Overview

Unit 6 is organized into five sections of which this overview and introduction are the first. The second section deals with the introductory concepts of data modelling. The third section is about the data modelling tool of Entity-Relationship (E-R) models. The fourth section lists step-by-step the process of constructing an E-R data model. The fifth section is the summary and a case study that demonstrates a data model for Orient-Pacific Insurance Corporation. A list of references, a glossary of terms and answers to the self-test close this last section.
Introduction

Information is the life-blood of nearly all organizations. As such, the data required by firms must be organized and logically documented in a manner that is flexible and supportive of the business processes. Data modelling is a tool to help us document and analyse the data. More specifically, we use Entity-Relationship diagrams, which are the most widely accepted modelling methodology, to represent data and show the relationships among them.

Comparatively, if we examine process models with data models, we can observe that the data model is less likely to change than processes. Another way to look at this issue is to compare a manual or paper-based organization to one that employs computer automation. Both organizations use the same data but the processes that operate on the corresponding data are different.

You should recall that in Unit 5 we looked into process models that describe the overall system functions and processes. In this Unit 6, we now examine more closely the ‘static’ elements of data and information that make up the data stores. A resultant data model for an organization is represented by an Entity-Relationship (E-R) diagram. The E-R data model will ultimately be used as a basis from which databases within an organization can be built.
Objectives

By the end of Unit 6, learners should be able to:

1. Define data modelling and its significance.
2. Relate data modelling and process modelling techniques.
3. Explain the benefits of data modelling.
4. Construct an Entity-Relationship (E-R) diagram.
5. Analyse a situation from a data perspective for constructing a data model.
6. Record the details of the data analysis to the E-R diagram.
Concept of data modelling

In the previous unit, we learned how the data flow diagram has helped us to define and understand the processes (data in motion) that occur within a system. It is data in usable form, information, that keeps organizational processes and the knowledge workers in motion. Process modelling represents one major but important aspect of systems analysis. In data modelling, we now need to identify and model the static data (data in storage) which serve as inputs and outputs to processes.

In modern systems analysis, we need to go beyond identifying processes that exist or need to exist, and to look at the data requirements of an organization. This is what data modelling is all about. Additionally, data modelling is seen at least as important as process modelling. Further, like process modelling, data modelling is enterprise-wide in scope, and it is likely to involve workgroups and be across departments.

Benefits of data modelling

Why is it that data modelling has become so important? Essentially, by restating the question to what are the benefits of data modelling, we can derive the answer:

1. Data models allow the analyst to identify business components or entities in greater clarity than DFDs because it is easier to specify a static view of data than a dynamic view of processes.

2. Data models, because of their factual or declarative nature, are usually easily identifiable. In contrast, a considerable effort is needed to identify a hierarchy of processes in process models.

3. Data models are typically a single layered diagram, so they are generally easier to assemble and read than multi-page DFDs.

4. Data models are easily prepared for existing and planned systems. These models can be easily ‘integrated’ to present an organization’s view of data.

Reading

‘An introduction to systems modelling,’ pp. 257–259 in your text. This reading introduces the relationship between systems modelling and data modelling.

‘Data modelling during systems analysis,’ pp. 187–188 in your text. This reading discusses the value of data modelling.
Entity-Relationship (E-R) Diagrams

Just as a DFD is a prime tool for process modelling, the Entity-Relationship model has been popularized as a principal tool for data modelling. It has the same characteristics as a DFD in that it allows you to clearly express your analysis in a non-redundant way that can be understood by both users and systems developers. An E-R Diagram is a map of data stored in an information system. It is closely related to data stores in a DFD.

Data modelling based on an E-R diagram uses the following abstractions to describe data:

- entities, which represent objects of interest to an organization
- relationships, which represent interactions between entities
- attributes, which are details or descriptions of entities and relationships (e.g. data dictionary information for a data store in DFDs).

The first two abstractions, entity and relationships represent the two basic diagramming constructs for an E-R diagram. Entities are typically diagrammed as a rectangle (traditional) or rounded box (Martin-style) — see Figure 6.1. Relationships are drawn as diamonds (traditional) or as a rectangle with a diamond outline embedded within — see Figure 6.2. These two components are joined (or related) by lines. Some symbols where the lines connect to the entities or relationships indicate their cardinality or numerical ratio of relationships. Though E-R diagrams are widely accepted by systems analysts and database designers, there exist some variations in their diagramming notations. We will use the Martin (information engineering) notation that is used in your textbook. Additionally two classifications of E-R diagrams will be used, one that does not specify the attribute details (E-R context diagram), and one that does specify the attribute details (E-R key-based diagram).

