Enterprise Data Model

• First step in database development
• Specifies scope and general content
• Overall picture of organizational data at high level of abstraction
• Entity-relationship diagram
• Descriptions of entity types
• Relationships between entities
• Business rules
Enterprise data model describes the high-level entities in an organization and the relationship between these entities.
Information Systems Architecture (ISA)

• Conceptual blueprint for organization’s desired information systems structure

• Consists of:
  – Data (e.g. Enterprise Data Model – simplified ER Diagram)
  – Processes – data flow diagrams, process decomposition, etc.
  – Data Network – topology diagram (like figure 1.7)
  – People – people management using project management tools (Gantt charts, etc.)
  – Events and points in time (when processes are performed)
  – Reasons for events and rules (e.g. decision tables)
Information Engineering

• A data-oriented methodology to create and maintain information systems

• Top-down planning: a generic IS planning methodology for obtaining a broad understanding of the IS needed by the entire organization

• Four steps to Top-Down planning:
  – Planning
  – Analysis
  – Design
  – Implementation
Information Systems Planning (Table 2-1)

- **Purpose:** align information technology with organization’s business strategies
- **Three steps:**
  1. Identify strategic planning factors
     - a. Goals
     - b. Critical success factors
     - c. Problem areas
  2. Identify corporate planning objects
     - a. Organizational units
     - b. Locations
     - c. Business functions
     - d. Entity types
  3. Develop enterprise model
     - a. Functional decomposition
     - b. Entity-relationship diagram
     - c. Planning matrices
Identify Strategic Planning Factors  
(Table 2-2)

- Organization goals – what we hope to accomplish
- Critical success factors – what MUST work in order for us to survive
- Problem areas – weaknesses we now have
Identify Corporate Planning Objects
(Table 2-3)

- Organizational units – departments
- Organizational locations
- Business functions – groups of business processes
- Entity types – the things we are trying to model for the database
- Information systems – application programs
Develop Enterprise Model

- Functional decomposition
  - See page 10 (Figure 2-2, text page 41).

- Enterprise data model
  - See page 3 (Figure 2-1, text page 38).

- Planning matrixes
  - See page 12 (Figure 2-3, text page 42).
Figure 2-2 -- Example of process decomposition of an order fulfillment function (Pine Valley Furniture)

Decomposition -- breaking large tasks into smaller tasks in a hierarchical structure chart.

- Order fulfillment
  - Fill customer orders
    - Receive sales order
    - Check customer credit
    - Create new customer
  - Ship customer orders
    - Check product availability
    - Create invoice
    - Create backorder
Planning Matrixes

• Describe relationships between planning objects in the organization

• Types of matrixes:
  – Function-to-data entity
  – Location-to-function
  – Unit-to-function
  – IS-to-data entity
  – Supporting function-to-data entity
  – IS-to-business objective
### Example business function-to-data entity matrix (Figure 2-3)

<table>
<thead>
<tr>
<th>Business Function (users)</th>
<th>Data Entity Types</th>
<th>Customer</th>
<th>Product</th>
<th>Raw Material</th>
<th>Order</th>
<th>Work Center</th>
<th>Work Order</th>
<th>Invoice</th>
<th>Equipment</th>
<th>Employee</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business Planning</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Product Development</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Materials Management</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Order Fulfillment</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Order Shipment</td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sales Summarization</td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Production Operations</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Finance and Accounting</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>
Two Approaches to Database and IS Development

• SDLC
  – System Development Life Cycle
  – Detailed, well-planned development process
  – Time-consuming, but comprehensive
  – Long development cycle

• Prototyping
  – Rapid application development (RAD)
  – Cursory attempt at conceptual data modeling.
  – Define database during development of initial prototype
  – Repeat implementation and maintenance activities with new prototype versions
**Systems Development Life Cycle**

**Figure 2-4**

- **Planning**
  - **Purpose**: To develop a preliminary understanding of a business situation and how information systems might help solve a problem or make an opportunity possible.
  - **Deliverable**: A written request to study the possible changes to an existing system or the development of a new system that addresses an information systems solution to the business problems or opportunities.

- **Analysis**
  - **Purpose**: To analyze the business situation thoroughly to determine requirements, to structure those requirements, and to select among competing system features.
  - **Deliverables**: The functional specifications for a system that meets user requirements and is feasible to develop and implement.

- **Design**
  - **Purpose**: To elicit and structure all information requirements; to develop all technology and organizational specifications.
  - **Deliverables**: Detailed functional specifications of all data, forms, reports, displays, and processing rules; program and database structures, technology purchases, physical site plans, and organizational redesigns.

- **Implementation**
  - **Purpose**: To write programs, build data files, test and install the new system, train users, and finalize documentation.
  - **Deliverables**: Programs that work accurately and to specifications, documentation, and training materials.

