Announcements

- Quiz – started Now in myCourses (closed book!)
- Teams formed - sit with your group from now on – when in classroom ;^)
- Project Deliverables

Agenda

Lecture

Team Logistics

- Come up with classwide discussion items

Homework

Organizing Domain Logic

SWEN-343

Software Engineering
Rochester Institute
of Technology

Learning Objectives

Organize ERP domain logic

For your project

Keep these activities in mind

(DETOUR) Data Models are Paramount for ERP

Review of Data Modeling

What do you remember?

Data Models

- A data model is a specification of the information content of a system
- A data model is built on a set of concepts that can be used to describe the structure of a database
    - Available data types, relationships, and constraints
    - Basic **operations** for create, read, update, delete (CRUD)
    - User-defined operations and **behavior** and **constraints**
- Data definition languages (DDLs) describe structure and content
- Data manipulation languages (DMLs) describe the create, read, update, and delete (CRUD) operations
    - Queries, procedures

Data Modeling (Development Phases)

Three phases of data model development

Data model **analysis**

<u>Conceptual data model</u>

Data model **design**

<u>Logical data model</u>

Data model **implementation**

<u>Physical data model</u>

Data Modeling (Development Phases)

Three phases of data model development (but do iterate!)

Data model **analysis**

Conceptual data model describes information in terms the users will understand

Entities (things, events, concepts), attributes (properties, characteristics) and relationships (associations)

Entity-Relationship diagrams, UML **class diagrams**, etc.

Problem analysis

Data model **design**

Logical data model describes information in a way that can be used to build a database

Relational models, object-oriented models, XML graph models, etc.

Solution design in the language of the database management system

Data model **implementation**

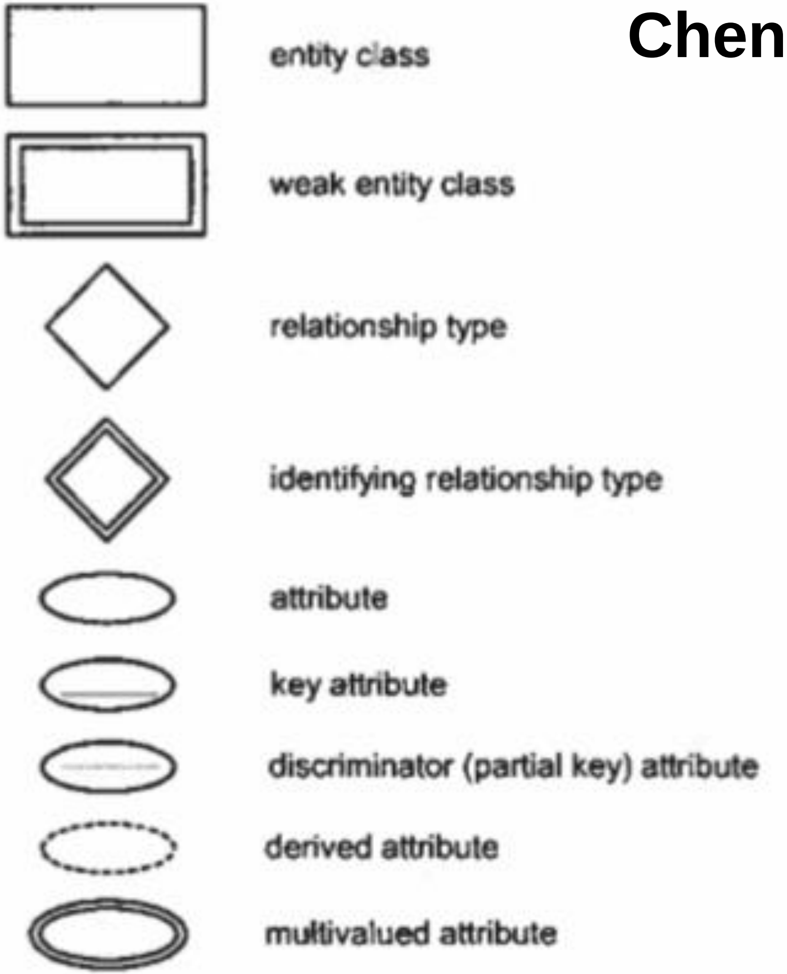
Physical data model describes information in terms of its representation in physical storage

File and record structure, data structure, access paths, indexes, etc.

