

DATA INPUT METHODS

LEARNING UNITS

7.1 Data Dictionary : its development and use

7.2 Data input methods : Batch and Interactive

7.3 Coding technique for unique data representation.

LEARNING GOALS

1. The need for a data dictionary for an application
2. How to develop a data dictionary for an application
3. Design of forms and screens for data input.
4. Need and methods of coding data elements.
5. Coding schemes for automatic error detection while inputting data
6. Need for and design of input data validation methods.

MOTIVATION

- During systems analysis it is essential for an analyst to decide the necessary and sufficient data for designing an application.

DFD gives the dataflows and stores of a system

- Individual data elements of dataflows and stores can be catalogued

- Such a catalogue with description of each element and their types will be an invaluable aid while designing a system.

MOTIVATION

- A catalogue will also bring out if any data is duplicated/missed
- A catalogue will also be an invaluable documentation of a system
- Such a catalogue is called Data dictionary-It is actually metadata,i.e.,data about data.

MOTIVATION

- After data dictionary is designed one needs to determine how the data is to be input.
- Data input methods depend on whether the data is filled in by customers in forms manually and later input by data entry operators or data is directly input by users on PC's.
- We thus need to understand both these methods.

MOTIVATION

- Unless data input is correct, results will be unreliable
- Information systems normally have a large volume of data
- Because of large volume special controls are needed to ensure correctness of data input - otherwise it is difficult to find which data is incorrect
- Thus it is important to design appropriate data input methods to prevent errors while entering data
- Key data elements are important to identify records. They need to be unique, concise and understandable by users. Thus we need to study methods of coding key data element

WHAT IS DATA DICTIONARY

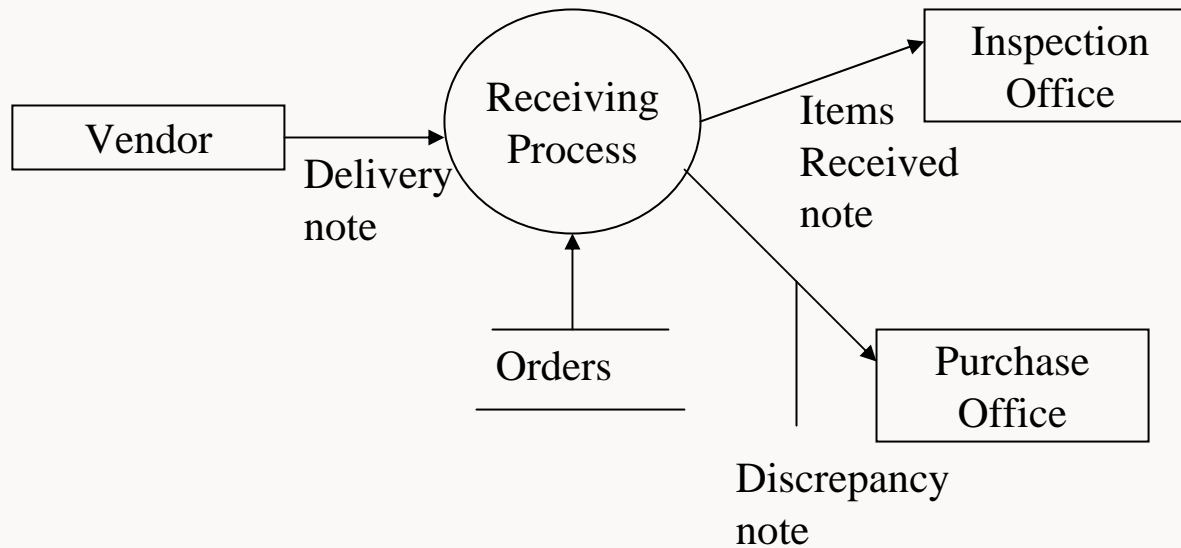
- Data dictionary is a catalogue of all data used in an application, their names, type and their origin.
- In other words it is data about data which is called metadata
- Data dictionary gives a single point reference of data repository of an organization
- It is thus an important documentation which would be useful to maintain a system

HOW IS DATA DICTIONARY DEVELOPED?

- Starting point is DFD

Example :

Consider the Receiving office process described in Module 3.
DFD is reproduced below.



We will recall the word statement from requirement specification now.

WORD STATEMENT OF REQUIREMENTS

- Vendor sends items with a delivery note while fulfilling an order (along with the physical items) to a receiving office.
- Receiving office compares a delivery note against order placed. If there is a discrepancy a discrepancy note is sent to purchase office.
- Actual items received note is sent to the inspection office along with items received.

