COMPUTER AIDED SOFTWARE ENGINEERING (CASE)

Dr. Ebadati
Kharazmi University
DATA MODELING

Chapter 6
KEY DEFINITIONS

• **Data model**
  • A formal way of representing the data that are used and created by a business system
  • Shows the people, places and things about which data is captured and the relationships among them.

• **Logical data model**
  • shows the organization of data without indicating how it is stored, created, or manipulated.
KEY DEFINITIONS

- **Physical data model**
  - shows how the data will actually be stored in databases or files.

- **Normalization** is the **process** analysts use to validate data models.

- Data models should **balance** with process models.
THE ENTITY-RELATIONSHIP DIAGRAM (ERD)
WHAT IS AN ERD?

• A picture showing the **information created, stored, and used** by a business **system**.
• **Entities** generally **represent similar** kinds of **information**
• **Lines** drawn **between entities** show **relationships** among the data
• High level business rules are also shown
• **Business rules** are *constraints* that are followed when the *system* is in operation.

• ERD symbols can show when one instance of an entity must exist for an instance of another to exist
  
  • *A doctor must exist before appointments for the doctor can be made*
USING THE ERD TO SHOW BUSINESS RULES

• **ERD symbols** can show when one instance of an **entity can be related to only one or many instances of another entity**
  - One doctor can have many patients; each patient may have only one primary doctor

• **ERD symbols show when the existence of an entity instance is optional for a related entity instance**
  - A patient may or may not have insurance coverage
AN ERD EXAMPLE
# ERD Elements

<table>
<thead>
<tr>
<th>Element</th>
<th>IDEFIX</th>
<th>Chen</th>
<th>Crow’s Foot</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>An ENTITY:</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>✓ Is a person, place, or thing</td>
<td>ENTITY-NAME</td>
<td>ENTITY-NAME</td>
<td>ENTITY-NAME</td>
</tr>
<tr>
<td>✓ Has a singular name spelled in all capital letters</td>
<td>Identifier</td>
<td></td>
<td><em>Identifier</em></td>
</tr>
<tr>
<td>✓ Has an identifier</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>✓ Should contain more than one instance of data</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>An ATTRIBUTE:</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>✓ Is a property of an entity</td>
<td>ENTITY-NAME</td>
<td>ATTRIBUTE-NAME</td>
<td>ATTRIBUTE-NAME</td>
</tr>
<tr>
<td>✓ Should be used by at least one business process</td>
<td>Attribute-name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>✓ Is broken down to its most useful level of detail</td>
<td>Attribute-name</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>A RELATIONSHIP:</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>✓ Shows the association between two entities</td>
<td>Relationship-name</td>
<td>Relationship-name</td>
<td>Relationship-name</td>
</tr>
<tr>
<td>✓ Has a parent entity and a child entity</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>✓ Is described with a verb phrase</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>✓ Has cardinality (1 : 1, 1 : N, or M : N)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>✓ Has modality (null, not null)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>✓ Is dependent or independent</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
• A person, place, event, or thing about which data is collected
• Must be multiple occurrences to be an entity
  • Example: If a firm has only one warehouse, the warehouse is not an entity. However, if the firm has several warehouses, the warehouse could be an entity if the firm wants to store data about each warehouse instance.
## Entities and Instances

<table>
<thead>
<tr>
<th>Entity</th>
<th>Example Instances</th>
</tr>
</thead>
<tbody>
<tr>
<td>Patient</td>
<td>John Smith</td>
</tr>
<tr>
<td></td>
<td>Susan Jones</td>
</tr>
<tr>
<td></td>
<td>Peter Todd</td>
</tr>
<tr>
<td></td>
<td>Dale Turner</td>
</tr>
<tr>
<td></td>
<td>Pat Turner</td>
</tr>
</tbody>
</table>
• Information captured about an entity
• Only those used by the organization should be included in the model
• Attribute names are nouns
• Sometimes entity name is added at the beginning of the attribute name for clarity
IDENTIFIERS

