**L1 Introduction**

**Functional Modeling**

Use Case Diagram: A graphic overview of the main functionality of a system/A picture showing system behavior along with thekey actors that interact with the system.

Use Case Description: Captures detailed interaction between user and the system

Activity Diagram: Describes workflow of the activities

**Structural Modeling**

Class Diagram: Describes the structure of the data that supports the biz processes

**Use Case:**

A depiction of a system’s behavior or functionality under various conditions as the system responds to requests from users.

**Actor**

An external entity that reacts with a system/An actor is a type or class of users.

**Extend Relationship**

An association between two use cases where one odds new behaviors or actions to the other.

**Include Relationship**

An association between two use cases where one use case uses the functionality contained in the other.

**Use Case Description**

Use Case Name/ Primary Actor/ Pre-conditions/ Post-condition/ Trigger/ Main Success Scenario/ Extension: The set of behaviors or functions in the use case that follow exceptions to the main success scenario.

**L2 Interaction Diagram**

**Behavior Model (Three Models in Info Sys: Functional Models/ Structural M/ Behavior M)**

1. Interaction Diagram
2. State Machine

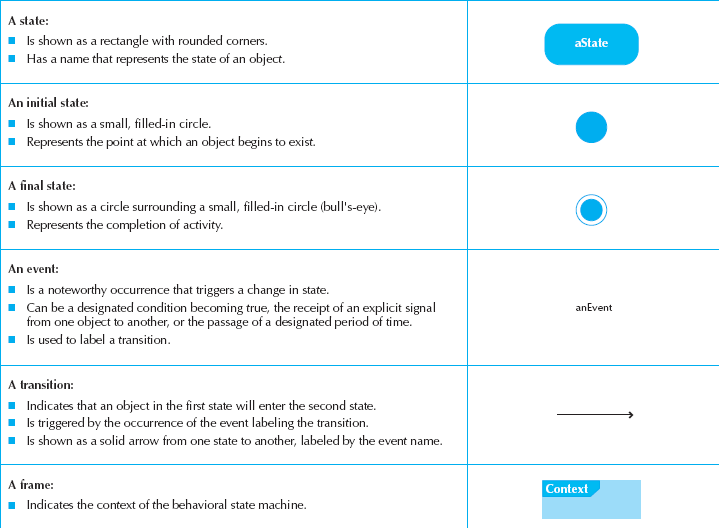
**Sequence Diagram:** Illustrate the objects that participate in a use case and the messages that pass between them over time

Elements: Actors &Objects/ Lifeline/ Execution Occurrence/ Message/ Guard Condition/ Frame

**Communication Diagram**

**Behavioral State Machine**

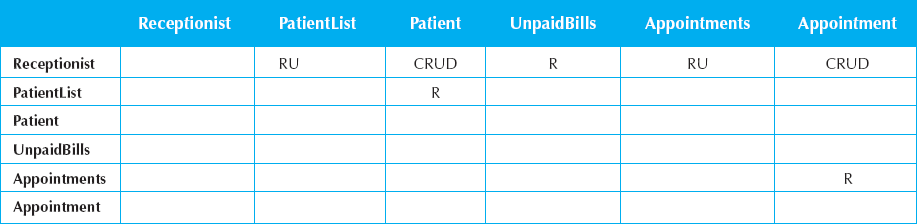
Elements: State of an object/ Event/ Transition/ Guard Condition



An Event is a notelorthy occurrence that triggers a change in state and is used to label a transition.

A Transition indicated that an object in the first state will enter the second state and is triggered by the occurrence of the event labeling the transition.

**CRUD stands for Create, Read, Update and Delete:** A CRUD matrix is adapted to OO systems development



**L3 System Design**

The purpose of the analysis phase is to figure out what the biz needs, mainly about functional requirements.

The purpose of design phase is to figure out how to provide it.

**Evolve Analysis Model into Design Model**

Verification& Validation through Walkthrough (a peer view of a product): To ensure that different models are consistent.

1. **Factoring (因式分解):** Separating out a module into a stand-alone module which can be a new class or a new method.

**Process1: Abstraction** (Abstract Class: A class that has no direct instance, but whose descendents may have direct instances)

* **Generalization** (a kind of)
  + Classes with similar attributes and methods => super class
  + Related classes with different attributes and methods => sub class
* **Aggregation** (logical part of)
* **Composition** (physical part of – Die togather)

Process2: Refinement

1. **Partitions (划分) (denormalization)**

Create ‘sub-systems’ or larger units

Group elements that collaborate with each other \*simplified modules lead to easy understanding \*allocation of work

1. **Layers (分层)**

Start considering non-functional requirement in the system environment such as Data Mgt, UI, Physical Architecture.

**Different Software Layers:** Foundation – Problem Domain – Data MGT – Human- Computer Interface (HCI) – Physical Architecture

**Packages Diagram**

It logically group UML elements to simplify UML diagrams/ Provide structure and less complex views of the new system.