Detailed design and implementation

Usually hidden (automatically managed by DBMS)

Chen-like notation



entity class

weak entity class

relationship type

identifying relationship type

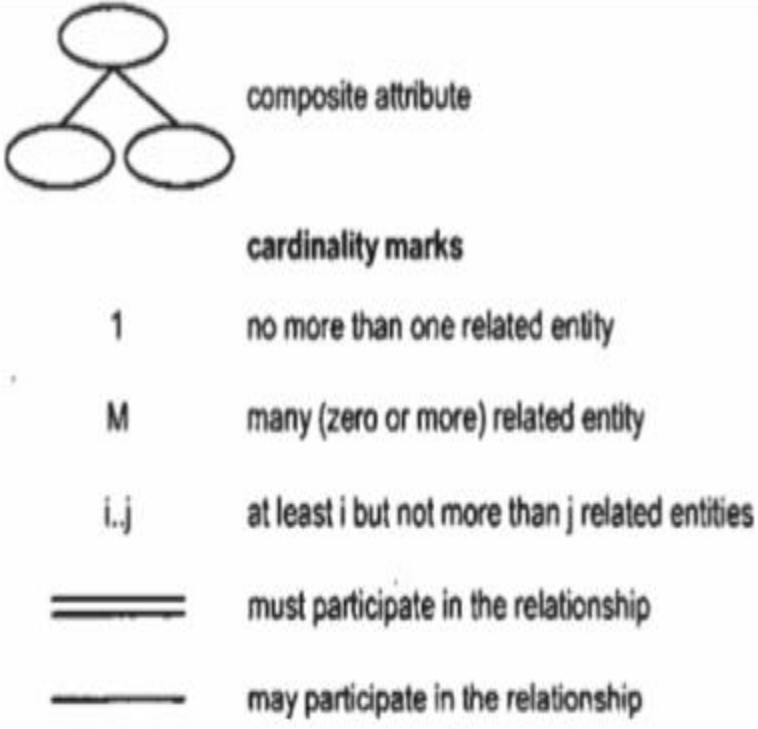
attribute

key attribute

discriminator (partial key) attribute

derived attribute

multivalued attribute



composite attribute

cardinality marks

1 no more than one related entity

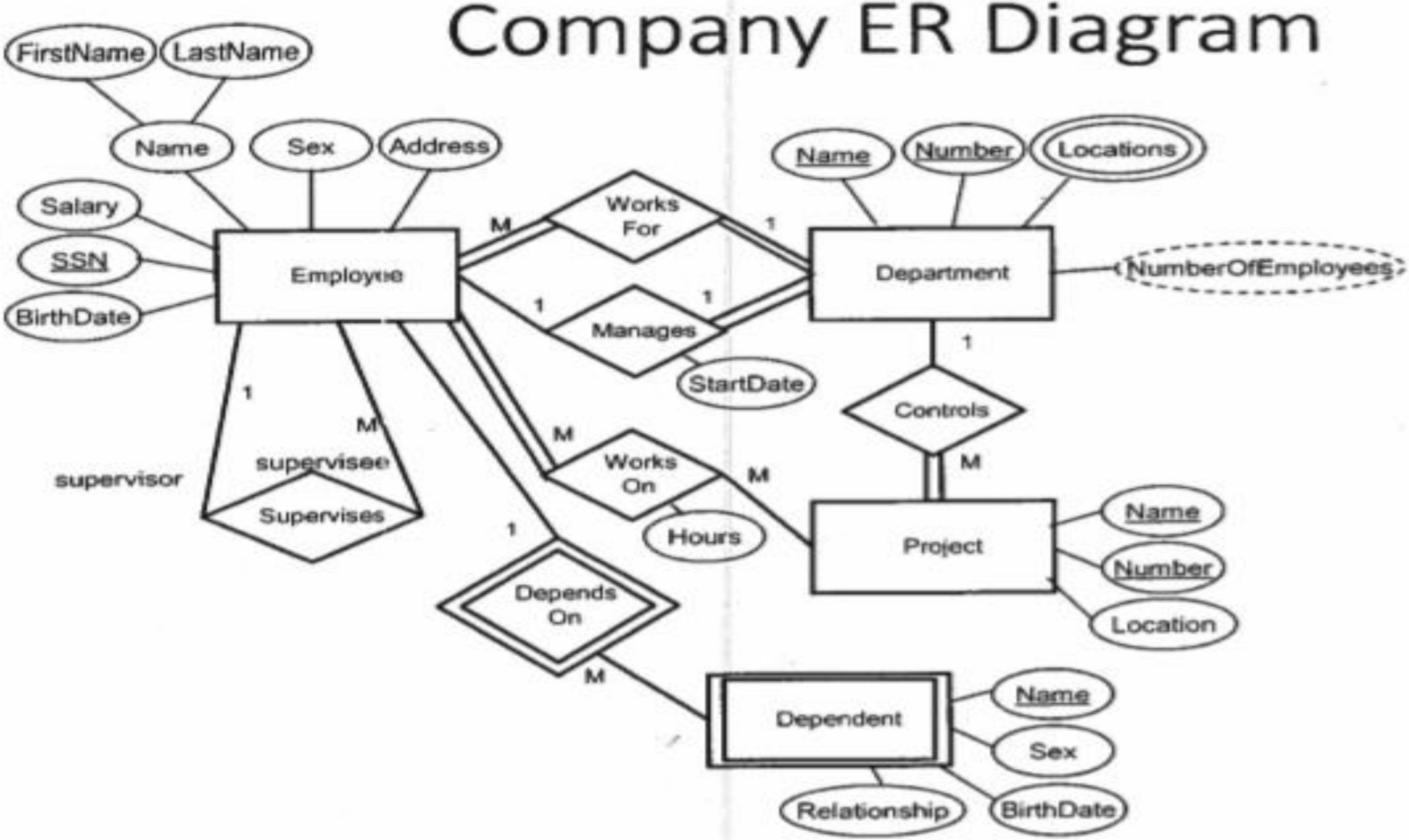
M many (zero or more) related entity

i..j at least i but not more than j related entities

must participate in the relationship

may participate in the relationship

Company ER Diagram



Logic Patterns

Describe functional algorithms or business logic that handle information exchange between database and user interface

Well organized Domain Logic is imperative for maintenance and scalability

→ VERY important for ERP

Domain Logic Introduction

Most business applications can be thought of as a series of transactions

Each interaction between client & server requires some logic

Can be simple or very complex

pulling data -> validations, complex calculations etc...

