Analysis and Design Models

- These models can be used both by the analysis and design processes.
- They are somewhat abstract and hence not necessarily biased towards implementation.
- They are somewhat concrete because they supply more detail than requirements.
All the three models use a data dictionary — a collection of data objects in the application domain.

- **Entity-Relationship Diagrams (ERD)** — a diagrammatic description of data objects
- **Data Flow Diagrams (DFD)** — indicate the processes involved and show the data flowing between the processes
- **State-Transition Diagrams (STD)** — a behavioral model of the system; captures the system states and potential dynamic changes
Data Object

A data object is a composite entity holding data.

✓ This is different from primitive entities that hold only one value.

✓ This is different from objects in object-oriented approach; data objects do not include associated operations.

Data Object Description

A data object consists of

- a unique identifier
- a set of attributes
- a set of relationships with other data objects

Example - Chair

1) identifier: part # is AZ12876
2) attributes: color is blue, type is "no-arm", ...
3) relationships: "is placed" in a "computer room"
Entity Relationship (ER) Diagram

Notations and semantics of entity-relationship diagrams

Entity (a named object) ................. entity name
Relationship ............................... relationship name
Association (entity/relationship) .....
Cardinality (of relationships between data objects)

One-to-one
- One car has one steering wheel.

One-to-many
- One person could have multiple cars.

Many-to-many
- A rental car may be driven by many people, and a person may rent many different cars.

Modality (of relationships between data objects)

Each data object has a modality for each relationship.
  Two data objects participating in the same relation may have different modalities.

Every modality is either MANDITORY or OPTIONAL.

Example
**Data Dictionary**

**Restaurant**
- **Id:** David’s Pizza Restaurant
- **Attributes:** food items prepared, special for the day, preferred customers list, …
- **Relationships:** <<fill in later>>

**Food**
- **Id:** veggie pizza
- **Attributes:** weight, cooking time, price, …
- **Relationships:** <<fill in later>>

**Customer**
- **Id:** Hai
- **Attributes:** favorite food, money to spend, …
- **Relationships:** <<fill in later>>
**Default notations in ERD diagrams**

If a cardinality is not specified, it is assumed to be 1.

If a modality is not specified, it is assumed to 0 (optional).

Specialized relationships do not have any names.
- Specified by empty diamonds
- See the next example

---

**ERD Example 2 - Car Dealership**

<table>
<thead>
<tr>
<th>Data Dictionary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Dealer</strong></td>
</tr>
<tr>
<td>• Id: Chen’s New BMW Cars - Wuhan</td>
</tr>
<tr>
<td>• Attributes: list of car inventory, list of customers, …</td>
</tr>
<tr>
<td>• Relationships: &lt;&lt;fill in later&gt;&gt;</td>
</tr>
<tr>
<td><strong>Customer</strong></td>
</tr>
<tr>
<td>• Id: TingTing</td>
</tr>
<tr>
<td>• Attributes: type of car to purchase, available money, …</td>
</tr>
<tr>
<td>• Relationships: &lt;&lt;fill in later&gt;&gt;</td>
</tr>
<tr>
<td><strong>Automobile</strong></td>
</tr>
<tr>
<td>• Id: Vehicle Identification Number</td>
</tr>
<tr>
<td>• Attributes: manufacturer, year, model, cost, …</td>
</tr>
<tr>
<td>• Relationships: &lt;&lt;fill in later&gt;&gt;</td>
</tr>
</tbody>
</table>
### ERD - Car Dealership

![ERD Diagram](image)

### ERD Example 3 - Personal Organizer

#### Data Dictionary

<table>
<thead>
<tr>
<th>Component</th>
<th>Attributes</th>
<th>Relationships</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>User</strong></td>
<td>Phone number, ...</td>
<td>Uses phone book</td>
</tr>
<tr>
<td><strong>Personal Organizer</strong></td>
<td>Phone contacts, appointment calendar</td>
<td>Used by user</td>
</tr>
<tr>
<td><strong>Phone contacts</strong></td>
<td>List of &lt;name, phone number, address&gt;</td>
<td>None</td>
</tr>
<tr>
<td><strong>Appointment calendar</strong></td>
<td>List of &lt;date &amp; time, purpose, name&gt;</td>
<td>None</td>
</tr>
</tbody>
</table>
ERD - Personal Organizer

Limitations of ERDs

✓ ERDs do not convey any information about the processes.

✓ ERDs cannot describe complex relationships.

