Universitas Esa Unggul

Smart, Creative and Entrepreneurial

CCA220-Analisis dan Perancangan system Informasi

www.esaunggul.ac.id

Dosen Pengampu :

5165-Kundang K Juman, Ir, MMSI

Prodi Teknik Informatika dan Sistem Informasi - Fakultas Ilmu Komputer

Chapter 8
Analyzing Systems Using Data Dictionaries

Systems Analysis and Design

Kendall & Kendall

Sixth Edition

Major Topics

- Data dictionary concepts
- Defining data flow
- Defining data structures
- Defining elements
- Defining data stores
- Using the data dictionary
- Data dictionary analysis

Data Dictionary

- Data dictionary is a main method for analyzing the data flows and data stores of data-oriented systems.
- The data dictionary is a reference work of data about data (metadata).
- It collects, coordinates, and confirms what a specific data term means to different people in the organization.

Reasons for Using a Data Dictionary

The data dictionary may be used for the following reasons:

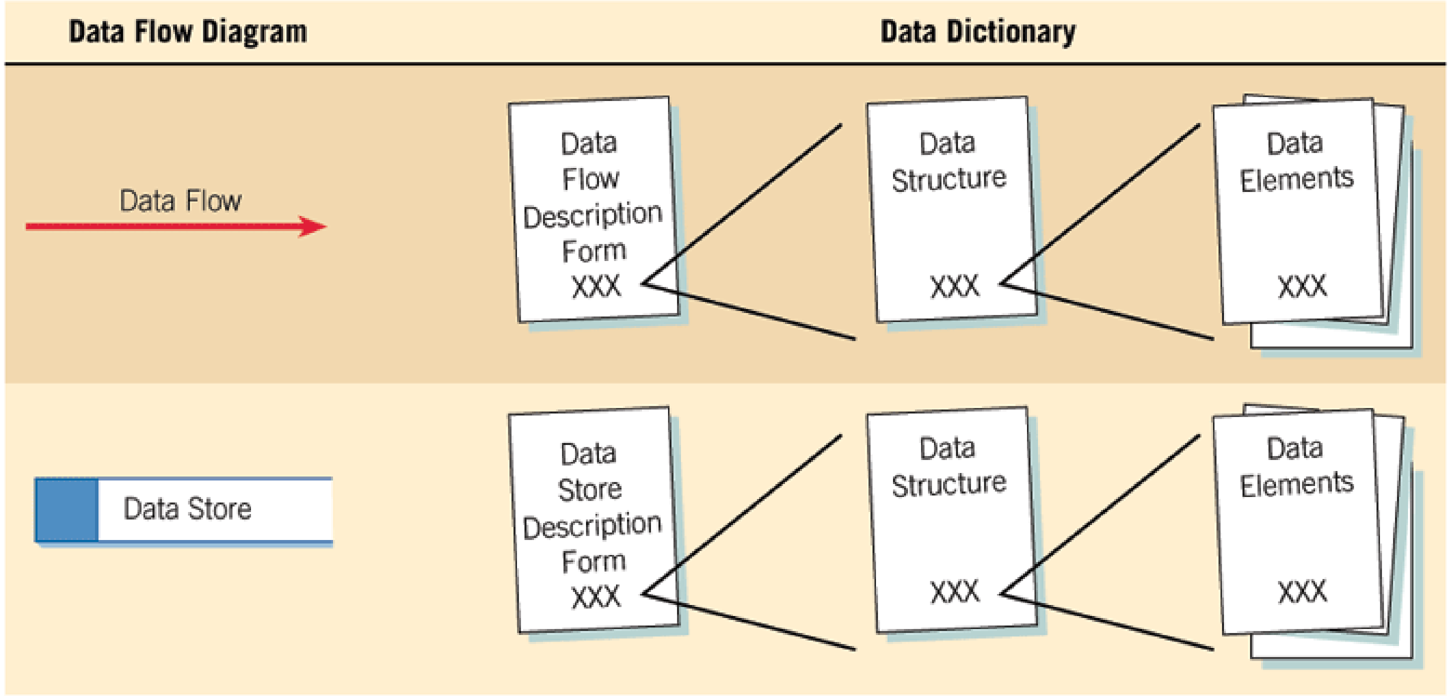
- Provide documentation.
- Eliminate redundancy.
- Validate the data flow diagram.
- Provide a starting point for developing screens and reports.
- To develop the logic for DFD processes.