DATA ELEMENTS IN DATA FLOW

From word statement we derive data elements in each data flow.

1. Delivery note

- Order no, Vendor name, Vendor address, item name, delivery date, quantity supplied, units

Item name and Vendor name may not be unique to ensure uniqueness we assign unique codes for them. Name is however still kept to aid people.

Thus delivery note is:

Delivery note = Order no + Vendor code + Vendor name + Vendor address + item code + item name + delivery date + quantity supplied + units.

DATA ELEMENTS IN DATA FLOW

Discrepancy note : Order no + Vendor code + Vendor name + Vendor address + item code + item name + delivery date + quantity supplied + units + excess/deficiency + no of days late/early.

Items received note = Delivery note

Data in data store

Order records = order no + vendor code + vendor name + vendor address + item code + item name + order date + qty ordered + units + delivery period.

DATA DICTIONARY FORMAT

Data dictionary gives in detail the characteristics of a data element.

Typical characteristics are:

Data name : Should be descriptive and self explanatory. This will help in documentation and maintenance

Data description : What it represents

Origin : Where the data originates

e.g. input from forms, comes from receiving office, keyed in by user etc.

Destination : Where data will flow and will be used (if any)

Data Type : numeric, alphanumeric, letters(or text), binary(0 or 1; True or False), Integer, Decimal fixed point, real(floating point), currency unit, date

TYPICAL CHARACTERISTICS OF DATA ELEMENTS(CONTD)

Length : no of columns needed

Limits on value : (if relevant)

e.g. upper and lower bounds of value (age>0,<100)

Remarks : (if any)

EXAMPLE OF DATA DICTIONARY ENTRY

Name : Order number

Description : Used to identify order given to vendor

Origin : Part of delivery note from vendor

Destination : Receiving process

Data type : Numeric Integer

Length : 8 digits

Limits on value : >000,<=999999999

Actual value not relevant.Used only as unique identifier

Remarks: It is a key field.

EXAMPLE OF DATA DICTIONARY ENTRY(CONTD

Name : Delivery date

Description : Date item is to be delivered

Origin : Part of delivery note from vendor. Is also in orders data store which is input to receiving process

Destination : Receiving process

Data type : Numeric Integer

Length : 8 digits

Limits on value : Date field in the form DDMMYYYY.

Should satisfy constraints of a date in calendar

Remarks: Blank fields not allowed.

e.g.05082004 is ok but not 582004

DATA DICTIONARY USES

- Data dictionary can be enormous in size.Requires careful development. However,it is centralized reference document.
- Invaluable resource to design
 - Input forms and screens
 - Data checking programs
 - Process specification
 - Database
- Very useful in understanding and maintaining system

DATA INPUT METHODS

- **ON-LINE** - User directly Enters data using screen prompts
- **OFF-LINE** -Forms filled by users- for example- candidates for admission to a college fill forms
- Data from forms keyed in by a data entry operator