**L4 Design Strategy**

**System Acquisition Strategy:**

1. Custom Development

Ad: Complete Control/ Highly meeting specialized requirements/ Flexible/ Easier to change

Disad: Time&money-consuming/ Backlogs/ departure of developers/ Insufficient technical expertise &experience

1. Package Software

Ad: Inexpensive/ Thoroughly tested (Best Practice)/ Sound component/ Allow for customization

Disad: Fixed functionalities/ Require changes in company’s biz processes/ Require workaround/ interface

1. Outsourcing

Ad: Technical expertise/ extend existing resources/ accountability

Disad: Risk of disclosing confidential info/ Lose control/ Language &Cultural barrier

Caution: fully understand needs and requirements/ carefully choose vendor/ payment style/ adequate communication/ experienced manager

Type of outsourcing: Complete, Nothing, Hybrid

**L5 Class Diagram and Method Design**

Classes from other layers are highly dependent on the **Problem Domain layer**.

**Class:** A logical grouping of objects that have the same or similar attributes, relationships, and behaviors.

**Object:** An entity that has a well-defined role in the application domain, and it has state, behavior, and identity characteristics.

**Operation / Method:** A function or a service that is provided by all the instances of a class.

* Constructor Operation
* Query Operation
* Update Operation

**Attribute:** Describes the object. +/ #/ -

**Relationship:**

* Generalization /Inheritance：See above.
* Aggregation:　See above.
* Association: A named relationship between or among object classes.
* Multiplicity: 1/ 0..\*/ 1..\*
* Underline Class attributes and operations!

**Class &Method Design**

* Encapsulation: Combination of data and process into an entity.
* Information Hiding: Only the information required to use a software module is publish to the user
* Polymorphism: A message can be interpreted differently by different classes of objects. (Ensure all methods with the same name implement that same **generic operation**! )
* Dynamic Binding: Delays typing or choosing a method for an object until run-time.

Design Criteria

* Coupling (the less the better)
* Interaction Coupling:
* Inheritance Coupling:
* Cohesion (the more the better)

**L6 Database Design**

**Relational Database (RDBMS)**

 1:1..N  1:0..N

**Normalization (标准化):** The process of converting complex data structures into simple, stable data structures.

* First Normal Form (1NF)

No multivalued fields.

* 2NF

Each non-primary key attribute is identified by the whole primary keys. (maybe many PKs leads to a attribute)

* 3NF

Non-primary key attributes do not depend on each other. i.e.no transitive dependency

**Foreign Key:** An attribute that appears as a non primary key attribute in one relation and as a primary key attribute in another relation.

**Optimizing Data Access Speed**

* **De-normalization:** The process of splitting or combining normalized relations into physical tables based on affinity of use of rows and fields. (When splitted into several tables, each of them should include the primary key)
* **Clustering**
* **Indexing:** A table used to determine the location of rows in a file that satisfy some condition.

**L7 User Interface Design**

**Principles for UI Design**

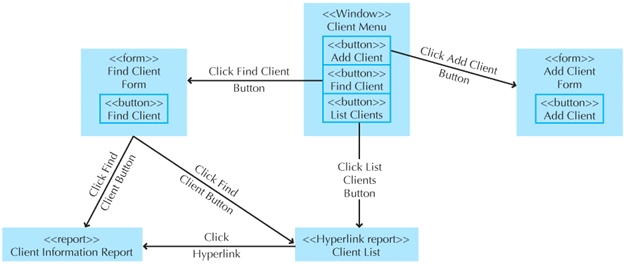
* **Layout**
* **Content Awareness**
* All interfaces should have titles
* Menus should show where we are and how to get there
* Fields labels
* Forms or reports should have preparation date
* **Aesthetics**
* Need to be functional and inviting to use
* Avoid squeezing in too much information
* Design text carefully and use colors &patterns with caution
* **User Experience**
* Put more emphasis on ease (frequent users’ habits) of use rather than ease of learning (novice)
* **Consistency**
* Enable user to predict what will happen to reduce learning curve.
* User standard interface objects: list boxes, put down menus
* Watch out for terminology
* **Minimize user effort**
* Three-click rule

**UI Design Process**

User Scenario Development/ Interface Structure Design/ Interface Standards Design/ Interface Design Prototype/ Interface Evaluation

* Interface Structure Design

Develop Window Navigation Diagram (WND)



* A **box** corresponds to a user interface component, such as a window, form, button, or report
* **Transitions** are modeled as a single-headed (return not required) or double-headed arrow (return required)
* **Stereotype** modeled as a text item enclosed in guillemets <<>>. It represents the type of user interface component
* Interface Evaluation (Approaches)

1. Heuristic

Compare the interface to a set of heuristics or principles

1. Walkthrough

Project team presents and walks through the prototype to a group of presenting users

1. Interactive

Users actually work with the prototype in a person session with project team

1. Formal Usability Testing

Users actually work with the prototype in a person session in a special lab with video and camera and user is giving a set of certain tasks

**Navigation Design**

**Basic Principles**

* Prevent Mistakes
* Simplify Recovery from mistakes
* Use Consistent Grammar Order