The Repository

- A data repository is a large collection of project information.
- It includes:
  - Information about system data.
  - Procedural logic.
  - Screen and report design.
  - Relationships between entries.
  - Project requirements and deliverables.
  - Project management information.

Data Dictionary and Data Flow Diagram

Figure 8.1 How data dictionaries relate to data flow diagrams.



Data Dictionary Contents

Data dictionaries contain:

- Data flow.
- Data structures.
- Elements.
- Data stores.

Defining Data Flow

- Each data flow should be defined with descriptive information and its composite structure or elements.
- Include the following information:
  - ID - identification number.
  - Label, the text that should appear on the diagram.
  - A general description of the data flow.

Defining Data Flow (Continued)

- The source of the data flow
  - This could be an external entity, a process, or a data flow coming from a data store.
- The destination of the data flow
- Type of data flow, either:
  - A record entering or leaving a file.
  - Containing a report, form, or screen.
  - Internal - used between processes.

Defining Data Flow (Continued)

- The name of the data structure or elements
- The volume per unit time
  - This could be records per day or any other unit of time.
- An area for further comments and notations about the data flow

Data Flow Example

Name	Customer Order
Description	Contains customer order information and is used to update the customer master and item files and to produce an order record.
Source	Customer External Entity
Destination	Process 1, Add Customer Order
Type	Screen
Data Structure	Order Information
Volume/Time	10/hour
Comments	An order record contains information for one customer order. The order may be received by mail, fax, or by telephone.

Defining Data Structures

- Data structures are a group of smaller structures and elements.
- An algebraic notation is used to represent the data structure.

Algebraic Notation

The symbols used are:

- Equal sign, meaning “consists of”.
- Plus sign, meaning "and”.
- Braces {} meaning repetitive elements, a repeating element or group of elements.
- Brackets [] for an either/or situation.
  - The elements listed inside are mutually exclusive.
- Parentheses () for an optional element.

Repeating Groups

- A repeating group may be:
  - A sub-form.
  - A screen or form table.
  - A program table, matrix, or array.
- There may be one repeating element or several within the group.

Repeating Groups (Continued)

- The repeating group may have:
  - Conditions.
  - A fixed number of repetitions.
  - Upper and lower limits for the number of repetitions.

Physical and Logical Data Structures

- Data structures may be either logical or physical.
- Logical data structures indicate the composition of the data familiar to the user.

Physical Data Structures

- Include elements and information necessary to implement the system
- Additional physical elements include:
  - Key fields used to locate records.
  - Codes to indicate record status.
  - Codes to identify records when multiple record types exist on a single file.
  - A count of repeating group entries.

Data Structure Example

Customer Order = Customer Number +
Customer Name +
Address +
Telephone +
Catalog Number +
Order Date +
{Order Items} +
Merchandise Total +
(Tax) +
Shipping and Handling +
Order Total +
Method of Payment +
(Credit Card Type) +
(Credit Card Number) +
(Expiration Date)

Structural Records

- A structure may consist of elements or smaller structural records.
- These are a group of fields, such as:
  - Customer Name.
  - Address.
  - Telephone.
- Each of these must be further defined until only elements remain.

General Structural Records

- Structural records and elements that are used within many different systems should be given a non-system-specific name, such as street, city, and zip.
- The names do not reflect a functional area.
- This allows the analyst to define them once and use in many different applications.

Structural Record Example

Customer Name = First Name +
(Middle Initial) +
Last Name

Address = Street +
(Apartment) +
City +
State +
Zip +
(Zip Expansion) +
(Country)

Telephone = Area code +
Local number

Defining Elements

- Data elements should be defined with descriptive information, length and type of data information, validation criteria, and default values.
- Each element should be defined once in the data dictionary.

Defining Elements (Continued)

- Attributes of each element are:
  - Element ID. This is an optional entry that allows the analyst to build automated data dictionary entries.
  - The name of the element, descriptive and unique
    - It should be what the element is commonly called in most programs or by the major user of the element.

