, programs
  - Assemble design elements into System Specification
- Present to steering committee
  - Go / No Go decision before entering final phase

28

**Pha cài đặt**  
**Implementation Phase**

- System Construction
  - Programming and testing
- System Installation
  - Training
  - Conversion to new system
- On-going system support

SỞ THÔNG TIN VÀ TRUYỀN THÔNG      BỘ MÔN CNTT - KHOA HỆ THỐNG THÔNG TIN      29

**1.2 Các hướng tiếp cận trong PTTKHT**

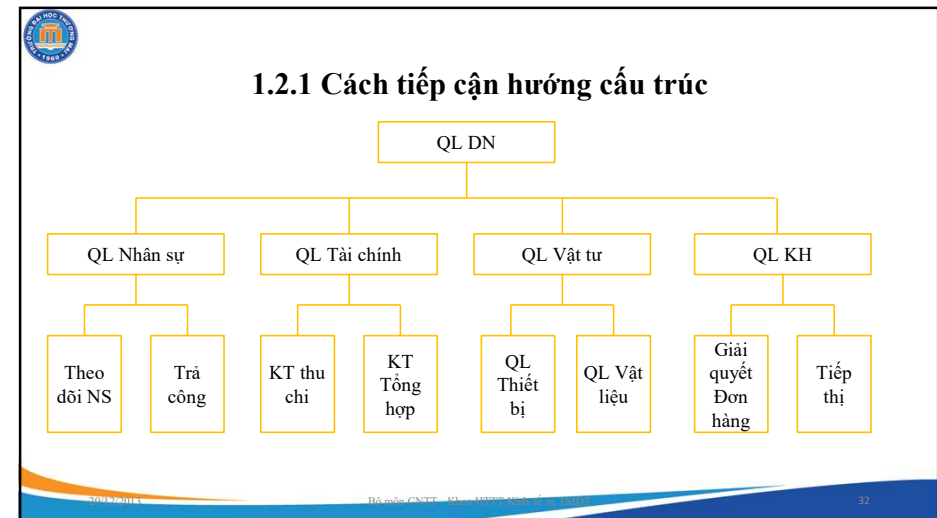
- 1.2.1. Tiếp cận hướng cấu trúc
- 1.2.2. Tiếp cận hướng đối tượng
- 1.2.3. Đánh giá các hướng tiếp cận

SỞ THÔNG TIN VÀ TRUYỀN THÔNG      BỘ MÔN CNTT - KHOA HỆ THỐNG THÔNG TIN      30

**1.2.1 Cách tiếp cận hướng cấu trúc**

- **Tư tưởng:** Lấy chức năng làm đơn vị phân rã HT
- **Đặc điểm:**
  - Dựa vào chức năng là chính
  - Phân rã chức năng và làm mịn dần theo cách thực hiện từ trên xuống
  - Các đơn thể chức năng trao đổi với nhau bằng cách truyền tham số hoặc sử dụng dữ liệu chung
  - Tính mở và thích nghi của HT bị hạn chế
  - Khả năng tái sử dụng bị hạn chế và không hỗ trợ cơ chế kế thừa

SỞ THÔNG TIN VÀ TRUYỀN THÔNG      BỘ MÔN CNTT - KHOA HỆ THỐNG THÔNG TIN      31







## 1.2.2 Cách tiếp cận hướng đối tượng

- **Tư tưởng:** Lấy thực thể/ đối tượng là đơn vị phân rã HT
- **Đặc điểm:**
  - Đặt trọng tâm vào dữ liệu
  - Xem HT như là tập các thực thể, đối tượng
  - Các lớp trao đổi với nhau bằng thông điệp
  - Tính mở và thích nghi của HT cao hơn
  - Hỗ trợ sử dụng lại và cơ chế kế thừa.



## 1.2.3. Đánh giá các hướng tiếp cận Tiếp cận hướng cấu trúc

- **Ưu điểm:**
  - Tư duy phân tích thiết kế rõ ràng.
  - Chương trình sáng sủa dễ hiểu.
  - Phân tích được các chức năng của hệ thống
  - Dễ theo dõi luồng dữ liệu.
- **Nhược điểm:**
  - Không hỗ trợ việc sử dụng lại. Các chương trình hướng cấu trúc phụ thuộc chặt chẽ vào cấu trúc dữ liệu và bài toán cụ thể, do đó không thể dùng lại modul nào đó trong phần mềm này cho phần mềm khác với các yêu cầu về dữ liệu khác.
  - Không phù hợp cho phát triển các phần mềm lớn.
  - Khó quản lý mối quan hệ giữa các modul và dễ gây ra lỗi trong phân tích cũng như khó kiểm thử và bảo trì.




## 1.2.3. Đánh giá các hướng tiếp cận Tiếp cận hướng đối tượng

- **Ưu điểm:**
  - Gần gũi với thế giới thực.
  - Tái sử dụng dễ dàng.
  - Đóng gói che giấu thông tin làm cho hệ thống tin cậy hơn.
  - Thừa kế làm giảm chi phí, hệ thống có tính mở cao hơn
  - Xây dựng hệ thống phức tạp
- **Nhược điểm:**
  - Phương pháp này khá phức tạp, khó theo dõi được luồng dữ liệu do có nhiều luồng dữ liệu ở đầu vào. Hơn nữa giải thuật lại không phải là vấn đề trọng tâm của phương pháp này.



## Lĩnh vực áp dụng


- Phương pháp hướng cấu trúc thường phù hợp với nhiều bài toán nhỏ, có luồng dữ liệu rõ ràng, cần phải tư duy giải thuật rõ ràng và người lập trình có khả năng tự quản lý được mọi truy cập đến các dữ liệu của chương trình.
- Phương pháp hướng đối tượng thường được áp dụng cho các bài toán lớn, phức tạp, hoặc có nhiều luồng dữ liệu khác nhau mà phương pháp cấu trúc không thể quản lý được. Khi đó người ta dùng phương pháp hướng đối tượng để tận dụng khả năng bảo vệ giữ liệu ngoài ra còn tiết kiệm công sức và tài nguyên



# Khởi tạo dự án Project Initiation

How projects get started


37



## Các dự án IS đến từ đâu? Where Do IS Projects Come From?

- Fulfill a **business need**
  - Enable a business initiative or strategy
  - Support a merger/acquisition
  - Fix a “point of pain”
  - Utilize a new technology
  - Outgrowth of Business Process Management (BPM)


38



## Quản trị quy trình kinh doanh là gì? What is BPM?

- Business Process Management: A methodology used by organizations to continuously improve end-to-end business processes
  - Internal and cross-organizational processes
  - Benefits include:
    - Enhanced process agility
    - Process alignment with industry “best practices”
    - Increased process efficiencies


39



## Quản trị quy trình kinh doanh là gì? What is BPM? (continued)

- Four-step continuous cycle:
  - Define and map the steps in a business process,
  - Create ways to improve on steps in the process that add value,
  - Find ways to eliminate or consolidate steps in the process that don’t add value,
  - Create or adjust electronic workflows to match the improved process maps.

40




## Xác định nhu cầu kinh doanh trong BPM

### BPM Identifies Business Needs

- ❑ Business Process Automation
  - “Create or adjust electronic workflows to match the improved process maps”
- ❑ Business Process Improvement
  - Study the business processes
  - Create new, redesigned processes to improve the process workflows, and/or
  - Utilize new technologies enabling new process structures
- ❑ Business Process Reengineering
  - Total overhaul of work processes

41




## Chúng ta đã có dự án chưa?

### Do We Have a Project Yet?

- ❑ Strong business need leads to a person or group stepping up as the Project Sponsor
  - Driving force behind project
  - Specifies overall business requirements
  - Determines business value
  - Formally requests a project via the **System Request**


42



## Bài tập

1. Phân tích và thiết kế HTTT Quản lý Nhân sự
2. Phân tích và thiết kế HTTT Quản lý kinh doanh
3. Phân tích và thiết kế HTTT Quản lý Khách sạn.
4. Sinh viên tự chọn hệ thống (và phải được giáo viên duyệt)


43




## Yêu cầu

- Phân tích thiết kế các hệ thống trên theo hướng đối tượng
- Sử dụng hệ thống biểu đồ UML
- Biên bản phân công công việc
- Bản demo: Thiết kế các giao diện và kịch bản sử dụng (optional)

44




## Chương 2. Ngôn ngữ mô hình hóa và công cụ phân tích thiết kế




## Nội dung

- 2.1. Ngôn ngữ mô hình hóa thống nhất UML
  - 2.1.1. Giới thiệu
  - 2.1.2. Các khái niệm cơ bản trong UML
  - 2.1.3. Các biểu đồ trong UML
- 2.2. Một số công cụ phân tích thiết kế
  - 2.2.1. Giới thiệu
  - 2.2.2. Công cụ phân tích
  - 2.2.3. Công cụ thiết kế



## 2.1. Ngôn ngữ mô hình hóa thống nhất UML

- 2.1.1. Giới thiệu
- 2.1.2. Các khái niệm cơ bản trong UML
- 2.1.3. Các biểu đồ trong UML



## 2.1.1. Giới thiệu về UML

- **Giới thiệu:** UML là ngôn ngữ chuẩn để viết kế hoạch chi tiết phần mềm, được dùng để mô hình hóa các hệ thống.
- **Xuất xứ:**
  - T1/1994: Grady Booch & Jim Rumbaugh (UML was developed by Grady Booch, Ivar Jacobson and James Rumbaugh (**The Three Amigos**))
  - T10/1995: Ivar Jacobsson
  - 14/11/1997: UML 1.1
  - T6/2003: UML 2.0
  - T2/2009: UML 2.2
  - T5/2010: UML 2.3
  - T3/2011: UML 2.4
  - T6/2015: UML 2.5

**Tổng quan về UML**

- OMG Standard (Object Management Group Standard)
- The Unified Modeling Language (UML)
- Data modeling, business modeling (work flows), object modeling, and component modeling
- UML aims to be a standard modeling language which can model concurrent and distributed systems
- UML is a de-facto industry standard
- UML models may be automatically transformed to other representations (e.g. Java, PHP)
- UML is extensible, through **profiles**, **stereotypes** and **tagged values**
- A lot of other languages are based on UML (SysML, SoaML)
- Usually default language for a lot of architecture frameworks (MoDAF, DoDAF, NAF)

Bà môn CNTT - Khoa CNTT - Trường ĐHQG TP. HCM

**2.1.2. Các khái niệm cơ bản trong UML**

- **Hướng nhìn (view):** cho phép biểu diễn nhiều hướng nhìn khác nhau của HT trong quá trình phát triển HT

The diagram illustrates four perspectives of a system centered around a central oval labeled "Góc nhìn ca sử dụng (ca sử dụng)". The four perspectives are:

- Góc nhìn TK (lớp, gói, đối tượng)**: Top-left perspective.
- Góc nhìn thực thi (thành phần)**: Top-right perspective.
- Góc nhìn quá trình (trình tự, giao tiếp, máy trạng thái, hoạt động)**: Bottom-left perspective.
- Góc nhìn bố trí (thành phần, bố trí)**: Bottom-right perspective.

Bà môn CNTT - Khoa CNTT - Trường ĐHQG TP. HCM

**Các phần tử mô hình và các quan hệ**

- **Các phần tử mô hình:** là các biểu tượng, khái niệm sử dụng trong mô hình
  - Phần tử cấu trúc: là các danh từ trong mô hình, biểu diễn các thành phần khái niệm hoặc vật lý. VD: lớp, giao diện, nút,...
  - Phần tử hành vi: là các động từ, biểu diễn hành vi theo tg, kg. VD: tương tác, trạng thái
  - Phần tử nhóm: là bộ phận tổ chức của mô hình. Duy nhất: gói
  - Chú thích: là bộ phận chú giải của mô hình. VD:

Bà môn CNTT - Khoa CNTT - Trường ĐHQG TP. HCM

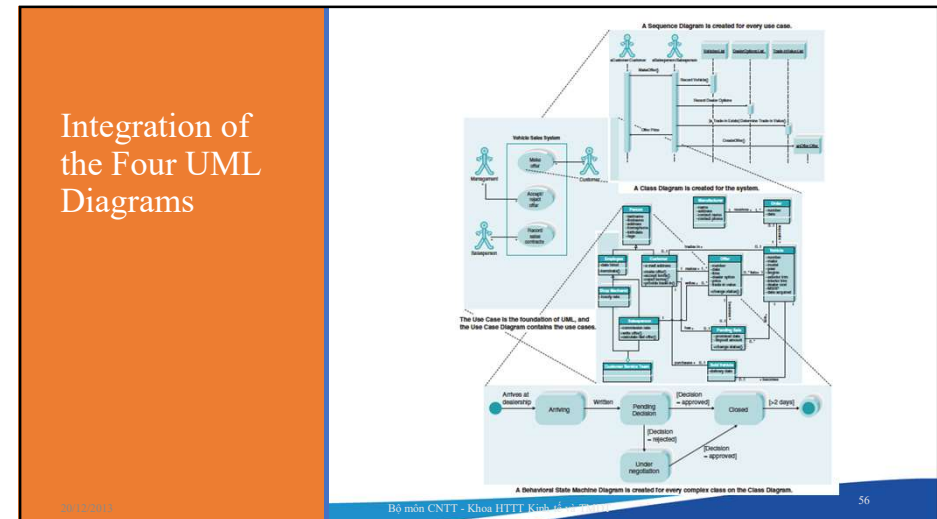
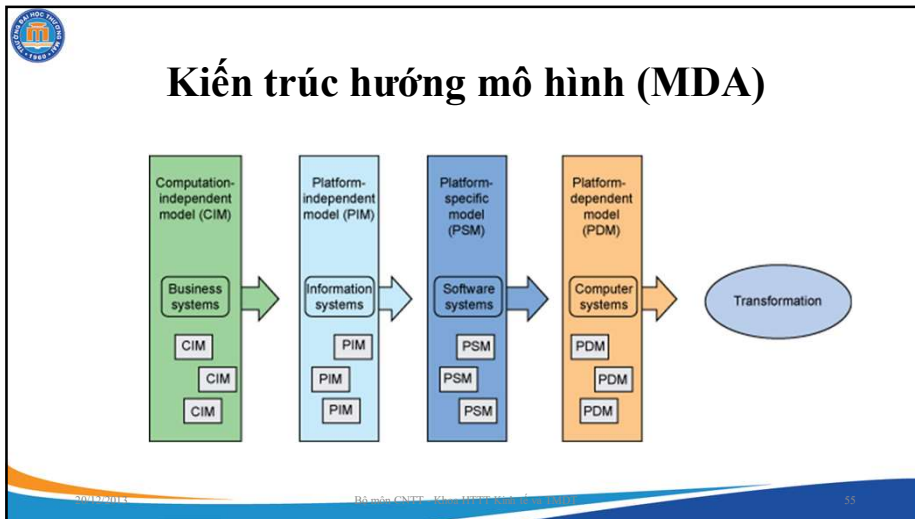
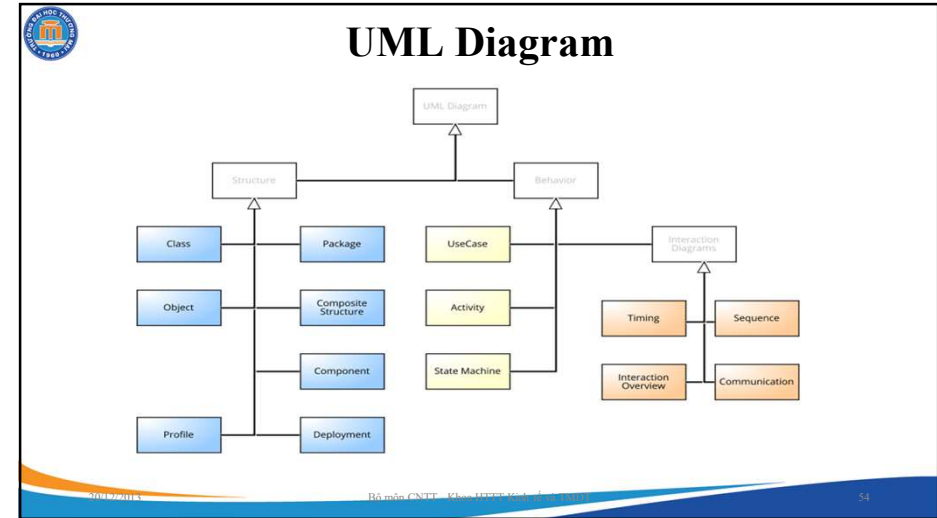
**Các phần tử mô hình và các quan hệ**

- **Các quan hệ:** gắn kết các phần tử lại với nhau
  - Phụ thuộc (dependency): là quan hệ ngữ nghĩa giữa hai phần tử trong đó phần tử độc lập sẽ tác động đến ngữ nghĩa của phần tử phụ thuộc.
  - Kết hợp/ (association): là quan hệ cấu trúc để mô tả tập liên kết giữa các đối tượng. (giữa chúng có sự gửi/ nhận thông điệp) → quan hệ tụ hợp/kết tập (aggregation) → quan hệ hợp thành (composition)
  - Khái quát hóa (generalization): là quan hệ trong đó đối tượng cụ thể sẽ kế thừa các thuộc tính và phương thức của đối tượng tổng quát
  - Hiện thực hóa: là quan hệ ngữ nghĩa giữa giao diện và lớp/thành phần hiện thực, giữa UC và hiện thực UC

Bà môn CNTT - Khoa CNTT - Trường ĐHQG TP. HCM

### 2.1.3. Các biểu đồ trong UML

- Ý nghĩa:** là ánh xạ của hệ thống, dùng để biểu diễn hệ thống đang xây dựng dưới các góc độ quan sát khác nhau thông qua các phần tử của mô hình.
- Biểu đồ Use case
- Biểu đồ lớp
- Biểu đồ trạng thái
- Biểu đồ tương tác
  - Biểu đồ tuần tự
  - Biểu đồ cộng tác
- Biểu đồ hoạt động
- Biểu đồ thành phần
- Biểu đồ triển khai



## Biểu đồ Use case (Use case diagram)

- **Ý nghĩa:**
  - Biểu diễn sơ đồ chức năng của hệ thống.
  - Mỗi usecase mô tả một chức năng mà HT cần phải có xét từ góc độ người dùng.
  - Các biểu đồ usecase có thể phân rã theo nhiều mức khác nhau.
- **Các phần tử mô hình:**
  - Tác nhân:
  - Các use case
  - Mối quan hệ giữa các use case:
    - Include: sử dụng
    - Extend: mở rộng
    - Generalization: kế thừa

Phần tử mô hình	Ý nghĩa	Cách biểu diễn	Ký hiệu trong biểu đồ
Usecase	Biểu diễn một chức năng xác định của hệ thống	Hình ellip chứa tên của các use case	
Tác nhân	Là một đối tượng bên ngoài hệ thống tương tác trực tiếp với các Usecase	Biểu diễn bởi một hình người tượng trưng	
Mối quan hệ giữa các use case	Tùy từng dạng quan hệ	Extend và Include có dạng mũi tên đứt nét, Generalization có dạng mũi tên tam giác	
Biên của hệ thống	Tách biệt phần bên trong và bên ngoài hệ thống	Được biểu diễn bởi một hình chữ nhật rỗng	

## Use cases

- Use cases are used to model the **requirements** of a system (what a system is supposed to do)
- The key concepts associated with use cases are **actors, use cases, and the subject.**
- describes the relationships among a set of use cases and the actors participating in these use cases
- it does NOT describe behavior or flows (**WHO and WHAT, not HOW**)

