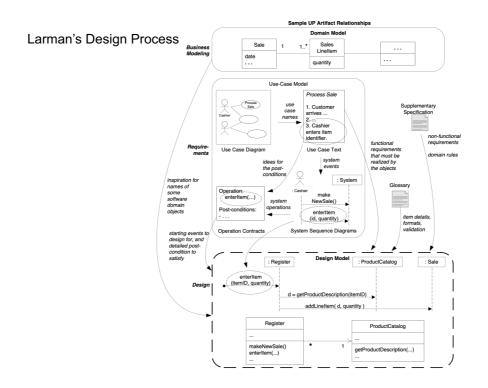
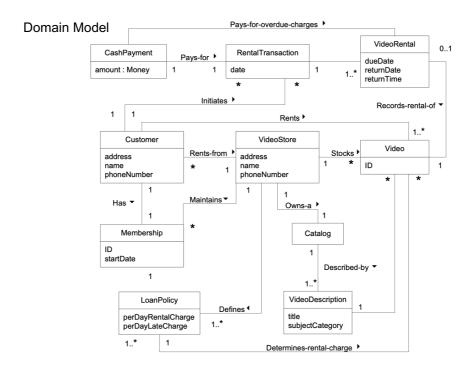
SOEN 343 Software Design

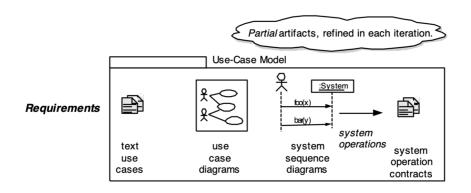
Section H Fall 2006 Dr Greg Butler

http://www.cs.concordia.ca/~gregb/home/soen343h-f06.html





Use Case Model



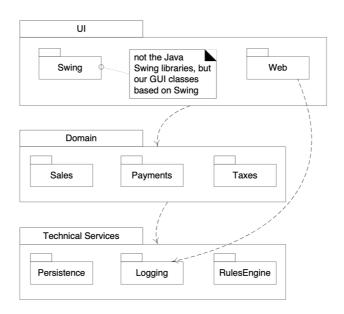
GUI windows reports speech interface HTML, XML, XSLT, JSP, Javascript, ... handles presentation layer requests workflow session state window/page transitions consolidation/transformation of disparate data for presentation handles application layer requests implementation of domain rules domain services (POS, Inventory) - services may be used by just one application, but there is also the possibility of multi-application services very general low-level business services used in many business domains CurrencyConverter (AKA Low-level Business Services) (relatively) high-level technical services and frameworks Persistence, Security

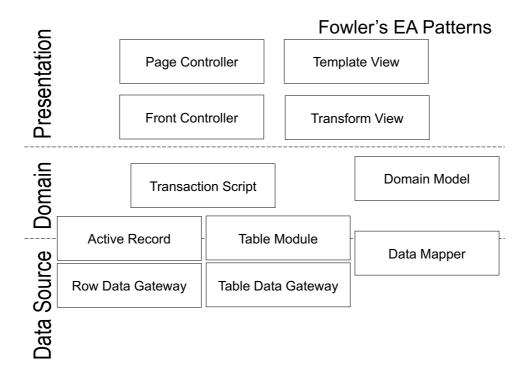
Foundation
(AKA Core Services, Base Services,
Low-level Technical Services/Infrastructure)
width implies range of applicability

Typical Software Architecture Layers

low-level technical services, utilities, and frameworks data structures, threads, math, file, DB, and network I/O

Typical Software Architecture Layers (Simplified)





What is Design?

Developing a blueprint (plan) for a mechanism that performs the required task,

- ... taking into account all the constraints, &
- ... making trade-offs between constraints when they are in conflict.

What is OO Analysis and Design

- Object-Oriented Analysis
- Object-Oriented Design
- Important domain concepts or objects?
- Design of software objects

- Vocabulary?
- ResponsibilitiesCollaborations
- Visualized in the Domain Model
- Design patterns
- Visualized in the Design Model

Important Concepts

Model

- · Abstraction hiding (unimportant) details
- Eg, cover of Larman's book

GRASP Principle

for assigning responsibility

Design pattern

- Solution to design problem in context
- · Eg, Command pattern

Responsibility-Driven Design (RDD)

- Detailed object design is usually done from the point of view of the *metaphor* of:
 - Objects have responsibilities
 - Objects collaborate
- Responsibilities are an abstraction.
 - The responsibility for persistence.
 - Large-grained responsibility.
 - The responsibility for the sales tax calculation.
 - More fine-grained responsibility.