IMS1002/CSE1205
Systems Analysis and Design
Lecture 2 & 3
Introduction to Data Modelling
Entity Relationship Modelling

Data Modelling
- Focus on the information aspects of the organisation
- In a database environment many applications share the same data
- The database is a common asset and corporate resource
- Corporate and application level data modelling

Conceptual Data Modelling
- A conceptual data model is a representation of organisational data
- Captures the structure, meaning and interrelationships amongst the data
- Independent of any data storage and access method, DBMS, platform issues
- Occurs in parallel with other systems analysis activities

Conceptual Data Modelling
- Identification of information requirements
- Allows integration of data across the organisation and across applications
- Allows sharing of data across organisation
- Helps eliminate problems of data inconsistency and duplication across the organisation

Entity Relationship (ER) Modelling
- Used for conceptual data modelling
  - has become a de facto industry standard especially for relational databases
- Diagrammatic technique used to represent
  - things of importance in an organisation
    - entities
  - the properties of those things
    - attributes
  - how they are related to each other
    - relationships

Conceptual Data Modelling
- Techniques
  - Entity Relationship (ER) modelling
  - Data Gathering - attributes
  - Normalisation
  - Data Structure Diagrams (DSD)
- Good modelling techniques are supported by rigorous standards and conventions to remove ambiguity and aid understanding
Entity Relationship (ER) Modelling

- Entity relationship (ER) models can be readily transformed into a variety of technical architectures.
- All information about the system’s data identified during conceptual data modelling must be entered into the data dictionary or repository.
- This assists in checking the consistency of data and process models.

Data "objects" or entities are things about which we wish to store information:
- things of interest to the organisation.

ER models show the major data objects and the associations between them:
- business rules, constraints.

ER models are useful in the initiation, analysis and design phases.

Entity

- Something of interest about which we store information:
  - EMPLOYEE
  - SALES ORDER
  - SUPPLIER

- Often identified from nouns used within the business application.

- Should be LOGICAL (not physical):
  - capture the essential nature of the data.

Entities are subjective (i.e. they reflect the needs and viewpoint of the system) and can be:
- Real
  - VEHICLE
- Abstract
  - QUOTA
- Event remembered
  - LOAN
- Role played
  - CUSTOMER
- Organisation
  - DEPARTMENT
- Geographical
  - LOCATION

Representing Entities

- We represent an entity by a named rectangle.
- Use a singular noun, or adjective + noun.
- Refer to one instance in naming convention.

Entity Types and Instances

- An entity type is a classification of entity instances:
  - BN Holdings
  - ABC Engineering
  - Acme Corp. Ltd.

- …are all instances of a Supplier – they share common characteristics.
For example, in a sales and inventory system there might be 3 physical forms of data—
- a stock file
- product brochures sent to customers enquiring about products
- a product range book used by salespeople when calling on customers to take orders
These could all be represented by one logical entity - PRODUCT

In a Student Records System there might be an entity type STUDENT which represents some of the data used in several physical forms of data—
- student re-enrolment forms
- subject class lists
- student results file
The ER model identifies the minimum set of data objects necessary to construct the data used within the system in its various physical forms.

Is an association between two entities. We may also wish to store information about the association. Often recognised by a verb—
- "entity" + verb + "entity"
  - eg CUSTOMER places ORDER
Relationships capture the structural associations between the data entities—
  - "business rules" of the system

We represent a relationship as a line between two entities.
The relationship is named by a meaningful verb or verb phrase which should indicate the meaning of the association.
Relationships are bi-directional so naming each end of the relationship conveys more meaning.

A relationship type is a classification of relationship instances.

<table>
<thead>
<tr>
<th>Relationship Type</th>
<th>Instance 1</th>
<th>Instance 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Marketing</td>
<td>employs</td>
<td>Sue Black</td>
</tr>
<tr>
<td>Finance</td>
<td>employs</td>
<td>Bill Brown</td>
</tr>
<tr>
<td>MIS</td>
<td>employs</td>
<td>John Smith</td>
</tr>
<tr>
<td>DEPT</td>
<td>employs</td>
<td>EMPLOYEE</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cardinality</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1:n</td>
<td>a CUSTOMER may place many ORDERs</td>
</tr>
<tr>
<td>1:1</td>
<td>an ORDER is placed by one CUSTOMER</td>
</tr>
<tr>
<td>n:m</td>
<td>an ITEM can appear on many ORDERs</td>
</tr>
</tbody>
</table>
**Types of Cardinalities**

- **One to One**
  - EMPLOYEE
  - PROJECT
  - Leads

- **One to Many**
  - EMPLOYEE
  - CUSTOMER
  - Leads

- **Many to Many**
  - EMPLOYEE
  - SUPPLIER
  - Supplies

**Nature of Relationships**

- Cardinality constraints enable us to precisely identify the nature of the relationship.
- We can indicate whether relationships are optional or mandatory.
  - A customer MAY place many sales orders.
  - Each sales order MUST be placed by one customer.

**ER Notations**

- **Notation used in Hoffer et al**
  - EMPLOYEE
  - COURSE
  - Attends

- **Notation used in Whitten et al**
  - EMPLOYEE
  - COURSE
  - Attends

**Degree of a Relationship**

- The degree of a relationship is the number of entity types that participate in the relationship.
- The most common relationships in ER modelling in practice are:
  - Unary (degree one)
  - Binary (degree two)
  - Ternary (degree three)
  - N-ary or higher degree

**Unary Relationships**

- An unary relationship is a relationship between instances of one entity type.
  - An instance of one entity has a relationship with one or more different instances of the same entity type.
  - Also called a recursive relationship.
Binary Relationships

- A binary relationship is a relationship between instances of two entity types and is the most common type of relationship encountered in practice.
- Each instance of one entity has a relationship with one or more instances of a second entity.