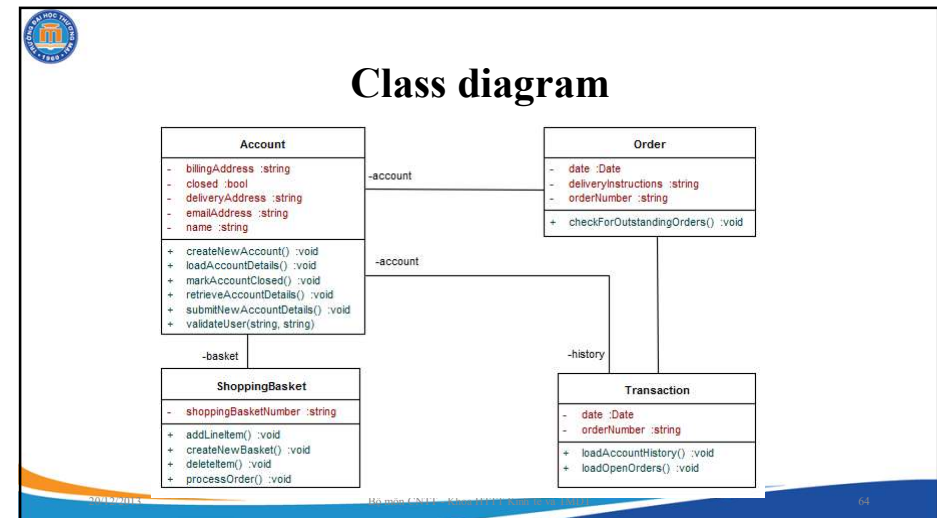
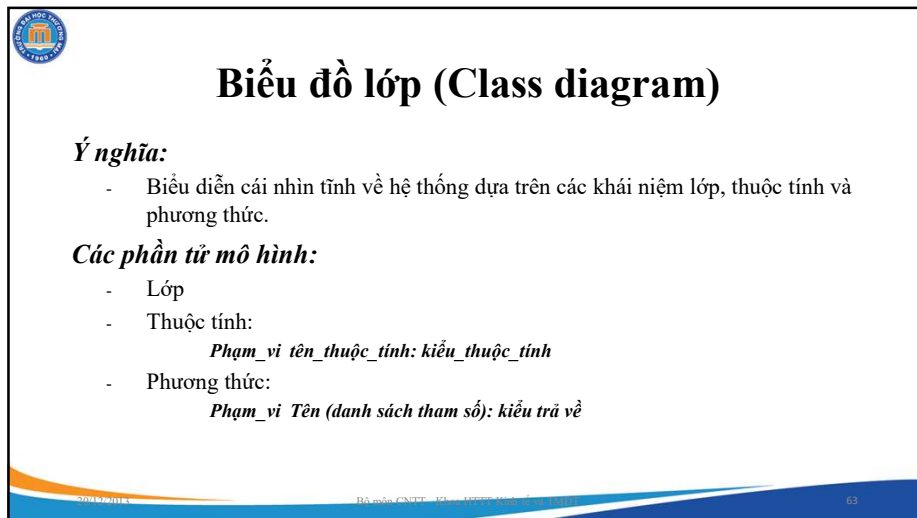
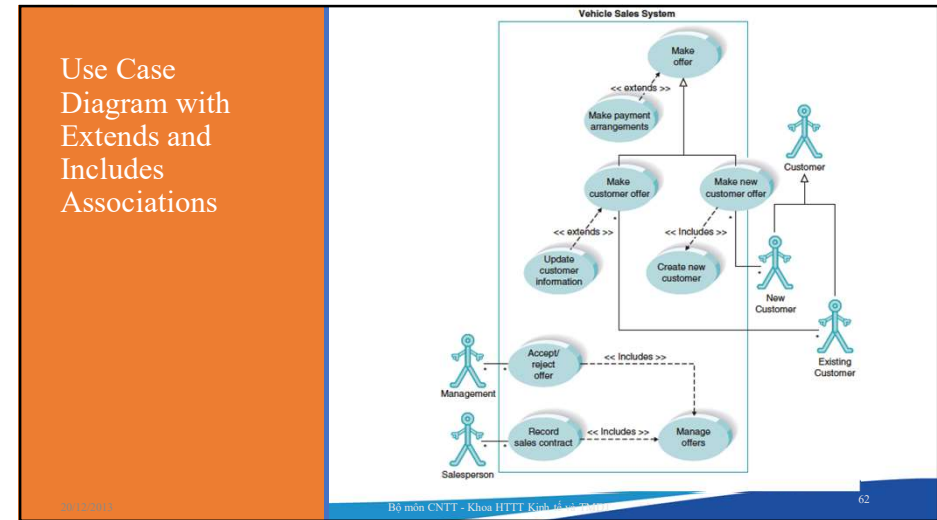
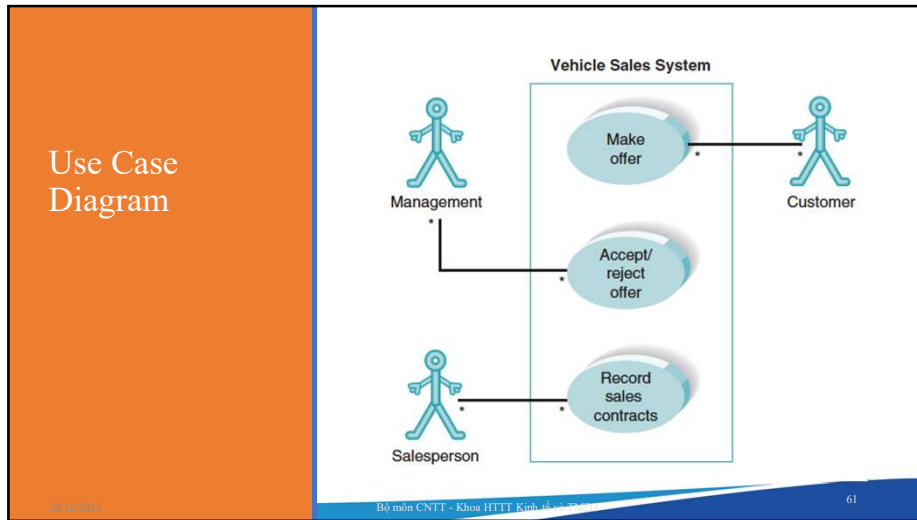
## Chi tiết trong các loại biểu đồ

**Actors**

**Use cases**

**Subjects**







## Biểu đồ lớp

- Các kiểu lớp trong UML**

STT	Kiểu lớp	Kí hiệu UML
1	Lớp thực thể	
2	Lớp biên (lớp giao diện)	
3	Lớp điều khiển	

BỘ MÔN CNTT - KHOA CNTT - ĐẠI HỌC ĐÀ NẴNG

## Biểu đồ lớp

**Các mối quan hệ trong biểu đồ lớp:**

- Khái quát hóa/tổng quát hóa (generalization):
- Quan hệ kết hợp (association):
  - Quan hệ cộng hợp/ tụ hợp (Aggregation):
  - Quan hệ hợp thành (composition):
- Quan hệ phụ thuộc (dependency):

BỘ MÔN CNTT - KHOA CNTT - ĐẠI HỌC ĐÀ NẴNG

# Multiplicity

Instance(s)	Representation of instance(s)	Diagram Involving Instance(s)	Explanation of Diagram
Exactly one	1		A department has one and only one boss.
Zero or more	0..*		An employee has zero to many children.
One or more	1..*		A boss is responsible for one or more employees.
Zero or one	0..1		An employee can be married to zero or one spouse.
Specified range	2..4		An employee can take between two and four vacations each year.
Multiple, disjoint ranges	1..3, 5		An employee is a member of one to three or five committees.

BỘ MÔN CNTT - KHOA CNTT - ĐẠI HỌC ĐÀ NẴNG

## Biểu đồ trạng thái (State Machine diagram)

**Ý nghĩa:**

- Biểu diễn các trạng thái và sự chuyển trạng thái của các lớp.
- Phạm vi của biểu đồ trạng thái?

```

stateDiagram-v2
    [*] --> Idle
    Idle --> Working : card in
    Working --> Idle : done
    Idle --> OutOfService : broken
    OutOfService --> Idle : fixed
    
```

BỘ MÔN CNTT - KHOA CNTT - ĐẠI HỌC ĐÀ NẴNG

## Biểu đồ trạng thái

**Các thành phần của biểu đồ trạng thái:**

- Trạng thái (state)
- Trạng thái con (substate)
- Các kiểu trạng thái
  - Trạng thái bắt đầu
  - Trạng thái kết thúc
  - Các chuyển tiếp (transition)
  - Sự kiện (event)
    - Call event
    - Signal event
    - Time event

69

## Biểu đồ trạng thái

Phần tử mô hình	Ý nghĩa	Biểu diễn	Ký hiệu
Trạng thái	Biểu diễn một trạng thái của đối tượng trong vòng đời của đối tượng đó	Hình chữ nhật vòng ở góc, gồm 3 phần: tên, các biến và các hoạt động	
Trạng thái khởi đầu	Khởi đầu vòng đời của đối tượng	Hình tròn đặc	●
Trạng thái kết thúc	Kết thúc vòng đời của đối tượng	Hai hình tròn lồng nhau	⦿
Chuyển tiếp (transition)	Chuyển từ trạng thái này sang trạng thái khác	Mũi tên liền nét với tên gọi là biểu diễn của chuyển tiếp đó	Tên chuyển tiếp →

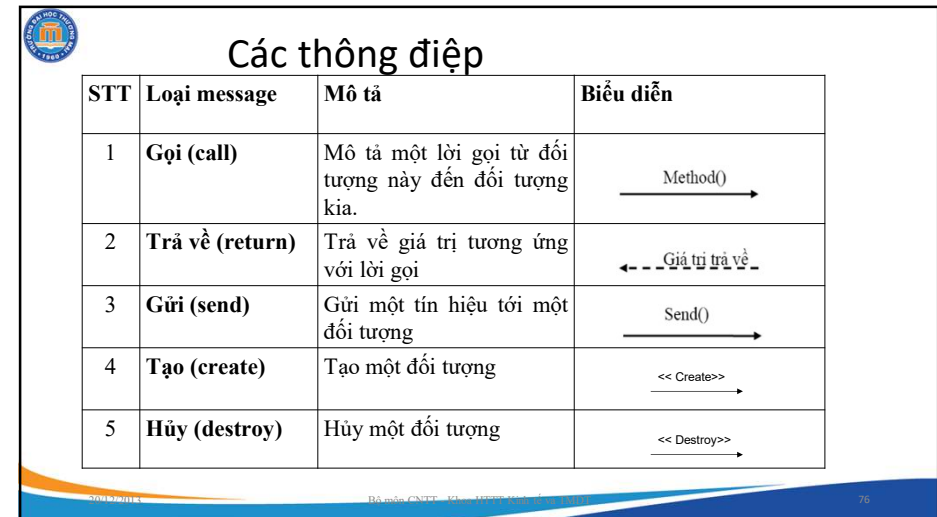
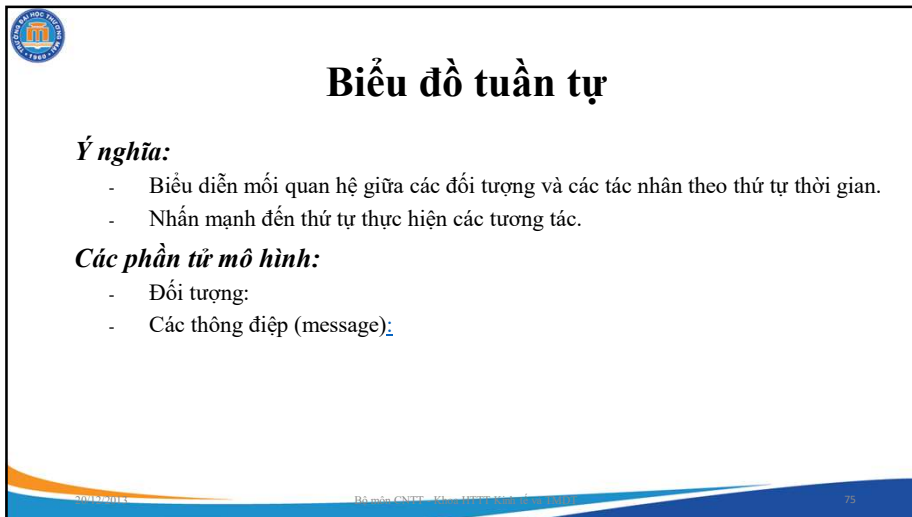
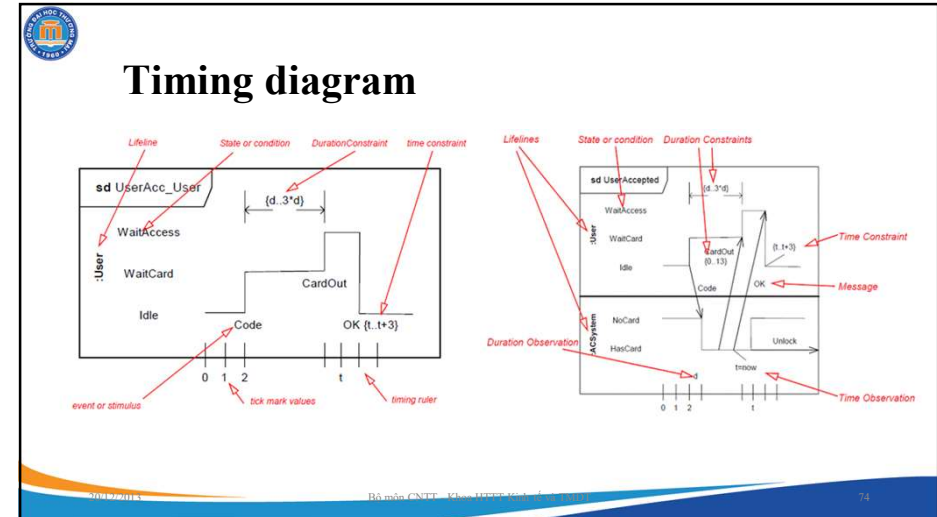
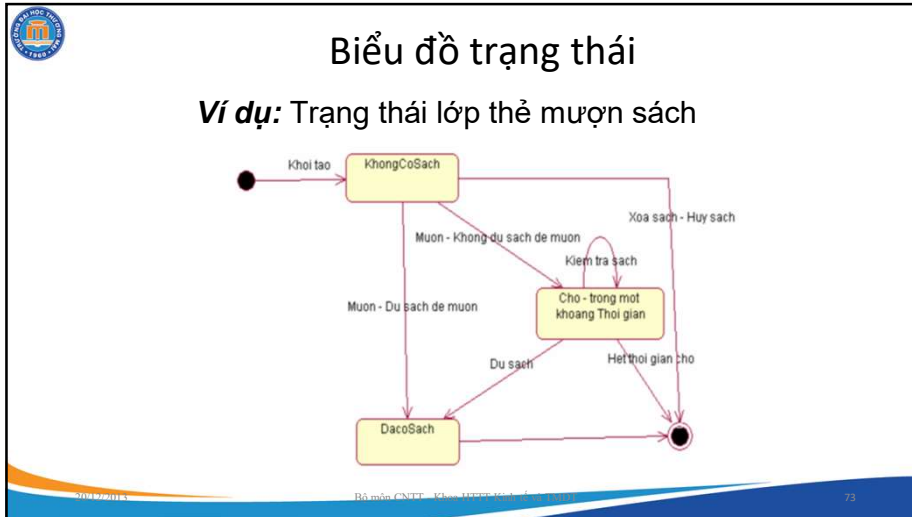
70

## Biểu đồ trạng thái

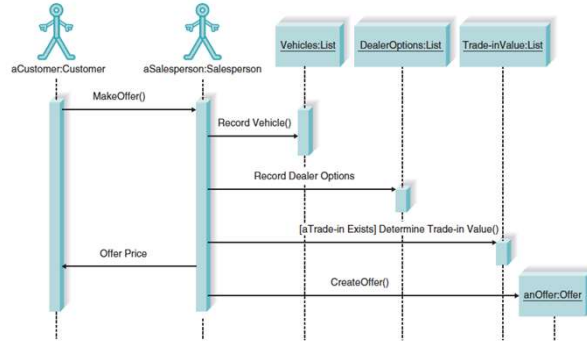
71

## Behavioral State Machine Diagram

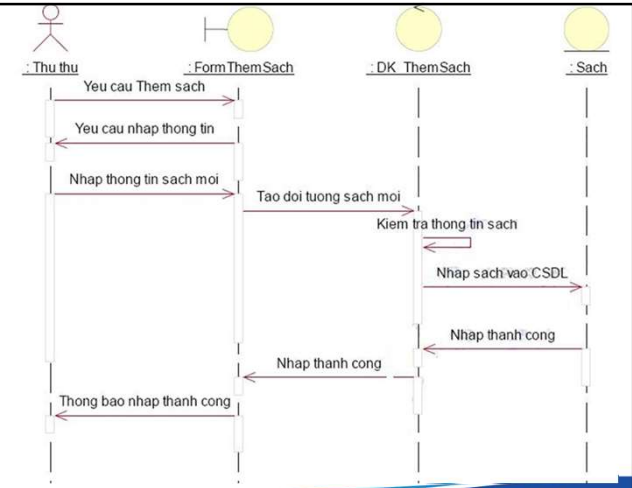
72



# Sequence Diagram



# Chức năng thêm sách



# Biểu đồ hoạt động Activity diagram

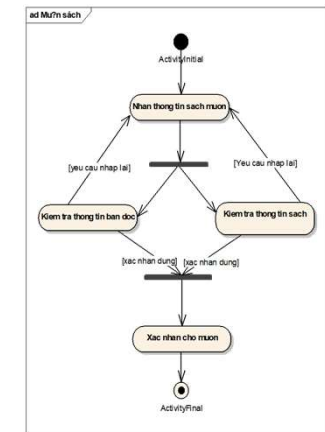
## Ý nghĩa:

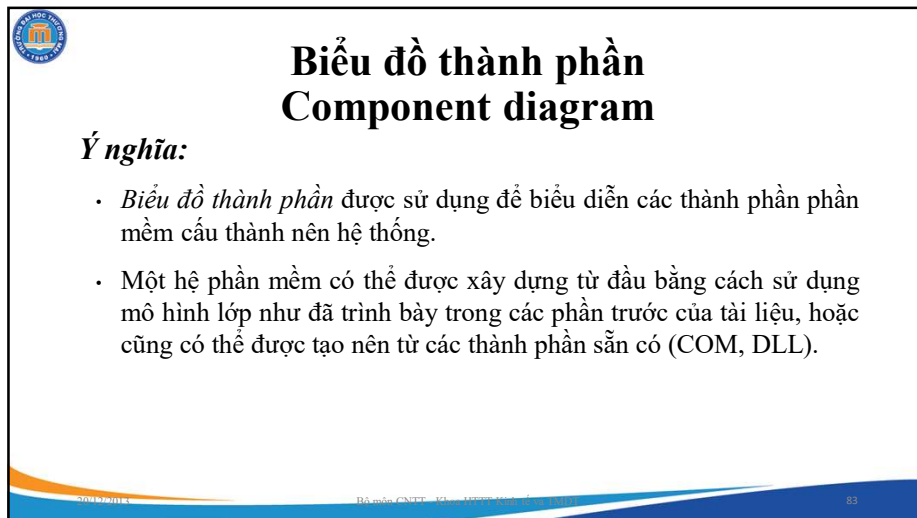
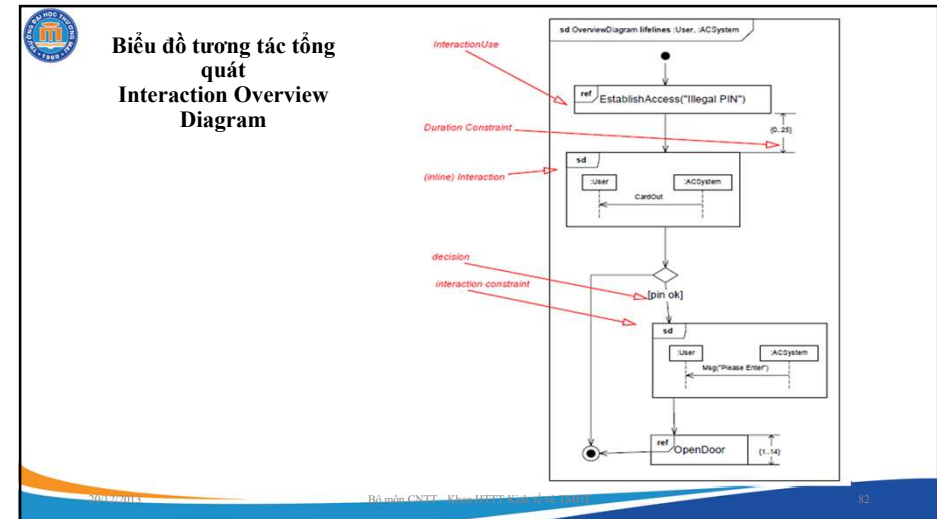
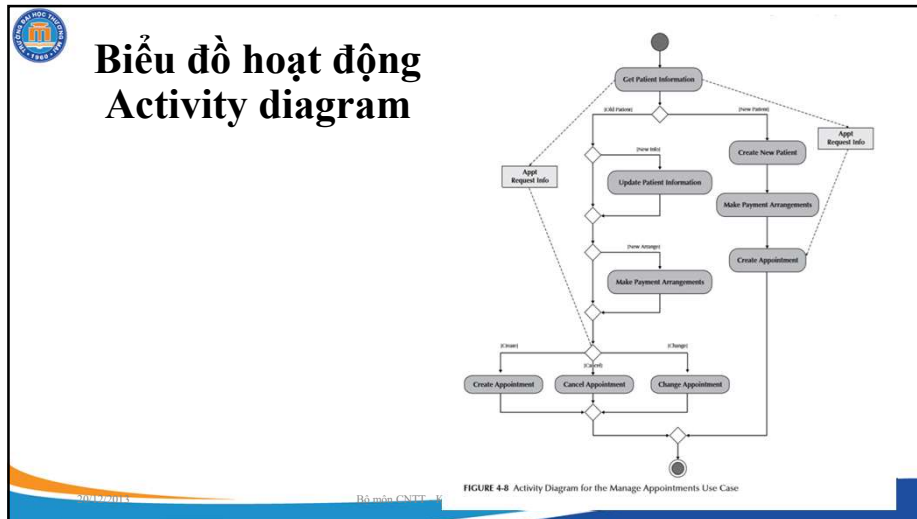
- Biểu đồ hoạt động biểu diễn các hoạt động và sự đồng bộ, chuyển tiếp các hoạt động của hệ thống trong một lớp hoặc kết hợp giữa các lớp với nhau trong một chức năng cụ thể.

