Detailed Process Definitions;
The Data Dictionary

Data Dictionary
- the data dictionary is a database or repository of information about objects identified during systems development
- every object (and each of its components) must have a definition in the data dictionary
- the data dictionary is a major source of documentation about the information system

Data Dictionary Entries for Components of DFDs
- the data dictionary must contain precise definitions of all components of all data flow diagrams:
  - to fully explain the meaning of the DFDs
  - to describe the contents of all data flows and data stores
  - to describe the processing that occurs in primitive processes
  - to ensure that names and meanings of components are used consistently (a common vocabulary)

Data Dictionary Entries
- a data dictionary entry must be included for each
  - data flow
  - data store
  - higher level process
  - primitive process
  - external agent (source/sink)

Data elements
- each data flow consists of a series of data elements
  - a data element is a unit of data that cannot be further broken down into meaningful units of data
  - each data element should also have an entry in the data dictionary
  - data flows and data stores are made up of data elements

Data Dictionary - data element entry

<table>
<thead>
<tr>
<th>DATA ELEMENT:</th>
<th>Product code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alias:</td>
<td>Inventory number, product number</td>
</tr>
<tr>
<td>Description:</td>
<td>Number to identify and differentiate each product held in warehouse</td>
</tr>
<tr>
<td>Values:</td>
<td>Must be a positive integer</td>
</tr>
<tr>
<td>Range:</td>
<td>00001 to 99999</td>
</tr>
</tbody>
</table>

AUTHOR: David Ross
DATE: 14 Oct 2002
**Data Flows**

- A data dictionary entry for a data flow describes the sequence of data elements and data structures in the data flow using the following connectors:
  - `=` is equivalent to
  - `+` and
  - `[ ]` select one of
  - `{ }` iterations of
  - `( )` optional
  - `*` comments

---

**Data Stores**

- A data store is made up of data flows and data elements
- Where a data store consists of a collection of data flows it is described as repetitions of that data flow
  - E.g. **Data Store:** {customer invoice}
- Data stores may also be described by listing their data elements
  - E.g. Customer deposit = account no + deposit date + deposit amount + account balance

---

**Describing Processes**

- Each process in higher level DFDs is defined by the DFD that decomposes the process at the next level down:
  - These are parent processes
- Each such process should have a data dictionary entry which includes a brief description of the overall nature and purpose of the process

---

**Example Data Dictionary Entry**

**Data Flow Entry**

- **Academic Consultancy**
- **Data Flow:** Sales order
- **Alias:** Customer order
- **Description:** Request for product from customer
- **Composition:**
  - Sales order = sales order no. +
    - Sales order date +
    - Customer number +
    - {account number} +
    - Customer name +
    - Customer address +
    - {Customer telephone no} +
    - {Item no + item desc + item price + item qty} +
    - Sales order total amount

**Data Store Entry**

- **Academic Consultancy**
- **Data Store:** Sales
- **Description:** Store of all pertinent data about product sales made to customers
- **Composition:** Sales = date +
  - (Product) +
  - Product quantity +
  - Sales total +
  - {Customer}
Data Dictionary - data store entry

**ACADEMIC CONSULTANCY**

**DATA PROCESS**

Treat patient

**AUTHOR:** David Ross  
**DATE:** 10 Oct 2002

- **Description:** Patient consultations are carried out to determine the causes of patients' illnesses/ medical problems. Further treatment/ follow up is recommended if appropriate. Details of consultations are recorded, and fees charged to the patient on the basis of these sessions.

- **Data required:** (Patient)+  
  Fee schedule

Describing External Agents

- Each external agent (source or sink) should have a data dictionary entry which describes its relationship with the system.

- **e.g.**  
  **Referring Doctors:**
  These are doctors who refer their patients to a specialist medical practitioner for treatment. They are usually general practitioners.

Data Dictionary - external agent entry

**ACADEMIC CONSULTANCY**

**DATA AGENT:** Referring doctor

**AUTHOR:** David Ross  
**DATE:** 14 June 2002

- **Alias:** Referrer, GP

- **Description:** Doctors who refer their patients for treatment.

Building and Maintaining the Data Dictionary

- Determine standard formats and information content for all types of data dictionary entries.

- Have a standard means of organising and storing the entries in the data dictionary.

- Ensure that all components of the DFDs have entries in the data dictionary and that they are kept up-to-date.

- Cross-referencing of entries in the data dictionary can help to check the completeness and consistency of the DFDs and other types of models.

Detailed Process Definitions

- The processing that occurs within the bottom level (primitive) processes in DFDs needs to be defined.

- Detailed process descriptions are also known as minispecs.

- Detailed process descriptions form part of the data dictionary: they define the contents of primitive processes.