![Binary Relationships Diagram]

Ternary Relationships

- A ternary relationship is a simultaneous relationship between instances of three entity types.
- A ternary relationship is NOT the same as three binary relationships between the same three entity types.

![Ternary Relationships Diagram]

Ternary Relationships

- A ternary relationship is a simultaneous relationship between instances of three entity types.
- A ternary relationship is NOT the same as three binary relationships between the same three entity types.

![Ternary Relationships Diagram]

Ternary Relationships

- A ternary relationship is a simultaneous relationship between instances of three entity types.
- A ternary relationship is NOT the same as three binary relationships between the same three entity types.

![Ternary Relationships Diagram]

Ternary Relationships

- A ternary relationship is a simultaneous relationship between instances of three entity types.
- A ternary relationship is NOT the same as three binary relationships between the same three entity types.

![Ternary Relationships Diagram]

Ternary Relationships

- A ternary relationship is a simultaneous relationship between instances of three entity types.
- A ternary relationship is NOT the same as three binary relationships between the same three entity types.

![Ternary Relationships Diagram]

Ternary Relationships

- A ternary relationship is a simultaneous relationship between instances of three entity types.
- A ternary relationship is NOT the same as three binary relationships between the same three entity types.

![Ternary Relationships Diagram]

Ternary Relationships

- A ternary relationship is a simultaneous relationship between instances of three entity types.
- A ternary relationship is NOT the same as three binary relationships between the same three entity types.

![Ternary Relationships Diagram]

Ternary Relationships

- A ternary relationship is a simultaneous relationship between instances of three entity types.
- A ternary relationship is NOT the same as three binary relationships between the same three entity types.

![Ternary Relationships Diagram]

Ternary Relationships

- A ternary relationship is a simultaneous relationship between instances of three entity types.
- A ternary relationship is NOT the same as three binary relationships between the same three entity types.

![Ternary Relationships Diagram]

Ternary Relationships

- A ternary relationship is a simultaneous relationship between instances of three entity types.
- A ternary relationship is NOT the same as three binary relationships between the same three entity types.

![Ternary Relationships Diagram]

Ternary Relationships

- A ternary relationship is a simultaneous relationship between instances of three entity types.
- A ternary relationship is NOT the same as three binary relationships between the same three entity types.

![Ternary Relationships Diagram]
Some data values vary over time and it may be important to store a history of data values to understand trends and for forecasting:
- for accounting purposes we are likely to need a history of costs of material and labour costs and the time period over which each cost was in effect
- reflect changes in price of retail items

Modelling time-dependent data can result in changes to entities, attributes and relationships.

One technique is to store a series of time stamped data values.

These values can either be represented as repeating data or as an additional entity called PRICE HISTORY.

Relationship cardinality can change.

Some entities can be generalised (or specialised) to form other entities.

An entity subtype is made up from some of the instances of the entity.

This is useful when one sub-type participates in a specific relationship not shared by other sub-types of the same entity.

The entity types:
- Motor Car
- Truck
- Train

...can be grouped together to form the generalised entity supertype
- Transport Vehicle

Example entity sub-type and super-type:
- the entity super-type EMPLOYEE includes the subtype SALESPERSON.
Entity subtypes are included in the ER model only when they are of use - they may participate in relationships and have additional attributes in their own right.

Entity subtypes may have multiple subtypes.
Entity subtypes may be nested.

Multiple entity subtypes should be:
- non-overlapping (disjoint)
- collectively exhaustive

This enables easier translation to a relational design.

Identify and list the major entities in the system.
Represent the entities by named rectangles.
Identify, draw, name, and quantify relationships.
Indicate mandatory optional nature of relationships.
Revise for entity subtypes where appropriate.

Fact-finding and information gathering techniques are used to determine the entities and relationships. Identify both existing and new information. Existing documents are particularly useful - forms, paper-based and computer files, reports, listings, data manuals, data dictionary. Existing and new business rules for information are often difficult to elicit from documents - it is essential to speak directly to the client.
ER Modelling Difficulties

- Is a given object an entity or relationship?
- Are two similar objects one entity or two?
- Is a given object an entity or an attribute of (data item about) an entity?
  - EMPLOYEE vs EMPLOYEE SPOUSE
- Do we need to store data about the object?
- What is the ‘best’ data model?

Quality Dimensions

- Correctness
- Completeness
- Understandability
- Simplicity
- Flexibility

ER Models and DFDs

- Do not to confuse entities with sources/sinks or relationships with data flows

TREASURER ACCOUNT EXPENSE EXPENSE REPORT

TREASURER is the person entering data; there is only one person and hence it is not an entity type
ACCOUNT has many account balance instances
EXPENSE has many expense transactions
EXPENSE REPORT contents are already in ACCOUNT and EXPENSE - it is not an entity type

Integrating ER Models and DFDs

All data elements represented in a DFD (data flows and data stores) MUST correspond to entities and their attributes in the ER model

Sales orders are made up of
- Order
- Line
- Product

Check sales order

Produced weekly based on

Order placed by

References