![Figure 6.1 Entity diagram construct](image1)

![Figure 6.2 Relationship diagram construct](image2)
Entity construct

An entity represents an object of interest to an organization. For example, a trading organization has customers about whom it wants to record data (attributes include name, address, and credit limit). This object, customer, is reflected in an E-R Diagram as an entity, CUSTOMER. This entity is exactly the same as a customer data store in a DFD. Another object of interest to a trading company is product (product name, location, size). This is an entity, PRODUCT, where data about each product is stored.

Entities can be declared for the following data abstractions and some examples of them are shown below:

- Persons: student, customer, spouse, and employee.
- Places: building, sales district, and department.
- Objects: part, computer system and motor vehicle.
- Events: invoice, flight, and accident.
- Concepts: course, bank account, and insurance policy.

Relationship construct

A relationship represents an interaction between entities. It is a logical association. For example, if a customer orders a product, a relationship ORDER is created to store data about that order. It relates CUSTOMER and PRODUCT entities. It records data such as date, amount ordered, price. An E-R Diagram for this relationship is shown in Figure 6.3.

```
CUSTOMER --ORDER-- PRODUCT
```

The naming convention used for entities and relationships is such that entity names are typically nouns while relationship names are typically verbs. A minor difficulty with E-R Diagrams is that the names of relationships are uni-directional. A CUSTOMER ORDERs a PRODUCT is fine, but a PRODUCT ORDERs a CUSTOMER makes no sense. Such modelling ambiguities in naming are deemed acceptable as the person reading the E-R diagram can usually tell the direction from the semantics.

There may be more than one relationship between two entities. For example, relationships INVOICE and DELIVERY both involve data about events that happen between CUSTOMER and PRODUCT. In both cases, the data used...
are different from the data about ORDER. Figure 6.4 illustrates two relationships between two entities.

![Figure 6.4 Sample E-R context diagram showing two relationships](image)

Clearly there are also entities that are not related to each other at all; for example, CUSTOMER and SUPPLIER. Thus an enterprise E-R diagram can have several non-linked sub-diagrams within it.

Another problem with E-R diagrams is that it may sometimes be difficult to distinguish between entities and relationships. For example, when CUSTOMERs ORDER PRODUCTs, they do so on an order form. An order form is a physical ‘object of interest to an organization’ — it is likely to be modelled as an entity! Similarly, relationships INVOICE and DELIVERY exist physically in form of an invoice and a delivery note; they are likely to be modelled as entities. Figure 6.5 shows an alternative view of the relationships shown in Figure 6.4. In this figure the relationships have effectively been changed from events to associations.

![Figure 6.5 Alternate sample E-R context diagram](image)

Neither of these views is ‘right’ or ‘wrong’. They are just different views of the same situation. As a systems analyst or data modeller, you decide which view is most appropriate for your current system. There is not necessarily a ‘right’ answer! This is one of the difficulties of systems analysis.

**Attributes, keys and instances**

Since an entity or relationship is a collection of data items about an object of interest to an organization, the attribute is a more formal data modelling term for these data items or fields. CUSTOMER-NAME is an attribute of
CUSTOMER entity, just as it is a field on a customer file or database. QUANTITY is an attribute of ORDER event relationship (or ORDER entity!), just as it is a field on an order file. See Figure 6.6 for attribute details in an E-R key-based diagram.

![Figure 6.6 Components of E-R key-based diagram](image)

In physical information systems, entities and relationships become files or databases. Each record on a file contains data about one particular, real-life object. In data modelling a record is called an **instance**. Customer Ken Leung is an instance of CUSTOMER. Figure 6.7 shows an example.

![Figure 6.7 An example of customer entity and an instance](image)

Every entity and relationship must have a unique **key** or **primary key (PK)**. A primary key uniquely identifies every instance. Without such a key, you cannot identify one CUSTOMER from another or one ORDER from another. Note that primary keys can be composed of more than one attribute, as in a relationship — this is sometimes called a **composite key**. Additionally an entity or relationship can have more than one primary key, in which case one is chosen as the primary key (by definition) and all others are classified as **alternate keys**.