Detailed Process Definitions

- Many techniques can be used to define the details of processing:

  - **e.g.** narrative text
    Structured English decision tables decision trees flow charts
Detailed Process Definitions

detailed process descriptions should:

- express what the process does (i.e. policy), not how the process is carried out (i.e. procedure)
- be in a form that can be easily understood and verified by both users and systems analysts
- be in a form that can be easily communicated to all potential stakeholders:
  e.g. end-users, systems analysts, managers, system designers, project leaders, programmers

Structured English

Structured English is a modified form of English with some major restrictions on vocabulary and structure:

- only action (imperative) verbs such as get, put, add, calculate, find, delete are used
- only nouns/noun phrases which refer to components of the DFDs should be used, i.e. data flows, data stores, data elements

Structured English

- sentences consist of action verbs and DFD components
- sentences are combined to form process descriptions using the control structures of sequence, condition, and repetition

Control Structures

- e.g.
  Select Case
  CASE 1 qty-in-stock is less than minimum-order-qty
      Do update stock-reorder indicator
  CASE 2 qty-in-stock is equal to minimum-order-qty
      Do nothing
  CASE 3 qty-in-stock is greater than minimum-order-qty
      Do nothing
  End Case

Data Dictionary - data store entry

<table>
<thead>
<tr>
<th>DATA PROCESS</th>
<th>AUTHOR: David Ross</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stock reorder</td>
<td>DATE: 10 Oct 2002</td>
</tr>
</tbody>
</table>

- Description: Current levels of stock in warehouse are matched to optimal requirements. Shortfalls are ordered immediately.
  - Select Case
    CASE 1 qty-in-stock is less than minimum-order-qty
        Do update stock-reorder indicator
    CASE 2 qty-in-stock is equal to minimum-order-qty
        Do nothing
    CASE 3 qty-in-stock is greater than minimum-order-qty
        Do nothing
  End Case
Control Structures

- Sequence is represented with one sentence following another in sequence:

  Add student to class list
  Decrease available-places
  Calculate class-fee

Control Structures

- Repetition uses Do-Until or Do-While loops:

  Do
  Accept customer-account-details
  Calculate daily-interest = daily balance * daily interest rate
  Add daily-interest to monthly-interest-due
  Until
  no more customer-accounts

Example Structured English

Accept sales-order
Find customer-details
If customer-details not found
  Then reject sales-order
Else
  Create sales-order-header
  Do while more sales-order-items
    find item-details
    calculate sales-order-item price = item price * order-qty
  Enddo
  Authorise sales-order
Endif

Guidelines for Structured English

- use indentation to indicate control structures and their scope: assists readability and understanding
- avoid more than three levels of nesting: complicated logic can be represented using other techniques
- Structured English descriptions should illustrate the logic of the processing, not the implementation of the processing

Decision Tables

- decision tables are useful for describing processes where several different conditions apply and the particular actions that are taken are determined by combinations of the values of the conditions
- decision tables are useful where the process logic is complex
- decision tables show all the possible choices and the conditions they depend on in a tabular form

Decision Tables

decision tables have three stubs (four quadrants):

<table>
<thead>
<tr>
<th>conditions</th>
<th>rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>actions</td>
<td></td>
</tr>
</tbody>
</table>

outcomes for each set of condition values
Example Decision Table

<table>
<thead>
<tr>
<th>avg account bal &gt; $1,000</th>
<th>Y</th>
<th>Y</th>
<th>Y</th>
<th>N</th>
<th>N</th>
<th>N</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>overdraft amount &lt; $50,000</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>previous paid-out loan</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>approve</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>conditional approval</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>reject</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Decision Trees

decision trees are an alternative graphical representation of a decision situation as a connected series of nodes and branches

- wholesale customer
  - local item: 15%
  - imported item: 10%
- retail customer
  - local item: 12%
  - imported item: 7%

Determine Customer Discount

Selecting Techniques for Process Descriptions

- Structured English is useful where a process has a sequence of activities and there is no more than three levels of nesting of decisions
- decision trees and decision tables are useful where a process involves a decision based on combinations of values of several conditions

Selecting Techniques for Process Descriptions

- decision trees visually distinguish the decision conditions and their values from the actions: they show the paths that decisions can follow but soon become cluttered if each condition has several possible values
- decision tables are able to show decisions involving many conditions each with many possible values

Overview of Process Modelling

- several techniques are available for representing the processing within systems
- the aim of process modelling in systems analysis is to define the processing that occurs within a system

Process Modelling Techniques

- models representing:
  - an overview of the system processing
  - the structure of the system processing
  - the flow of data into and out of the processes
  - the detailed logic of the processes

  can be constructed
References