Types of Navigation Controls

* Languages (command/ natural)
* Menus
* Direct Manipulation

L8 Input Design

Basic Principles:

* Online Processing VS. Batch Processing
* Online Processing: Immediately records the transaction in the appropriate database/ Used when Real-time is required
* Batch Processing: Collects data over time and enter them into the system at one time in a batch/ Used when Real-time is not required
* Capture data at the source

Source Data Automation: Barcode Reader/ Magnetic Stripe Reader/ Smart Card/ RFID

* Minimize Keystrokes

Use list selection and default values

**Input Validation:** **Completeness Check/ Format Check** (fields are numeric or coded data)/ **Range Check** (all numeric data)/ **Check Digit Check/ Consistency Check** (When data are related)/ **Database Check**

**L9 Output Design**

**Basic Principles**

* **Understand Report Usage**

Usage/ Frequency/ Distribution/ Real-time or Batch/ Hard or Softcopy/ Sort Sequence/ Content/ Layout

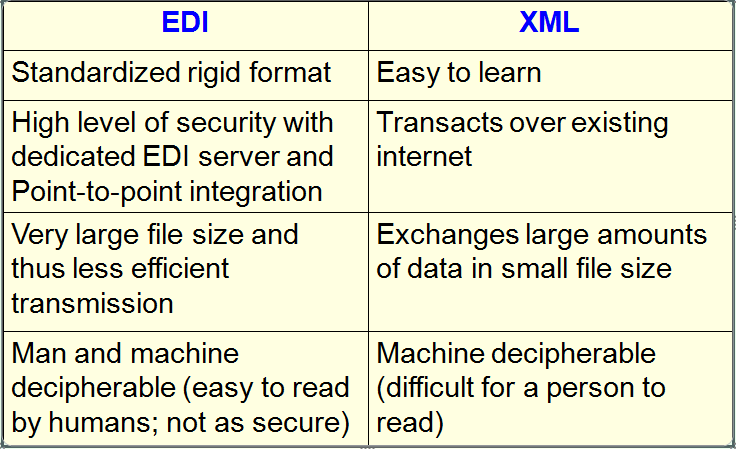
* **Manage Information Load**

Info needed, not available

* **Minimize Bias**

**Electronic Data Interchange (EDI):** The transfer of structured data, by agreed message standards, from one computer system to another (can be from one trading partner to another trading partner) without human intervention but by electronic means.

**Extensible Markup Language:** a set of rules for encoding documents electronically



**Physical Architecture Layer Design**

The plan for how the system ill be distributed across the computers and what hard/software will be used.

The goal is to what parts of the application software will be assigned to what hardware.

**Components:**

Software: Data Storage/ Data Access Logic/ Application Logic/ Presentation Logic

Hardware: Client Computers/ Servers/ Connecting Network

**Three Principal Architecture**

* Server-Based A
* Client-Based A
* Client-Server A (most common): Balance the processing between the client and the server
* Client is responsible for the presentation logic
* Server is responsible for the data access logic and data storage
* The application logic may be on either the client or the server or be split between both.
* AD: Scalable: Easy to increase or decrease the processing capabilities of the server

Failover: When one server is down, others can still function

* DISAD: Complicated/ update on all clients

**Non-functional Requirements** (which have a great impact on choosing architectures)

\*Functional Requirements are under Problem Domain area

* Operational Requirements: Technical Environment/ System Integration/ Portability/ Maintainability
* Performance: Speed/ Capacity/ Availability and Reliability
* Security
* Cultural and Political

**L11 Designing Test and Documentation**

**Testing Planning**

**Categories of Testing**:

* **Unit:** Test each unit to ensure that it performance its function/ Focus on single class
* Black Box Testing: Examines externally visible behaviors of a class/ Driven by method contracts
* White Box Testing: Examines the internal of a class/ Driven by method specifications
* **Integration:** Assess whether a set of class that must work together do so without errors
* **System:** Conducted by system analysts to ensure that all modules work together without errors
* **Acceptance:** Done by users with support from the project team to confirm that the system is complete, meet the biz need, and is acceptable to the users.
* Alpha Testing: Repeats tests with made-up data by users
* Bate Testing: Users test the system with real data

**Developing Documentation** (must be done through the system development)

* System Documentation: for programmers and analysts
* User Documentation: for users
  + Reference Documents: Tell users how to perform specific tasks
  + Procedure Manuals: Describe how to perform business tasks
  + Tutorial: Teach users how to use major components of a system

**L13 Conversion Strategy & Change**

**Change MGT:**

* Unfreezing: Loosening up people’s habits and norms
* Moving: System transition
* Refreezing: Institutionalize and make efficient new ways of doing things

**Migration Planning:**

* Installing hard/software
* Data conversion
* User training
* Motivating employees

**Conversion Style**

* Direct conversion
* Parallel conversion

**Conversion Location**

* Pilot conversion
* Phased conversion
* Simultaneous conversion