A reason for having separate data modelling jargon is to keep analysis distinct from design, to keep analysis logical not physical. It is quite likely that entities will not be implemented as files/DBs exactly as you analyse them. They are designed to fit physical constraints.

Note that two types of E-R diagrams are classified — context and key-based diagrams. The difference is that in the context-based E-R diagram, relationships are not really shown but **cardinality** is presented. In a key-based E-R diagram, relationships and all the attributes are specified including primary keys.
Notice that entities are ‘loosely coupled’ and ‘highly cohesive’:

- loosely coupled because entities have relationships with many other entities in a data model. For example, a CUSTOMER is related to ORDER and INVOICE (if modelled as entities).

- highly cohesive because all attributes of an entity are in one place, not scattered over a number of places. Furthermore, all attributes in an entity have to do only with that object — there are no attributes about anything else stored in a particular entity. For example, a CUSTOMER entity refers to attributes of CUSTOMER and nothing else. If a customer changes address, you capture that change in the entity CUSTOMER; however, if a customer places an order, nothing about customer object has changed and no data in CUSTOMER entity should change.

Like entities, relationships are highly cohesive. All attributes of a relationship are stored in that particular relationship. However, unlike entities, relationships are tightly coupled. They point specifically to key attributes of two related entities, for example, CUSTOMER-ID and PRODUCT-NO on an ORDER relationship between entities CUSTOMER and PRODUCT respectively.

**Cardinality**

Entities are related to each other not only by name but by number as well. For example, a CUSTOMER has many ORDERs and an ORDER has many PRODUCTS. Your data model must also describe these relationships.

Cardinality is the term used to describe the numerical aspect of relationships. One-to-one, one-to-many or many-to-many relationships are possible. Please note the following cardinality notation used in Martin E-R diagrams in Figure 6.8.
Cardinality is the term used to describe the numerical aspect of relationships. One-to-one, one-to-many or many-to-many relationships are possible.

A one-to-one relationship implies that attributes of one entity have exactly a one-to-one correspondence with another entity. This is very unusual. It often indicates that your model should regard these entities as the same! If a one-to-one relationship occurs (Figure 6.9), you should determine if there should be only one entity not two (or if two with a one-to-one relationship is correct). See Figure 6.10 for the conversion of the one-to-one relationship of Figure 6.9 into one entity.
A one-to-many relationship implies that one instance of entity A has a relationship with many instances in entity B, and no other relationship the other way around. The relationship implies a foreign key relationship where an attribute in entity B is common with the primary key in entity A. See Figure 6.11 for an example — where attribute DEPT of EMPLOYEE entity is a foreign key to the same attribute that is the key of DEPARTMENT entity.

A many-to-many relationship implies that one instance of entity A can have a relationship with many instances in entity B, and, conversely, one instance of entity B can have a relationship with many instances in entity A. In Figure 6.12 a relationship MEMBER_OF is created with the primary keys of all involved entities (EMPLOYEE and PROJECT) along with relevant attributes.

Since a relationship is supposed to link two entities, there is one occasion where this is not the case. A relationship can exist that links to just one entity, actually it links two instances of the same entity — this is what is called a recursive relationship.
Figure 6.13 is an example of a recursive relationship for PART that states that a part (e.g., engine) is itself a component of other PARTs (e.g. block, cylinders, and pistons).

This relationship form is useful and needed to represent relationships of component structures, and generalization/specialization/classification hierarchies.

![Diagram](image)

**Figure 6.13** Recursive relationship for the concept of parts made of smaller parts

---

**Worked example**

Referring to Figures 6.9 to 6.12, try to explain the E-R diagrams in English.

**Answer**

For Figure 6.9:

- ‘A person is an employee.’
- ‘An employee is a person.’

For Figure 6.10:

- ‘An employee is a person.’

For Figure 6.11:

- ‘A department consists of many employees.’
- ‘An employee can only belong to one department.’

For Figure 6.12:

- ‘A project involves many employees.’
- ‘An employee can be involved in many projects.’
How to construct E-R data models

The following are the general steps involved in developing an E-R model.