Domain Logic Patterns

Transaction Script

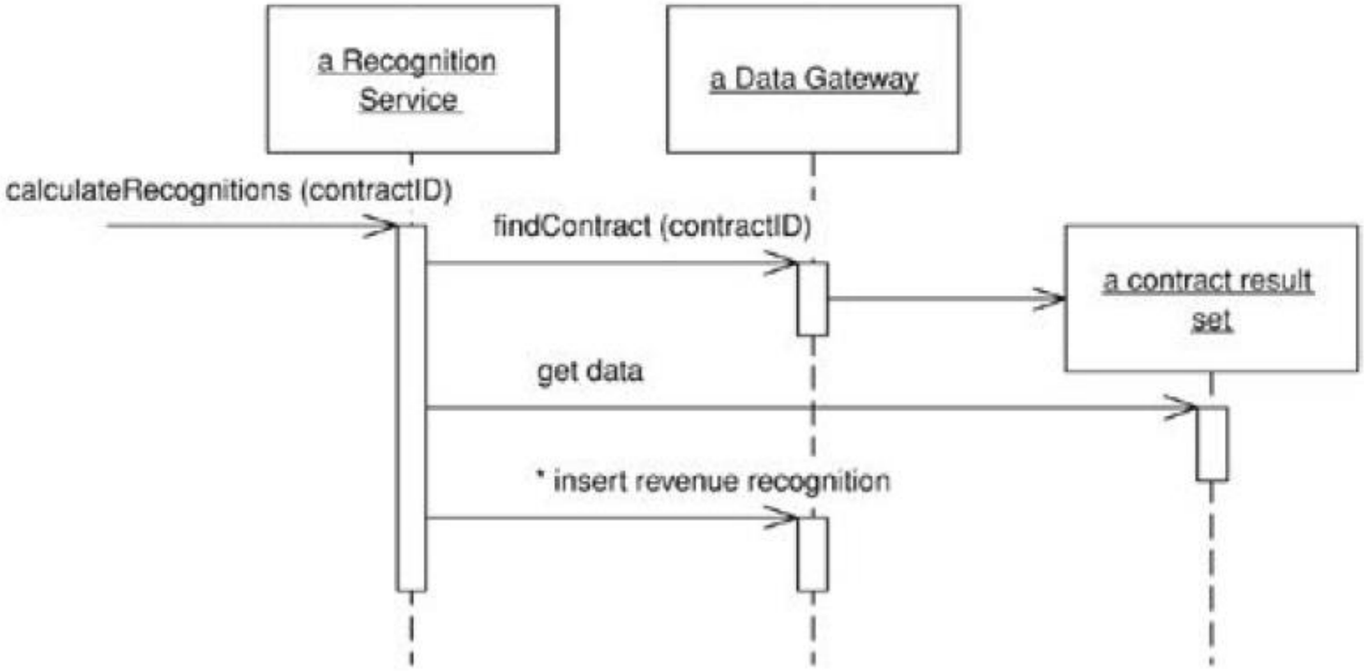
Domain Model

Table Module

Transaction Script

- Transactions carry out functionality
- Organizes logic using procedures
  - Each procedure handles a single request from presentation
  - Processes it with validations & calculations
- Single procedure for each action.
  - Open a connection & then close it
    - getAge()
    - updateAge()
    - getName()
    - ……..

Transaction Script


a Recognition Service
a Data Gateway
calculateRecognitions (contractID)
findContract (contractID)
a contract result set
get data
* insert revenue recognition

Transaction Benefits

Simple, easily understandable

Works well with simple data source layer

Obvious how to set transaction boundaries

Great for simple applications.

Transaction Drawbacks

?

Transaction Drawbacks

Duplicated code with several transactions doing the same/similar things

Reasonably sized applications will likely be tangled web, without a clear structure.

Domain Model

Objects…...

