Entity Relationship Diagrams

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.
Using the ERD to Show Business Rules

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

DOCTOR
- DOC_physicianidnumber
- DOC_firstname
- DOC_lastname
- DOC_address
- DOC_city
- DOC_state
- DOC_zipcode
- DOC_homephone
- DOC_pagernumber
- DOC_primspecialty

APPOINTMENT
- APP_date
- APP_time
- DOC_physicianidnumber
- APP_duration
- APP_reason
- APP_status

PATIENT
- PAT_idnumber
- PAT_firstname
- PAT_lastname
- PAT_address
- PAT_city
- PAT_state
- PAT_zipcode
- PAT_homephone
- PAT_pagernumber
- PAT_birthdate

BILL
- BIL_number
- BIL_amountinsured
- BIL_amountnotinsured
- BIL_datesent
- BIL_status

PAYMENT
- PAY_receiptnumber
- PAY_amount
- PAY_date
- PAY_method

INSURANCE COMPANY
- INS_name
- INS_benefitscontract
- INS_phonenumber
- INS_claimsaddress
- INS_claimsummaryinformation

generates/is generated by

is scheduled by/includes

schedules/is scheduled by

makes/is made by

makes/is made by

insures/is insured by

pays/is paid by
### ERD Elements

<table>
<thead>
<tr>
<th></th>
<th>IDEFIX</th>
<th>Chen</th>
<th>Crow’s Foot</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>An ENTITY:</strong></td>
<td>An ENTITY:</td>
<td>An ENTITY:</td>
<td>An ENTITY:</td>
</tr>
<tr>
<td></td>
<td>✓ Is a person, place, or thing</td>
<td>✓ Has a singular name</td>
<td>✓ Identifier</td>
</tr>
<tr>
<td></td>
<td>✓ Has a singular name</td>
<td>✓ spelled in all capital letters</td>
<td></td>
</tr>
<tr>
<td></td>
<td>✓ Has an identifier</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>✓ Should contain more than one</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>instance of data</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>An ATTRIBUTE:</strong></td>
<td>An ATTRIBUTE:</td>
<td>An ATTRIBUTE:</td>
<td>An ATTRIBUTE:</td>
</tr>
<tr>
<td></td>
<td>✓ Is a property of an entity</td>
<td>✓ Should be used by at least</td>
<td>✓ Attribute-name</td>
</tr>
<tr>
<td></td>
<td>✓ Should be used by at least</td>
<td>one business process</td>
<td></td>
</tr>
<tr>
<td></td>
<td>one business process</td>
<td>✓ Is broken down to its most</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>useful level of detail</td>
<td></td>
</tr>
<tr>
<td><strong>A RELATIONSHIP:</strong></td>
<td>A RELATIONSHIP:</td>
<td>A RELATIONSHIP:</td>
<td>A RELATIONSHIP:</td>
</tr>
<tr>
<td></td>
<td>✓ Shows the association</td>
<td>✓ Shows the association</td>
<td>✓ Shows the association</td>
</tr>
<tr>
<td></td>
<td>between two entities</td>
<td>between two entities</td>
<td>between two entities</td>
</tr>
<tr>
<td></td>
<td>✓ Has a parent entity and a</td>
<td>✓ Has a parent entity and a</td>
<td></td>
</tr>
<tr>
<td></td>
<td>child entity</td>
<td>child entity</td>
<td></td>
</tr>
<tr>
<td></td>
<td>✓ Is described with a verb</td>
<td>✓ Is described with a verb</td>
<td></td>
</tr>
<tr>
<td></td>
<td>phrase</td>
<td>phrase</td>
<td></td>
</tr>
<tr>
<td></td>
<td>✓ Has cardinality (1 : 1, 1 : N,</td>
<td>✓ Has cardinality (1 : 1, 1 : N,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>or M : N)</td>
<td>or M : N)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>✓ Has modality (null, not null)</td>
<td>✓ Has modality (null, not null)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>✓ Is dependent or independent</td>
<td>✓ Is dependent or independent</td>
<td></td>
</tr>
</tbody>
</table>
Entity

- 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>
Attributes

- 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_Iddnumber
  - PAT_lastname
  - PAT_firstname

- **Identifier to be Added Later**
  - PAT_lastname
  - PAT_firstname
Relationships

- 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)
Modality

Refers to whether or not an instance of a child entity can exist without a related instance in the parent entity

- **Not Null** means that an instance in the related entity must exist for an instance in another entity to be valid
- **Null** means that no instance in the related entity is necessary for an instance in another entity to be valid
M : N Relationships

**PATIENT**
- PAT_idnumber
- PAT_firstname
- PAT_lastname
- PAT_address
- PAT_city
- PAT_state
- PAT_zipcode
- PAT_homephone
- PAT_birthday

