IMS1002 – Data Modelling

**Lecture 3**

**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
- Helps eliminate problems of data inconsistency and duplication across the organisation

**Conceptual Data Modelling**

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

**Entity Relationship Modelling**

- Used for conceptual data modelling
- 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
Entity Relationship 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.

Entity Relationship Modelling

- Data “objects” or entities are things about which we wish to store information.
- ER models show the major data objects and the associations between them.
- ER models are useful in the initiation, analysis and design phases.

Entity

- Something of interest about which we store information.
  eg. EMPLOYEE, SALES ORDER, SUPPLIER.
- Often identified from nouns used within the business application.
- Should be LOGICAL (not physical).

Identifying Entities

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

   CUSTOMER  PART-TIME EMPLOYEE

Entity Types and Instances

- An entity type is a classification of entity instances.
  eg BN Holdings
  ABC Engineering
  Acme Corp. Ltd.
  SUPPLIER
Entity Types are Logical

- E.g. 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

  which could be represented by one logical entity PRODUCT

Entity Types are Logical

- E.g. 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.

Relationship

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

Representing Relationships

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

  SUPPLIER supplies ITEM

Relationship Types and Instances

- A relationship type is a classification of relationship instances

  Marketing employs Sue Black
  Finance employs Bill Brown
  MIS employs John Smith
  DEPT employs EMPLOYEE

Cardinalities in Relationships

- The cardinality of a relationship is the number of instances of one entity type that may be associated with each instance of the other entity type
  
  eg a CUSTOMER may place many ORDERs
  an ORDER is placed by one CUSTOMER
  an ITEM can appear on many ORDERs
Examples of Cardinalities

**One to One**
- **EMPLOYEE**
  - led by
  - leads
- **PROJECT**

**One to Many**
- **CUSTOMER**
  - placed by
  - places
- **SALES ORDER**
- **ITEM**

**Many to Many**
- **SUPPLIER**
  - supplied by
  - supplies
- **CUSTOMER**
  - placed by
- **SALES ORDER**

Nature of Relationships

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

![Diagram of relationships](image)

Notations

- **EMPLOYEE**
  - attends
  - attends
- **COURSE**

- **EMPLOYEE**
  - Is attended by
  - attends
- **COURSE**

Relationship Degree

- 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)

Unary relationships

- A unary relationship is a relationship between instances of one entity type (also called a recursive relationship)

![Diagram of unary relationships](image)
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.

![Movie has copy of video tape](image)

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.

![Movie has copy of video tape](image)

Ternary relationships

Triplets e.g.
- Mary uses COBOL on HR Project

Example ER model

Multiple Relationships

- It is common to have two or more relationships between the same entities.
- They represent different business rules.

![Example ER model](image)

Associative Entities (Gerunds)

- An associative entity (or gerund) is a relationship that a data modeller decides to model as an entity type
- As both entities and relationships can have attributes, this is possible

![Associative Entities (Gerunds)](image)
Modelling Time-dependent Data

- Some data values vary over time and it may be important to store a history of data values to understand trends and for forecasting. E.g., 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.

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

Entity subtypes and supertypes

- 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.

- Example entity subtype:

  the entity type EMPLOYEE includes the subtype SALESPERSON

Entity Subtypes

- Entity subtypes are included in the ER model only when they are of use - they may participate in relationships and have additional attributes.
Multiple entity subtypes

- Entity types may have multiple subtypes
- Entity subtypes may be nested

Entity Subtypes

Multiple entity subtypes should be
- Non-overlapping (disjoint)
- Collectively exhaustive
  This enables easier translation to a relational design

Building a Basic ER Model

- 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

Example ER model

- Airline ticketing model

Eliciting Information for an ER Model

- 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 e.g. 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?
  e.g. EMPLOYEE and 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 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

Integration of ER Models with DFDs

- All data elements represented in data flow diagrams for a system (data flows and data stores) **MUST** correspond to entities and their attributes in the ER model

References

Barker, R. (1989) CASE*METHOD Entity Relationship Modelling, Addison-Wesley, Wokingham UK. Chapters 4, 5