## Các phần tử mô hình:

- Hoạt động:
- Thanh đồng bộ hóa:
- Điều kiện
- Các luồng (swimlane)

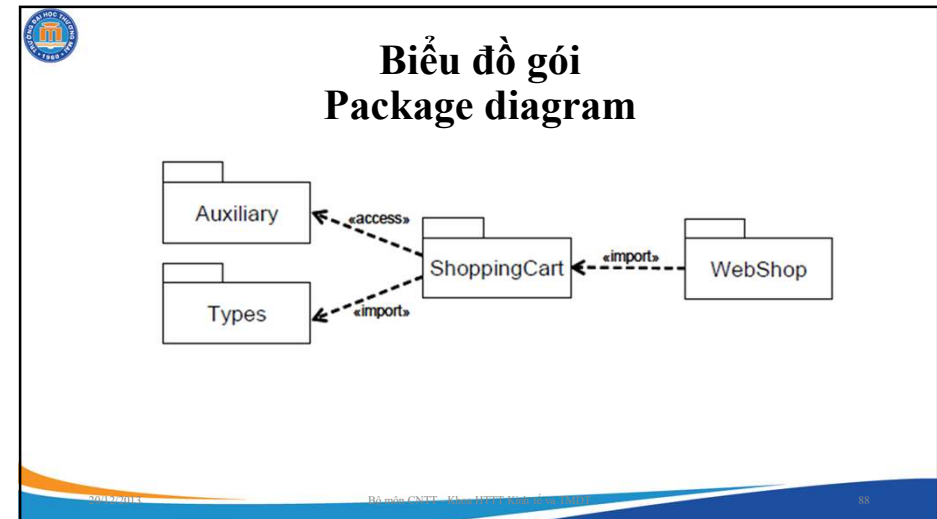
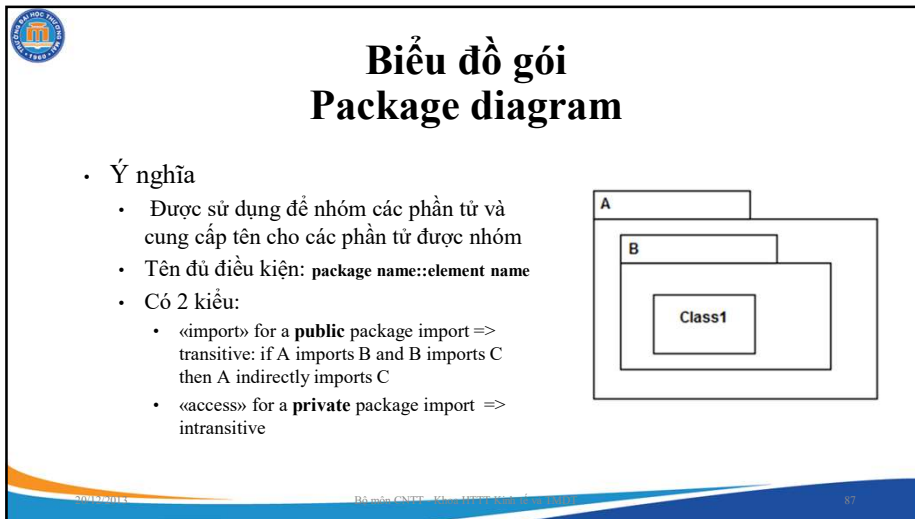
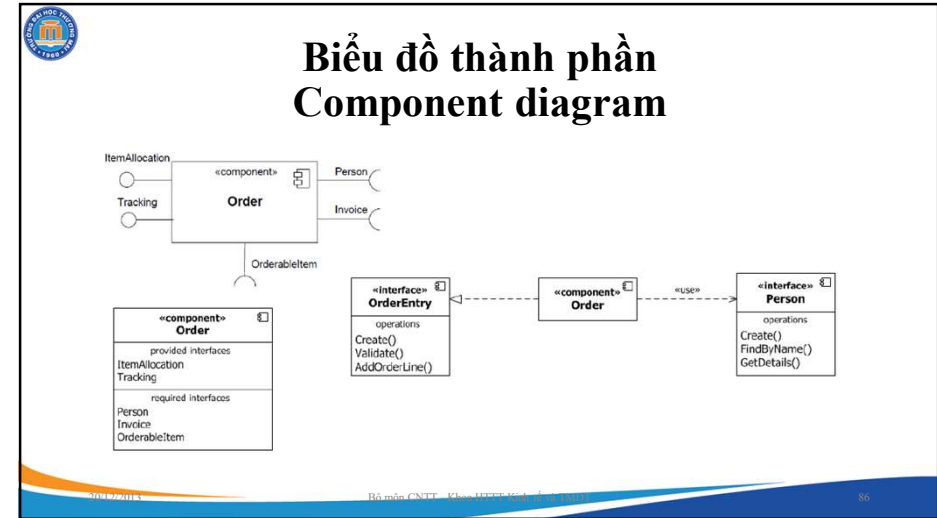
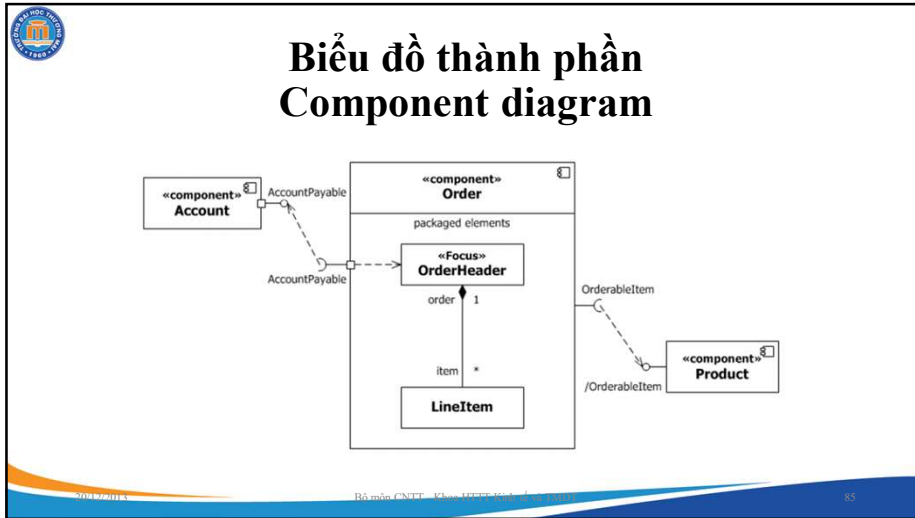
Phần tử mô hình	Ý nghĩa	Ký hiệu trong biểu đồ
Hoạt động	Mô tả hoạt động gồm tên hoạt động và đặc tả của nó	NewActivity
Trạng thái khởi đầu		●
Trạng thái kết thúc		●
Thanh đồng bộ ngang	Mô tả thanh đồng bộ nằm ngang	—
Thanh đồng bộ hòa dọc	Mô tả thanh đồng bộ theo chiều thẳng đứng	
Chuyển tiếp		→
Quyết định	Mô tả một lựa chọn điều kiện	◇
Các (swimlane)	luồng Phân tác các lớp đối tượng khác nhau tồn tại trong biểu đồ hoạt động	Phân cách nhau bởi một đường kẻ dọc từ trên xuống dưới biểu đồ

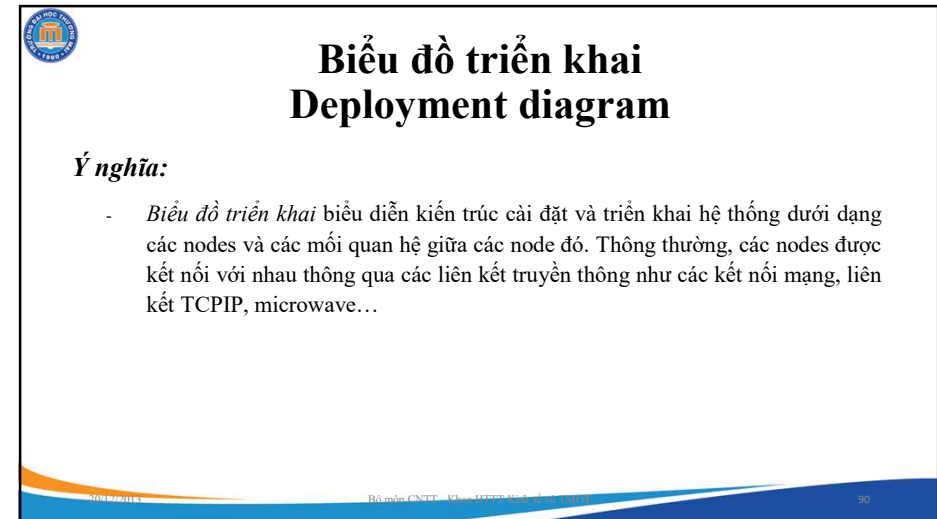
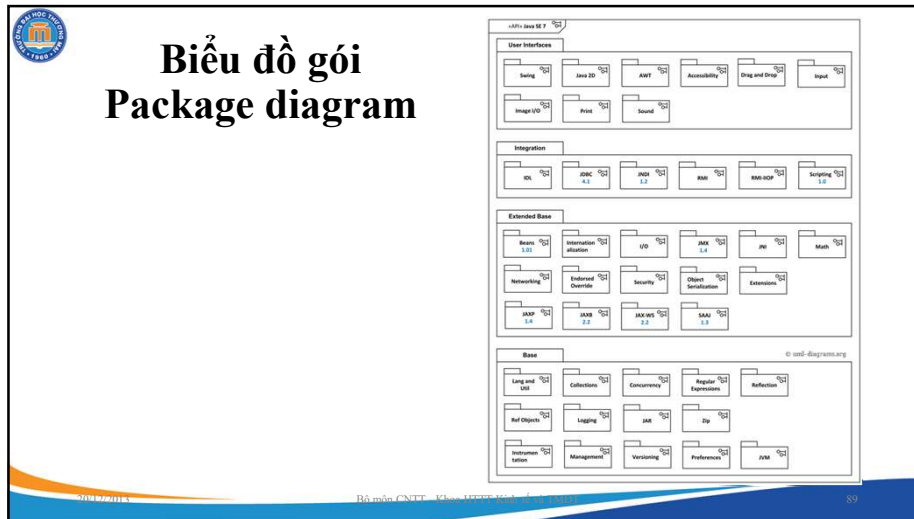




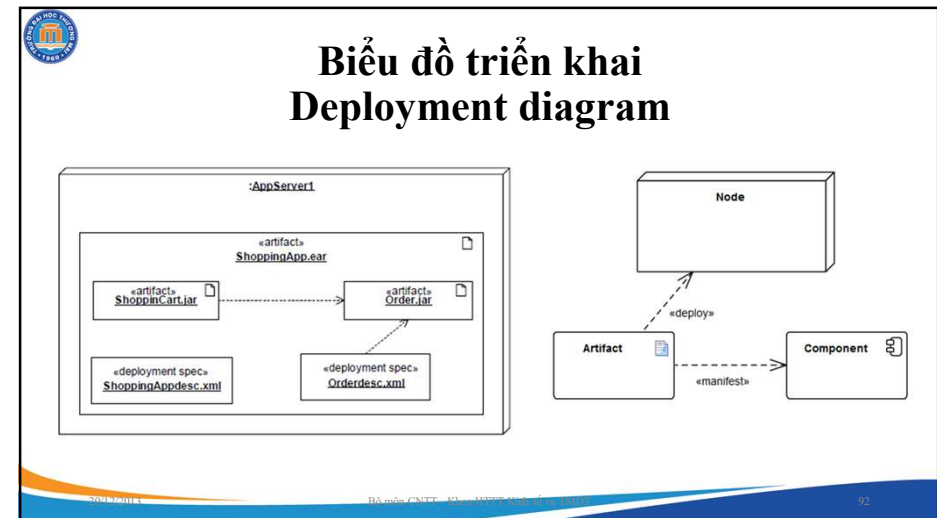
Phân tử mô hình	Ý nghĩa	Ký hiệu trong biểu đồ
Thành phần	Mô tả một thành phần của biểu đồ, mỗi thành phần có thể chứa nhiều lớp hoặc nhiều chương trình con.	Component
Giao tiếp	Mô tả giao tiếp gắn với mỗi thành phần. Các thành phần trao đổi thông tin qua các giao tiếp	
Mối quan hệ phụ thuộc giữa các thành phần	Mối quan hệ giữa các thành phần (nếu có)	
Gói (Package)	Được sử dụng để nhóm một số thành phần lại với nhau	NewPackage

**Bảng 2.7: Các ký hiệu của biểu đồ thành phần**

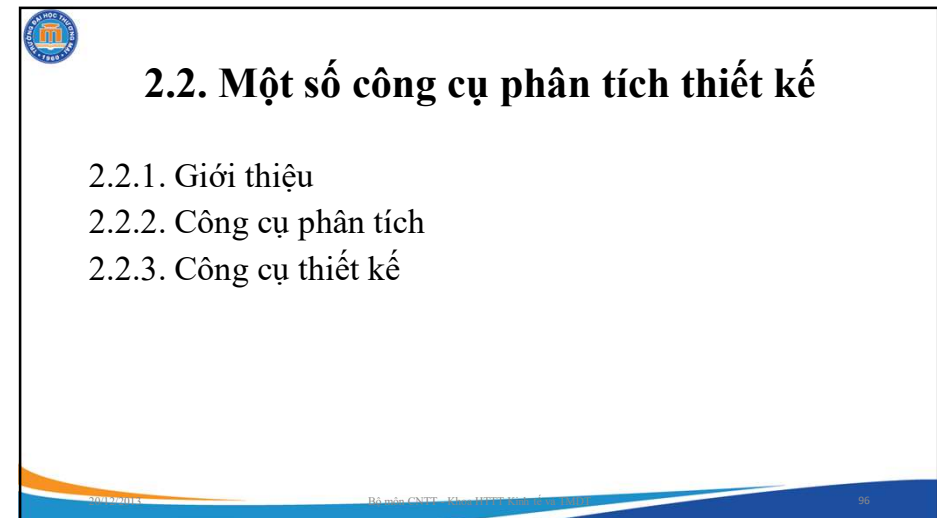
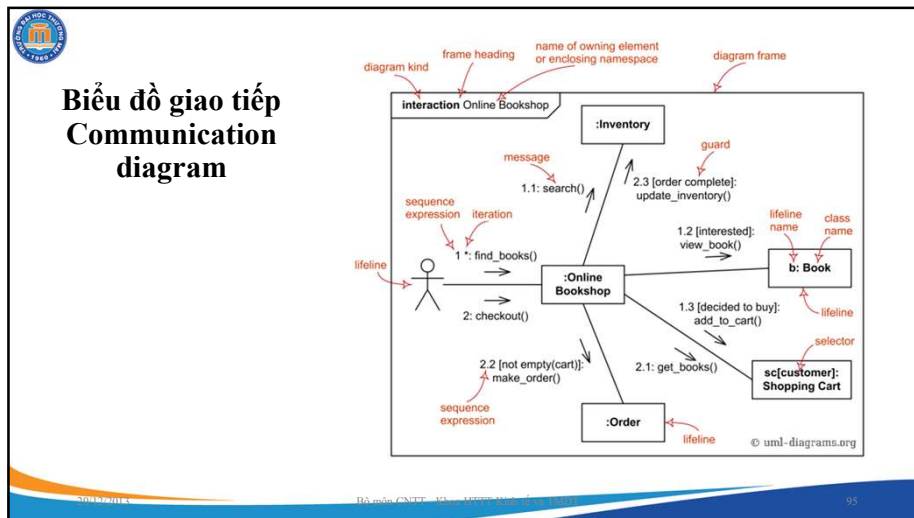
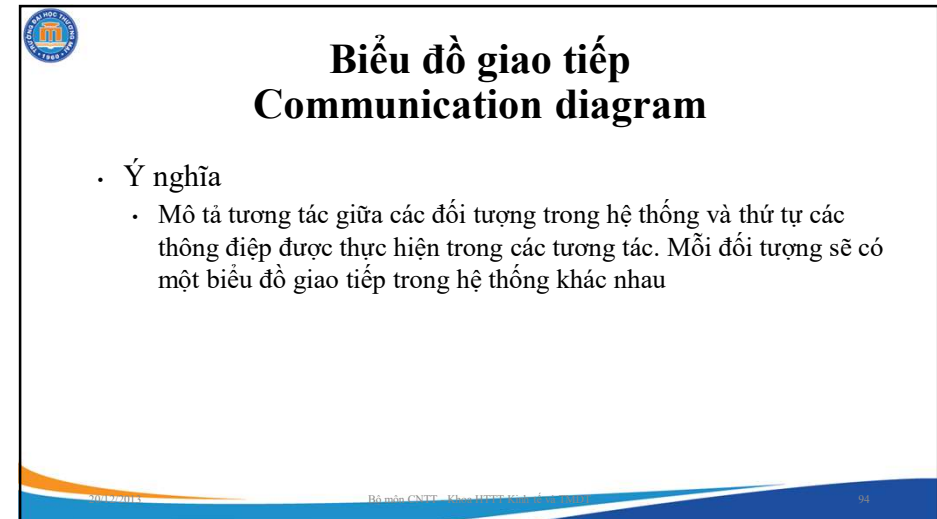
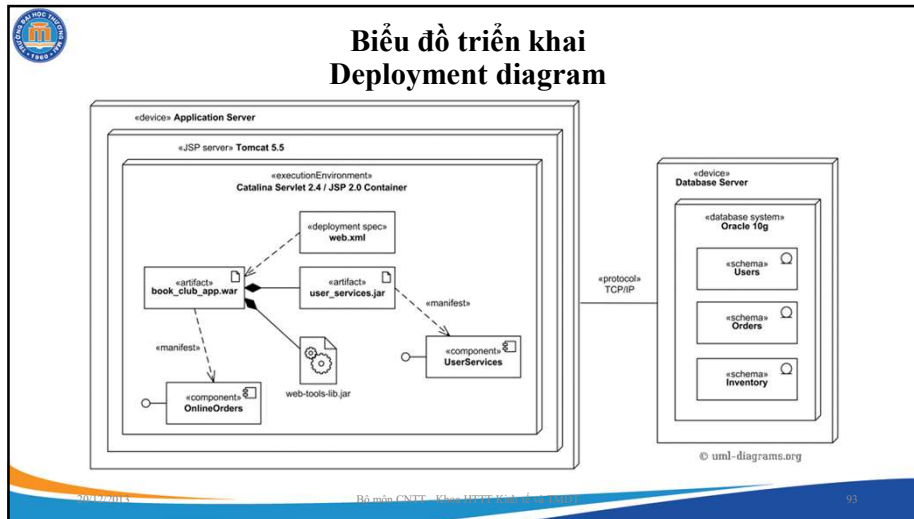




Phần tử mô hình	Ý nghĩa	Ký hiệu trong biểu đồ
Các nodes (hay các thiết bị)	Biểu diễn các thành phần không có bộ vi xử lý trong biểu đồ triển khai hệ thống	
Các bộ xử lý	Biểu diễn các thành phần có bộ xử lý trong biểu đồ	
Các liên kết truyền thông	Nối các thành phần của biểu đồ triển khai hệ thống. Thường mô tả một giao thức truyền thông cụ thể.	

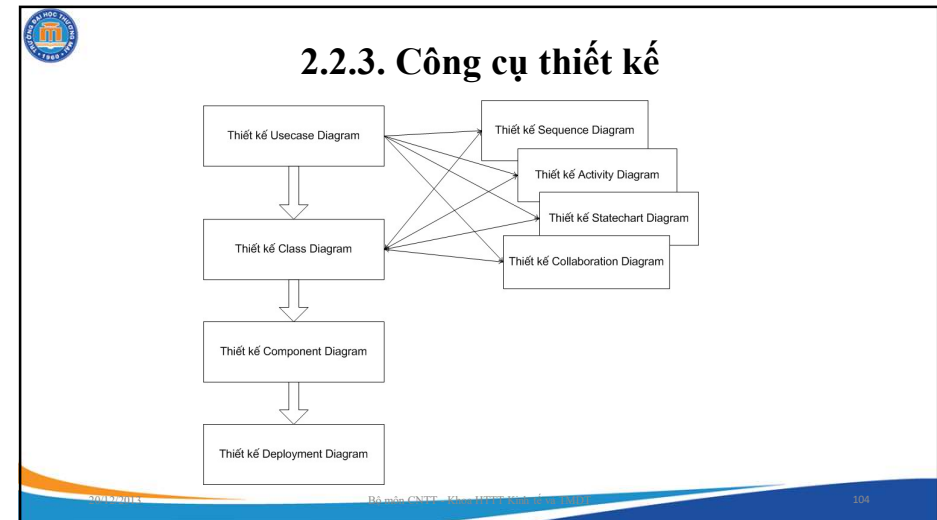
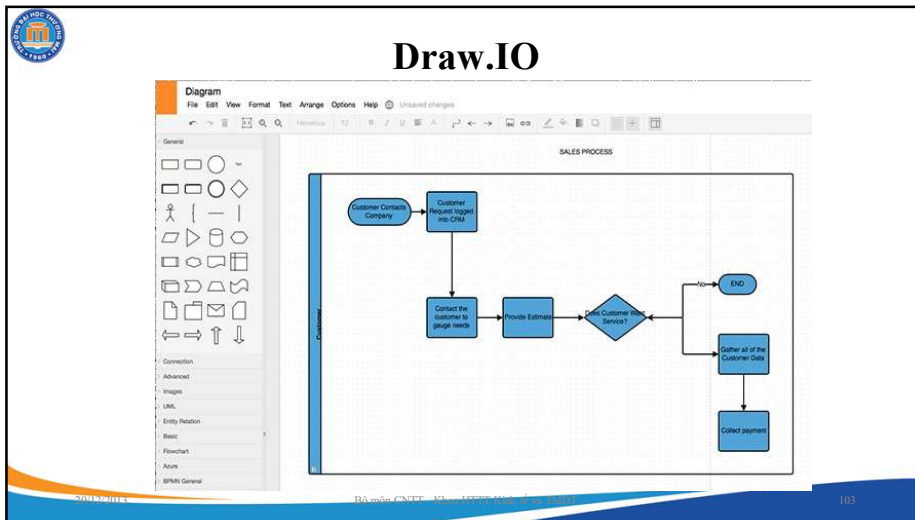
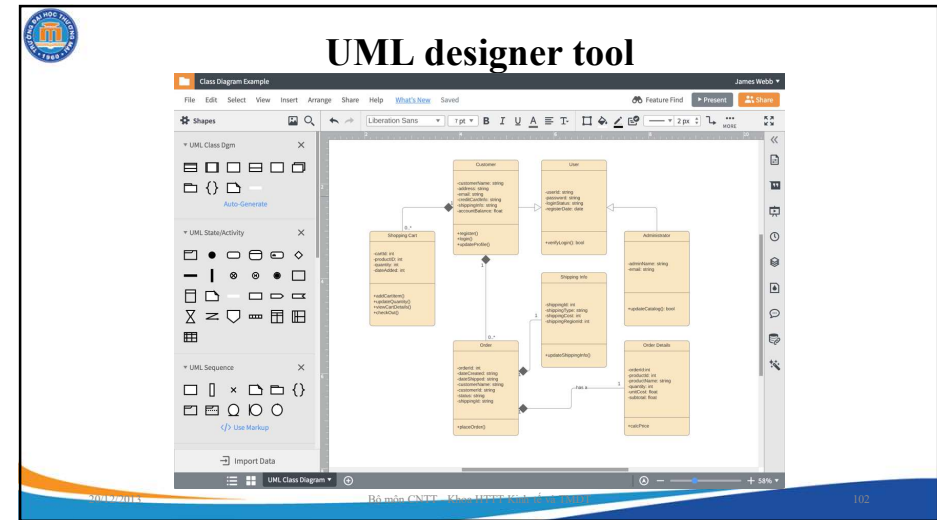
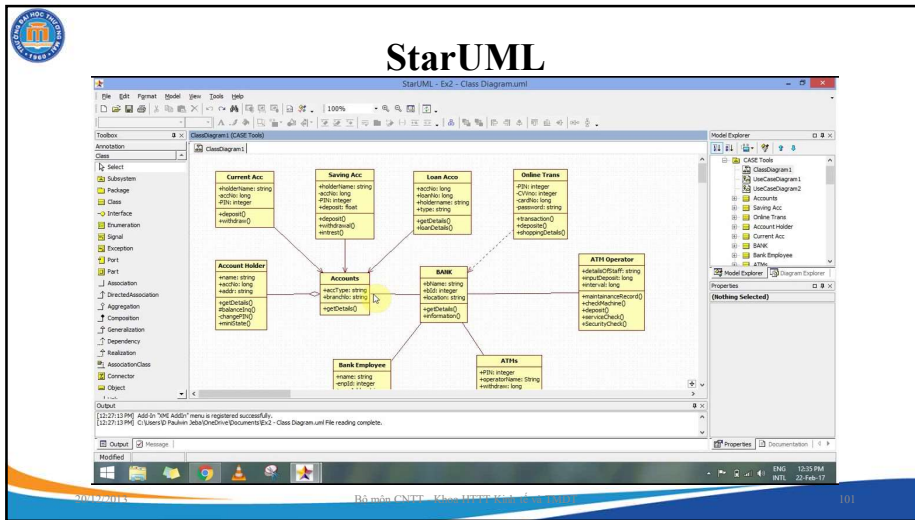
















## Câu hỏi ôn tập Chương 2


- Trình bày các khái niệm cơ bản về UML?
- Rational Rose có vai trò như thế nào trong phân tích, thiết kế HTTT theo HĐT?
- Các công cụ phân tích ?