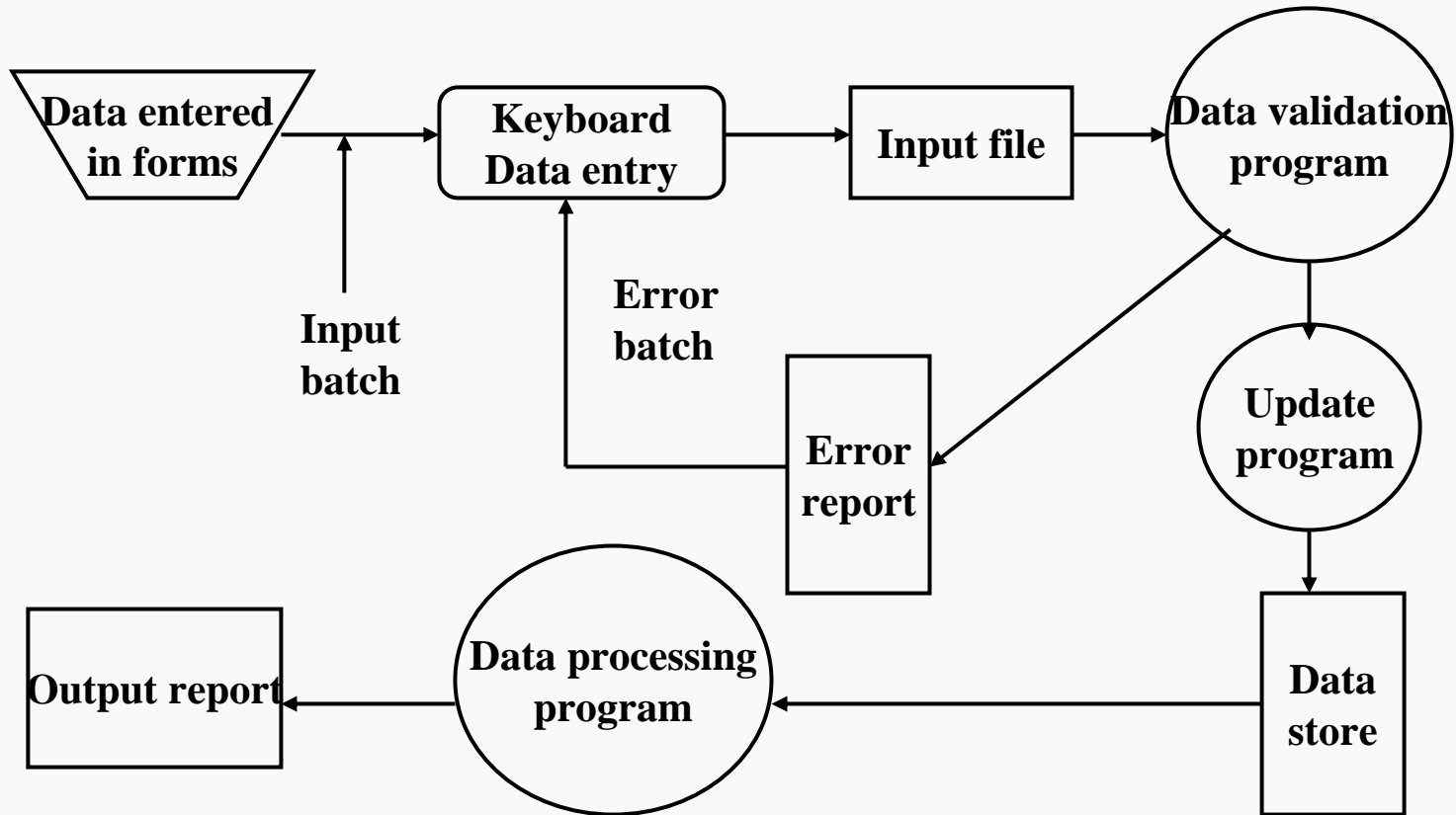
ERROR SOURCES

- Errors in on-line data entry due to poor screen design. System should inform the user immediately when wrong data is input
- Errors in off-line data entry due to bad form design and human errors by users and data entry operator
- System should prevent user making mistakes by
 - Good form design by leaving enough space for writing legibly
 - Clear instructions to fill form
- System should prevent data entry operator making mistakes by
 - Good form design
 - Reducing key strokes
 - Immediate error feedback

OFF LINE DATA ENTRY - PROBLEMS

- Forms batched
- Desirable for the machine to give message when input is wrong. Not always possible
- Error found after elapse of time
- Need good controls to automatically detect and if possible correct errors

BATCH DATA ENTRY



BATCH DATA ENTRY

Name

.....									

Address

.....									
.....									
.....									

Bad design : Tendency will be to fill name on top line. Not enough space for letters of address

Tick as applicable

Individual
Hindu undivided family
Parent/Guardian of minor

Bad design : Choices are not codified. Data entry operator will be confused.

BATCH DATA ENTRY

Enter date

Day	month	Year

(Good design)

Enter date

--	--	--	--	--	--	--	--	--	--

(Bad design)

Enter time

Hr	Min	Sec

(Good design)

Enter time

--	--	--	--	--	--	--	--	--	--

(Bad design)

COMPUTER READABLE FORMS

- As manual data input from forms are slow and expensive attempts have been made to automate form reading using scanners
- Needs hand writing recognition and correct form alignment – Not very successful
- However if forms require just darkening some pre-defined areas they can be machine read and interpreted.
- Example – Multiple choice questions in exams where specific boxes are darkened based on the choice.

INTERACTIVE DATA INPUT

- Advent of PC's and client/server model in computer networks, interactive data input is now widely used
- Advantages are instant response when data is input so that errors are immediately corrected.
- Flexibility in screen design which minimizes manual effort
- Use of mouse and icons simplifies pre-determined choices of data

INTERACTIVE DATA INPUT

Three main models of interactive data input :

- Menus
- Templates
- Commands

MODELS OF DATA INPUT

MENUS

User presented several alternatives and asked to type his/her choice

EXAMPLE

SELECT ALTERNATIVE

Type 1 For entering new student record

Type 2 For deleting student record

Type 3 For changing student record

Your choice

MODELS OF DATA INPUT

TEMPLATE

- Template analogous to form
- Has features to reject incorrect data input using built-in program
- User friendly visual presentation

Example

Roll no

Name

FIRST NAME/INITIALS

LAST NAME

Dept code

CE	CS	ME	EE	IT
----	----	----	----	----

CODES

Year

Hostel code

A	B	C	D
---	---	---	---

Pre-programmed to reject incorrect Roll no,Dept code,Year, Hostel code

MODELS OF DATA INPUT

Interactive commands guides user through alternatives

Example

Computer : Did you request deletion of record ?

