Analyzing Systems Using Data Dictionaries
Analyzing Systems
Using Data Dictionaries

SOURCE: Systems Analysis and Design, 9e
Kendall & Kendall, Copyright © 2014 Pearson Education, Inc. Publishing as Prentice Hall
Learning Objectives

- Understand how analysts use of data dictionaries for analyzing data-oriented systems.

- Understand the concept of a repository for analysts’ project information and the role of CASE tools in creating them.

- Create data dictionary entries for data processes, stores, flows, structures, and logical and physical elements of the systems being studied, based on DFDs.

- Recognize the functions of data dictionaries in helping users update and maintain information systems.
Data flow diagrams can be used to catalog:

- Data processes
- Flows
- Stores
- Structures
- Elements

Cataloging takes place with the data dictionary
Major Topics

- The data dictionary
- The data repository
- Defining data flow
- Defining data structures
- Defining data elements
- Defining data stores
- Using the data dictionary
- XML
The Data Dictionary

- A reference work of data about data (metadata)
- Collects and coordinates data terms, and confirms what each term means to different people in the organization
Need for Understanding the Data Dictionary

- Provide documentation
- Eliminate redundancy
- Validate the data flow diagram
- Provide a starting point for developing screens and reports
- Determine the contents of data stored in files
- To develop the logic for DFD processes
- Create XML
The Data Repository

- A data repository is a large collection of project information
- It includes:
  - Information about the data maintained by the system
  - Procedural logic and use cases
  - Screen and report design
  - Data relationships
  - Project requirements and final system deliverables
  - Project management information
How Data Dictionaries Relate to Data Flow Diagrams (Figure 8.1)
Data Dictionary Categories

- Data flows
- Data structures
- Elements
- Data stores
Defining the Data Flow

- ID—identification number
- Unique descriptive name
- A general description of the data flow
- The source of the data flow
- The destination of the data flow
- Type of data flow
- The name of the data structure describing the elements
- The volume per unit time
- An area for further comments and notations
An Example of a Data Flow Description from World’s Trend Catalog Division (Figure 8.3)
Describing Data Structures

- Data structures are made up of smaller structures and elements
- An algebraic notation is used to describe data structures
Algebraic Notation

- Equal sign means “is composed of”
- Plus sign means “and”
- Braces {} mean repetitive elements
- Brackets [] for an either/or situation
- Parentheses () for an optional element
Data Structure Example for Adding a Customer Order at World’s Trend Catalog Division (Figure 8.4)
Structural Records

- A structure may consist of elements or structural records
- These are a group of elements, such as:
  - Customer name
  - Address
  - Telephone
- Each of these must be further defined until they are broken down into their component elements
Structural Records Used in Different Systems

- Structural records and elements that are used within many different systems are 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
Logical and Physical Data Structures

- **Logical:**
  - Show what data the business needs for its day-to-day operations

- **Physical:**
  - Include additional elements necessary for implementing the system
Physical Data Structures

- Key fields used to locate records
- Codes to identify record status
- Transaction codes to identify different record types
- Repeating group entries
- Limits on items in a repeating group
- Password
An Element Description Form Example from World’s Trend Catalog Division (Figure 8.6)
Data Element Characteristics

- Element ID
- The name of the element
- Aliases
- A short description of the element
- Element is base or derived
- Element length
- Type of data
- Input and output formats
- Validation criteria
- Default value
- An additional comment or remark area
Element ID

- Optional entry
- Allows the analyst to build automated data dictionary entries
The Name of the Element

- **Should be:**
  - Descriptive
  - Unique

- **Based on what the element is commonly called in most programs or by the major user of the element**
Aliases

- Synonyms or other names for the element
- Names used by different users in different systems
- A CUSTOMER NUMBER may also be called a RECEIVABLE ACCOUNT NUMBER or a CLIENT NUMBER
Short Description of the Element

- An example might be:
  - Uniquely identifies a customer who has made any business transactions within the last five years
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 a series of decision-making statements.
Element Length

What should the element length be?

