Analysis Modeling
Structured Analysis

Objectives of the Analysis Model

1. To describe what the customer requires
2. To establish a basis for creation of a software design
3. To define a set of requirements that can be validated once the software is built

Structure of the Analysis Model
Structure of the Analysis Model (Contd)

Data dictionary
- core of the model
- repository that contains descriptions of all data objects consumed or produced by the software

Data Dictionary

- Name - the primary name of the data or control item, the data store or an external entity
- Alias - other names used for the first entry
- Where used/how used - a listing of the processes that use the data or control item and how it is used (e.g., input to the process, output from the process, as a store, as an external entity
- Content description - a notation for representing content
- Supplementary information - other information about data types, preset values (if known), restrictions or limitations, and so forth

Example of Data Dictionary Entry

<table>
<thead>
<tr>
<th>Name:</th>
<th>telephone number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aliases:</td>
<td>none</td>
</tr>
<tr>
<td>Where used/ how used:</td>
<td>assess against set-up (output)</td>
</tr>
<tr>
<td></td>
<td>dial phone (output)</td>
</tr>
<tr>
<td>Description:</td>
<td>telephone number = [local number</td>
</tr>
<tr>
<td></td>
<td>local number = prefix + access number</td>
</tr>
<tr>
<td></td>
<td>long distance number = 1 + area code + local number</td>
</tr>
<tr>
<td></td>
<td>area code = [800</td>
</tr>
<tr>
<td></td>
<td>prefix = “a three digit number that never starts with 0 or 1”</td>
</tr>
<tr>
<td></td>
<td>access number = “any four number string”</td>
</tr>
</tbody>
</table>
Structure of the Analysis Model (Contd)

Entity-Relationship Diagram (ERD)
- Data modeling
- Relationship between data objects
- Attributes of each data object can be described using a data object description

Data Flow Diagram (DFD)
- Functional modeling and information flow
- Indicates how data are transformed as they move through the system
- Depicts the functions that transform the data flow
- Each function description is contained in a process specification

State – Transition Diagram (STD)
- Behavioral modeling
- Indicates how the system behaves as a consequence of external events
- Additional information about control aspects of the software is contained in the control specification
Data Modeling - ERD

- What are the primary data objects to be processed by the system?
- What is the composition of each data object and what attributes describe the object?
- Where do the objects currently reside?
- What is the relationship between each object and other objects?
- What is the relationship between the objects and the processes that transform them?

Data Modeling - ERD

- Defines all data that are input, stored, transformed, and produced within an application.
- Especially useful for applications in which data and relationships that govern data are complex.
- Unlike DFD, ERD considers data independently of the processing that transforms the data.

Data Objects, Attributes, & Relationships

- Data object – representation of almost any composite information that must be understood by software.
- Composite information – number of different properties or attributes.
- Data object description incorporates the data object and all of its attributes.
- Relationships are always defined by the context of the problem being analyzed.
Data Objects, Attributes, & Relationships

Objects:
- Name
- Address
- Age
- Driver’s license number

Attributes:
- Make
- Model
- ID number
- Body type
- Color

Relationships:
- Owns

---

Data Objects

- External entity – anything that produces or consumes information
  - Thing (e.g., car)
  - Occurrence (e.g., telephone call)
  - Event (e.g., alarm)
  - Role (e.g., student)
  - Organizational unit (e.g., Dept. CSEE)
  - Place (e.g., ESB)
  - Structure (e.g., students file)

---

Data Objects

- Data objects are frequently represented as a table

<table>
<thead>
<tr>
<th>Make</th>
<th>Model</th>
<th>ID#</th>
<th>Body type</th>
<th>Color</th>
<th>Owner</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lexus</td>
<td>LS400</td>
<td>AB123</td>
<td>Sedan</td>
<td>White</td>
<td>RSP</td>
</tr>
<tr>
<td>Volkswagen</td>
<td>Beetle</td>
<td>X456</td>
<td>Sports</td>
<td>Pink</td>
<td>VP</td>
</tr>
<tr>
<td>BMW</td>
<td>750L</td>
<td>XZ765</td>
<td>Coupe</td>
<td>Red</td>
<td>GG</td>
</tr>
<tr>
<td>Ford</td>
<td>Taurus</td>
<td>Q12A5</td>
<td>Sedan</td>
<td>White</td>
<td>MDG</td>
</tr>
</tbody>
</table>
Attributes