Type Y or N

User : Y

Computer : Give student roll no

User : 56743

Computer : Is name of the student A.K.Jain?

Type Y or N

User : Y

Computer : Is he 1st year student

Type Y or N

User : Y

Computer : Shall I delete name?

User : Y

MODELS OF DATA INPUT

- Normally all three models will occur together in application .In other words Menu, Forms and Commands are not mutually exclusive
- Graphical user interface design very rich area-languages such as Visual Basic simplifies design of user interface
- We have given only a flavor of the topic

WHY DO WE NEED CODES?

NEED FOR CODING

- **UNIQUE IDENTIFIER**

 - Example Roll no instead of name

- **CROSS REFERENCING BETWEEN APPLICATIONS**

 - unique Roll no may be used in examination records,accounts, health centre

- **EFFICIENT STORAGE AND RETRIEVAL**

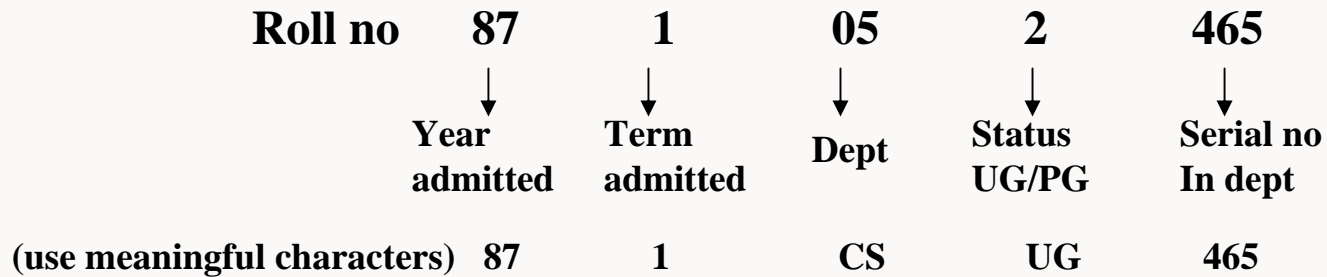
 - Codes concise- a long name will have a shorter roll no

WHAT ARE THE REQUIREMENTS OF A GOOD CODE?

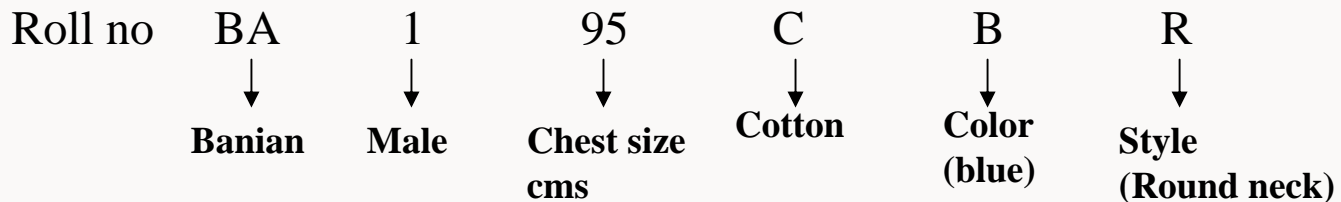
- **CONCISE** - Smallest length to reduce storage and data input effort
- **EXPANDABLE** - Add new members easily
- **MEANINGFUL**- Code must convey some information about item being coded
- **COMPREHENSIVE** - Include all relevant characteristics of item being coded
- **PRECISE** - Unique, unambiguous code

WHAT METHODS DO WE USE TO CODE

- 1) SERIAL NO: Assign serial number to each item
- 2) BLOCK CODES: Blocks of serial numbers assigned to different categories.
- 3) GROUP CLASSIFICATION CODE- Groups of digits/characters assigned for different characteristics



- 4) SIGNIFICANT CODES - Some or all parts given values



CHARACTERISTICS OF CODES

Characteristics →

<u>Codes</u>	Concise	Expandable	Meaningful	Comprehensive	Precise
SERIAL NO	Yes	Yes	No	No	Yes
BLOCK CODES	Moderate	Yes	No	No	Yes
GROUP CLASSIFICATION CODE	No	Yes	Yes	Yes	Yes
SIGNIFICANT CODE	No	Yes	Yes	Yes	Yes