**SYMPTOM**
- SYM_name
- SYM_description

**DOCTOR**
- DOC_physicianidnumber
- DOC_firstname
- DOC_lastname
- DOC_address
- DOC_city
- DOC_state
- DOC_zipcode
- DOC_homephone
- DOC_pagerphone
- DOC_specialty

**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
Step 1: 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
Step 2: Add Attributes and Assign Identifiers

Identify attributes of the entity that are relevant to the system under development

- Check the process model repository entries for details on data flows and data stores
- Check the data requirements of the requirements definition
- Interview knowledgeable users
- Perform document analysis on existing forms and reports

Select the entity’s identifier
Step3: 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
Advanced Syntax

**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

**PATIENT**
- PAT_idnumber
- PAT.firstname
- PAT.lastname
- PAT.address
- PAT.state
- PAT.zipcode
- PAT.homephone
- PAT.birthdate

**SYMPTOM**
- SYM.name
- SYM.description

**PATIENT**
- PAT_idnumber
- PAT_firstname
- PAT_lastname
- PAT_address
- PAT_state
- PAT_zipcode
- PAT_homephone
- PAT_birthdate

**PATIENT_SYMPTOM**
- PAT_idnumber
- SYM_name
- PS_date

**SYMPTOM**
- SYM_name
- SYM_description
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.

6. This model assumes that a teacher can only work for one university—what about those with joint appointments? An assumption should be stated on the model or in the documentation so that this business rule can be confirmed.
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>Step Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 Normal Form</td>
<td><strong>Do any attributes have multiple values for a single instance of an entity?</strong>&lt;br&gt;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.&lt;br&gt;No: The data model is in 1NF.</td>
</tr>
<tr>
<td>1 Normal Form</td>
<td><strong>Is the identifier comprised of more than one attribute? If so, are any attribute values dependent on just part of the identifier?</strong>&lt;br&gt;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.&lt;br&gt;No: The data model is in 2NF.</td>
</tr>
<tr>
<td>2 Normal Form</td>
<td><strong>Do any attribute values depend on an attribute that is not the entity’s identifier?</strong>&lt;br&gt;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.&lt;br&gt;No: The data model is in 3NF.</td>
</tr>
<tr>
<td>3 Normal Form</td>
<td><strong>Do any attributes have multiple values for a single instance of an entity?</strong>&lt;br&gt;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.&lt;br&gt;No: The data model is in 1NF.</td>
</tr>
</tbody>
</table>
Unnormalized Entity

Begin with an entity from the logical data model

<table>
<thead>
<tr>
<th>SPECIAL ORDER</th>
</tr>
</thead>
<tbody>
<tr>
<td>*Special Order Date</td>
</tr>
<tr>
<td>*Customer Last Name</td>
</tr>
<tr>
<td>*Customer First Name</td>
</tr>
<tr>
<td>Customer Phone</td>
</tr>
<tr>
<td>Customer Address</td>
</tr>
<tr>
<td>Customer Birthdate</td>
</tr>
<tr>
<td>Customer Book Preferences</td>
</tr>
<tr>
<td>Book ISBN1</td>
</tr>
<tr>
<td>Book Name1</td>
</tr>
<tr>
<td>Book Author1</td>
</tr>
<tr>
<td>Book Publication Year1</td>
</tr>
<tr>
<td>Book Author University1</td>
</tr>
<tr>
<td>Book ISBN2</td>
</tr>
<tr>
<td>Book Name2</td>
</tr>
<tr>
<td>Book Author2</td>
</tr>
<tr>
<td>Book Publication Year2</td>
</tr>
<tr>
<td>Book Author University2</td>
</tr>
<tr>
<td>Book ISBN3</td>
</tr>
<tr>
<td>Book Name3</td>
</tr>
<tr>
<td>Book Author3</td>
</tr>
<tr>
<td>Book Publication Year3</td>
</tr>
<tr>
<td>Book Author University3</td>
</tr>
<tr>
<td>Store Name</td>
</tr>
<tr>
<td>Store Manager</td>
</tr>
<tr>
<td>Store Location</td>
</tr>
<tr>
<td>Special Order Status</td>
</tr>
<tr>
<td>Special Order Days on Order</td>
</tr>
</tbody>
</table>
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 concatenated identifier, look for attributes that depend only on part of the identifier. If found, remove to 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.
Partial Process Model and CRUD Matrix
Balancing ERDs with DFDs

- 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!
Summary

- 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.
All rights reserved. Reproduction or translation of this work beyond that permitted in Section 117 of the 1976 United States Copyright Act without the express written permission of the copyright owner is unlawful.

Request for further information should be addressed to the Permissions Department, John Wiley & Sons, Inc.

The purchaser may make back-up copies for his/her own use only and not for redistribution or resale.

The Publisher assumes no responsibility for errors, omissions, or damages, caused by the use of these programs or from the use of the information contained herein.