BỘ MÔN CNTT - KHOA HỆ THỐNG THÔNG TIN



## CHƯƠNG 3: PHÂN TÍCH HỆ THỐNG THEO HƯỚNG ĐỐI TƯỢNG


BỘ MÔN CNTT - KHOA HỆ THỐNG THÔNG TIN



## Tổng quan về pha phân tích Overview of the Analysis Phase

- Goal is to develop a clear understanding of the new system's requirements
  - Understand the “As-Is” system
  - Identify Improvements
  - Develop the “To-Be” system concept
- Use critical thinking skills to determine the true causes of problems
- Apply knowledge of IS and business to outline ways to solve the problems in the new system


BỘ MÔN CNTT - KHOA HỆ THỐNG THÔNG TIN



## Nội dung

- 3.1. Phân tích yêu cầu hệ thống
- 3.2. Biểu đồ usecase
- 3.3. Biểu đồ lớp
- 3.4. Biểu đồ tương tác
- 3.5. Biểu đồ trạng thái và biểu đồ hoạt động

BỘ MÔN CNTT - KHOA HỆ THỐNG THÔNG TIN




# Yêu cầu của hệ thống

## The Systems Request

The business reasons for the new system

BỘ MÔN CNTT - KHOA CNTT, ĐẠI HỌC ĐHQG TP. HCM 109



# Yêu cầu hệ thống

## System Request

- ❑ Describes business reasons for project
- ❑ Defines system's expected value
  - Force the sponsor to formalize his/her ideas
  - Provide a framework for collecting initial project information
  - Standardize information to be used by steering (approval) committee
- ❑ Lists project's key elements

BỘ MÔN CNTT - KHOA CNTT, ĐẠI HỌC ĐHQG TP. HCM 110

## Elements of the Systems Request

Element	Description	Examples
Project Sponsor	The person who initiates the project and who serves as the primary point of contact for the project on the business side	Several members of the finance department Vice president of marketing CIO CEO
Business Need	The business-related reason for initiating the system	Reach a new market segment Offer a capability to keep up with competitors Improve access to information Decrease product defects Streamline supply acquisition processes
Business Requirements	The new or enhanced business capabilities that the system will provide	Provide online access to information Capture customer demographic information Include product search capabilities Produce performance reports Enhance online user support
Business Value	The benefits that the system will create for the organization	3% increase in sales 1% increase in market share Reduction in headcount by 5 FTEs \$200,000 cost savings from decreased supply costs \$150,000 savings from removal of outdated technology
Special Issues or Constraints	Issues that pertain to the approval committee's decision	Government-mandated deadline for May 30 System needed in time for the Christmas holiday season Top-level security clearance needed by project team to work with data

FTE, full-time equivalent.

BỘ MÔN CNTT - KHOA CNTT, ĐẠI HỌC ĐHQG TP. HCM 111

## Systems Request for Tune Source Music Download System

System Request—Digital Music Download Project	
<b>Project Sponsor:</b>	Carly Edwards, Assistant Vice President, Marketing
<b>Business Need:</b>	This project has been initiated to create the capability of selling digital music downloads to customers through kiosks in our stores and over the Internet using our web site. Currently, <ul style="list-style-type: none"> <li>• Customers have many alternatives for downloading music and we need to provide this capability to retain our competitive position.</li> <li>• Our music archive of rare and hard-to-find music is underutilized.</li> </ul>
<b>Business Requirements:</b>	Using this system over the Web or in-store kiosks, customers will be able to search for and purchase digital music downloads. The specific functionality that the system should have includes the following: <ul style="list-style-type: none"> <li>• Search for music in our digital music archive.</li> <li>• Listen to music samples.</li> <li>• Purchase individual downloads at a fixed fee per download.</li> <li>• Establish a customer subscription account permitting unlimited downloads for a monthly fee.</li> <li>• Purchase music download gift cards.</li> </ul>
<b>Business Value:</b>	We expect that Tune Source will increase sales by enabling existing customers to purchase specific digital music tracks and by reaching new customers who are interested in our unique archive of rare and hard-to-find music. We expect to gain a new revenue stream from customer subscriptions to our download services. We expect some increase in cross-selling, as customers who have downloaded a track or two of a CD decide to purchase the entire CD in a store or through our web site. We also expect a new revenue stream from the sale of music download gift cards. Conservative estimates of tangible value to the company include the following: <ul style="list-style-type: none"> <li>• \$757,500 in sales from individual music downloads</li> <li>• \$950,000 in sales from customer subscriptions</li> <li>• \$205,000 in additional in-store or web site CD sales</li> <li>• \$153,000 in sales from music download gift cards</li> </ul>
<b>Special Issues or Constraints:</b>	<ul style="list-style-type: none"> <li>• The marketing department views this as a strategic system. To prevent significant customer attrition, this project should be completed as soon as possible.</li> </ul>

BỘ MÔN CNTT - KHOA CNTT, ĐẠI HỌC ĐHQG TP. HCM 112

## Estimating Business Value

- Identify sources: increased sales; decreased costs; reduced headcount; lower turnover...
- Assign values as initial estimates

	Sales Projections			
	Individual Downloads	Subscriptions	Cross-Selling of CDs	Gift Cards
High-level estimate (prob. = 25%)	\$900,000	\$1,100,000	\$250,000	\$180,000
Medium-level estimate (prob. = 60%)	750,000	950,000	200,000	150,000
Low-level estimate (prob. = 15%)	550,000	700,000	150,000	120,000
Weighted average expected sales	\$757,500	\$950,000	\$205,000	\$153,000

20/11/2014

Bộ môn CNTT - Khoa CNTT - Trường ĐHQG TP. HCM

113



## Phân tích tính khả thi Feasibility Analysis

Is this project really worth doing...

Can we do this project...

Will the organization accept this if we go ahead...

20/11/2014

Bộ môn CNTT - Khoa CNTT - Trường ĐHQG TP. HCM

114



## Phân tích tính khả thi Feasibility Analysis

- Detailed business case for the project
  - Technical feasibility
  - Economic feasibility
  - Organizational feasibility
- Compiled into a feasibility study
- *Critically important* to reassess feasibility throughout the project

20/11/2014

Bộ môn CNTT - Khoa CNTT - Trường ĐHQG TP. HCM

115




## Khả thi về mặt công nghệ Technical Feasibility: *Can We Build It?*

- Sources of Technical Risk:
  - Users' and analysts' lack of familiarity with the business application area
  - Lack of familiarity with technology
    - Have we used it before? How new is it?
  - Project size
    - Number of people, time frame, distinct features
  - Compatibility with existing systems
    - Degree of integration required

20/11/2014

Bộ môn CNTT - Khoa CNTT - Trường ĐHQG TP. HCM

116



## Khả thi về mặt tài chính

### Economic Feasibility: *Should We Build It?*

- Identify costs and benefits
- Assign values to costs and benefits
- Determine cash flow
- Assess financial viability
  - Return on investment
  - Break even point
  - Net present value

117

## Costs and Benefits

- Include development and operational costs
- Consider tangible and intangible benefits

1. Identify Costs and Benefits	List the tangible costs and benefits for the project. Include both one-time and recurring costs.
2. Assign Values to Costs and Benefits	Work with business users and IT professionals to create numbers for each of the costs and benefits. Even intangibles should be valued if at all possible.
3. Determine Cash Flow	Forecast what the costs and benefits will be over a certain period, usually, 3–5 years. Apply a growth rate to the values, if necessary.
4. Assess Project's Economic Value	Evaluate the project's expected returns in comparison to its costs. Use one or more of the following evaluation techniques: <ul style="list-style-type: none"> <li>• Return on Investment (ROI): Calculate the rate of return earned on the money invested in the project, using the ROI formula.</li> <li>• Break-Even Point (BEP): Find the year in which the cumulative project benefits exceed cumulative project costs. Apply the breakeven formula, using figures for that year. This calculation measures how long it will take for the system to produce benefits that cover its costs.</li> <li>• Net Present Value (NPV): Restate all costs and benefits in today's dollar terms (present value), using an appropriate discount rate. Determine whether the total present value of benefits is greater than or less than the total present value of costs.</li> </ul>

118


## Cost-Benefit Analysis

- Discounted cash flow method preferred
- NPV preferred

	2015	2016	2017	2018	2019	Total
<b>Benefits</b>						
Increased sales		500,000	530,000	561,800	595,508	
Reduction in customer complaint calls*		70,000	70,000	70,000	70,000	
Reduced inventory costs		68,000	68,000	68,000	68,000	
<b>Total Benefits<sup>§</sup></b>	<b>638,000</b>	<b>668,000</b>	<b>699,800</b>	<b>733,508</b>		
<b>Present Value Total Benefits</b>	<b>601,887</b>	<b>594,518</b>	<b>587,566</b>	<b>581,007</b>		<b>2,364,978</b>
<b>Development Costs</b>						
2 Servers @ \$125,000	250,000	0	0	0	0	
Printer	100,000	0	0	0	0	
Software licenses	34,825	0	0	0	0	
Server software	10,945	0	0	0	0	
Development labor	1,236,525	0	0	0	0	
<b>Total Development Costs</b>	<b>1,632,295</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	
<b>Operational Costs</b>						
Hardware		50,000	50,000	50,000	50,000	
Software		20,000	20,000	20,000	20,000	
Operational labor		115,000	119,600	124,384	129,359	
<b>Total Operational Costs</b>		<b>185,000</b>	<b>189,600</b>	<b>194,384</b>	<b>199,359</b>	
<b>Total Costs</b>	<b>1,632,295</b>	<b>185,000</b>	<b>189,600</b>	<b>194,384</b>	<b>199,359</b>	
<b>Present Value Total Costs</b>	<b>1,632,295</b>	<b>174,528</b>	<b>168,743</b>	<b>163,209</b>	<b>157,911</b>	<b>2,296,686</b>
<b>NPV (PV Total Benefits – PV Total Costs)</b>						<b>68,292</b>

\*Customer service values are based on reduced costs of handling customer complaint phone calls.  
<sup>§</sup>An important yet intangible benefit will be the ability to offer services that our competitors currently offer.

119




## Khả thi về mặt tổ chức

### Organizational Feasibility: *If We Build It, Will They Come?*

- Strategic alignment
  - Are project goals aligned with business strategy?
- Evaluate effect on various stakeholders
  - Strong and influential project champion?
  - Strong and widespread organizational management support?
  - Receptive / resistant system users?

120





## Organizational Feasibility

### *If We Build It, Will They Come?*

- Strategic alignment
  - Close alignment with strategy increases the likelihood of success
- Stakeholder groups can be influenced
  - Presentations describing and promoting benefits
  - Emphasizing personal benefits as well as organizational benefits
  - Prototypes help prove the system concept
  - Real user involvement throughout project

121




## Đánh giá tính khả thi

### Feasibility Assessment: Summing It Up

- All projects have feasibility risks
  - Our goal is to know the risks we face and the significance of those risks
  - Project Sponsor, Project Manager, and other team members need this awareness
  - Once risks are known, steps can be taken to mitigate the risks
    - For example, if unfamiliar with a new technology
      - Provide enough budget for training
      - Provide enough budget to hire consultants with expertise
      - Allow more schedule time to move up the learning curve
      - Use a methodology that incorporates experimentation

122




## Đánh giá tính khả thi

### Feasibility Assessment: Summing It Up

- Essential to continuously review and revise the feasibility assessment
  - How well are we managing the risks we previously identified? Are adjustments needed?
    - Risk is being managed
    - Risk is not well managed and needs further attention
  - Are there any new risks that have appeared?
    - If so, what are the actions needed to address those risks?
    - Budgetary and schedule effect?


123



## Lựa chọn và quản lý dự án

### Project Selection and Management


Systems analysis and design, 6<sup>th</sup> edition  
Dennis, wixom, and roth



## Các vấn đề khi lựa chọn dự án Project Selection Issues

- Ways to Characterize Projects
  - Size
  - Cost
  - Purpose
  - Length
  - Risk
  - Scope
  - Economic Value


125



## Quản lý danh mục dự án Project Portfolio Management

- PPM software collects and manages information about all projects – on-going and awaiting approval.
- Companies stay up to date on projects and adapt to changing conditions.
- Features: project prioritization, employee allocation, real-time project monitoring, flagging cost and time variances, monitoring economic feasibility.


126



## Khởi tạo kế hoạch cho dự án Creating the Project Plan

- Once a project is approved, the project manager must:
  - Select the best project methodology
  - Develop a project work plan
  - Establish a staffing plan
  - Create ways to coordinate and control the project

127



## Lựa chọn phương thức cho dự án Selecting a Project Methodology

- Methodology: A formalized approach to implementing the SDLC
  - A series of steps to perform and deliverables to produce
- Methodology Sources
  - Internally developed by organizations
  - Consulting firms
  - Software vendors
  - Government agencies

128



## Những vấn đề khi lựa chọn phương thức cho dự án Selecting a Project Methodology - Issues

- ❑ These factors influence the best choice:
  - Clarity of User Requirements
  - Familiarity with Technology
  - System Complexity
  - System Reliability
  - Time Frame
  - Schedule Visibility

## Mô hình hóa phát triển các hệ thống Structured Systems Development

- ❑ Based upon SDLC
- ❑ Assumes a project phase is complete before moving to the next phase
  - Waterfall Development
  - Parallel Development
  - V-model
- ❑ Goal – doing each phase thoroughly before moving forward ensures correct and high-quality outcomes

## Waterfall Development Methodology

- Move from phase to phase
- Emphasis on deliverables from one phase flowing into the next phase

```

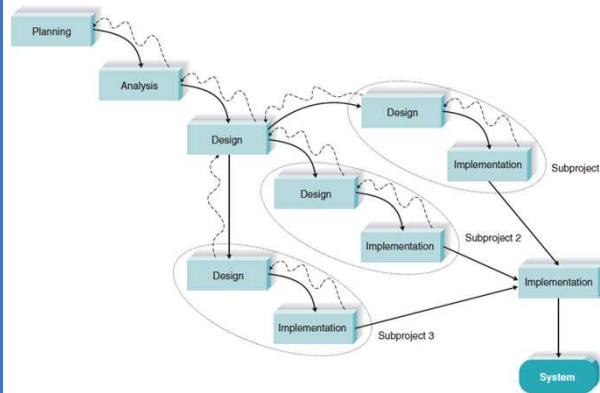
graph TD
    Planning[Planning] --> Analysis[Analysis]
    Analysis --> Design[Design]
    Design --> Implementation[Implementation]
    Implementation --> System[System]
  
```

## Waterfall Methodology Assessment

STRENGTHS	WEAKNESSES
<ul style="list-style-type: none"> <li>☑ System requirements identified long before construction begins</li> <li>☑ Requirements are “frozen” as project proceeds – no moving targets allowed</li> </ul>	<ul style="list-style-type: none"> <li>☑ Must wait a long time before there is “visible” evidence of the new system</li> <li>☑ Takes a long time from start to finish</li> </ul>

## Parallel Development Methodology

- Subdivide the project into subprojects that can be worked on at the same time.
- Reduce the overall project length



20/12/2014

Bộ môn CNTT - Khoa CNTT - Trường ĐHQG TP. HCM

133

## Parallel Methodology Assessment

### STRENGTHS

- ☑ Reduces overall project time (compared to Waterfall)
- ☑ Reduces the need for rework; with shorter time frame, less chance of requirements changing

### WEAKNESSES

- ☑ Creating subprojects requires careful design decisions
- ☑ Integrating subprojects at the end can be complex and difficult

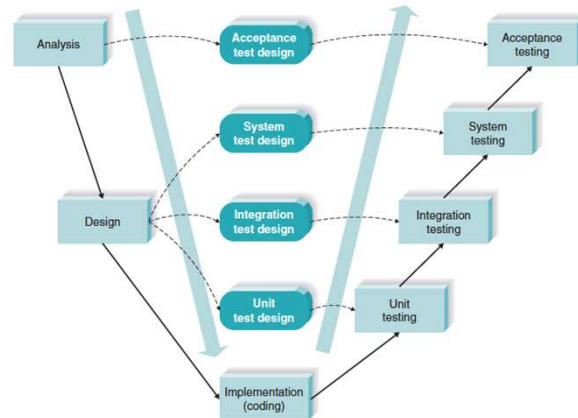
20/12/2014

Bộ môn CNTT - Khoa CNTT - Trường ĐHQG TP. HCM

134

## V-Model Development Methodology

- Emphasizes system quality through text plan development



20/12/2014

Bộ môn CNTT - Khoa CNTT - Trường ĐHQG TP. HCM

135

## V-Model Methodology Assessment

### STRENGTHS

- ☑ Simple and straightforward
- ☑ Quality improves through the emphasis on testing
- ☑ Including Quality Assurance expertise early in the project strengthens system quality

### WEAKNESSES

- ☑ Rigid
- ☑ Difficult to use in a dynamic business environment

20/12/2014

Bộ môn CNTT - Khoa CNTT - Trường ĐHQG TP. HCM

136

## Rapid Application Development

- ❑ Incorporate special techniques and tools:
  - CASE tools
  - JAD sessions
  - Visual programming languages
  - Code generators
- ❑ Goal – get some portion of system developed quickly and in the users' hands

## Three RAD Approaches

- ❑ Iterative development
  - A series of versions developed sequentially
- ❑ System Prototyping
  - Create prototype (model) of system and “grow” it into the final system
- ❑ Throw-away prototyping
  - Prototype alternative designs in an experimental way
  - Build system following prototype design but discard the actual prototype

### Iterative Development Methodology

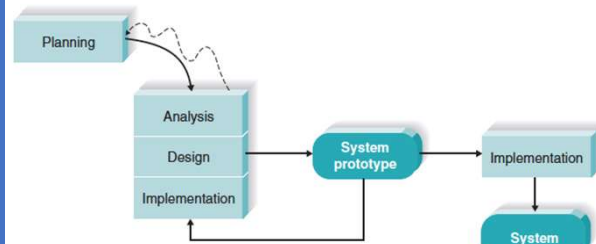
- RAD approach
- Develop system in series of versions

### Iterative Development Methodology Assessment

STRENGTHS	WEAKNESSES
<ul style="list-style-type: none"> <li>☑ Users get a system to use quickly</li> <li>☑ Users identify additional needs for later versions based on real experiences with current version</li> </ul>	<ul style="list-style-type: none"> <li>☑ Users faced with using an incomplete system for a time</li> <li>☑ Users must be patient and wait for fully-functional system</li> </ul>

## System Prototyping Development Methodology

- RAD approach
- Create a rough version of system quickly and "grow" it into final system with repetitive refinement



20/12/2014

Bộ môn CNTT - Khoa CNTT - Trường ĐHQG Hà Nội

141

## System Prototyping Methodology Assessment

### STRENGTHS

- ☑ Users get to work with prototype very quickly
- ☑ Feedback cycles let users identify changes and refine real requirements

### WEAKNESSES

- ☑ Superficial analysis may cause problems
- ☑ Initial design decisions may be poor
- ☑ Overlooked features may be hard to add later

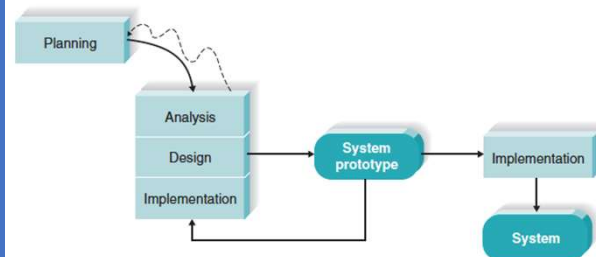
20/12/2014

Bộ môn CNTT - Khoa CNTT - Trường ĐHQG Hà Nội

142

## Throwaway Prototyping Development Methodology

- RAD approach
- Adds emphasis on experimenting with design options before design is finalized
- Design options are thrown-away, but learning from them is factored into final design



20/12/2014

Bộ môn CNTT - Khoa CNTT - Trường ĐHQG Hà Nội

143

## Throwaway Prototyping Methodology Assessment

### STRENGTHS

- ☑ Uncertainty is minimized
- ☑ Important issues are understood before building the final system

### WEAKNESSES

- ☑ May take longer (compared to system prototyping)

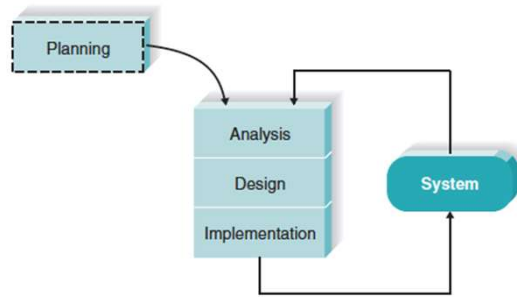
20/12/2014

Bộ môn CNTT - Khoa CNTT - Trường ĐHQG Hà Nội

144

## Agile Development Methodologies

- Extreme Programming (XP), Scrum, and others
- Focus on short cycles that produce a complete software product
- Highly adaptable in dynamic environments



20/12/2014

Bộ môn CNTT - Khoa CNTT - Trường ĐHQG Hà Nội

145

## Agile Methodologies Assessment

### STRENGTHS

- ☑ Fast delivery of results
- ☑ Works well in projects with undefined or changing requirements

### WEAKNESSES

- ☑ Requires discipline
- ☑ Significant user involvement is essential
- ☑ Initial high learning curve
- ☑ Works best in smaller projects

20/12/2014

Bộ môn CNTT - Khoa CNTT - Trường ĐHQG Hà Nội

146

## Selection Summary

Ability to develop systems	Waterfall	Parallel	V-Model	Iterative	System Prototyping	Throwaway Prototyping	Agile Development
With unclear user requirements	Poor	Poor	Poor	Good	Excellent	Excellent	Excellent
With unfamiliar technology	Poor	Poor	Poor	Good	Poor	Excellent	Poor
That are complex	Good	Good	Good	Good	Poor	Excellent	Poor
That are reliable	Good	Good	Excellent	Good	Poor	Excellent	Good
With a short time schedule	Poor	Good	Poor	Excellent	Excellent	Good	Excellent
With schedule visibility	Poor	Poor	Poor	Excellent	Excellent	Good	Good

20/12/2014

Bộ môn CNTT - Khoa CNTT - Trường ĐHQG Hà Nội

147

## Các nhiệm vụ trong quản lý dự án Project Management Tasks

Preparing to launch the project

20/12/2014

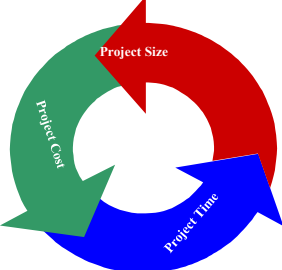
Bộ môn CNTT - Khoa CNTT - Trường ĐHQG Hà Nội

148

## Project Manager's Balancing Act

- Project Management involves making trade-offs...