ERROR DETECTION CODE

- Incorrect data entry can lead to chaos
- Mistakes occur as volume of data processed is large
- Need to detect and if possible correct errors in data entry
- Error detected by introducing controlled redundancy in codes
- Error control digits added based on statistics of types of errors normally committed during data entry

MODULUS 11 CHECK DIGIT SYSTEM

- Error detection digit added at the end of a numeric code
- Code designed in such a way as to detect all single transcription and single transposition errors which is 95% of all errors

Single transcription error- $\rightarrow 49687 \rightarrow 48687$

Single transposition error $\rightarrow 45687 \rightarrow 48657$

- Given code 49687 modulus check digit obtained as follows: Multiply each digit by Weights of 2,3,4 etc starting with least significant digit

$$7*2+8*3+6*4+9*5+4*6=131$$

$$131/11=11,\text{remainder } 10; \text{ or } 131 \bmod (11) =10;$$

$$(11-10)=1 \text{ append it to the code}$$

- The code with check digit=496871
- If remainder is 1 then append $(11-1)=10$ code as X

ERROR DETECTION

496871 \longrightarrow 486871

Correct code

Code as entered

$$\begin{aligned} \text{Error detection} &- 1*1 + 7*2 + 8*3 + 6*4 + 8*5 + 4*6 \\ &= 127/11 \text{ Remainder } \neq 0 \Rightarrow \text{Error} \end{aligned}$$

496871 \longrightarrow 416879

$$\begin{aligned} \text{Error detection} &- 9*1 + 7*2 + 8*3 + 6*4 + 1*5 + 4*6 \\ &= 100/11 \text{ Remainder } \neq 0 \Rightarrow \text{Error} \end{aligned}$$

WHY DOES MODULUS 11 CHECK DIGIT WORK

- Given d_n, d_{n-1}, \dots, d_1 where d_1 is the check digit

$$\sum_{i=1}^n W_i d_i \pmod{N} = 0 \text{ by design}$$

What should be the values of N & W_i s
Single transcription error: d_k become t

$$\sum_{i=1}^n W_i d_i = \sum_{i=1}^n W_i d_i + t W_k - W_k d_k$$

As $\sum_{i=1}^n W_i d_i \pmod{N} = 0$ $(t - d_k) W_k \pmod{N} \neq 0$

$(t - d_k) W_k \neq p \cdot N$ where p is any integer

Conditions

- $0 < W_k < N$
- As $[t - d_k] < 10$ and $W_k < N$, $N > 10$
- Product of integer not a prime $\Rightarrow N$ a prime
- Smallest prime $> 10 = 11 \Rightarrow N = 11$

WHY DOES MODULUS 11 CHECK DIGIT WORK

Single transposition error

Let d_k and d_m get interchanged

$$[\sum_{i=1}^n W_i d_i + (d_k W_m + d_m W_k - d_k W_k - d_m W_m)] \bmod N \neq 0$$

$$\text{Or}(d_k - d_m)(W_m - W_k) \neq p \cdot N$$

1. $(W_m - W_k) \neq 0 \Rightarrow$ Weights distinct
2. $(d_k - d_m) < 10$ If $N > 10$ equation satisfied
3. If N prime product cannot be prime
therefore $N = 11$ satisfies conditions

OTHER CHECKING SYSTEMS

- USE MODULO N CHECK WITH N PRIME > LARGEST CODE CHARACTER VALUE

- FOR HEXADECIMAL CODES SYMBOLS = 16, N = 17

- FOR ALPHANUMERIC CODES 26 LETTERS

10 DIGITS

—————
36 SYMBOLS
—————

- Therefore N=37.

VALIDATING INPUT DATA

- WHEN LARGE VOLUME OF DATA IS INPUT SPECIAL PRECAUTIONS NEEDED TO VALIDATE DATA
- VALIDATION CHECKS :
 - SEQUENCE NUMBERING - detects missing record
 - BATCH CONTROL - Use batch totals
 - DATA ENTRY AND VERIFICATION-Dual input
 - RECORD TOTALS-Add individual values for checking
 - MODULUS 11 CHECK DIGIT

CHECKS ON INDIVIDUAL FIELDS

- Radix errors - For example seconds field cannot exceed 60, month field cannot exceed 12
- Range check - Fields should be within specified range
- Reasonableness check - Telephone bill cannot be more than 10 times average bill of last few months
- Inconsistent data - For example : 31-04-99
- Incorrect data- Batch total checks this
- Missing data - Batch control data checks this
- Inter field relationship check -
- For example - Student of 8th class cannot have age > 25