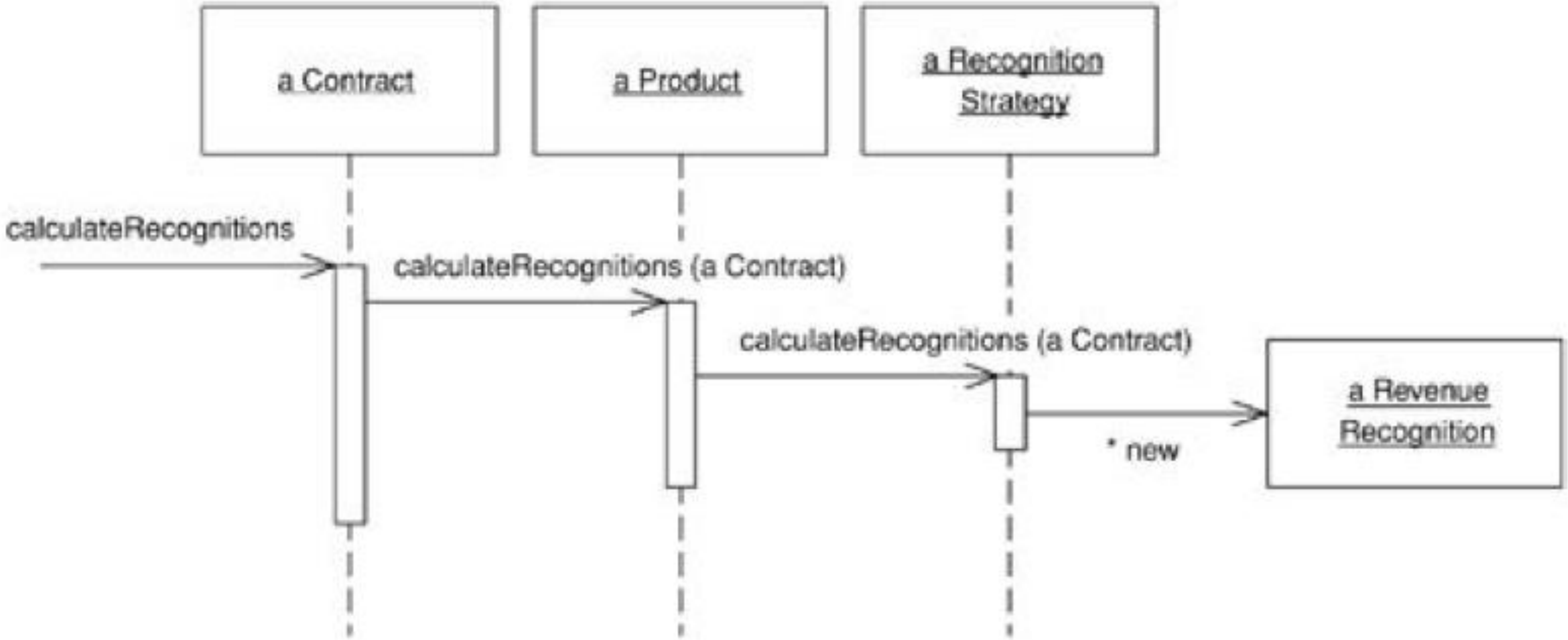
Build model of domain

- Organized around nouns

Contains logic for validations and calculations

Each object takes part in logic it is relevant to.

Domain Model



Domain Model Benefits

Instead of one routine handle all logic for user action, each object takes a part of the logic that is relevant to it.

Allow for complex logic to be handled in well organized manner.

Domain Model Drawbacks

- ***Relatively*** complex to use
  - Takes time to get used to
- Database mapping
  - Richer the domain model, the more complex the mapping

Domain Model

Interface with relational dbs can be tough

Example: If you have many orders, a domain model will have one order object per order