Modifying one element requires adjusting the others



149

## Project Estimation

- The process of assigning projected values for time and effort
- Sources of estimates
  - Methodology in use
  - Actual previous projects
  - Experienced developers
- Estimates begin as a range and become more specific as the project progresses
  - Industry standards
  - Function point estimation (Appendix 2A)

150

## Project Estimates Using Industry Standard Percentages

INDUSTRY STANDARD PERCENTAGES

	Planning	Analysis	Design	Implementation
Typical industry standards for business applications (%)	15	20	35	30
Estimates based on actual figures for first stage of SDLC (person-months)	Actual: 4	Estimated: 5.33	Estimated: 9.33	Estimated: 8

SDLC, systems development life cycle.

EXAMPLE

- IF 4 months are required for **Planning**, then
- $15\% X = 4$ , where  $X$  = overall length of project
- $X = 4 / 15\%$
- $X = 26.66$  months for entire project
- Therefore:
  - Planning (15%): 4 months
  - Analysis (20%): 5.33 months
  - Design (35%): 9.33 months
  - Implementation (30%): 8 months
- Total Project Length: 26.66 months

151

## Identifying Tasks

- Use established guidelines – existing methodologies
- Use analogies – model previous projects' task lists
- Top-down approach – break high level tasks into smaller, detailed tasks
- Organize into work breakdown structure

152

## Typical Workplan Entry

Task Information	Example
Name of the task	Perform economic feasibility
Start date	Jan 5, 2016
Completion date	Jan 19, 2016
Person assigned to the task	Project sponsor Mary Smith
Deliverable(s)	Cost-benefit analysis
Completion status	Complete
Priority	High
Resources needed	Spreadsheet software
Estimated time	16 hours
Actual time	14.5 hours

13/08/2016

Bộ môn CNTT - Khoa CNTT - Trường ĐHQG Hà Nội

153

## Project Work Plan

Task ID	Task Name	Assigned To	Estimated			Actual			Dependency	Status
			Duration (days)	Start Date	Finish Date	Start Date	Finish Date	Duration variance		
1	Design Phase		31	Mon 11/14/16	Wed 12/28/16					Open
1.1	Develop database design document	Megan	9	Mon 12/5/16	Thurs 12/15/16					Open
1.1.1	Staging database design	Megan	9	Mon 12/5/16	Thurs 12/15/16					Open
1.1.2	Suppense database design	Megan	9	Mon 12/5/16	Thurs 12/15/16					Open
1.2	Develop specs-handling design document	Megan	9	Fri 12/14/16	Wed 12/28/16				1.1.1, 1.1.2	Open
1.2.1	Rejects-handling engine design	Megan	9	Fri 12/16/16	Wed 12/28/16				1.1.1, 1.1.2	Open
1.3	Develop OLAP design document	Joachim	9	Fri 12/16/16	Wed 12/28/16				1.1.1, 1.1.2	Open
1.3.1	Universe design	Joachim	9	Fri 12/16/16	Wed 12/28/16					Open
1.4	Develop OLAP design part 1	Kevin	8	Fri 12/9/16	Tues 12/20/16					Open
1.4.1	High-priority reports design	Kevin	8	Fri 12/9/16	Tues 12/20/16					Open
1.5	Develop application design document	Tomas	9	Fri 12/16/16	Wed 12/28/16					Open
1.5.1	Group consolidation and corporate reporting GCCCR maintenance application design	Tomas	9	Fri 12/16/16	Wed 12/28/16					Open

Bộ môn CNTT - Khoa CNTT - Trường ĐHQG Hà Nội

154

## Project Work Plan

Task ID	Task Name	Assigned To	Estimated			Actual			Dependency	Status
			Duration (days)	Start Date	Finish Date	Start Date	Finish Date	Duration variance		
1.6	Extract, transform, load (ETL) design document	Joachim	2	Thu 12/29/16	Fri 12/30/16					Open
1.6.1	Data export utility design	Joachim	2	Thu 12/29/16	Fri 12/30/16					Open
1.7	Application design document	Mel-ling	26	Mon 11/14/16	Tue 12/20/16					Open
1.7.1	Web entry application UI design	Mel-ling	26	Mon 11/14/16	Tue 12/20/16					Open
1.7.2	Web entry application UI design sign-off	Mel-ling	1	Wed 11/30/16	Wed 11/30/16					Open
1.7.3	Web entry forms and database model validation	Kevin	11	Wed 11/23/16	Thu 12/8/16					Open
1.8	Functional requirements document	Charrelle	9	Mon 12/13/16	Thu 12/22/16					Open
1.8.1	Application design	Charrelle	9	Mon 12/13/16	Thu 12/22/16					Open
1.8.1.1	User authentication	Charrelle	4	Mon 12/12/16	Thu 12/15/16					Open
1.8.1.2	Call logging	Charrelle	2	Fri 12/16/16	Mon 12/19/16					Open
1.8.1.3	Search	Charrelle	3	Tue 12/20/16	Thu 12/22/16					Open

(Thanks to Priya Padmashiban for suggesting this example.)

13/08/2016

Bộ môn CNTT - Khoa CNTT - Trường ĐHQG Hà Nội

155

## 3.1. Phân tích yêu cầu hệ thống

- 3.1.1. Xác định yêu cầu hệ thống
- 3.1.2. Mô hình hoá nghiệp vụ
- 3.1.3. Các hướng nhìn trong phân tích

13/08/2016

Bộ môn CNTT - Khoa CNTT - Trường ĐHQG Hà Nội

156





### 3.1.1 Xác định các yêu cầu hệ thống

Cơ sở xác định yêu cầu hệ thống

Các phương pháp xác định yêu cầu truyền thống.

Các phương pháp xác định yêu cầu hiện đại



### Cơ sở xác định yêu cầu hệ thống

- **Xuất phát:** Xác định hiện trạng
  - HT tổ chức đang hoạt động
  - HT tổ chức mới hình thành
- **Mục tiêu:**
  - Tiếp cận chuyên môn nghiệp vụ, môi trường hoạt động của HT
  - Tìm hiểu các chức năng, nhiệm vụ và cung cách hoạt động của HT
  - Xác định những chỗ hợp lý của HT → kế thừa, chỗ bất hợp lý → cần khắc phục.



### Yêu cầu là gì? What is a Requirement?


- A statement of what the system must do; or
- A statement of characteristics the system must have
- Types of requirements:
  - what the business needs (business requirements);
  - what the users need to do (user requirements);
  - what the software should do (functional requirements);
  - characteristics the system should have (nonfunctional requirements); and
  - how the system should be built (system requirements).



### Các yêu cầu người dùng User Requirements

- What the user needs to do to complete a needed job or task
- Focus on user tasks that are integral to business operations
- Understanding user tasks helps reveal ways that the new system can support those tasks





## Các yêu cầu chức năng Functional Requirements

- ❑ A process the system should perform as a part of supporting a user task, or
- ❑ Information the system should provide as the user performs a task
- ❑ Specify the support the system will provide to the user in fulfilling his/her work tasks

161

## More on Functional Requirements

- Process-oriented
- Information-oriented

Functional Requirement	Description	Examples
Process-oriented	A process the system must perform; a process the system must do	<ul style="list-style-type: none"> <li>• The system must allow registered customers to review their own order history for the past 3 years.</li> <li>• The system must check incoming customer orders for inventory availability.</li> <li>• The system should allow students to view a course schedule while registering for classes.</li> </ul>
Information-oriented	Information the system must contain	<ul style="list-style-type: none"> <li>• The system must retain customer order history for 3 years.</li> <li>• The system must include real-time inventory levels at all warehouses.</li> <li>• The system must include budgeted and actual sales and expense amounts for the current year and 3 previous years.</li> </ul>

162



## Các yêu cầu phi chức năng Nonfunctional Requirements

- ❑ Behavioral properties the system must have
  - Operational – physical and technical operating environment
  - Performance – speed, capacity, and reliability needs
  - Security – access restrictions, needed safeguards
  - Cultural and political – issues that will affect the final system
- ❑ Nonfunctional requirements are discussed in Chapter 8 (Architecture Design)

163


## More on Nonfunctional Requirements

- Behavioral properties the system must have

Nonfunctional Requirement	Description	Examples
Operational	The physical and technical environments in which the system will operate	<ul style="list-style-type: none"> <li>• The system will run on Android mobile devices.</li> <li>• The system should be able to integrate with the existing inventory system.</li> </ul>
Performance	The speed, capacity, and reliability of the system	<ul style="list-style-type: none"> <li>• The system should be compatible with any Web browser.</li> <li>• Any interaction between the user and the system should not exceed 2 seconds.</li> <li>• The system downloads new status parameters within 5 minutes of a change.</li> <li>• The system should be available for use 24 hours per day, 365 days per year.</li> <li>• The system supports 300 simultaneous users from 9–11 A.M.; 150 simultaneous users at all other times.</li> </ul>
Security	Who has authorized access to the system under what circumstances	<ul style="list-style-type: none"> <li>• Only direct managers can see staff personnel records.</li> <li>• Technicians can see only their own work assignments.</li> <li>• The system includes all available safeguards from viruses, worms, Trojan horses, etc.</li> </ul>
Cultural and Political	Cultural and political factors and legal requirements that affect the system	<ul style="list-style-type: none"> <li>• The system should be able to distinguish between US currency and currency from other nations.</li> <li>• Company policy is to buy computers only from Dell.</li> <li>• Country managers are permitted to authorize custom user interfaces within their units.</li> <li>• Personal information is protected in compliance with the Data Protection Act.</li> </ul>

Source: The Atlantic Systems Guild, [www.systemsguild.com](http://www.systemsguild.com)


164



## Các yêu cầu về tài liệu Documenting Requirements

- ❑ Requirements definition report
  - Text document listing requirements in outline form
  - Organized in logical groupings
  - Priorities may be included
- ❑ Key purpose is to define the project scope
  - what is included
  - what is not included.

165




## Comparing Techniques

- Depth
- Breadth
- Integration
- User involvement
- Cost

Type of information	Interviews	Joint Application Design	Questionnaires	Document Analysis	Observation
	As-is, improvements, to-be	As-is, improvements, to-be	As-is, improvements	As-is	As-is
Depth of information	High	High	Medium	Low	Low
Breadth of information	Low	Medium	High	High	Low
Integration of information	Low	High	Low	Low	Low
User involvement	Medium	High	Low	Low	Low
Cost	Medium	Low-Medium	Low	Low	Low-Medium

166



## Các chiến lược phân tích yêu cầu Requirements Analysis Strategies

Ways to discover true underlying requirements

167



## Các yêu cầu cải tiến nhỏ To Identify Small Improvements

- ❑ Problem Analysis
  - Ask users to identify problems and solutions
  - Improvements tend to be small and incremental
  - Rarely finds improvements with significant business value
- ❑ Root Cause Analysis
  - Challenge assumptions about why problem exists
  - Trace symptoms to their causes to discover the “real” problem

168



## Các yêu cầu cải tiến trung bình To Identify Moderate Improvements

- ❑ Goal is to improve efficiency and effectiveness
- ❑ Expect moderate changes to existing systems
- ❑ Expect moderate impact and value to organization
- ❑ Types of activities:
  - Duration Analysis
  - Activity-Based Costing
  - Informal Benchmarking


169



## Các yêu cầu cải tiến lớn To Identify Major Improvements

- ❑ Goal is radical redesign of business processes
- ❑ Expect significant impact and value to organization
- ❑ Existing system is “obliterated:
- ❑ Activities focus on envisioning the business in new ways:
  - Outcome Analysis
  - Technology Analysis
  - Activity Elimination


170



## Phân tích kết quả Outcome Analysis

- ❑ Consider desirable outcomes from customers’ perspective
- ❑ Consider what the organization *could* enable the customer to do
- ❑ Brainstorm on desirable customer outcomes enabled by IS


171



## Cơ sở xác định yêu cầu hệ thống

- **Các nguồn điều tra**
  - Những người dùng hệ thống
  - Các sổ sách tài liệu
  - Các chương trình máy tính
  - Các tài liệu mô tả quy trình, chức trách
  - Các thông báo
- **Các bước xác định yêu cầu**
  - Khảo sát hiện trạng
  - Xác định yêu cầu chức năng
    - Về chức năng nghiệp vụ
    - Về chức năng hệ thống
  - Xác định yêu cầu phi chức năng


172



## Các phương pháp xác định yêu cầu

- Nghiên cứu các tài liệu viết
- Sử dụng phiếu thăm dò
- Phỏng vấn
- Quan sát
- ???


173



## 3.1.2. Mô hình hoá nghiệp vụ

- Tìm hiểu cấu trúc và khía cạnh động của tổ chức triển khai.
- Xác định vấn đề thực tại, các cải tiến để nâng cao hiệu quả tổ chức.
- Người dùng cuối và người phát triển có cái nhìn chung về tổ chức.
- Nắm bắt yêu cầu hệ thống cần hỗ trợ cho tổ chức


174



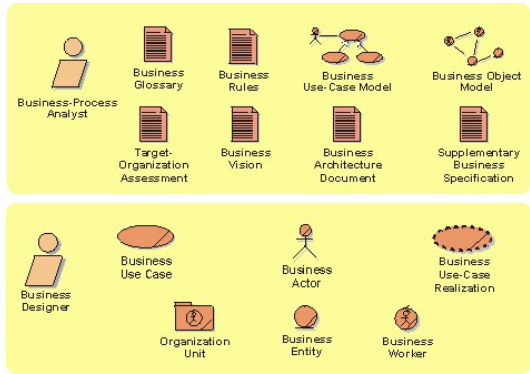
## 3.1.2. Mô hình hoá nghiệp vụ (tt) – Chế tác

Chế tác RUP	Nỗ lực
Business Vision	Low
Business Use Case Model (in Rose)	Medium
Business Use Case Model Survey	Low
Business Object Model	Medium
Business Object Model Survey	Low

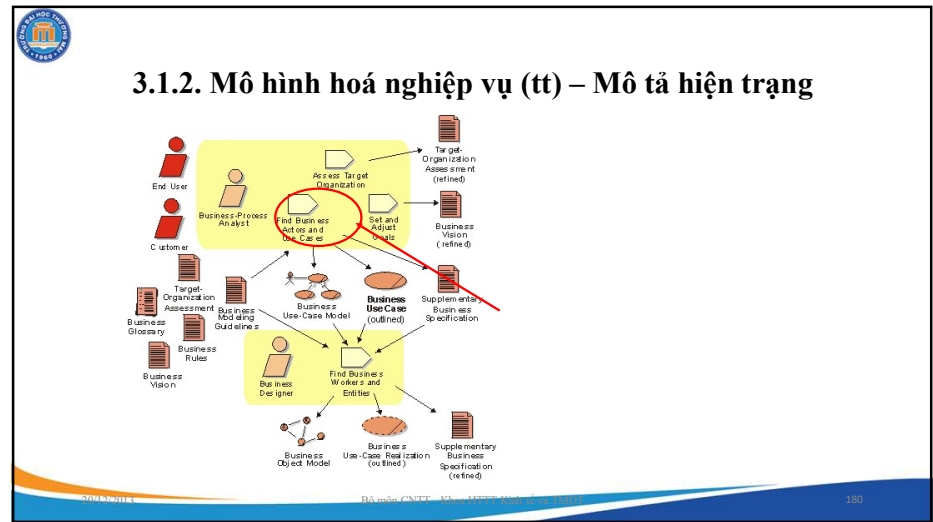
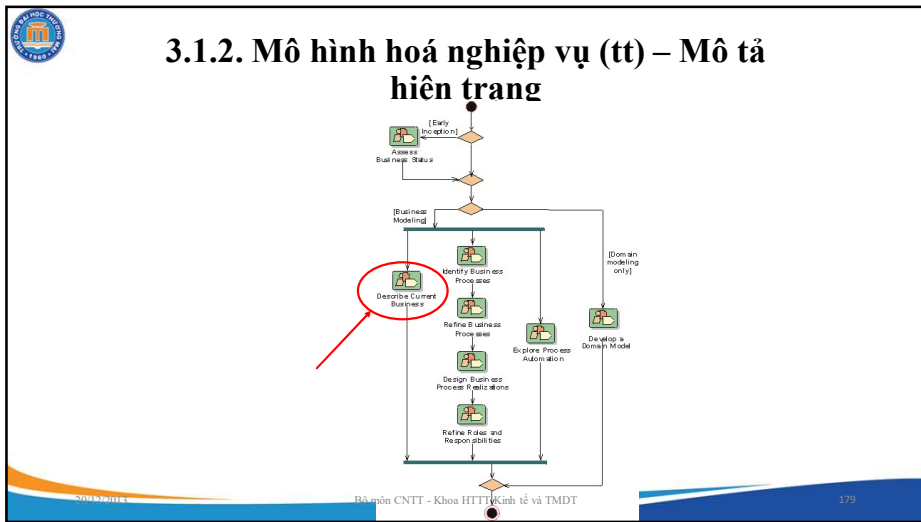
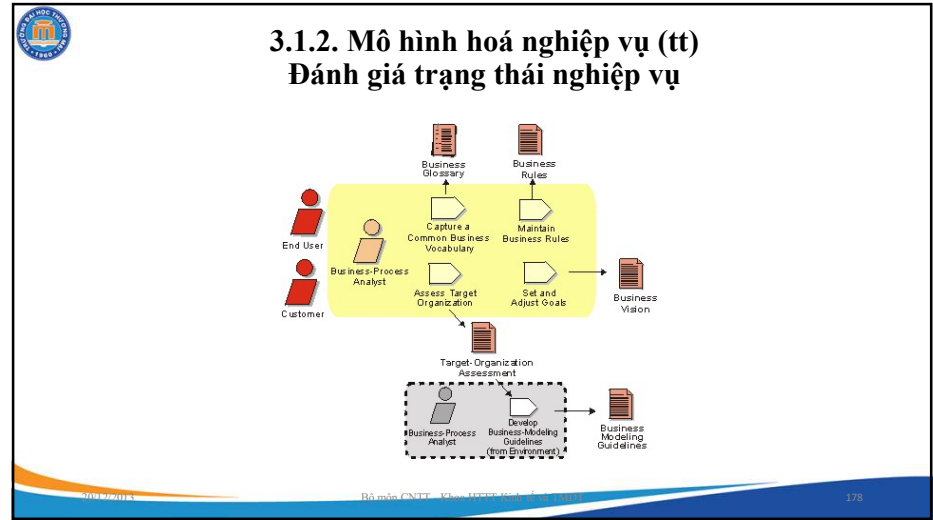
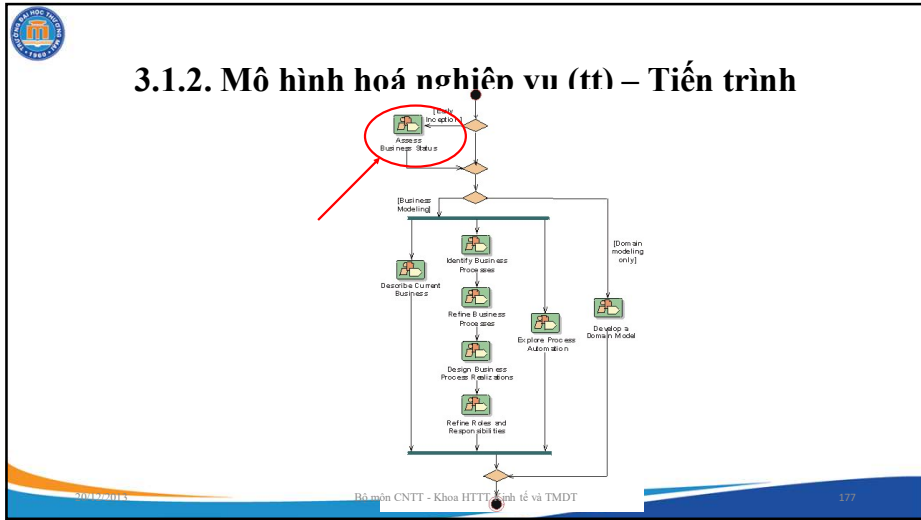
175



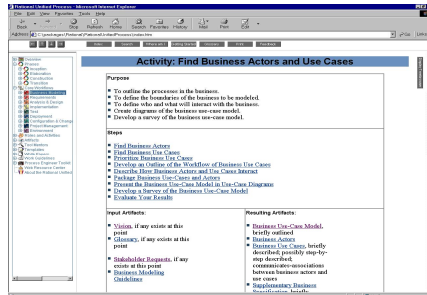
## 3.1.2. Mô hình hoá nghiệp vụ (tt) – Luồng công việc



176



### 3.1.2. Mô hình hoá nghiệp vụ (tt)



Lưu ý:

- Tốc độ quan trọng hơn chi tiết trong pha khởi đầu
- Làm sao nhận được bức tranh tổng thể đúng

## 3.2. Biểu đồ Usecase

### • Các bước xây dựng

- Xác định tác nhân và các use case
- Xác định các mối quan hệ và phân rã biểu đồ Usecase
- Mô tả các use case thông qua các kịch bản
- Kiểm tra và hiệu chỉnh biểu đồ

### • Tìm tác nhân:

- Ai sẽ là người sử dụng các chức năng của HT
- Ai cần sự hỗ trợ của HT để thực hiện công việc hàng ngày?
- Ai cần bảo trì, quản trị và đảm bảo HT hoạt động?
- HT sẽ tương tác với các HT nào khác?
- Ai hay cái gì quan tâm đến kết quả HT sinh ra

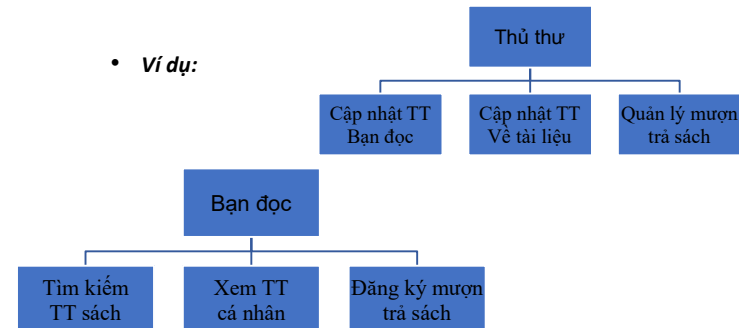
## 3.2. Biểu đồ Usecase

### • Tìm use case

- Tác nhân cần chức năng nào từ HT?
- Tác nhân cần phải xem, cập nhật hay lưu trữ thông tin gì trên HT?
- Tác nhân cần thông báo cho HT những sự kiện gì, sự kiện đó đại diện cho chức năng nào?
- HT có cần thông báo với tác nhân khi có sự thay đổi không?
- HT có chức năng gì để đơn giản hóa nhiệm vụ của tác nhân?
- Các tác nhân và Usecase còn có thể sinh ra bởi sự kiện nào khác (sự kiện thời gian, tác động của chức năng khác)?
- Hệ thống cần thông tin đầu vào đầu ra nào?