Develop a context E-R diagram of the organization:

1. Identify the general entities of the organization. This can be ascertained from a JAD session, reviewing existing computer and paper-based files/databases, and from the data flows/data stores in the organizations DFDs.

2. Determine the relationships between each of these entities. Start from the logical associations that should relate one entity to another.

3. Draw the context E-R diagram of the organization.

Develop a key-based E-R diagram that specifies the details:

4. Identify all the attributes associated with each entity or relationship. This can be ascertained from existing databases and from forms used in the organization.

5. Identify the primary key for each entity or relationship. Additionally, find all related attributes or foreign keys that are shared between entities and relationships. Identify cardinalities between entities as well.

6. Draw the key-based E-R diagram.

**Reading**

‘How to construct data models’, pp. 277–286 in your text. In this reading you will be given a more detailed treatment on the construction of data models.

Additional details related to data model

The above steps to create an enterprise E-R diagram should be sufficient to depict the data used within an organization. In the eventual movement from the logical E-R diagram to one that is more ‘physical’ in the direction of implementation, additional data details need to be specified. Thus modern E-R models include additional documentation that supports the following data aspects:

**Data type and domain**: For each attribute there can be a type (integer, real, character, date, etc.) and a bounded set of values for its validity. This information will be useful for database development and/or system development stages that will build controls to check for valid inputs.

**Data security and privacy**: There may be laws or policies that restrict user access to certain fields or records. It is better to specify this constraint in a
data dictionary where it will be obvious to developers when a new process is created that uses that data.

**Data timing**: Some data characteristics fluctuate markedly over time. As an extreme example, taxation returns peak at one time of the year and is almost absent at other times. This has a great bearing on the way data is handled in systems design.

**Data volume**: Volumes of data also have a great bearing on the way data is handled in systems design. Expected growth in data volumes is also important.

**Data constraints**: Some data elements have a limited range of values or might be related to other elements. For example an element that records sex may only be male or female.

---

**Reading**

‘Looking ahead to systems design’, p. 276 in your text. Table 7.4 presents the additional details that may be called for in the development of fully-specified data models and associated documentation.

---

**Self-test**

1. Why is data modelling important?

2. What is the difference between an enterprise E-R model, E-R context model and E-R key-based model?

3. How do you go about identifying entities and relationships in an organization?

4. What purpose does an E-R model serve once it is completed at this stage?

5. Given the following data attributes and entities, indicate which attributes could be identifiers for each of the entities. Draw a rough draft ER diagram.

**PC maintenance system**

**Entity: equipment**
- serial number
- model number
- manufacturer’s name
- item description
- installed in serial number (if applicable)

**Entity: software package**
- serial number
- package name
- package type
- version number
- registration date
date purchased date purchased
supplier name supplier name
supplier address supplier address
purchase price purchase price
replacement cost one or more of:
current end-user location current end-user
current responsible end-user current location

**Entity: spare part**

internal part number
vendor part number
supplier part number
part description
used in one or more of:
manufacturer’s name
model number

**Entity: warranty**

serial number
rmodel number
manufacturer’s name
manufacturer’s location
item description
date purchased
date warranty expires
warranty service provider’s name
warranty service provider’s address
warranty service provider’s tel. no.

**Entity: maintenance contract**

serial number
model number
manufacturer’s name
item description
maintenance purchase date
maintenance start date
maintenance finish date
labour covered?
parts covered?
Summary

You use systems analysis to discover information systems requirements and to specify them in a way that can be communicated unambiguously to both users and systems developers. Systems analysis comprises of both process analysis and data analysis. Data analysis defines the underlying structure on which processes perform. It is enterprise-wide in nature and is a modelling process.

A conceptual (logical) model of a system has five components. You learned about three of these components in Unit 5: Data Flow Diagrams, process descriptions, and the data dictionary. These are tools of process analysis. They are used to identify and specify processes (or functions or actions) necessary to capture data and to turn that data into information. The fourth component is a data model, in the form of an Entity-Relationship diagram. The fifth component is the network model of data distribution to be discussed in the next unit. These five components are highly integrated. They form documentation that you develop and maintain in the life-cycle of systems analysis. Together, they allow you to represent processes and data that comprise an information system.