To overcome this problem, we use the TABLE MODULE

Table Module

Similar to domain as both have classes.

Domain = 1 instance for ***each*** occurrence.

- New Dog object for each dog

Table = 1 instance for ***all*** occurrences.

- 1 Dog table

Table Module

- Much like a middle ground between script and domain.
- Provides structure and easy to find duplication.
- Single instance that handles the business logic for all rows in a database table or view
  - One object handles all orders

Table Module Drawbacks

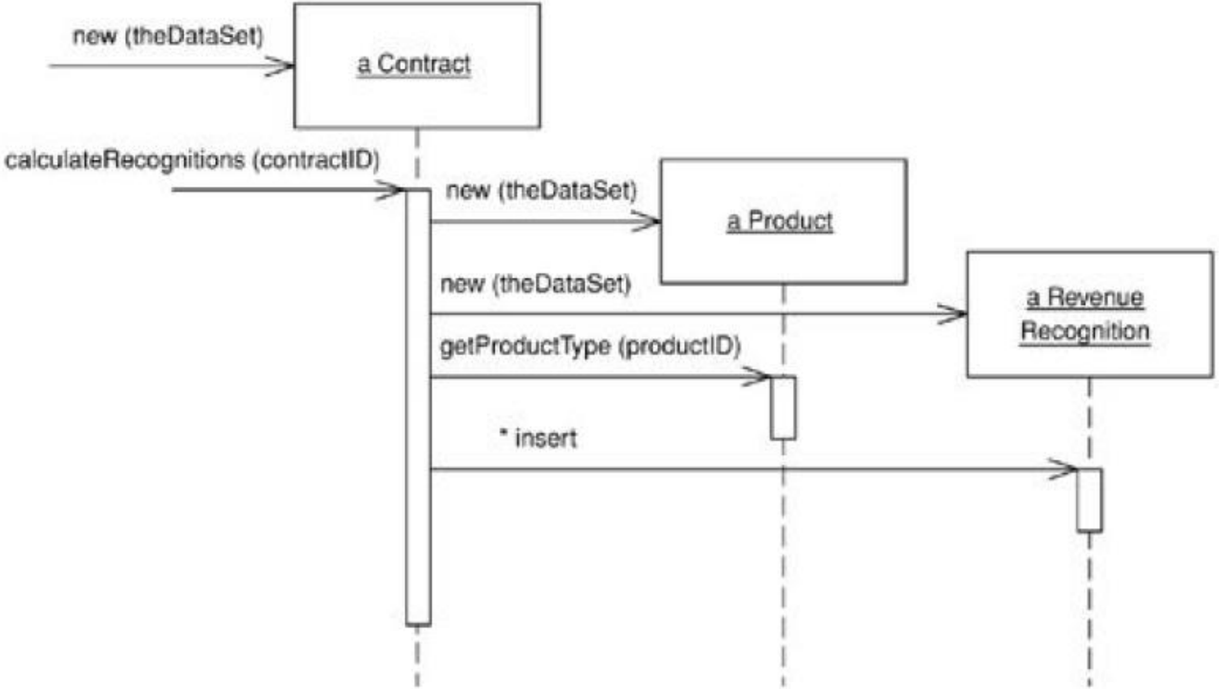
- Lose much of the benefits of domain models.
  - Logic, inheritance and other OO patterns.