✓ Attributes are part of the data dictionary, but not shown in the diagram

✓ ERDs are useful for data abstractions – mainly used in database modeling
Data Flow Diagram (DFD)

DFD Notation

- Process .............  
- Data store..........  
- External entity ....  
- Data Flow ..........  
- Control .............  
DFD Semantics

External entity
  - is External to the software; not created by the developers
  - interacts with the software, but developer does not necessarily know its behavior

Process
  - represents an activity or a group of activities that are part of the system being developed

DFD Semantics (cont’d)

Data Store
  - is a passive entity used for storing and retrieving data
  - has a format not specified by the DFD

Data Flow
  - represents any movement of data
    1. from one process to another process
    2. from process to or from data store
    3. from process to or from external entity
    4. between two external entities
    5. between an external entity and a data store

Control Flow
  - used by a master process to invoke a slave process
DFD Example 1 - Home Alarm System

A computerized home alarm system includes a front panel. This front panel has a keypad and a display unit. Users can configure the system with passwords. All interactions with the alarm system are done through the front panel. When an intruder enters the home, the sensors (part of the alarm system hardware) identify the intrusion and the alarm system raises an alarm. At the same time, the alarm system automatically dials one or more predefined numbers stored in the configuration.

© Roger Pressman, 1994

Home Alarm System - Data Dictionary

<table>
<thead>
<tr>
<th>Entities</th>
<th>Processes</th>
<th>Data Store</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key pad</td>
<td>User interface</td>
<td>Configuration Data</td>
</tr>
<tr>
<td>Display</td>
<td>Configure the system</td>
<td></td>
</tr>
<tr>
<td>Sensor</td>
<td>Validate the password</td>
<td></td>
</tr>
<tr>
<td>Alarm</td>
<td>Activate / deactivate</td>
<td></td>
</tr>
<tr>
<td>Telephone</td>
<td>Display messages</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Monitor sensors</td>
<td></td>
</tr>
</tbody>
</table>
Hints for Constructing a DFD

1) Begin with data dictionary and continue to add to it during the DFD construction.
2) Start with external entities.
3) Use noun phrases to label data flows and verb phrases to label processes.
4) Don’t focus on error flows.
5) Try to avoid issues of control.
DFD Example 2 - a Library

A library maintains a collection of books. The information about all books is kept in a database. The information about users is kept in another database. A user of the library can borrow a book, return a book and reserve a book. Assume that there is no limit to the number of books a user can borrow. Develop a DFD for this problem.

Library - Data Dictionary

DFD shown only for borrowing books from the library.

<table>
<thead>
<tr>
<th>Entities</th>
<th>Processes</th>
<th>Data Stores</th>
</tr>
</thead>
<tbody>
<tr>
<td>User</td>
<td>Verify user</td>
<td>Users Database</td>
</tr>
<tr>
<td>Librarian</td>
<td>Verify book</td>
<td>Books Database</td>
</tr>
<tr>
<td></td>
<td>Update user details</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Issue book</td>
<td></td>
</tr>
</tbody>
</table>
DFD – Library (Borrow operation)

State Diagrams
State Diagram Terms

What is a state?
- A state represents some condition that takes place during software execution (sometimes only for a short time).

What is a state transition?
- As time goes on a computer (i.e., the software) changes from one state to another. Any change in states is called a state transition.

What is an event?
- An event is something that causes a state transition.

What is an action?
- An action is some operation that takes place while in a state.

State Diagram Notation

[Diagram showing state transition from Hungry to do/eat pizza, with an event labeled 'ate too much']
In the Wuhan cafeteria, a server places food on a tray. The customer must insert a meal card and will have the cost of the meal deducted from the card.
**Detailed Design Example - Calculator**

1. **1st Operand**
   - **entry** / oprnd1 ← 0
   - **do** / display oprnd1
   - **click digit (0 - 9)**
   - **click digit (0 - 9)**

2. **1st Operand More Digits**
   - **entry** / oprnd1 ← oprnd1*10 + digit
   - **do** / display oprnd1
   - **click digit (0 - 9)**

3. **2nd Operand**
   - **entry** / op ← operator; oprnd2 ← 0
   - **do** / display oprnd2
   - **click digit (0 - 9)**
   - **click digit (0 - 9)**

4. **2nd Operand More Digits**
   - **entry** / oprnd2 ← oprnd2*10 + digit
   - **do** / display oprnd2

5. **Calculate Result**
   - **entry** / result ← oprnd1 <op> oprnd2
   - **oprnd1 ← 0**
   - **do** / display result
   - **click =**
   - **click digit (0 - 9)**
   - **click C**
   - **click C**
   - **click C**

**State Diagram**

Unlike ERDs and DFDs a state diagram captures the passage of time.

State diagram focuses on control flow, not data -- good for time sensitive software, not good for database-centered software.

State diagrams can be too close to implementation if too much detail is included.