If you recall, the tools of process analysis allow you to model business processes. Process modelling is local in nature. Processes are specific to a business area or business function of an organization. In contrast, the tools of data analysis allow you to model business data. Data modelling is enterprise-wide in nature where you try to show how data are shared across functions of an organization. From a bottom-up perspective, data models are later used to define the underlying data structure on which processes operate. You should note that a deeper understanding of data modelling is essential to successful systems analysis. If you need more help in this area, please consult any database textbooks. The results of the data model and analysis discussed here will be directly used in database design that is covered later in Unit 11.

In the Case Study section of Unit 6 (below) you were able to practise using data modelling techniques in a ‘real world’ environment. We should emphasise that the skills required in data analysis come from practice. You should attempt to analyse any system you encounter in using these techniques.

In Units 5 and 6 you have developed the modelling skills necessary to represent processes and data in an information system. Unit 5 discussed process analysis techniques that allow you to specify what the system needs to do. Unit 6 examined the structure of a system in terms of its data. Your next unit, Unit 7, covers the modelling of data, applications, and people in a distributed networked environment that is representative of most organizations today. Units 5 to 7 together form a coherent methodology for systems analysis.
References

You should be able to meet the learning objectives of this course and successfully complete your assessment on the basis of your study units and your textbook. These references are not prescribed reading; they are provided to enable you to develop your knowledge beyond the requirements of this course.

Systems analysis and design


Glossary

This glossary provides brief definitions of the main technical terms you encountered in Unit 6. The definitions given here are offered in addition to, or in place of those found in your textbook.

**Alternate key:** An alternate key is any potential key of an entity that is not selected to become the primary key.

**Attribute:** A field or descriptive property of an entity. Synonyms include data item, data element, field, and property. It represents the smallest piece of data that has meaning to the end-users and the business.

**Cardinality:** A term to describe the numerical aspect of relationships between entities. Possible relationships are one-to-one, one-to-many and many-to-many.

**Composite key:** A group of attributes that uniquely identifies an instance of an entity. Synonyms include compound key and concatenated key.

**Data constraints:** Limited range of values of data elements.

**Data model:** A graphic that defines system data entities and their relationships. The most popular of data modelling techniques is Entity-Relationship diagrams.

**Data security and privacy:** Policies, procedures, and technical measures used to prevent unauthorized access, alteration, theft, or physical damage to data.

**Data timing:** The frequency and the instant that data appears.

**Data volume:** Data volume refers to the size and quantity of data.

**Domain:** Defines what values an attribute can legitimately take on.

**Entity:** A class of persons, places, objects, events, or concepts about which we need to capture and store data.

**E-R modelling:** A data modelling technique that models data as entities and relationships. It also has a corresponding diagram notation.

**Foreign keys:** It is an attribute duplicated in one entity or relationship that is a key of another related entity. A foreign key (always in a child entity) always matches the primary key (in a parent entity).

**Highly cohesive:** Entities are highly cohesive when all attributes of an entity are in one place, and there are no attributes about anything else stored in a particular entity.

**Instance:** Data about one particular, real-life object or event. In a physical system it is represented by a record on a file.

**Key:** Attribute, or group of attributes, that assumes a unique value for each entity instance. It is sometimes called an identifier.
Loosely coupled: Entities are loosely coupled when they have clearly defined relationships with many other entities in a data model.

Many-to-many relationship: One in which many instances of one entity are associated with many instances of another entity.

One-to-one relationship: A one-to-one relationship implies that key of one entity has exactly a one-to-one correspondence with another entity.

One-to-many relationship: A one-to-many relationship implies that an instance of an entity has a relationship with many instances in another entity. Conversely, an instance of the latter can have a relationship with many instances in the former.

Primary key: Attribute(s) that uniquely identifies instances of an entity (or relationship).

Recursive relationship: A recursive relationship is a relationship that exists between different instances of the same entity.

Relationship: An interaction between entities. It can be a logical association or an event.
Answer key for self-test questions

1. Data modelling is important because:
   - it allows greater clarity when identifying business components or entities. This helps analysts to get to the enterprise data model more clearly and precisely.
   - it allows modellers to capture data models more quickly, especially from a hierarchy of processes.
   - data models are typically single-layered diagrams, and are easier to assemble and read than multi-page DFDs.
   - data models can be easily ‘integrated’ to present an organization’s view of data.