Table Module

Recordsets & Record Tables in .Net

Works with many existing technologies.

Table Module



Software Engineering
Rochester Institute of Technology

Table vs. Domain

A *Table Module* organizes domain logic with one class per table in the database, and a single instance of a class contains the various procedures that will act on the data. The primary distinction with *Domain Model* is that, if you have many orders, a *Domain Model* will have one order object per order while a *Table Module* will have one object to handle all orders.

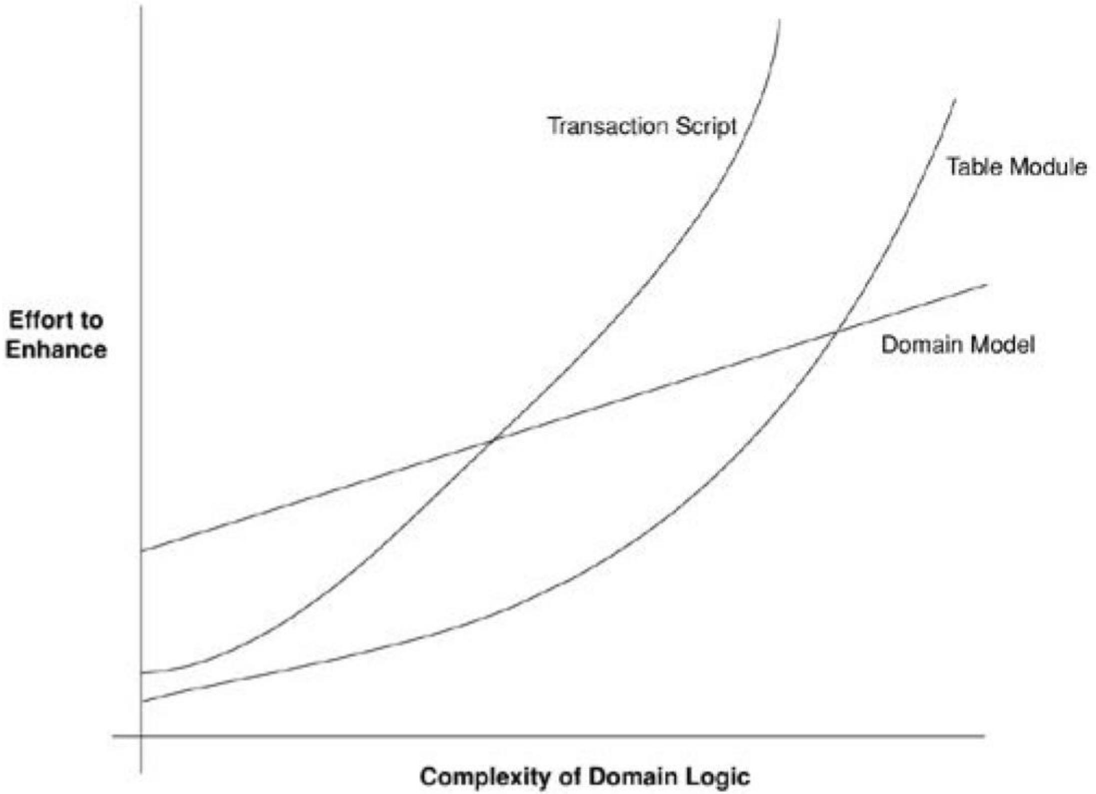
When to use table?