Defining Elements (Continued)

- Aliases, which are synonyms or other names for the element
- These are names used by different users within different systems
- Example, a Customer Number may be called a:
  - Receivable Account Number.
  - Client Number.

Defining Elements (Continued)

- A short description of the element
- Whether the element is base or derived
  - A base element is one that has been initially keyed into the system.
  - A derived element is one that is created by a process, usually as the result of a calculation or some logic.
- The length of an element

Determining Element Length

What should the element length be?

- Some elements have standard lengths, such as a state abbreviation, zip code, or telephone number.
- For other elements, the length may vary and the analyst and user community must decide the final length.

Determining Element Length (Continued)

- Numeric amount lengths should be determined by figuring the largest number the amount will contain and then allowing room for expansion.
- Totals should be large enough to accommodate the numbers accumulated into them.
- It is often useful to sample historical data to determine a suitable length.

Determining Element Length

Element	Length	Percent of data that will fit within the length
Last Name	11	98%
First Name	18	95%
Company Name	20	95%
Street	18	90%
City	17	99%

Data Truncation

- If the element is too small, the data will be truncated.
- The analyst must decide how this will affect the system outputs.
- If a last name is truncated, mail would usually still be delivered.
- A truncated email address or Web address is not usable.

Data Format

- The type of data, either numeric, date, alphabetic or alphanumeric or other microcomputer formats
- Storage type for numeric data
  - Mainframe: packed, binary, display.
  - Microcomputer (PC) formats.
  - PC formats depend on how the data will be used, such as Currency, Number, or Scientific.

Personal Computer Formats

Bit - A value of 1 or 0, a true/false value

Char, varchar, text - Any alphanumeric character

Datetime, smalldatetime - Alphanumeric data, several formats

Decimal, numeric - Numeric data that is accurate to the least significant digit
Can contain a whole and decimal portion

Float, real - Floating point values that contain an approximate decimal value

Int, smallint, tinyint - Only integer (whole digit) data

Money, smallmoney - Monetary numbers accurate to four decimal places

Binary, varbinary, image - Binary strings (sound, picture, video)

Cursor, timestamp, uniqueidentifier - A value that is always unique
within a database

Defining Elements - Format

- Input and output formats should be included, using coding symbols:
  - Z - Zero suppress.
  - 9 – Number.
  - X – Character.
  - X(8) - 8 characters.
  - . , - Comma, decimal point, hyphen.
- These may translate into masks used to define database fields.

Defining Elements - Validation

- Validation criteria must be defined.
- Elements are either:
  - Discrete, meaning they have fixed values.
    - Discrete elements are verified by checking the values within a program.
    - They may search a table of codes.
  - Continuous, with a smooth range of values.
    - Continuous elements are checked that the data is within limits or ranges.

Defining Elements

- Include any default value the element may have
- The default value is displayed on entry screens
- Reduces the amount of keying
  - Default values on GUI screens
    - Initially display in drop-down lists
    - Are selected when a group of radio buttons are used

Defining Elements (Continued)

- An additional comment or remarks area.
- This might be used to indicate the format of the date, special validation that is required, the check-digit method used, and so on.

Data Element Example

Name	Customer Number
Alias	Client Number
Alias	Receivable Account Number
Description	Uniquely identifies a customer that has made any business transaction within the last five years.
Length	6
Input Format	9(6)
Output Format	9(6)
Default Value	
Continuous/Discrete	Continuous
Type	Numeric
Base or Derived	Derived
Upper Limit	<999999
Lower Limit	>18
Discrete	Value/Meaning
Comments	The customer number must pass a modulus-11 check-digit test.

Defining Data Stores

- Data stores contain a minimal of all base elements as well as many derived elements.
- Data stores are created for each different data entity; that is, each different person, place, or thing being stored.

Defining Data Stores (Continued)

- Data flow base elements are grouped together and a data store is created for each unique group.
- Since a data flow may only show part of the collective data, called the user view, you may have to examine many different data flow structures to arrive at a complete data store description.

Data Store Definition

- The Data Store ID
- The Data Store Name, descriptive and unique
- An Alias for the file
- A short description of the data store
- The file type, either manual or computerized

Data Store Definition (Continued)

- If the file is computerized, the file format designates whether the file is a database file or the format of a traditional flat file.
- The maximum and average number of records on the file
- The growth per year
  - This helps the analyst to predict the amount of disk space required.