• One or more attributes can serve as the entity identifier, uniquely identifying each entity instance

• Concatenated identifier consists of several attributes

• An identifier may be ‘artificial,’ such as creating an ID number

• Identifiers may not be developed until the Design Phase
CHOICES FOR IDENTIFIERS

- Concatenated Identifier
  - PAT_lastname
  - PAT_firstname

- Single Identifier
  - PAT_idnumber
  - PAT_lastname
  - PAT_firstname

- Identifier to be Added Later
  - PAT_lastname
  - PAT_firstname
• Associations between entities
• The **first entity** in the relationship is the *parent* entity; the **second entity** in the relationship is the *child entity*
• Relationships should have active verb names
• Relationships go in both directions
• Cardinality
  • refers to the number of times instances in one entity can be related to instances in another entity
    • One instance in an entity refers to one and only one instance in the related entity (1:1)
    • One instance in an entity refers to one or more instances in the related entity (1:N)
    • One or more instances in an entity refer to one or more instances in the related entity (M:N)
M : N RELATIONSHIPS

PATIENT
*PAT_idnumber
PAT_firstname
PAT_lastname
PAT_address
PAT_city
PAT_state
PAT_zipcode
PAT_homephone
PAT_birthdate

SYMPTOM
*SYM_name
SYM_description

DOCTOR
*DOC_physicianidnumber
DOC_firstname
DOC_lastname
DOC_address
DOC_city
DOC_state
DOC_zipcode
DOC_homephone
DOC_pagernumber
DOC_primaryspecialty

SPECIALTY
*SPE_name
SPE_description

presents/is presented by
has qualified for/is associated with
THE DATA DICTIONARY AND METADATA

• **Metadata** is *information stored* about *components* of the data model

• Metadata is stored in the data dictionary so it can be shared by developers and users throughout the SDLC

• A complete, shareable data dictionary helps improve the quality of the system under development
DATA DICTIONARY ENTRY FOR THE PATIENT ENTITY (SHOWN USING ERWIN)

The patient entity refers to people who have scheduled an appointment. It does not include future patients (people who have not yet made an appointment).
BUILDING AN ENTITY-RELATIONSHIP DIAGRAM (ERD)
ERD BASICS

• Drawing the ERD is an iterative process of trial and revision
• ERDs can become quite complex
STEPS IN BUILDING ERDS

• Identify the entities
• Add attributes and assign identifiers
• Identify relationships
IDENTIFY THE ENTITIES

- Identify major categories of information
  - If available, check the process models for data stores, external entities, and data flows
  - Check the major inputs and outputs from the use cases

- Verify that there is more than one instance of the entity that occurs in the system
IDENTIFY RELATIONSHIPS

• **Start** with an **entity** and **identify all entities** with which it shares relationships

• Describe the relationship with the appropriate verb phrase

• Determine the cardinality and modality by discussing the business rules with knowledgeable users
ERD BUILDING TIPS

• Data stores of the DFD should correspond to entities
• Only include entities with more than one instance of information
• Don’t include entities associated with implementation of the system (they will be added later)
ADVANCED SYNTAX

• **Independent Entity**
  • Can exist without the help of another entity
  • Identifiers created from the entity’s own attributes
  • Attributes from other entities are not needed to uniquely identify instances of these entities

• **Dependent Entity**
  • Relationships when a child entity does require attributes from the parent entity to uniquely identify an instance
**Intersection Entity**

- Exists in order to capture some information about the relationship that *exists between two other entities*. Typically, intersection entities are added to a data model to store information about two entities sharing an *M : N relationship*. 
ADVANCED SYNTAX – RESOLVING AN M : N RELATIONSHIP
VALIDATING AN ERD
DESIGN MODELING GUIDELINES
SUMMARY

1. Country only has one instance (i.e., Mexico). This entity is not needed.

2. If teachers are called “Professors”, then the ERD should contain an entity called “Professor” to remain consistent.

3. Why are all of these attributes being captured about university? Will it be necessary to store the founder and first president of each university? If not, these attributes should be removed from the ERD.