Highly based on table-oriented data, so use it when we access tabular data using a recordset

Fit business logic into the application in a well organized manner, but do not lose the way various elements work on the tabular data..

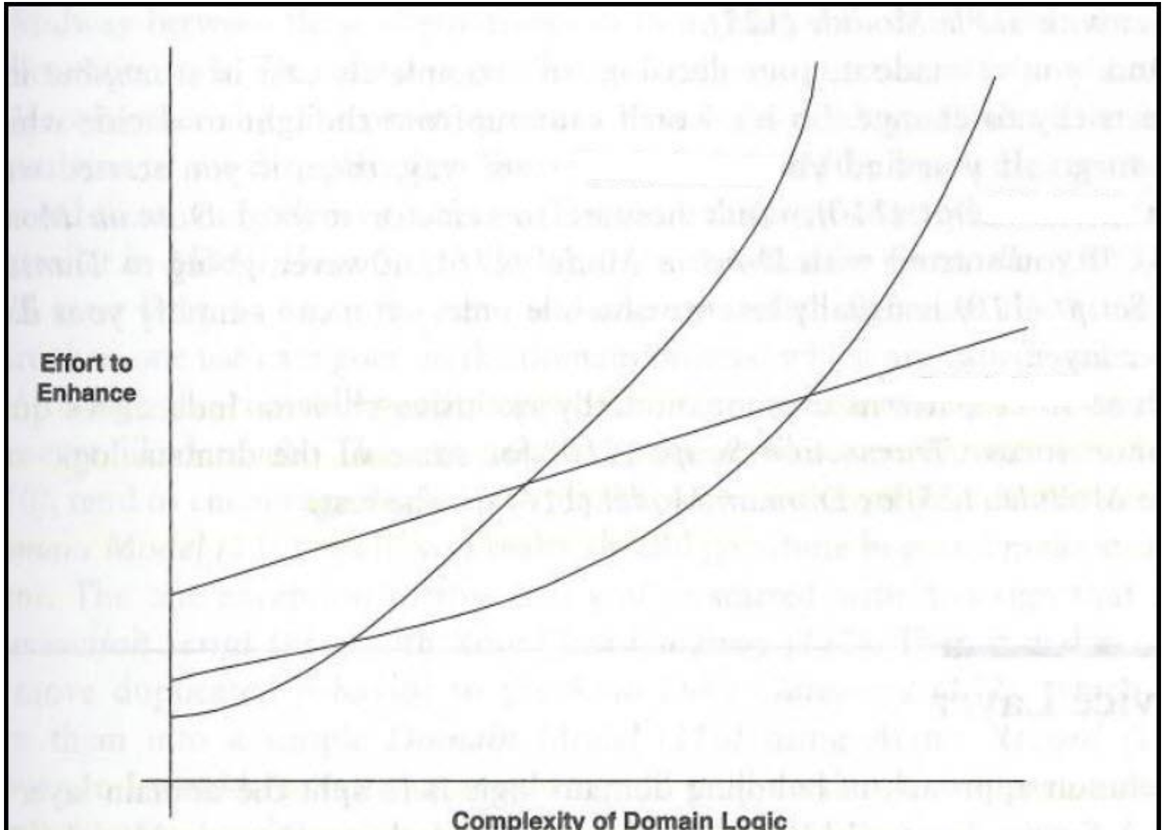
Which to Choose?

Relative Level of Effort vs. Domain Complexity



Software Engineering
Rochester Institute of Technology

Relative Level of Effort vs. Domain Complexity



Which to use?

- Application complexity
  - How to determine?
- Team familiarity with each
- Changing from one to the other can be expensive.
- Can mix and match
  - Application does not need to use one or the other.

Quiz - Script

What were the 3 primary patterns discussed today?

Recap - Transaction

What is it?

When is it used?

What are its drawbacks?

Recap - Domain

What is it?

When is it used?

What are its drawbacks?

Software Engineering

Rochester Institute of Technology

Recap - Table

What is it?

When is it used?

What are its drawbacks?

Resources

http://martinfowler.com/eaaCatalog/