2. The difference between an enterprise E-R model, E-R context model, and E-R key-based lies in the levels of details they supplied to their readers.
   - An enterprise E-R model shows all the entities and relationships within an organization. It typically identifies only the most fundamental of entities. The entities are not described in terms of keys and attributes.
   - An E-R context model only includes entities and relationships but no attributes.
   - An E-R key-based model eliminates non-specific relationships, adds associative entities, and include primary and alternate keys, as well as foreign keys. The key-based model will also include precise cardinalities.

3. An entity is a class of persons, places, objects, events, or concepts about which we need to capture and store data. It represents an object of interest to an organization.

   A relationship is a natural business association that exists between one or more entities. The relationship may represent an event that links the entities or merely a logical affinity that exists between the entities. In short, it is an interaction between entities.

   It is sometimes difficult to distinguish between entities and relationships. In reality, many physical objects form a relationship between entities. These physical objects should be treated as relationships but they are often seen as ‘objects of interests to an organization’. For example, INVOICE links CUSTOMER and PRODUCT because it involves data about events that happen between the two entities. However, INVOICE exists as a physical form, an invoice, and is likely to be modelled as an entity. Therefore, whether an object is an entity or relationship depends largely on the view of the systems analyst.

4. An E-R model depicts data in terms of the entities and relationships described by the data. Once it is completed, its data can be used to build the databases later in the design and construction phases.
5  Equipment = manufacturer’s name +
    model number +
    serial number

Software package = package name +
    serial number

Spare part = internal part number

Warranty = manufacturer’s name +
    model number

Maintenance contract = manufacturer’s name +
    model name +
    serial number +
    maintenance purchase date
Unit 6

Data modelling
Course Team (Original course)
Project manager: Steve Elliot, OUHK
Course developers: University of Technology, Sydney
Craig McDonald, Consultant
David Wilson, Consultant
Course designer: Shannon Timmers, OUHK

Course Team (Revised course)
Course developers: Ali F Farhoomand, Consultant
Lester W Yee, Consultant
Course designer: Linda Chung, OUHK
Course coordinator: John Mak, OUHK
Member: Nigel Leung, OUHK

External course assessor
Professor N V Balasubramanian, City University of Hong Kong

Production
ETPU Publishing Team

Acknowledgement
The OUHK gratefully acknowledges a grant of $100,000,000 from the Hong Kong Government designated for course development.

B329 Systems Analysis and Design was developed in part using resources provided by this generous grant.

Copyright © The Open University of Hong Kong, 1998.

All rights reserved.
No part of this material may be reproduced in any form by any means without permission in writing from the President, The Open University of Hong Kong,
The Open University of Hong Kong
30 Good Shepherd Street
Ho Man Tin, Kowloon
Hong Kong
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview</td>
<td>1</td>
</tr>
<tr>
<td>Introduction</td>
<td>2</td>
</tr>
<tr>
<td>Objectives</td>
<td>3</td>
</tr>
<tr>
<td><strong>Concept of data modelling</strong></td>
<td>4</td>
</tr>
<tr>
<td>Benefits of data modelling</td>
<td>4</td>
</tr>
<tr>
<td><strong>Entity-Relationship (E-R) Diagrams</strong></td>
<td>5</td>
</tr>
<tr>
<td>Entity construct</td>
<td>6</td>
</tr>
<tr>
<td>Relationship construct</td>
<td>6</td>
</tr>
<tr>
<td>Attributes, keys and instances</td>
<td>7</td>
</tr>
<tr>
<td>Cardinality</td>
<td>9</td>
</tr>
<tr>
<td><strong>How to construct E-R data models</strong></td>
<td>13</td>
</tr>
<tr>
<td>Additional details related to data model</td>
<td>13</td>
</tr>
<tr>
<td><strong>Summary</strong></td>
<td>16</td>
</tr>
<tr>
<td><strong>References</strong></td>
<td>17</td>
</tr>
<tr>
<td><strong>Glossary</strong></td>
<td>18</td>
</tr>
<tr>
<td><strong>Answer key for self-test questions</strong></td>
<td>20</td>
</tr>
</tbody>
</table>