4. The attributes in the subject entity are poorly labeled. For one, we have no way of knowing to which entity they belong if they stood alone—it would be helpful to begin each attribute with SUB_. Also, what is area? A word like department or field of research may be more descriptive.

5. The name attribute really should be broken down into last name and first name—otherwise, there would be no way to manipulate names in the system. For example, there would be no way to sort by last name if it was combined with someone’s first name.
NORMALIZATION

• Technique used to validate data models
• Series of rules applied to logical data model to improve its organization
• Three normalization rules are common
# NORMALIZATION STEPS

<table>
<thead>
<tr>
<th>Normal Form</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 Normal Form</td>
<td>Do any attributes have multiple values for a single instance of an entity? Yes: Remove the repeating attributes and repeating groups. Create an entity that describes the attributes. Usually you will need to add a relationship to connect the old and new entities. No: The data model is in 1NF.</td>
</tr>
<tr>
<td>1 Normal Form</td>
<td>Is the identifier comprised of more than one attribute? If so, are any attribute values dependent on just part of the identifier? Yes: Remove the partial dependency. Move the attributes to an entity in which their values are dependent on the entire identifier. Usually you will need to create a new entity and add a relationship to connect the old and new entities. No: The data model is in 2NF.</td>
</tr>
<tr>
<td>2 Normal Form</td>
<td>Do any attribute values depend on an attribute that is not the entity’s identifier? Yes: Remove the transitive dependency or derived attribute. Move the attributes to an entity in which their values are dependent on the identifier. Usually you will need to create a new entity and add a relationship to connect the old and new entities. No: The data model is in 3NF.</td>
</tr>
<tr>
<td>3 Normal Form</td>
<td></td>
</tr>
</tbody>
</table>
UN-NORMALIZED ENTITY

Begin with an entity from the logical data model.
FIRST NORMAL FORM (1NF)

Look for repeating groups of attributes and remove them into separate entities.
SECOND NORMAL FORM (2NF)

If an entity has a joint identifier, look for attributes that depend only on part of the identifier. If found, remove to a new entity.
THIRD NORMAL FORM (3NF)

Look for attributes that depend only on another non-identifying attribute. If found, remove to new entity. Also remove any calculated attributes.
• All analysis activities are interrelated
• Process models contain two data components
  • Data flows and data stores
• The DFD data components need to balance the ERD’s data stores (entities) and data elements (attributes)
• Many CASE tools provide features to check for imbalance
• Check that all data stores and elements correspond between models
  • Data that is not used is unnecessary
  • Data that has been omitted results in an incomplete system
• Do not follow thoughtlessly -- check that the models make sense!
The ERD is the most common technique for drawing data models. The building blocks of the ERD are:

- **Entities** describe people, places, or things
- **Attributes** capture information about the entity
- **Relationships** associate data across entities

Intersection, dependent, and independent entities must be recognized.

The **ERD must be balanced with the DFD**.
ERD DRAW TOOL

• http://www.aquafold.com/aquadatastudio/er_modeler.html
ERD EXAMPLE
Dimensional Model of Store Sales System

**Store Sales Fact**
- PK, FK1: **Store_key**
- PK, FK2: **Employee_key**
- PK, FK3: **Customer_key**
- PK, FK4: **Date_key**
- PK, FK5: **Product Key**
- PK, FK5: **Department_key**
- PK, FK6: **Time_key**

**Sales Person Dimension**
- PK: **Employee_key**
  - EmployeeID
  - Full Name

**Time Dimension**
- PK: **Time_key**
  - Time

**Customer Dimension**
- PK: **Customer_key**
  - CustomerID
  - Full Name

**Store Dimension**
- PK: **Store_key**
  - StoreID
  - Store Name

**Product Dimension**
- PK: **Product Key**
  - ProductID
  - Product Name
  - DepartmentID
  - Department Name
موافق باشید