- Some elements have standard lengths, state abbreviations, zip codes, or telephone numbers.
- For other elements, the length may vary and the analyst and user community must decide the final length.
Element Length Considerations

- Numeric amount lengths
- Name and address fields
- Other fields
## Name and Address Length

<table>
<thead>
<tr>
<th>Element</th>
<th>Length</th>
<th>Percent of data that will fit (United States)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Last Name</td>
<td>11</td>
<td>98</td>
</tr>
<tr>
<td>First Name</td>
<td>18</td>
<td>95</td>
</tr>
<tr>
<td>Company Name</td>
<td>20</td>
<td>95</td>
</tr>
<tr>
<td>Street</td>
<td>18</td>
<td>90</td>
</tr>
<tr>
<td>City</td>
<td>17</td>
<td>99</td>
</tr>
</tbody>
</table>
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
Type of Data

- **Alphanumeric or text data**

- **Formats**
  - Mainframe: packed, binary, display
  - Microcomputer (PC) formats
  - PC formats, such as Currency, Number, or Scientific, depend on how the data will be used
Some Examples of Data Formats Used in PC Systems (Figure 8.7)

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bit</td>
<td>A value of 1 or 0, a true/false value</td>
</tr>
<tr>
<td>Char, varchar, text</td>
<td>Any alphanumeric character</td>
</tr>
<tr>
<td>Datetime, smalldatetime</td>
<td>Alphanumeric data, several formats</td>
</tr>
<tr>
<td>Decimal, numeric</td>
<td>Numeric data that are accurate to the least significant digit; can contain a whole and decimal portion</td>
</tr>
<tr>
<td>Float, real</td>
<td>Floating-point values that contain an approximate decimal value</td>
</tr>
<tr>
<td>Int, smallint, tinyint</td>
<td>Only integer (whole digit) data</td>
</tr>
<tr>
<td>Currency, money, smallmoney</td>
<td>Monetary numbers accurate to four decimal places</td>
</tr>
<tr>
<td>Binary, varbinary, image</td>
<td>Binary strings (sound, pictures, video)</td>
</tr>
<tr>
<td>Cursor, timestamp, uniqueidentifier</td>
<td>A value that is always unique within a database</td>
</tr>
<tr>
<td>Autonumber</td>
<td>A number that is always incremented by one when a record is added to a database table</td>
</tr>
</tbody>
</table>
## Format Character Codes
(Figure 8.8)

<table>
<thead>
<tr>
<th>Formatting Character</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>May enter or display/print any character</td>
</tr>
<tr>
<td>9</td>
<td>Enter or display only numbers</td>
</tr>
<tr>
<td>Z</td>
<td>Display leading zeros as spaces</td>
</tr>
<tr>
<td>,</td>
<td>Insert commas into a numeric display</td>
</tr>
<tr>
<td>.</td>
<td>Insert a period into a numeric display</td>
</tr>
<tr>
<td>/</td>
<td>Insert slashes into a numeric display</td>
</tr>
<tr>
<td>–</td>
<td>Insert a hyphen into a numeric display</td>
</tr>
<tr>
<td>V</td>
<td>Indicate a decimal position (when the decimal point is not included)</td>
</tr>
</tbody>
</table>
Validation Criteria

- Ensure that accurate data are captured by the system
- Elements are either:
  - Discrete, meaning they have fixed values
  - Continuous, with a smooth range of values
Default Value

- 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
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 Stores

- Data stores are created for each different data entity being stored.

- When data flow base elements are grouped together to form a structural record, a data store is created for each unique structural record.

- Because a given data flow may only show part of the collective data that a structural record contains, many different data flow structures may need to be examined to arrive at a complete data store description.
Describing the Data Store

- The data store ID
- The data store name
- An alias for the table
- A short description of the data store
- The file type
- File format
Describing the Data Store (continued)

- The maximum and average number of records on the file as well as the growth per year
- The file or data set name specifies the file name, if known
- The data structure should use a name found in the data dictionary
- Primary and secondary keys
- Comments
Example of a Data Store Form for World’s Trend Catalog Division (Figure 8.9)
Creating the Data Dictionary

- **Data dictionary entries**
  - Created after the data flow diagram is completed
  - or
  - Created as the data flow diagram is being developed

- **Created using a top-down approach**
Two Data Flow Diagrams and Corresponding Data Dictionary Entries for Producing an Employee Paycheck (Figure 8.11)
Analyzing Input and Output

- A descriptive name for the input or output
- The user contact responsible
- Whether the data is input or output
- The format of the data flow
- Elements indicating the sequence of the data on a report or screen
- A list of elements
An Example of an Input/Output Analysis Form for World’s Trend Catalog Division (Figure 8.12)