Data Store Definition (Continued)

- The data set name specifies the table or file name, if known.
  - In the initial design stages, this may be left blank.
- The data structure should use a name found in the data dictionary.

Data Store Definition - Key Fields

- Primary and secondary keys must be elements (or a combination of elements) found within the data structure.
- Example: Customer Master File
  - Customer Number is the primary key, which should be unique.
  - The Customer Name, Telephone, and Zip Code are secondary keys.

Data Store Example - Part 1

ID	D1
Name	Customer Master
Alias	Client Master
Description	Contains a record for each customer
File Type	Computer
File Format	Database
Record Size	200
Maximum Records	45,000
Average Records	42,000
Percent Growth/Year	6%

Data Store Example - Part 2

Data Set/Table Name	Customer
Copy Member	Custmast
Data Structure	Customer Record
Primary Key	Customer Number
Secondary Keys	Customer Name, Telephone, Zip Code
Comments	The Customer Master file records are copied to a history file and purged if the customer has not purchased an item within the past five years. A customer may be retained even if he or she has not made a purchase by requesting a catalog.

Data Dictionary and Data Flow Diagram Levels

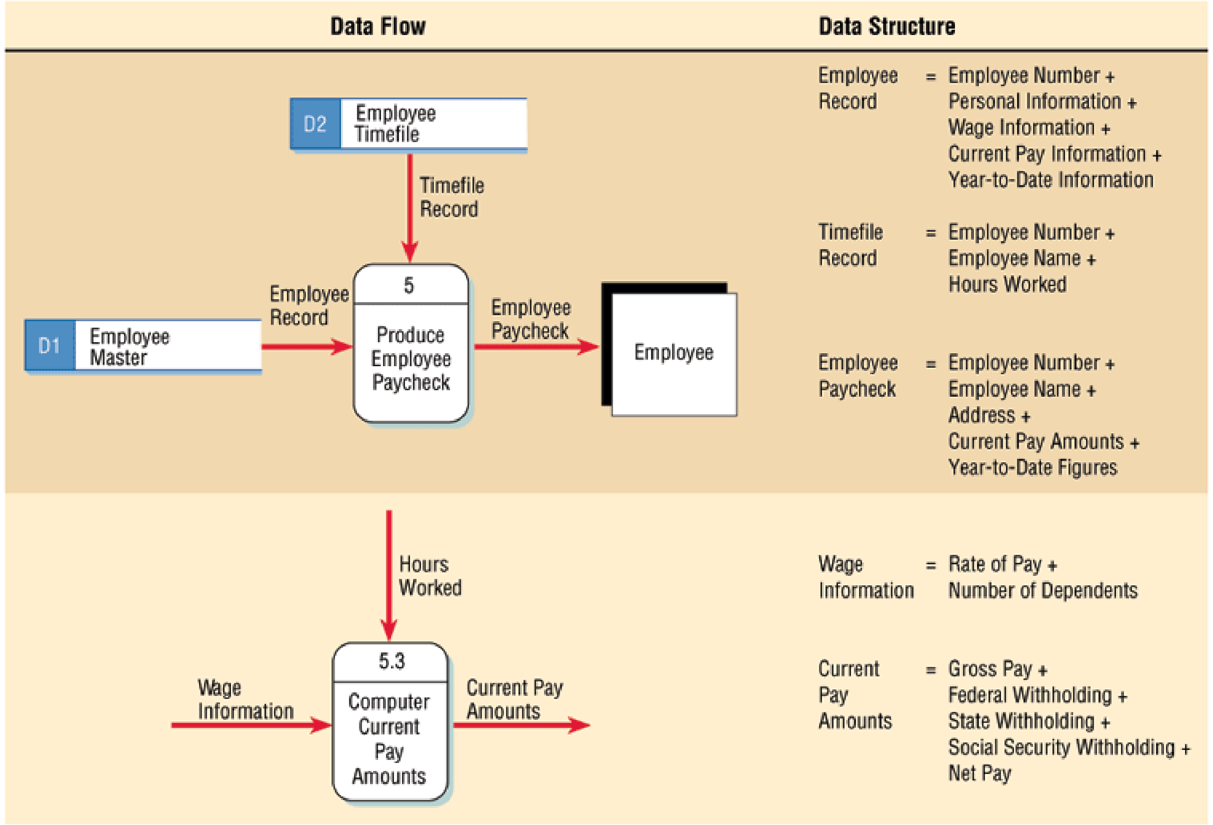
- Data dictionary entries vary according to the level of the corresponding data flow diagram.
- Data dictionaries are created in a top-down manner.
- Data dictionary entries may be used to validate parent and child data flow diagram level balancing.