## 3.2. Biểu đồ Usecase

### • Ví dụ:



## 3.2. Biểu đồ Use case

- **Xác định mối quan hệ**
  - Các mối quan hệ được sử dụng?
    - Quan hệ Include:
    - Quan hệ extend:
    - Quan hệ generalization:
    - Quan hệ kết hợp: biểu diễn mỗi liên hệ
- **Trình tự phân rã biểu đồ use case**
  - Xác định biểu đồ ở mức tổng quát
  - Phân rã use case ở mức cao
  - Phân rã use case cho đến khi gặp use case ở mức lá
  - Hoàn thiện biểu đồ

## 3.2. Biểu đồ Usecase

- **Ở mức tổng quát:**
  - Các chức năng chính
  - Có mức khái quát cao
  - Dễ dàng nhìn thấy trên quan điểm của tác nhân
- **Ví dụ:**
  - + Đăng nhập
  - + Cập nhật
  - + Tìm kiếm

## 3.2. Biểu đồ Usecase


- **Ở mức cao:**
  - Sử dụng quan hệ extend
  - Thêm vào các usecase cha
- **Ví dụ:**
  - Cập nhật bao gồm
    - Cập nhật bạn đọc
    - Cập nhật tài liệu
- **Ở mức lá:**
  - Các chức năng thực sự tương tác với tác nhân
- **Ví dụ:**
  - Cập nhật bạn đọc:
    - Thêm bạn đọc
    - Thay đổi thông tin bạn đọc.
    - Xóa bạn đọc

## Biểu đồ Use case theo kịch bản

Use Case Name	Maintain Available Rental Unit Information	Make Old Patient Appr
ID	1	2
Importance Level	High	Low
Primary Actor	Apartment Owner	Old Patient
Use Case Type	Detail, Internal	Detail, Internal
Subscribers and Interest	Apartment Owner - wants to advertise available apartments Campus Housing Service - provides a service that enables the apartment owners to rent their available apartments	Subscribers and Interest: Old Patient - wants to make, change, or cancel an appointment Doctor - wants to ensure patient's needs are met in a timely manner
Brief Description	This use case describes how the campus housing service can maintain an up-to-date listing of available apartments.	This use case describes how we make an appointment as well as changing or canceling an appointment for a previously made patient.
Trigger	Apartment Owner wants to add or delete an available apartment	Patient calls and asks for a new appointment or asks to cancel or change an existing appointment
Type	External	External
Relationships	Association: Apartment Owner Include: External Generalization: Manage Appointments	Association: Old Patient Include: External Generalization: Manage Appointments
Normal Flow of Events	1. The Patient contacts the office regarding an appointment. 2. The Patient provides the Receptionist with his or her name and address. 3. If the Patient's information has changed... 4. If the Patient's payment arrangements has changed... 5. The Receptionist asks Patient if he or she would like to make a new appointment, cancel an existing appointment, or change an existing appointment.	1. The Patient contacts the office regarding an appointment. 2. The Receptionist matches the Patient's desired appointment times with available dates and times and schedules the new appointment. 3. Cancel Appointment. 1. The Receptionist asks the Patient for the old appointment time. 2. The Receptionist finds the correct appointment in the appointment file and cancels it. 3. Change Appointment. 1. The Receptionist performs the S.1, cancel appointment sub-flow. 2. The Receptionist performs the S.2, new appointment sub-flow.
Sub-Flows	S.1: New Appointment 1. The Receptionist asks the Patient for possible appointment times. 2. The Receptionist matches the Patient's desired appointment times with available dates and times and schedules the new appointment. S.2: Cancel Appointment 1. The Receptionist asks the Patient for the old appointment time. 2. The Receptionist finds the correct appointment in the appointment file and cancels it. S.3: Change Appointment 1. The Receptionist performs the S.1, cancel appointment sub-flow. 2. The Receptionist performs the S.2, new appointment sub-flow.	S.1: New Appointment 1. The Receptionist asks the Patient for possible appointment times. 2. The Receptionist matches the Patient's desired appointment times with available dates and times and schedules the new appointment. S.2: Cancel Appointment 1. The Receptionist asks the Patient for the old appointment time. 2. The Receptionist finds the correct appointment in the appointment file and cancels it. S.3: Change Appointment 1. The Receptionist performs the S.1, cancel appointment sub-flow. 2. The Receptionist performs the S.2, new appointment sub-flow.
Alternative/Exceptional Flows	S.1.1a: The Receptionist proposes some alternative appointment times based on what is available in the appointment schedule. S.1.1a: The Patient chooses one of the proposed times or decides not to make an appointment.	S.1.1a: The Receptionist proposes some alternative appointment times based on what is available in the appointment schedule. S.1.1a: The Patient chooses one of the proposed times or decides not to make an appointment.








## Câu hỏi ôn tập Chương 3

1. Trình bày phương pháp xác định yêu cầu nghiệp vụ?
2. Trình bày các loại biểu đồ trong PTTK HĐT?
3. Vẽ các biểu đồ tương ứng cho bài toán xây dựng HTTT Quản lý Nhân sự

Bộ môn CNTT - Khoa CNTT - Trường Đại học FPT


193



## Chương 4: Thiết kế hệ thống theo hướng đối tượng

Bộ môn CNTT - Khoa CNTT - Trường Đại học FPT

194




## Chuyển đổi từ yêu cầu sang thiết kế Transition from Requirements to Design

Brief preview

Bộ môn CNTT - Khoa CNTT - Trường Đại học FPT

195




## Key Ideas

- ❑ In *Systems Analysis* we figure out...
  - What the business needs
- ❑ In *System Design* we figure out...
  - How to build the system that fulfills those needs
- ❑ All of the “logical” work from Systems Analysis is converted to the “physical”

Bộ môn CNTT - Khoa CNTT - Trường Đại học FPT

196



## Key Definitions

- Design phase
  - Decide *how* to build the system
  - Create *system requirements* that describe all technical details for building the system
- System specification
  - Final deliverable from design phase
  - Conveys exactly what system the development team will implement during the implementation phase

Bộ môn CNTT - Khoa CNTT - Học Viện Kỹ Thuật Điện Tử



## Các bước trong pha thiết kế Design Phase Steps


- Determine system acquisition strategy (make, buy, or outsource)
- Determine the technical architecture for the system
- Address security concerns and globalization issues
- Make hardware and software selections
- Determine the way that users will interact with the system (interface, inputs, and outputs)
- Design the programs for the underlying processes
- Design the way data will be stored
- Create final deliverable - the *system specification*

Bộ môn CNTT - Khoa CNTT - Học Viện Kỹ Thuật Điện Tử

## Elements of System Specification

- Recommended System Acquisition Strategy
- System Acquisition Weighted Alternative Matrix
- Architecture Design
- Hardware and Software Specification
- Interface Design
- Physical Process Model
- Program Design Specifications
- Physical Data Model
- Data Storage Design
- Updated CRUD Matrix
- Updated CASE Repository Entries

Bộ môn CNTT - Khoa CNTT - Học Viện Kỹ Thuật Điện Tử



## Nội dung

- 4.1. Thiết kế các hệ thống con
- 4.2. Thiết kế giao diện người dùng và thiết kế lớp
- 4.3. Thiết kế việc lưu trữ các dữ liệu
- 4.4. Mô hình hóa cài đặt hệ thống

Bộ môn CNTT - Khoa CNTT - Học Viện Kỹ Thuật Điện Tử



## Giới thiệu

- Sau khi xác định và phân tích yêu cầu hệ thống, chúng ta chuyển sang pha thiết kế và cài đặt hệ thống. Thiết kế kiến trúc hệ thống là giai đoạn sớm nhất trong quy trình thiết kế hệ thống. Thiết kế kiến trúc cung cấp cho chúng ta bản đặc tả về kiến trúc hệ thống, bao gồm những hệ thống con nào, tương tác với nhau ra sao, framework hỗ trợ điều khiển tương tác giữa các hệ thống con như thế nào ...

2018/2019

Bộ môn CNTT - Khoa CNTT - Trường ĐHQGHN

201



## Thiết kế kiến trúc Architecture Design

Systems analysis and design, 6<sup>th</sup> edition

Dennis, wixom, and roth

2018/2019

Bộ môn CNTT - Khoa CNTT - Trường ĐHQGHN

204



## Key Definitions

- Architecture design
  - Plans for how the system will be distributed across computers and what hardware and software will be used for each computer.
- Hardware and software specification
  - Describes the hardware/software components in detail to aid those responsible for purchasing those products.

2018/2019

Bộ môn CNTT - Khoa CNTT - Trường ĐHQGHN

203




## Objective of Architecture Design

- Assign the software components of the information system to the hardware devices of the system in the most advantageous way.
- The major *architectural components* of any system are the software and the hardware.

2018/2019

Bộ môn CNTT - Khoa CNTT - Trường ĐHQGHN


204



## Architectural Components

- Software systems can be divided into four basic functions:
  - **Data storage.**
  - **Data access logic:** the processing required to access stored data.
  - **Application logic:** the logic documented in the DFDs, use cases, and functional requirements.
  - **Presentation logic:** the display of information to the user and the acceptance of the user's commands.


205



## Architectural Components (cont'd)

- The three primary hardware components:
  - **Client computers:** Input-output devices employed by users (e.g., PCs, laptops, handheld and mobile devices, smart phones)
  - **Servers:** Larger multi-user computers used to store software and data.
  - **Network:** Connects the computers.


206



## Client-Server Architectures

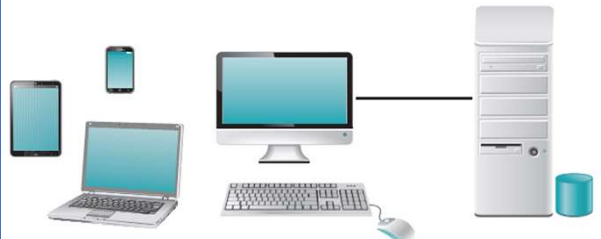
- Client-server architectures balance the processing between client devices and one or more server devices.
- Generally, clients are responsible for the presentation logic, and
- the server(s) are responsible for the data access logic and data storage.
- Application logic location varies depending on the C-S configuration chosen.

207



## Two-Tiered Client-Server Architecture

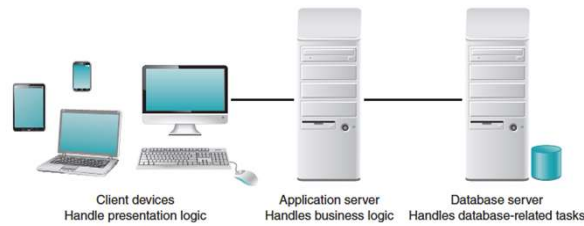
- Thick client – most of application logic on the client side (shown here)
- Thin client – little application logic on the client side; most shifted to server side



208

## Three-Tiered Client-Server Architecture

- Adds "specialized" servers – one for application logic; one for data base tasks



20/12/2013

Bộ môn CNTT - Khoa CNTT - Trường ĐHQG Hà Nội

209

## n-Tiered Client-Server Architecture

- Adds "specialized" servers – one for Web-related business logic; one for application logic; one for data base tasks



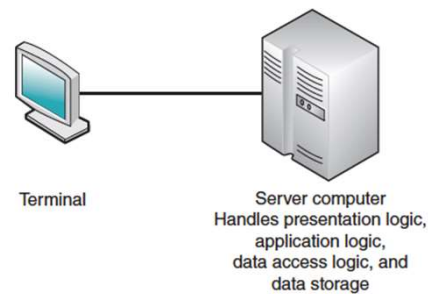
20/12/2013

Bộ môn CNTT - Khoa CNTT - Trường ĐHQG Hà Nội

210

## Server-Based Architecture

Zero-client used today in virtual desktop infrastructure (VDI)



20/12/2013

Bộ môn CNTT - Khoa CNTT - Trường ĐHQG Hà Nội

211



## Mobile Application Architecture

- Rich client – involves processing on the mobile device using its resources. Presentation logic, business logic, and data access logic on the client side.
- Thin Web-based client – business and data access logic on the server side; always connected to server.
- Rich Internet application – browser-based; uses some technologies on client device to provide a rich user interface (e.g., Flash).

20/12/2013

Bộ môn CNTT - Khoa CNTT - Trường ĐHQG Hà Nội

212



## Mobile Application Options

- ❑ Native app – written to run on specific device with specific operating system.
- ❑ Cross-platform frameworks – develop in web-based technologies and use framework to deploy to multiple devices.
- ❑ Mobile Web app – browser-based; platform independent. Most limited user experience.



## Selecting an Architecture Design

- ❑ Lower costs often used to justify choice of client-server
- ❑ Recommended selection process:
  - Expand nonfunctional requirement details
  - Base architecture selection on the detailed nonfunctional requirements
    - Operational,
    - Performance,
    - Security, and
    - Cultural/political



## Operational Requirements


Requirement	Definition	Example
<b>Technical Environment</b>	Special hardware, software, and network requirements imposed by business requirements	All office locations have always-on network connection permitting real-time database updates
<b>System Integration</b>	The extent to which the system will operate with other systems	The system will read and write to the main inventory database
<b>Portability</b>	The extent to which the system will need to operate in other environments	The system must operate with mobile devices (Android and iOS)
<b>Maintainability</b>	Expected business changes to which the system should be able to adapt	The system must accommodate new manufacturing plants



## Performance Requirements

Requirement	Definition	Example
<b>Speed</b>	Time within which the system must perform its function	Network transaction response time $\leq 4$ seconds
<b>Capacity</b>	Total and peak number of users and the volume of data expected	Maximum of 2000 simultaneous users at peak use times
<b>Availability and Reliability</b>	Extent to which the system will be available to the users and the permissible failure rate due to errors	99% uptime performance






## Security Requirements

Requirement	Definition	Example
<b>System Value Estimates</b>	Estimated business value of the system and its data	A complete loss of all system data would cost \$20 million
<b>Access Control</b>	Limitations on who can access what data	Inventory item changes can be made only by managers for items in their own department
<b>Encryption and Authentication</b>	Defines what data will be encrypted where and whether authentication will be needed for user access	Data will be encrypted from the user's computer to the Web site to provide secure ordering
<b>Virus Control</b>	Controls to limit viruses	All uploaded files will be checked for viruses before being saved in the system

Bộ môn CNTT - Khoa CNTT - Trường Đại học Khoa học Tự nhiên TP.HCM



## Cultural/Political Requirements


Requirement	Definition	Example
<b>Multilingual</b>	The language(s) the system users will need	The system will operate in English, French, and Spanish
<b>Customization</b>	Specification of what aspects of the system can be changed by local users	Country managers will be able to define new fields in the product database to capture country-specific information
<b>Making Unstated Norms Explicit</b>	Explicitly stating assumptions that differ from country to country	All weights will be stated in pounds and in kilograms
<b>Legal</b>	The laws and regulations that impose system requirements	Personal customer information cannot be transferred from European Union countries to US

Bộ môn CNTT - Khoa CNTT - Trường Đại học Khoa học Tự nhiên TP.HCM

## Nonfunctional Requirements and the Architecture Design

Requirements	Server-Based	Thin Client-Server	Thick Client-Server
<b>Operational Requirements</b>			
System Integration Requirements	✓	✓	✓
Portability Requirements		✓	
Maintainability Requirements	✓	✓	
<b>Performance Requirements</b>			
Speed Requirements		✓	✓
Capacity Requirements		✓	✓
Availability/Reliability Requirements	✓	✓	✓
<b>Security Requirements</b>			
High System Value	✓	✓	
Access Control Requirements	✓		
Encryption/Authentication Requirements		✓	✓
Virus Control Requirements	✓		
<b>Cultural/Political Requirements</b>			
Multilingual Requirements		✓	
Customization Requirements		✓	
Making Unstated Norms Explicit		✓	
Legal Requirements	✓	✓	✓

Bộ môn CNTT - Khoa CNTT - Trường Đại học Khoa học Tự nhiên TP.HCM



## Đặc tả phần cứng và phần mềm Hardware Software Specification

Outlining needs in new system

Bộ môn CNTT - Khoa CNTT - Trường Đại học Khoa học Tự nhiên TP.HCM



## HW/SW Specification Purpose

- Used if new hardware or software must be purchased
- Communicates project needs
- Actual acquisition of hardware and software may be done by a purchasing department -- especially in larger firms.

## Sample HW/SW Specification

	Standard Client	Standard Web Server	Standard Application Server	Standard Database Server
Operating System	• Windows 7 Pro	• Linux	• Linux	• Linux
Special Software	• Real Audio • Adobe Acrobat Reader	• Apache	• Java	• Oracle
Hardware	• 500-GB disk drive • Intel® Core™ i3-4130 dual core processor • 22-inch LCD Monitor	• 500-GB disk drive • Quad-core Xeon	• 500-GB disk drive • Six-core Xeon	• 1-TB disk drive • RAID • Eight core Xeon
Network	• Always-on Broadband, preferred • Dial-up at 56 Kbps, possible with some performance loss	• Dual 100 Mbps Ethernet	• Dual 100 Mbps Ethernet	• Dual 100 Mbps Ethernet



## Thiết kế giao diện người dùng User Interface Design


Systems analysis and design, 6<sup>th</sup> edition

Dennis, wixom, and roth



## Key Definitions

- System Interface: “connections” with other systems, where systems exchange information with each other. Designed as a part of program design.
- User Interface: “connections” with users. Focus of this chapter.
  - The **navigation mechanism** provides the way for users to tell the system what to do
  - The **input mechanism** defines the way the system captures information
  - The **output mechanism** defines the way the system provides information to users or other systems



## Usability Concept

- The system is easy to use and easy to learn
- Tasks are completed more efficiently and with more accuracy
- Mistakes with system are reduced
- User satisfaction with new system is increased
- Adoption of system is more likely


225



## Nguyên lý thiết kế giao diện Interface Design Principles

General guidelines for user interface design


226



## Principles for User Interface Design

- Layout
- Content awareness
- Aesthetics
- Usage level
- Consistency
- Minimize user effort


227



## Layout Concepts

- The screen is often divided into three boxes
  - Navigation area (top)
  - Status area (bottom)
  - Work area (middle)
- Information can be presented in multiple areas
- Like areas should be grouped together

228



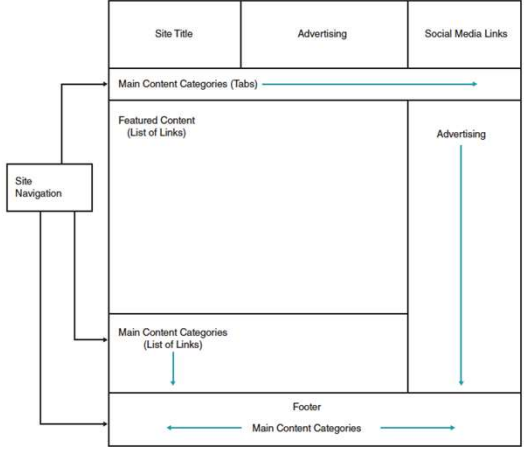
## More Layout Concepts

- Areas and information should minimize user movement from one to another
- Ideally, areas will remain consistent in
  - Size
  - Shape
  - Placement for entering data
  - Reports presenting retrieved data

229

## Model Layout for Web Page

○ Note use of multiple layout areas for site navigation

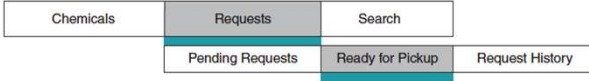


230


## Content Awareness

○ Note the use of highlighting to indicate menu selections

○ Breadcrumbs provide additional clues on navigational path




231



## Usage Level

- Some people will be frequent, heavy users of the system
- Frequent users desire **ease of use** – quick and easy completion of job tasks
- Other people may use the system infrequently
- Infrequent users desire **ease of learning** – quick and easy ways to figure out what to do.

232



## Consistency

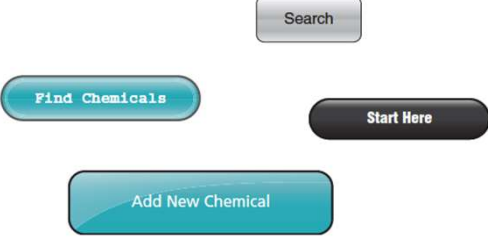
- Elements are the same throughout the application
- Enables users to predict what will happen
- Reduces learning curve
- Considers elements within an application and across applications
- Pertains to many different levels
  - Navigation controls
  - Terminology
  - Report and form design

233




## Example of Inconsistent Elements

Note the different button styles, colors, and font styles.




234



## Minimize Effort

- Three clicks rule
  - Users should be able to go from the start or main menu of a system to the information or action they want in no more than three mouse clicks or three keystrokes

235



## Special Issues of Touch Screen Design

- Ideal for information display but not data entry.
- Place content at top and navigation controls at bottom so finger does not obscure content area.
- Place labels on top of navigation controls.
- Size objects correctly for “fat fingers.”
- Include adequate spacing between objects.