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Length</th>
<th>B/D</th>
<th>Edit Criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current Date</td>
<td>6</td>
<td>B</td>
<td>(System Supplied)</td>
</tr>
<tr>
<td>Customer Number</td>
<td>6</td>
<td>B</td>
<td>(Include Check Digit)</td>
</tr>
<tr>
<td>Customer First Name</td>
<td>20</td>
<td>B</td>
<td>Not Spaces</td>
</tr>
<tr>
<td>Customer Last Name</td>
<td>18</td>
<td>B</td>
<td>Not Spaces</td>
</tr>
<tr>
<td>Street</td>
<td>1</td>
<td>B</td>
<td>A through Z or Space</td>
</tr>
<tr>
<td>Apartment</td>
<td>20</td>
<td>B</td>
<td>Not Spaces</td>
</tr>
<tr>
<td>City</td>
<td>20</td>
<td>B</td>
<td>Not Spaces</td>
</tr>
<tr>
<td>State</td>
<td>20</td>
<td>B</td>
<td>Not Spaces</td>
</tr>
<tr>
<td>Zip</td>
<td>5</td>
<td>B</td>
<td>Not Spaces</td>
</tr>
<tr>
<td>Order Number</td>
<td>9</td>
<td>B</td>
<td>Valid State Abbrev.</td>
</tr>
<tr>
<td>Order Date</td>
<td>6</td>
<td>D</td>
<td>Numeric; Last 4 Oct.</td>
</tr>
<tr>
<td>Order Total</td>
<td>8</td>
<td>D</td>
<td>&gt;= 0</td>
</tr>
<tr>
<td>Previous Payment Amount</td>
<td>9</td>
<td>D</td>
<td>Format: 9 (7) V99</td>
</tr>
<tr>
<td>Total Amount Owell</td>
<td>5</td>
<td>D</td>
<td>Format: 9 (7) V99</td>
</tr>
<tr>
<td>Comment</td>
<td>60</td>
<td>B</td>
<td>Format: 9 (7) V99</td>
</tr>
</tbody>
</table>

Comments: Print one page for each customer. If there are more items that fill on a page, continue on a second page.
Developing Data Stores

- Represent data at rest
- Contain information of a permanent or semipermanent (temporary) nature
- When data stores are created for only one report or screen, we refer to them as “user views”
Using the Data Dictionary

- To have maximum power, the data dictionary should be tied into a number of systems programs

- May be used to
  - Create screens, reports, and forms
  - Generate computer language source code
  - Analyze the system design, detecting flaws and areas that need clarification
Create Screens, Reports, and Forms

- Use the element definition and their lengths
- Arrange the elements in a pleasing and functional way using design guidelines and common sense
- Repeating groups become columns
- Structural records are grouped together on the screen, report, or form
Analyze the System Design, Detecting Flaws and Areas that Need Clarification

- All base elements on an output data flow must be present on an input data flow to the process producing the output.
- A derived element should be created by a process and should be output from at least one process into which it is not input.
- The elements that are present in a data flow coming into or going out of a data store must be contained in the data store.
Using Data Dictionaries to Create XML

- XML is used to exchange data between businesses
- XML addresses the problem of sharing data when users have different computer systems and software or different database management systems
- XML documents may be transformed into different output formats
- XML is a way to define, sort, filter, and translate data into a universal data language that can be used by anyone
- XML may be created from databases, a form, software programs, or keyed directly into a document, text editor, or XML entry program
Using Data Dictionaries to Create XML (continued)

- The data dictionary is an ideal starting point for developing XML content
- A standard definition of the data is created using a set of tags that are included before and after each data element or structure
- XML elements may also include attributes
- The XML document tends to mirror the data dictionary structure
Using a Data Dictionary Entry to Develop XML Content: The XML Document Mirrors the Data Dictionary Structure (Figure 8.16)
XML Document Type Definitions

- Used to determine if the XML document content is valid
- DTDs may be created using the data dictionary
- DTD may be used to validate the XML document
A Document Type Definition for the Customer XML Document
(Figure 8.17)
XML Schemas

- A more precise way to define the content of an XML document
- Includes exact number of times an element may occur
- Includes type of data within elements
Summary

- **The data dictionary**
  - A reference work containing data about data
  - Includes all data items from data flow diagrams

- **Repository**
  - A larger collection of project information

- **Defining data structures**

- **Defining elements**
Summary (continued)

- Defining data stores
- Data dictionary entries
- Using the data dictionary
- Data dictionary analysis
- Data dictionary to XML
This work is protected by United States copyright laws and is provided solely for the use of instructors in teaching their courses and assessing student learning. Dissemination or sale of any part of this work (including on the World Wide Web) will destroy the integrity of the work and is not permitted. The work and materials from it should never be made available to students except by instructors using the accompanying text in their classes. All recipients of this work are expected to abide by these restrictions and to honor the intended pedagogical purposes and the needs of other instructors who rely on these materials.

Copyright © 2014 Pearson Education, Inc.
Publishing as Prentice Hall