Data Dictionary and Data Flow Diagram Levels (Continued)

- Whole structures, such as the whole report or screen, are used on the top level of the data flow diagram.
  - Either the context level or diagram zero
- Data structures are used on intermediate-level data flow diagram.
- Elements are used on lower-level data flow diagrams.

Data Dictionary and Data Flow Diagram Levels

Figure 8.13 Two data flow diagrams and corresponding data dictionary entries for producing an employee paycheck.



Data Structure

Employee Record = Employee Number + Personal Information + Wage Information + Current Pay Information + Year-to-Date Information

Timefile Record = Employee Number + Employee Name + Hours Worked

Employee Paycheck = Employee Number + Employee Name + Address + Current Pay Amounts + Year-to-Date Figures

Wage Information = Rate of Pay + Number of Dependents

Current Pay Amounts = Gross Pay + Federal Withholding + State Withholding + Social Security Withholding + Net Pay

Creating Data Dictionaries

1. Information from interviews and JAD sessions is summarized on Input and Output Analysis Forms.
   - This provides a means of summarizing system data and how it is used.
2. Each structure or group of elements is analyzed.

Creating Data Dictionaries (Continued)

- 3. Each element should be analyzed by asking the following questions:
  - Are there many of the field?
    - If the answer is yes, indicate that the field is a repeating field using the { } symbols.
  - Is the element mutually exclusive of another element?
    - If the answer is yes, surround the two fields with the [|] symbols.

Creating Data Dictionaries (Continued)

  - Is the field an optional entry or optionally printed or displayed?
    - If so, surround the field with parenthesis ( ).
- 4. All data entered into the system must be stored.
  - Create one database table or file for each different type of data that must be stored.
  - Add a key field that is unique to each table.

Determining Data Store Contents

- Data stores may be determined by analyzing data flows.
- Each data store should consist of elements on the data flows that are logically related, meaning they describe the same entity.

Maintaining the Data Dictionary

- To have maximum power, the data dictionary should be tied into other programs in the system.
- When an item is updated or deleted from the data dictionary it is automatically updated or deleted from the database.

Using the Data Dictionary

Data dictionaries may be used to:

- Create reports, screens, and forms.
- Generate computer program source code.
- Analyze the system design for completion and to detect design flaws.

Creating Reports, Screens, Forms

To create screens, reports, and forms:

- Use the element definitions to create fields.
- Arrange the fields in an aesthetically pleasing screen, form, or report, using design guidelines and common sense.
- Repeating groups become columns.
- Structural records are grouped together on the screen, report, or form.

Data Dictionary Analysis

- The data dictionary may be used in conjunction with the data flow diagram to analyze the design, detecting flaws and areas that need clarification.

Data Dictionary Analysis (Continued)

- Some considerations for analysis are:
  - All base elements on an output data flow must be present on an input data flow to the process producing the output.
  - Base elements are keyed and should never be created by a process.

Data Dictionary Analysis (Continued)

- A derived element should be output from at least one process that it is not input into.
- The elements that are present on a data flow into or coming from a data store must be contained within the data store.

Extensible Markup Language (XML)

- XML is used to exchange data between businesses.
- An XML document may be transformed into different formats.
- The transformation may limit the data seen by a user.
- XML may be sorted, filtered, and translated.

Using Data Dictionaries to Create XML

- The data dictionary is an ideal starting point for developing XML.
- Data names are stored within tags, a less than and greater than symbol.
- <customer> or <lastName>
- The data dictionary is organized using structures, which are included in XML.

XML Document Type Definition (DTD)

- A DTD is used to ensure that the XML data conforms to the order and type of data specified in the DTD.
- DTD’s may be created using the data dictionary.