- Define properties of a data object
- Take on one of three different characteristics
  - Name an instance of the data object
  - Describe the instance
  - Make reference to another instance in another table
- One or more attributes must be defined as an identifier which becomes key when we want to find an instance of the data object
- The set of attributes is determined through understanding of the problem context

Relationships

![Relationship Diagram]

Cardinality

- Cardinality is the specification of the number of occurrences of one object that can be related to the number of occurrences of another object
  - “one” or “many”
- One-to-one (1:1)
- One-to-many (1:N)
- Many-to-many (M:N)
Modality

- Modality is
  - 1 if an occurrence of the relationship is mandatory
  - 0 if there is no explicit need for the relationship to occur or the relationship is optional

Cardinality and Modality

Each faculty member advises many students, each student has only one advisor.

Every faculty member may not be advisor, each student must have an advisor.

Example - ERD
Example - ERD

- Manufacturer
- Shipper
- Dealership
- Car

Functional Modeling and Information Flow - DFD

- External entity: Producer or consumer of information that resides outside the bounds of the system to be modeled
- Process: A transformer of information (a function) that resides within the bounds of the system to be modeled
- Data store: A repository of data that is to be stored for use by one or more processes; May be as simple as a buffer or queue or as sophisticated as a relational database
Functional Modeling and Information Flow - DFD

- DFD – graphical technique that depicts information flow and the transforms that are applied as data move from input to output
- Can be used at any level of abstraction
  - A level 0 DFD, also called a fundamental system model or context diagrams represents the entire software system as a single bubble with input and output data indicated by incoming and outgoing arrows respectively
  - A level 1 DFD might contain five or six bubbles with interconnecting arrows; each of the processes represented at level 1 are subfunctions of the overall system depicted in the context model

Example 1 - DFD

Example 2 – DFD: First Refinement
Example 2 – DFD: Second Refinement

Functional Modeling and Information Flow - DFD

- DFD is not sufficient to describe requirements
  - What is the content of the data implied by the arrow or depicted by the store?
    Described in data dictionary
  - What are the processing details implied by a bubble within a DFD?
    Described in processing specification

Behavioral Modeling - STD
**Behavioral Modeling - STD**

- STD – represents the behavior of a system by depicting its states and the events that cause system to change state.
- STD also indicates the actions (e.g., process activation) taken as a consequence of a particular event.
- STD does not show the flow of data within the system.
- State – any observable mode of behavior; STD indicates how the system moves from state to state.

**Relationship between Functional and Behavioral Models**

- DFD
- Process spec
- Data input
- Data output
- Process activators
- Control spec
- Control input
- Control output
- Data conditions

**Example – STD for Photocopier Software**

- Event that causes transition to occur
- Action that occurs as a consequence of the event
- Event that causes transition to occur
SafeHome Security System Example

SafeHome software enables the homeowner to configure the security system when it is installed, monitors all sensors connected to the security system, and interacts with the homeowner through a keypad and function keys contained in the SafeHome control panel.

During installation, the SafeHome control panel is used to "program" and configure the system. Each sensor is assigned a number and type, a master password is programmed for arming and disarming the system, and telephone number(s) are input for dialing when sensor event occurs.

When a sensor event is recognized, the software invokes an audible alarm attached to the system. After a delay time that is specified by the homeowner during the system configuration activities, the software dials a telephone number of a monitoring service, provides information about the location reporting the nature of the event that has been detected. The number will be redialed every 20 seconds until telephone connection is obtained.

All interaction with SafeHome is managed by a user-interaction subsystem that reads input provided through the keypad and function keys, displays prompting messages and system status on the LCD display.