- **Maintenance**
  - **Purpose**: To monitor the operation and usefulness of a system, and to repair and enhance the system.
  - **Deliverables**: Periodic audits of the system to demonstrate whether the system is accurate and still meets needs.
Database Development Activities During The Systems Development Life Cycle

**Figure 2-5**

**Enterprise modeling**
- Analyze current data processing
- Analyze the general business functions and their database needs
- Justify need for new data and databases in support of business

**Conceptual data modeling**
- Identify scope of database requirements for proposed information system
- Analyze overall data requirements for business function(s) supported by database

**Database maintenance**
- Analyze database and database applications to ensure that evolving information requirements are met
- Tune database for improved performance
- Fix errors in database and database applications and recover database when it is contaminated

**Database implementation**
- Code and test database processing programs
- Complete database documentation and training materials
- Install database and convert data from prior systems

**Conceptual data modeling, cont’d.**
- Develop preliminary conceptual data model, including entities and relationships
- Compare preliminary conceptual data model with enterprise data model
- Develop detailed conceptual data model, including all entities, relationships, attributes, and business rules
- Make conceptual data model consistent with other models of information system
- Populate repository with all conceptual database specifications

**Logical database design**
- Analyze in detail the transactions, forms, displays, and inquiries (database views) required by the business functions supported by the database
- Integrate database views into conceptual data model
- Identify data integrity and security requirements, and populate repository

**Physical database design and definition**
- Define database to DBMS (often generated from repository)
- Decide on physical organization of data
- Design database processing programs
Figure 2-6 The prototyping methodology and database development process:

1. Identify problem
2. Develop initial prototype
3. Convert to operational system
4. Implement and use prototype
5. Revise and enhance prototype
6. Working prototype
7. New requirements
8. Problems
9. Next version
10. Initial requirements

If prototype is inefficient, go to step 3.
Figure 2-6 The prototyping methodology and database development process

- Conceptual data modeling
  - Analyze requirements
  - Develop preliminary data model

1. Identify problem → Develop initial prototype
2. Develop initial prototype → Working prototype
3. Working prototype → Implement and use prototype
4. Implement and use prototype → Revise and enhance prototype
5. Revise and enhance prototype → Next version
6. Next version → Problems
7. Problems → New requirements
8. New requirements → Identify problem
9. Convert to operational system
10. If prototype is inefficient → Implement and use prototype
Figure 2-6 The prototyping methodology and database development process

Conceptual data modeling
- Analyze requirements
- Develop preliminary data model

Identify problem → Develop initial prototype

Logical database design
- Analyze requirements in detail
- Integrate database views into conceptual data model

Physical database design and definition
- Define new database contents to DBMS
- Decide on physical organization for new data
- Design database processing programs

Convert to operational system

Working prototype

Database implementation
- Code database processing
- Install new database contents, usually from existing data sources

Implement and use prototype

If prototype is inefficient

New requirements

Revise and enhance prototype

Problems → Next version

Database maintenance
- Analyze database to ensure it meets application needs
- Fix errors in database
Figure 2-6 The prototyping methodology and database development process

Conceptual data modeling
- Analyze requirements
- Develop preliminary data model

Identify problem

Initial requirements

Develop initial prototype

Logical database design
- Analyze requirements in detail
- Integrate database views into conceptual data model

Physical database design and definition
- Define new database contents to DBMS
- Decide on physical organization for new data
- Design database processing programs

Convert to operational system

Working prototype

Database implementation
- Code database processing
- Install new database contents, usually from existing data sources

If prototype is inefficient

Database maintenance
- Analyze database to ensure it meets application needs
- Fix errors in database

Implement and use prototype

Problems

Next version

New requirements

Revise and enhance prototype
Figure 2-6 The prototyping methodology and database development process

- Conceptual data modeling:
  - Analyze requirements
  - Develop preliminary data model

- Database maintenance:
  - Tune database for improved performance
  - Fix errors in database

- Logical database design:
  - Analyze requirements in detail
  - Integrate database views into conceptual data model

- Physical database design and definition:
  - Define new database contents to DBMS
  - Decide on physical organization for new data
  - Design database processing programs

- Database implementation:
  - Code database processing
  - Install new database contents, usually from existing data sources

- Database maintenance:
  - Analyze database to ensure it meets application needs
  - Fix errors in database

- Initial requirements:
  - Develop initial prototype

- Working prototype:
  - Convert to operational system

- If prototype is inefficient:
  - Implement and use prototype

- New requirements:
  - Revise and enhance prototype

- Next version:
  - Problems

- Next version:
  - Problems
Packaged Data Models

- Model components that can be purchased, customized, and assembled into full-scale data models

- Advantages
  - Reduced development time
  - Higher model quality and reliability

- Two types:
  - Universal data models
  - Industry-specific data models
CASE

• Computer-Aided Software Engineering (CASE) – software tools providing automated support for systems development

• Three database features:
  – Data modeling – entity-relationship diagrams
  – Code generation – SQL code for table creation
  – Repositories – knowledge base of enterprise information
Managing Projects

- Project – a planned undertaking of related activities to reach an objective that has a beginning and an end
- Involves use of review points for:
  - Validation of satisfactory progress
  - Step back from detail to overall view
  - Renew commitment of stakeholders
- Incremental commitment – review of systems development project after each development phase with rejustification after each phase
Managing Projects: People Involved

• Systems analysts
• Database analysts
• Users
• Programmers
• Database/data administrators
• Systems programmers, network administrators, testers, technical writers
Gantt Chart

Sales Promotion Tracking

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Requirements Collection</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Screen Design</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Report Design</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Database Design</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>User Documentation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Programming</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Testing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Installation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Date: 4/1/98 8:00am

Critical: [ ]
Noncritical: [ ]
Progress: [ ]
Milestone: [ ]
Summary: [ ]
Rolled Up: [ ]

Shows time estimates of tasks
PERT chart (Program Evaluation and Review Technique)

Shows dependencies between tasks