236

## Android Device Common Hand Gestures

Gesture	Meaning/Function
Touch/tap	Triggers element's default functionality
Long press	Triggers data selection mode
Swipe/drag	Scrolls through content. Swipe is quick and the effect continues after finger is picked up. Drags are slower, more precise, and stop when the finger is picked up.
Long press drag	Rearranges or moves element
Double touch/tap	Scales up around the target
Double touch drag	Scales content by pushing away or pulling closer.
Two-finger pinch open	Zooms into content
Two-finger pinch close	Zooms out of content

Source: Android Gestures, developer.android.com/design/patterns/gestures

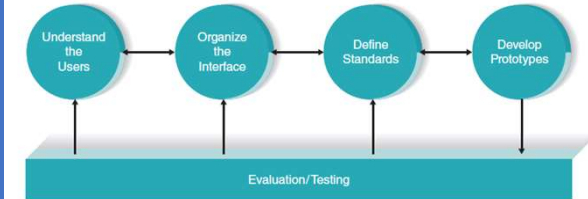
20/12/2014

Bộ môn CNTT - Khoa CNTT - Trường ĐHQGHN

237

## User Interface Design Process

- Understand the Users
- Organize the Interface
- Define Standards
- Develop Prototypes
- Evaluation / Testing



20/12/2014

Bộ môn CNTT - Khoa CNTT - Trường ĐHQGHN

238

## Understand the Users

- Users likely will have very different goals and intentions when using the system.
- Use **personas** to develop characterizations of various user groups.
  - Interests
  - Typical behaviors
  - Goals and objectives
  - Expectations
- Plan a user interface that will be satisfying for that particular user group.



20/12/2014

Bộ môn CNTT - Khoa CNTT - Trường ĐHQGHN

239

## Personas for Tune Source

### Angelica, the Browsing Shopper

*"I like to take my time, do my research, and consider my options before buying"*

**Knowledge:** Not familiar with, or superficial knowledge of the artists and music selections offered by Tune Source.  
**Tasks:** Arrives at the site with the intent to look around. Wants to explore a number of options and thoroughly research those options by looking at material and listening to samples. Wants to be able to save some selections to consider for a period of time before buying.  
**Interests:** Enjoys music. Has eclectic tastes. Not content with mainstream choices. Likes to own some out-of-the-ordinary music selections.  
**Characteristics:** Unhurried; deliberate; thoughtful.

### Carl, the Hurry Up Shopper

*"I know what I want, and I want to buy it quickly"*

**Knowledge:** Very familiar with the artists and music selections offered by Tune Source.  
**Tasks:** Arrives at the site with a specific purchase in mind. Wants to select the item quickly, pay for it quickly, and initiate the download immediately.  
**Interests:** May have studied music or work in some aspect of the music industry.  
**Characteristics:** Impatient; dislikes unexpected delays; expects things to work seamlessly.

Bộ môn CNTT - Khoa CNTT - Trường ĐHQGHN

240

## Use Scenarios for Tune Source

**Use Scenario: The Browsing Shopper**  
User is not sure what they want to buy and will browse through several tunes.

1. User may search for a specific artist or browse through a music category (1.2).
2. User will likely read the basic information for several tunes, as well as the marketing material for some. He or she will likely listen to music samples and browse related tunes (1.3).
3. User will put the tune in the shopping cart (1.3) and will continue browsing (1.2).
4. Eventually, the user will want to purchase the download, but will probably want to look through the shopping cart, possibly discarding some tunes first (1.3).

**Use Scenario: The Hurry-Up Shopper**  
User knows exactly what he or she wants and wants it quickly.

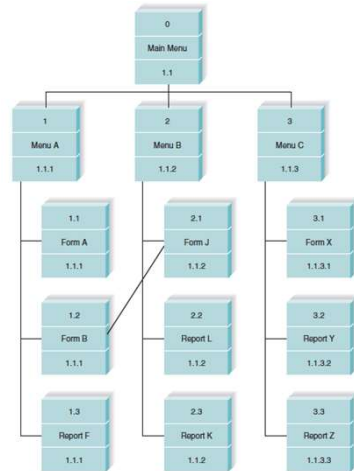
1. User will search for a specific artist or tune (1.2).
2. User will look at the price and enough other information to confirm that the tune is the desired one (1.3).
3. User will want to buy the download (process 2) or move on to other web sites.

The numbers in parentheses refer to process numbers in the DFD.

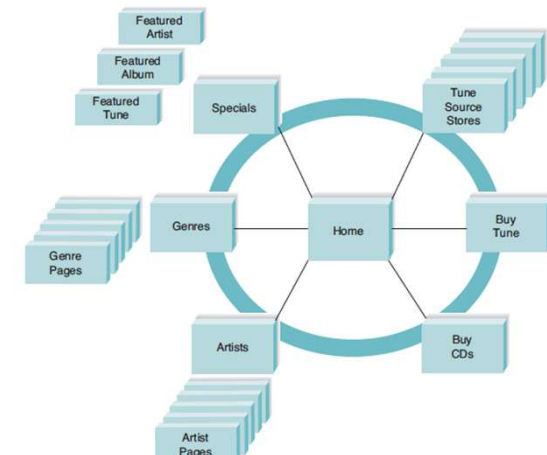
## Organize the Interface

- Define the basic components of the interface and how they work together to provide functionality to users.
- Use **Interface Structure Diagram (ISD)**
- Shows how all screens, forms, and reports are related
- Shows how user moves from one to another
- Similar to DFD in using boxes and lines
  - Boxes denote screens
  - Lines show movement from one to another
- Different from DFD in having no standard rules or format

## Interface Structure Diagram for Tune Source

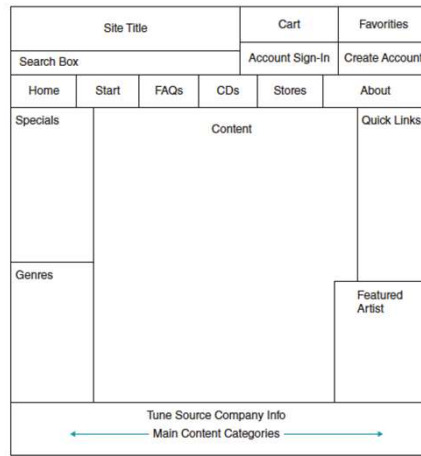


## Site Map for Tune Source





# Wireframe Diagram for Tune Source

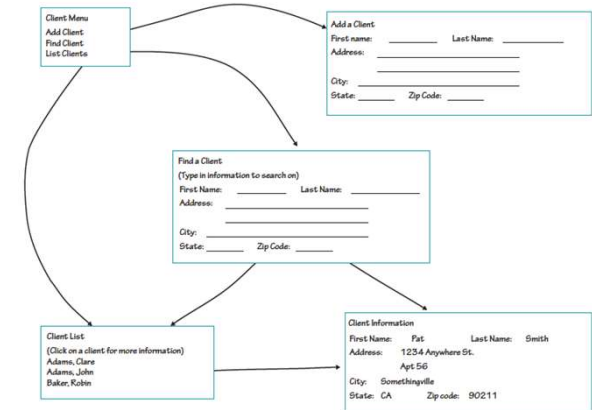


20/12/2014

Bộ môn CNTT - Khoa CNTT - Trường Đại học Sư phạm Hà Nội

245

# Storyboard Example

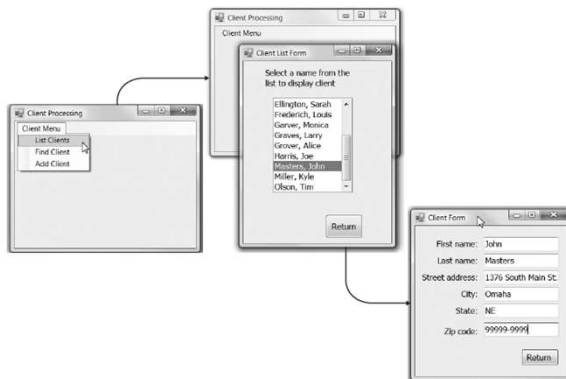


20/12/2014

Bộ môn CNTT - Khoa CNTT - Trường Đại học Sư phạm Hà Nội

246

# Language Prototype Example



20/12/2014

Bộ môn CNTT - Khoa CNTT - Trường Đại học Sư phạm Hà Nội

247

# Thiết kế chương trình Program Design

Systems analysis and design, 6<sup>th</sup> edition  
Dennis, wixom, and roth





## The Physical Process Model

- Show the implementation details and explain how the system will work, including
  - Actual, specific technology
  - Format of information
  - Human interaction with system

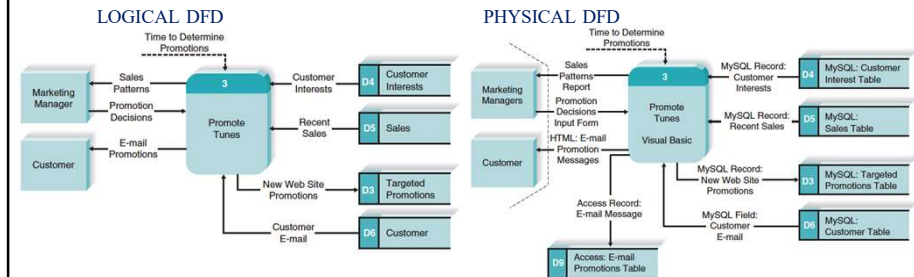
SOT/2011/13

Đà Nẵng CNTT - Khoa CNTT - Trường ĐHTN

249



## Contrasting a Logical and Physical DFD



SOT/2011/13

Đà Nẵng CNTT - Khoa CNTT - Trường ĐHTN

250



## Key Definitions

- *Program design* - creating instructions for the programmers
- *The top-down, modular approach* - begin with the “big picture” and gradually add detail
- *Program design document* – all structure charts and specifications needed by programmers to implement the system

SOT/2011/13

Đà Nẵng CNTT - Khoa CNTT - Trường ĐHTN

251



## The Structure Chart

- Important program design technique
- Shows all components of code in a hierarchical format
  - Sequence
  - Selection
  - Iteration
- Illustrates the organization and interactions of the different program modules

SOT/2011/13

Đà Nẵng CNTT - Khoa CNTT - Trường ĐHTN

252

## Example Structure Chart

Structure Chart Element	Purpose	Symbol
<b>Every module:</b> <ul style="list-style-type: none"> <li>Has a number.</li> <li>Has a name.</li> <li>Is a control module if it calls other modules below it.</li> <li>Is a subordinate module if it is controlled by a module at a higher level.</li> </ul>	Denotes a logical piece of the program.	
<b>Every library module has:</b> <ul style="list-style-type: none"> <li>A number.</li> <li>A name.</li> <li>Multiple instances within a diagram.</li> </ul>	Denotes a logical piece of the program that is repeated within the structure chart.	
<b>A loop:</b> <ul style="list-style-type: none"> <li>Is drawn with a curved arrow.</li> <li>Is placed around trees of one or more modules that are repeated.</li> </ul>	Communicates that a module(s) is repeated.	
<b>A conditional face:</b> <ul style="list-style-type: none"> <li>Is drawn with a diamond.</li> <li>Includes modules that are invoked on the basis of some condition.</li> </ul>	Communicates that subordinate modules are invoked by the control module based on some condition.	
<b>A data couple:</b> <ul style="list-style-type: none"> <li>Contains an arrow.</li> <li>Contains an empty circle.</li> <li>Names the type of data that is being passed.</li> <li>Can be passed up or down.</li> <li>Has a direction that is denoted by the arrow.</li> </ul>	Communicates that data are being passed from one module to another.	
<b>A control couple:</b> <ul style="list-style-type: none"> <li>Contains an arrow.</li> <li>Contains a flag or flag that is being passed.</li> <li>Should be passed up, not down.</li> <li>Has a direction that is denoted by the arrow.</li> </ul>	Communicates that a message or a system flag is being passed from one module to another.	
<b>An off-page connection:</b> <ul style="list-style-type: none"> <li>Is denoted by the hexagon.</li> <li>Is used when the diagram is too large to fit everything on the same page.</li> </ul>	Identifies when parts of the diagram are continued on another page of the structure chart.	
<b>An on-page connection:</b> <ul style="list-style-type: none"> <li>Is denoted by the circle.</li> <li>Has a flag.</li> <li>Is used when the diagram is too large to fit everything on the same spot on a page.</li> </ul>	Identifies when parts of the diagram are continued somewhere else on the same page of the structure chart.	

253

## Building the Structure Chart

- Processes in the DFD tend to represent one module on the structure chart
  - Afferent processes – provide inputs to system
  - Central processes – perform critical system operations
  - Efferent processes – handle system outputs
- The DFD leveling can correspond to the structure chart hierarchy

254

## Types of Cohesion

- Functional
- Sequential
- Communicational
- Procedural
- Temporal
- Logical
- Coincidental

Good  
↓  
Bad

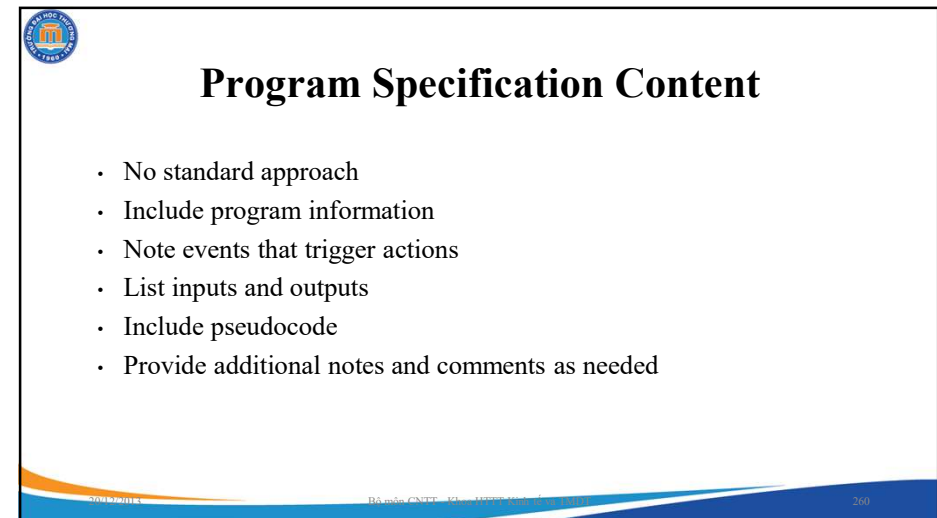
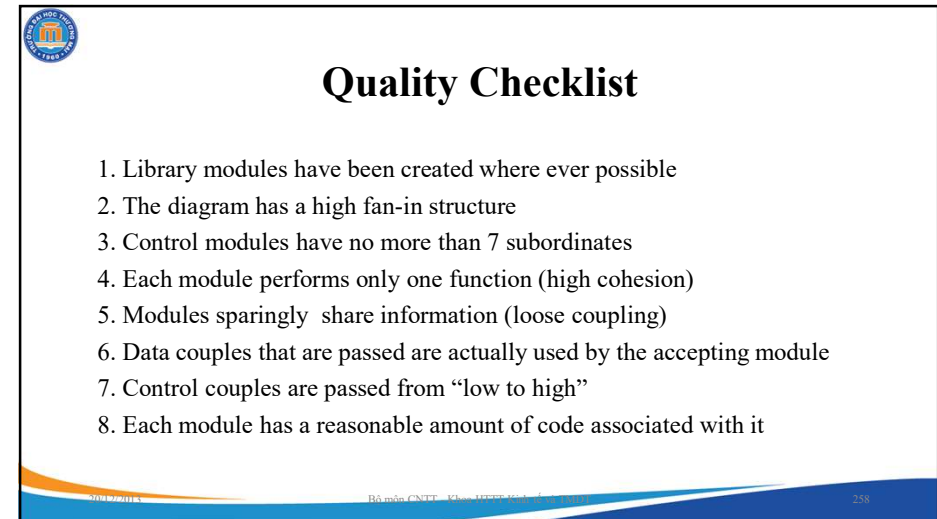
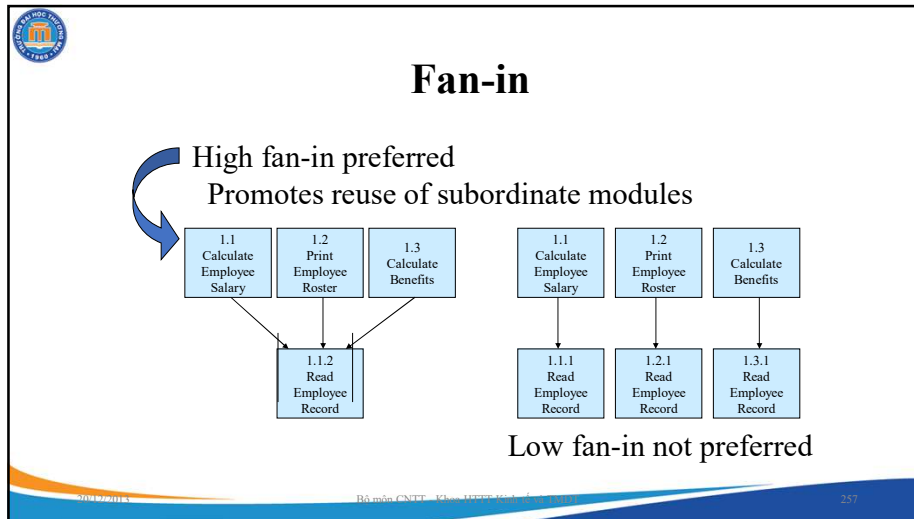
255

## Types of Coupling

- Data
- Stamp
- Control
- Common
- Content

Good  
↓  
Bad

256



## Sample Pseudocode

```

Calculate_discount_amount (total_price, discount_amount)
If total_price < 50 THEN
    discount_amount = 0
ELSE
    If total_price < 500 THEN
        discount_amount = total_price *.10
    ELSE
        discount_amount = total_price *.20
    END IF
END IF
END
  
```

20/12/2014

Bộ môn CNTT - Khoa CNTT - Trường ĐHQG Hà Nội

261

## Sample Program Specification

Program Specification 1.1.2.2 for Digital Music Download System

Module Name: Find\_Tune\_By\_Title  
 Purpose: Display basic tune information, using a title input by the user  
 Programmer: John Smith  
 Date due: April 26, 2015

C  HTML/PHP  Visual Basic  Javascript

Events  
 search by title push-button is clicked  
 search by title hyperlink is selected

Input Name	Type	Provided by	Notes
Tune title	String (50)	Program 1.1.2	

Output Name	Type	Used by	Notes
Tune ID	String (10)	Program 1.1.2	
Not_found	Logical	Program 1.1.2	Used to communicate when tune is not found.

Pseudocode  
 (Find\_tune module)  
 not\_found = True  
 For all tune titles in Available\_Tunes table  
 If user title matches tune title, save tune ID  
 not\_found = False  
 End If  
 End For  
 Return

Business rule: If no matching tunes are found, the "Artist of the week" will appear to the user.  
 Note: A control couple containing a not\_found flag should be included from 1.1.2.2 to 1.1.2 to instruct 1.1.2 to display a not found message to the user and the Artist of the week.

20/12/2014

Bộ môn CNTT - Khoa CNTT - Trường ĐHQG Hà Nội

262



# Thiết kế lưu trữ dữ liệu Data Storage Design

Systems analysis and design, 6<sup>th</sup> edition  
 Dennis, wixom, and roth



## Key Concepts

- **Data storage function:** how data is stored and handled by programs that run the system.
- **Data storage design:**
  - select the data storage format;
  - convert the logical data model into a *physical* data model to reflect implementation decisions;
  - ensure that DFDs and ERDs balance; and
  - design the selected data storage format to optimize its processing efficiency.

20/12/2014

Bộ môn CNTT - Khoa CNTT - Trường ĐHQG Hà Nội

264



## DATA STORAGE FORMATS

- Types of data storage formats:
  - Files**: electronic lists of data, optimized to perform a particular transaction.
  - Database**: a collection of groupings of information that are related to each other in some way.
- Database Management System (DBMS)**: software that creates and manipulates the databases.



## Files

- Data file**: an electronic list of information that is formatted for a particular transaction.
- Sequential organization is typical.
- Record associations with other records created by **pointers**.
- Also called **linked lists** because of the way the records are linked together using pointers.




## Types of Files

- Master files** – store core information that is important to the application.
- Look-up files** – contain static values.
- Transaction files** – store information that can be used to update a master file.
- Audit files** – record “before” and “after” images of data as the data is altered.
- History files** (or archive files) – store past transactions.

## Appointment File Example

Appointment Date	Appointment Time	Duration	Reason	Patient ID	First Name	Last Name	Phone Number	Doctor ID	Doctor Last Name
11/23/2015	2:30	0.25 hour	Flu	758843	Patrick	Dennis	548-9456	V524625587	Vroman
11/23/2015	2:30	1 hour	Physical	136136	Adelaide	Kin	548-7887	T445756225	Tantalo
11/23/2015	2:45	0.25 hour	Shot	544822	Chris	Pullig	525-5464	V524625587	Vroman
11/23/2015	3:00	1 hour	Physical	345344	Felicia	Marston	548-9333	B544742245	Brousseau
11/23/2015	3:00	0.5 hour	Migraine	236454	Thomas	Bateman	667-8955	V524625587	Vroman
11/23/2015	3:30	0.5 hour	Muscular	887777	Ryan	Nelson	525-4772	V524625587	Vroman
11/23/2015	3:30	0.25 hour	Shot	966233	Peter	Todd	667-2325	T445756225	Tantalo
11/23/2015	3:45	0.75 hour	Muscular	951657	Mike	Morris	663-8944	T445756225	Tantalo
11/23/2015	4:00	1 hour	Physical	223238	Ellen	Whitener	525-8874	B544742245	Brousseau
11/23/2015	4:00	0.5 hour	Flu	365548	Jerry	Starla	548-9887	V524625587	Vroman
11/23/2015	4:30	1 hour	Minor surg	398633	Susan	Perry	525-6632	V524625587	Vroman
11/23/2015	4:30	0.5 hour	Migraine	222577	Elizabeth	Gray	667-8400	T445756225	Tantalo
11/24/2015	8:30	0.25 hour	Shot	858756	Elias	Awad	663-6364	T445756225	Tantalo
11/24/2015	8:30	1 hour	Minor surg	232158	Andy	Ruppel	525-9888	V524625587	Vroman
11/24/2015	8:30	0.25 hour	Flu	244875	Rick	Grenci	548-2114	B544742245	Brousseau
11/24/2015	8:45	0.5 hour	Muscular	655683	Eric	Meier	667-0254	T445756225	Tantalo
11/24/2015	8:45	1 hour	Physical	447521	Jane	Pace	548-0025	B544742245	Brousseau
11/24/2015	9:30	0.5 hour	Flu	554263	Trey	Maxham	663-8547	V524625587	Vroman



## Databases

- There are many types of databases:
  - Legacy database
  - Relational database
  - Object database
  - Multidimensional database
  - NoSQL database

269

## Appointment Database

Appointment Date	Appointment Time	Duration	Reason	Patient ID	Doctor ID
11/23/2015	2:30	0.5 hour	Flu	708043	V524625587
11/23/2015	2:30	1 hour	Physical	136136	T445762225
11/23/2015	2:45	0.25 hour	Shot	544822	V524625587
11/23/2015	3:00	1 hour	Physical	345344	8544782246
11/23/2015	3:00	0.5 hour	Migrane	236454	V524625587
11/23/2015	3:30	0.5 hour	Muscular	887777	V524625587
11/23/2015	3:30	0.25 hour	Shot	966253	T445762225
11/23/2015	3:45	0.75 hour	Muscular	951657	T445762225
11/23/2015	4:00	1 hour	Physical	222298	8544782246
11/23/2015	4:00	0.5 hour	Flu	365548	V524625587
11/23/2015	4:30	1 hour	Minor surg	398633	V524625587
11/23/2015	4:30	0.5 hour	Migrane	222577	T445762225
11/24/2015	8:30	0.25 hour	Shot	858758	T445762225
11/24/2015	8:30	1 hour	Minor surg	222158	V524625587
11/24/2015	8:30	0.25 hour	Flu	244876	8544782246
11/24/2015	8:45	0.5 hour	Muscular	855683	T445762225
11/24/2015	8:45	1 hour	Physical	447521	8544782246
11/24/2015	9:30	0.5 hour	Flu	554263	V524625587


Tables related by patient ID

Patient ID	First Name	Last Name	Phone Number
136136	Adelaide	Kim	545-7887
222577	Elizabeth	Gray	667-8400
222298	Eileen	Whitener	525-8074
222158	Andy	Ruppel	525-9088
236454	Thomas	Bushman	667-8855
244876	Rick	Greenl	548-2114
244344	Patricia	Manson	548-9333
365548	Jerry	Stacia	548-9887
398633	Buban	Ferry	525-0622
447621	Janae	Pike	548-0205
544822	Chris	Puig	525-6464
554263	Thy	Martinez	663-8547
665683	Eric	Melzer	667-0254
708043	Patrick	Dieryn	548-9456
858758	Clara	Amel	663-5564
887777	Ryan	Nelson	525-4772
951657	Mina	Morris	663-8944
966253	Peier	Toed	667-2525

Tables related by doctor ID

Doctor ID	Last Name
8544782246	Brunswick
T445762225	Tarvato
V524625587	Woman


270



## Object Databases

- Based on object orientation: that all things should be treated as *objects* that have both data (attributes) and processes (behaviors).
- *Object-oriented database management system (OODBMS)* are mainly used to support multimedia applications or systems that involve complex data.
- Play a minor role in the DBMS market at this time.

271



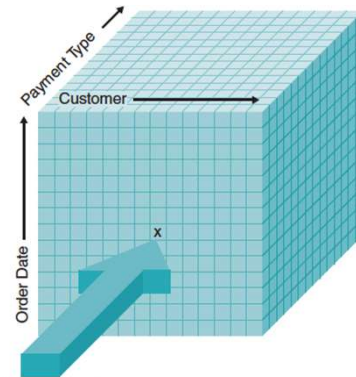
## Multidimensional Databases

- A type of relational database used extensively in data warehousing.
- *Data warehousing* is the practice of taking and storing data in a data warehouse (i.e., a large database) that supports *business intelligence (BI)* systems.
- *Data marts* are smaller databases based on data warehouse data; support BI for specific departments or functional areas of the organization.

272



## Multidimensional Database



*Last quarter, how many customers placed more than one order, using an American Express card?*

20/12/2014

Bộ môn CNTT - Khoa CNTT - Trường ĐHQG Hà Nội

273

## Selecting a Storage Format

- Each of the file and database data storage format has its strengths and weaknesses.
- Factors to consider in selecting a storage format:
  - Data Types
  - Type of Application System
  - Existing Storage Formats
  - Future Needs

20/12/2014

Bộ môn CNTT - Khoa CNTT - Trường ĐHQG Hà Nội

274

## Comparing Storage Formats

	Files	Legacy databases	Relational databases	Object databases	Multidimensional databases	NoSQL databases
Major strengths	Files can be designed for fast performance; good for short-term data storage.	Very mature products	Leader in the database market; can handle fast updating and querying needs	Able to handle complex data	Configured to answer business intelligence questions quickly	Designed for huge, varied data sets
Major weaknesses	Redundant data; data must be updated using programs.	Not able to store data as efficiently; limited future	Cannot handle complex data	Limited market acceptance; skills are hard to find.	Highly specialized user skills are hard to find	New in the market, highly specialized user skills are hard to find
Data types supported	Simple	Not recommended for new systems	Simple	Complex (e.g., video, audio, images)	Aggregated	Mixed data sets with structured and unstructured components
Application system types supported	Transaction processing	Not recommended for new systems	Transaction processing and decision making	Transaction processing	Business intelligence	Business intelligence; finding patterns and relationships in mixed data
Existing data formats	Organization dependent	Organization dependent	Organization dependent	Organization dependent	Organization dependent	Organization dependent
Future needs	Limited future prospects	Poor future prospects	Good future prospects	Uncertain future prospects	Uncertain future prospects	New, uncertain future prospects

20/12/2014

Bộ môn CNTT - Khoa CNTT - Trường ĐHQG Hà Nội

275

## 4.1. Thiết kế các hệ thống con

- Quy trình thiết kế nhằm xác định các hệ thống con cấu tạo lên hệ thống đề xuất và framework giúp điều khiển các hệ thống con và giao tiếp giữa chúng được gọi là quy trình thiết kế kiến trúc. Kết quả của quy trình thiết kế này là bản đặc tả về kiến trúc phần mềm.
- Thiết kế kiến trúc là pha sớm nhất trong quy trình thiết kế hệ thống. Thiết kế kiến trúc thường được thực hiện song song với một số hành động đặc tả. Nó bao gồm có việc phát hiện các thành phần chính của hệ thống và giao tiếp giữa chúng.

20/12/2014

Bộ môn CNTT - Khoa CNTT - Trường ĐHQG Hà Nội

276



## 4.1. Thiết kế các hệ thống con

- Nếu chúng ta có được bản thiết kế kiến trúc rõ ràng thì ta sẽ thấy được các ưu điểm của nó trong những hoạt động sau:
  - Giao tiếp giữa các stakeholder:
    - Kiến trúc hệ thống thường được sử dụng làm tâm điểm của các buổi thảo luận giữa các stakeholder.
  - Phân tích hệ thống:
    - Tức là phân tích để xác định liệu hệ thống có thoả mãn các yêu cầu phi chức năng của nó hay không.
  - Tái sử dụng với quy mô lớn:
    - Kiến trúc có thể được tái sử dụng trong nhiều hệ thống.



## 4.1. Thiết kế các hệ thống con

- Các đặc điểm của kiến trúc hệ thống:
  - Hiệu năng: hạn chế các thao tác phức tạp và tối thiểu hoá giao tiếp.
  - Bảo mật: sử dụng kiến trúc phân lớp với nhiều kiểm soát chặt chẽ ở các lớp sâu hơn.
  - An toàn.
  - Sẵn dùng.
  - Có khả năng bảo trì.




## 4.1. Thiết kế các hệ thống con

- Các mô hình kiến trúc cơ bản:
  - Mô hình cấu trúc tĩnh: mô tả các thành phần hệ thống chính.
  - Mô hình quy trình động: biểu diễn quy trình cấu trúc của hệ thống.
  - Mô hình giao diện: định nghĩa tập hợp các giao diện của hệ thống con
  - Mô hình quan hệ: biểu diễn quan hệ giữa các hệ thống con.
  - Mô hình phân tán: biểu diễn cách cài đặt các hệ thống con trên máy tính.



## 4.2. Thiết kế giao diện và thiết kế lớp


- Nguyên tắc quan trọng khi xây dựng một hệ thống phần mềm, đó là:
  - Người sử dụng không quan tâm đến cấu trúc bên trong của hệ thống, đơn giản hay phức tạp; cái mà họ có thể đánh giá được và cảm nhận được chính là giao diện tương tác giữa hệ thống và người sử dụng.
  - Nếu người sử dụng cảm thấy giao diện không thích hợp, khó sử dụng thì rất có thể họ sẽ không sử dụng cả hệ thống; cho dù hệ thống đó có đáp ứng tất cả các chức năng nghiệp vụ mà họ muốn. Và như vậy, dự án của chúng ta sẽ thất bại.
- Vì tầm quan trọng của giao diện người dùng, nên chúng ta có cả một chương để nói về chúng. Trong chương này, chúng ta sẽ nghiên cứu những vấn đề sau:
  - Các yếu tố liên quan đến giao diện người dùng
  - Quy trình xây dựng giao diện người dùng



## 4.2.1. Thiết kế giao diện người dùng

- Tổng quan:
  - Giao diện người dùng cần phải được thiết kế sao cho phù hợp với kỹ năng, kinh nghiệm và sự trông đợi của người sử dụng nó.
  - Người sử dụng hệ thống thường đánh giá hệ thống thông qua giao diện hơn là chức năng của nó. Giao diện của hệ thống nghèo nàn có thể khiến người sử dụng tạo ra các lỗi hết sức nghiêm trọng. Đó là lý do tại sao nhiều hệ thống phần mềm không bao giờ được sử dụng.
- Mục tiêu:
  - Hiểu được sự ảnh hưởng của người sử dụng tới giao diện
  - Một số nguyên tắc khi thiết kế giao diện người dùng
  - Phân loại các khả năng tương tác giữa người và máy để thiết kế giao diện cho phù hợp
  - Biết cách biểu diễn thông tin cho phù hợp với người sử dụng


Đà Nẵng CNTT - Khoa CNTT - Học Viện Kỹ Thuật Điện Tử



## 4.2.1. Thiết kế giao diện người dùng

- Tác nhân con người trong thiết kế giao diện:
  - Một nhân tố quan trọng ảnh hưởng tới quá trình thiết kế giao diện đó chính là người sử dụng hệ thống. Do đó, chúng ta phải tìm hiểu một số đặc điểm của người sử dụng có liên quan đến giao diện hệ thống:
    - Khả năng nhớ tức thời của con người bị hạn chế: con người chỉ có thể nhớ ngay khoảng 7 loại thông tin. Nếu ta biểu diễn nhiều hơn 7 loại, thì có thể khiến người sử dụng không nhớ hết và gây ra các lỗi.
    - Người sử dụng có thể gây ra lỗi: khi người sử dụng gây ra lỗi khiến hệ thống sẽ hoạt động sai, những thông báo không thích hợp có thể làm tăng áp lực lên người sử dụng và do đó, càng xảy ra nhiều lỗi hơn.
    - Người sử dụng là khác nhau: con người có những khả năng khác nhau. Những người thiết kế không nên chỉ thiết kế giao diện phù hợp với những khả năng của chính họ.
    - Người sử dụng thích các loại tương tác khác nhau: một số người thích hình ảnh, văn bản, âm thanh ...


Đà Nẵng CNTT - Khoa CNTT - Học Viện Kỹ Thuật Điện Tử



## Giao diện người dùng (tt2)

- Các nguyên tắc thiết kế giao diện:
  - Thiết kế giao diện phải phụ thuộc vào yêu cầu, kinh nghiệm và khả năng của người sử dụng hệ thống.
  - Người thiết kế cũng nên quan tâm đến những giới hạn vật lý và tinh thần của con người và nên nhận ra rằng con người luôn có thể gây ra lỗi.
  - Không phải tất cả các nguyên tắc thiết kế giao diện đều có thể được áp dụng cho tất cả các giao diện.


Đà Nẵng CNTT - Khoa CNTT - Học Viện Kỹ Thuật Điện Tử



## Giao diện người dùng (tt3)

- Các nguyên tắc thiết kế giao diện (tt1):
  - Sau đây là các nguyên tắc thiết kế giao diện:
    - Sự quen thuộc của người sử dụng: giao diện phải được xây dựng dựa trên các thuật ngữ và các khái niệm mà người sử dụng có thể hiểu được hơn là những khái niệm liên quan đến máy tính.
      - Ví dụ: hệ thống văn phòng nên sử dụng các khái niệm như thư, tài liệu, cặp giấy ... mà không nên sử dụng những khái niệm như thư mục, danh mục ...
    - Thống nhất: hệ thống nên hiển thị ở mức thống nhất thích hợp.
      - Ví dụ: các câu lệnh và menu nên có cùng định dạng ...
    - Tối thiểu hoá sự bất ngờ: nếu một yêu cầu được xử lý theo cách đã biết trước thì người sử dụng có thể dự đoán các thao tác của những yêu cầu tương tự.


Đà Nẵng CNTT - Khoa CNTT - Học Viện Kỹ Thuật Điện Tử



## Giao diện người dùng (tt4)

- Các nguyên tắc thiết kế giao diện (tt2):
  - Sau đây là các nguyên tắc thiết kế giao diện (tt1):
    - Khả năng phục hồi: hệ thống nên cung cấp một số khả năng phục hồi từ lỗi của người sử dụng và cho phép người sử dụng khôi phục lại từ chỗ bị lỗi. Khả năng này bao gồm cho phép làm lại, hồi lại những hành động như xoá, huỷ ...
    - Hướng dẫn người sử dụng: như hệ thống trợ giúp, hướng dẫn trực tuyến ...
    - Tính đa dạng: hỗ trợ nhiều loại tương tác cho nhiều loại người sử dụng khác nhau.
      - Ví dụ: nên hiển thị phông chữ lớn với những người cận thị.


285



## 4.2.1. Thiết kế giao diện người dùng

- Các nguyên tắc thiết kế giao diện (tt3):
  - Sau đây là các nguyên tắc thiết kế giao diện (tt2):
    - Tương tác giữa người sử dụng và hệ thống được chia thành 5 loại sau:
      - Vận hành trực tiếp
      - Lựa chọn menu
      - Điền vào biểu mẫu (Form)
      - Ngôn ngữ ra lệnh
      - Ngôn ngữ tự nhiên


286



## 4.2.1. Thiết kế giao diện người dùng

- Biểu diễn thông tin
  - Biểu diễn thông tin có liên quan tới việc hiển thị các thông tin trong hệ thống tới người sử dụng. Thông tin có thể được biểu diễn một cách trực tiếp hoặc có thể được chuyển thành nhiều dạng hiển thị khác như: dạng đồ họa, âm thanh ...
  - Thông tin cần biểu diễn được chia thành hai loại:
    - Thông tin tĩnh: được khởi tạo ở đầu của mỗi phiên. Nó không thay đổi trong suốt phiên đó và có thể là ở dạng số hoặc dạng văn bản.
    - Thông tin động: thay đổi trong cả phiên sử dụng và sự thay đổi này phải được người sử dụng quan sát.
    - Biểu diễn digital hay analogue?

287



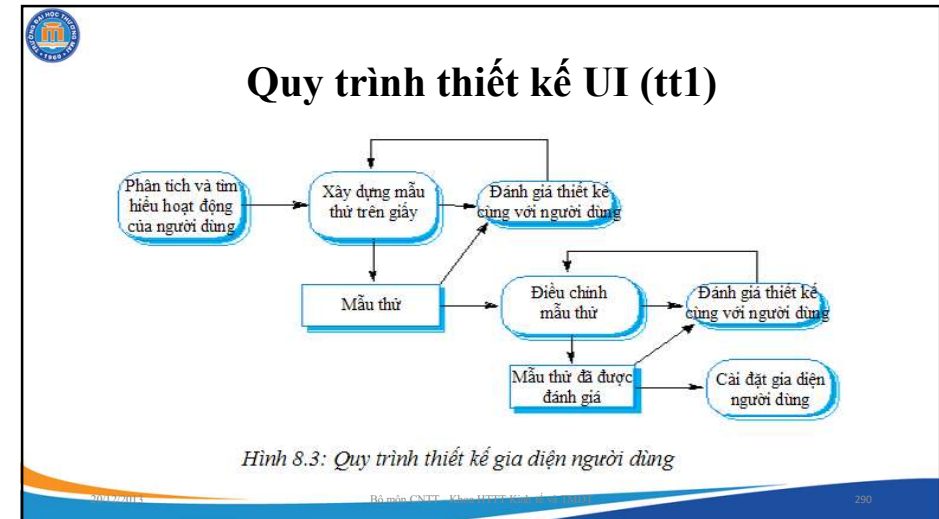
## 4.2.1. Thiết kế giao diện người dùng

- Biểu diễn thông tin (tt1)
  - Các nhân tố ảnh hưởng tới việc hiển thị thông tin:
    - Người sử dụng thích hiển thị một phần thông tin hay quan hệ dữ liệu?
    - Giá trị của thông tin thay đổi nhanh như thế nào? Sự thay đổi đó có cần phải thể hiện ngay lập tức hay không?
    - Người sử dụng có phải thực hiện các hành động để đáp ứng với sự thay đổi không?
    - Có phải là giao diện vận hành trực tiếp không?
    - Thông tin ở dạng văn bản hay dạng số? Các giá trị quan hệ có quan trọng không?
    - Biểu diễn digital hay analogue?

288

**Quy trình thiết kế UI**

- Giới thiệu:
  - Thiết kế giao diện người dùng là một quy trình lặp lại bao gồm sự cộng tác giữa người sử dụng và người thiết kế.
  - Trong quy trình này gồm 3 hoạt động cơ bản:
    - Phân tích người sử dụng: tìm hiểu những gì người sử dụng sẽ làm với hệ thống.
    - Lập mẫu thử hệ thống: xây dựng một tập các mẫu thử để thử nghiệm
    - Đánh giá giao diện: thử nghiệm các mẫu thử cùng với người sử dụng.



**Quy trình thiết kế UI (tt2)**

- Mục tiêu:
  - Hiểu được quy trình thiết kế giao diện người dùng
  - Nắm được chi tiết từng hoạt động trong quy trình thiết kế giao diện người dùng
  - Với mỗi hoạt động, chúng ta có rất nhiều cách để thực hiện. Do đó, phải có khả năng lựa chọn phương pháp nào là thích hợp nhất cho từng hoàn cảnh cụ thể.

**Quy trình thiết kế UI (tt3)**

- Phân tích người sử dụng
  - Nếu ta không hiểu rõ những gì người sử dụng muốn làm với hệ thống, thì ta sẽ không thể thiết kế được một giao diện hiệu quả.
  - Phân tích người sử dụng phải được mô tả theo những thuật ngữ để người sử dụng và những người thiết kế khác có thể hiểu được.
  - Các **ngữ cảnh** mà ta mô tả thao tác ở trong đó là một trong những cách mô tả phân tích người dùng. Ta có thể lấy được rất nhiều yêu cầu của người sử dụng từ đó.



### 4.3. Thiết kế việc lưu trữ các dữ liệu

- Có 2 dạng lưu trữ chính:
  - Lưu trữ dưới dạng Tập tin
  - Lưu trữ dưới dạng CSDL
- Lưu trữ dưới dạng tập tin
  - Thường chỉ thích hợp với một số PM
    - Chú trọng rất nhiều vào các xử lý và hình thức giao diện.
    - Thường các thông tin được tiếp nhận và xử lý ngay.
    - Ví dụ: các game nhỏ, ...
- Lưu trữ dưới dạng CSDL rất thông dụng.

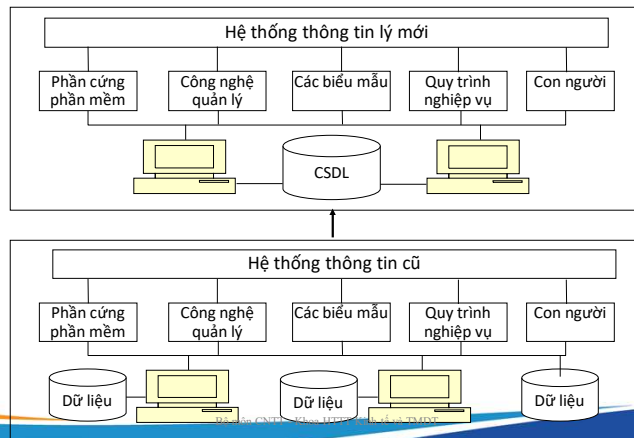


### 4.3. Thiết kế việc lưu trữ các dữ liệu

- Các bước thực hiện:
  - Bước 1: Chọn 1 YC và xác định sơ đồ logic cho YC đó.
  - Bước 2: Bổ sung thêm 1 YC và xem lại sơ đồ logic
    - Nếu sơ đồ logic vẫn đáp ứng được thì tiếp tục bước 3 (không thêm gì cả).
    - Nếu sơ đồ logic không đáp ứng được thì bổ sung vào
      - Ưu tiên 1: thuộc tính mới
      - Ưu tiên 2: thành phần mới cùng với các thuộc tính và liên kết tương ứng.
  - Bước 3: Quay lại bước 2 cho đến khi đã xem xét đầy đủ YC.
  - Bước 4: Tìm và liệt kê các RBTN, RBNC



### 4.4. Mô hình hóa cài đặt hệ thống



### Bài tập

- Câu 1:
  - Nếu một số trường hợp không thể cung cấp giao diện người dùng một cách nhất quán
- Câu 2:
  - Hãy đề xuất một vài giao diện người dùng thích hợp cho hệ thống bán sách trực tuyến, dành cho những khách hàng là người khuyết tật.
- Câu 3:
  - Cho biết những ưu điểm của việc hiển thị thông tin một cách trực quan. Lấy ví dụ để minh họa cho tính vượt trội của việc hiển thị thông tin trực quan so với các thông tin thống kê như văn bản, số liệu.



## Bài tập (tt1)

- Câu 4:
  - Nêu những chú ý khi sử dụng màu để thiết kế giao diện
- Câu 5:
  - Viết kịch bản tương tác của hệ thống dịch vụ đặt mua vé trước và thực hiện thanh toán bằng thẻ tín dụng.