DATADICT - A DATA ANALYSIS AND LOGICAL DATABASE DESIGN TOOL

Tan Tahn Joo, Tan Kah Poh, Goh Ah Moi

SYSTEMS & COMPUTER ORGANISATION/NATIONAL COMPUTER BOARD

ABSTRACT

*The **S&C/NCB Project Manager** addresses a system development methodology, which is adapted from IBM's Business Systems Planning, Yourdon's Structured Analysis and Structured Design, Infocom's Information Engineering, and Jackson Structured Programming. **DATADICT** is an automated tool designed to support the S&C/NCB Project Manager.*

*This article describes the data driven approach adopted in our methodology and how DATADICT is designed as a **documentation** as well as an **analytical** and **design tool** for logical data analysis and design.*

INTRODUCTION

The term "Data Engineering" has become very popular nowadays to refer to works relating to the analysis and design of data as an information resource to the organisation. In the S&C/NCB Project Manager, data is taken as the nucleus for the entire system development life cycle. Starting from the inception of Information Systems Planning (ISP), to the Structured Analysis and Logical Design (SALD) of a functional system, a vast amount of information is represented as data. In ISP, data to support the corporate mission, goals and operating strategies is identified. In SALD, data is analysed and manipulated to suit the functional requirements within the application system. The mathematician Codd called it a normalisation process, where data redundancy is removed to obtain a stable logical data model. The completeness and correctness of the data model is verified against the usage paths and transactional frequencies of the transactions that have been identified earlier in the analysis phase. After logical data analysis and design, the consolidated data model obtained is ready to be coded and stored in the computer according to the physical data base design carried out on a particular data base management system.

DATA MODELLING TOOL

A **data entity** refers to a logical grouping of similar things, events or objects. **Data attributes** describe the characteristics of this data entity, e.g.CUSTOMER-NAME is a data attribute describing the data entity CUSTOMER. There can be at most two entities to form a relationship, which can be one-to-one, one-to-many or many-to-many. The relationship may be mandatory or optional as represented by "1" or "0" respectively. A **data model** is a graphical representation of these data entities with their relationships indicated. An example of a data model is shown below with its entity listing shown in Figure 5.

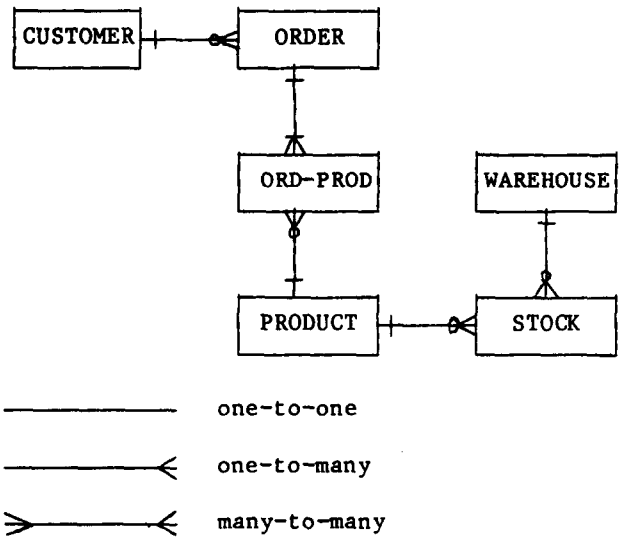


Figure 1

METHODOLOGY - S&C/NCB Project Manager

The S&C/NCB Project Manager is **OBJECTIVE, DATA** and **USER** driven (Figure 2).

ISP uses a top-down approach starting with the strategic requirements planning through identification of corporate purpose and mission statement, and corporate objectives. It also identifies the business processes which are required to be performed in the organisation to manage the key resources and supporting resources of the organisation. It then identifies the data required to support these corporate objectives. The main deliverable at this step is the **initial corporate data model.**

The study then continues with the identification of goals and operating strategies in each business unit of the organisation. More operational data is identified to form a **business unit data model.** The final step will be the integration of these business unit data models and the resolution of the conflicts or duplications of data among them. Due to the voluminous amount of data at this time, an automated tool to aid in the integration process will be very much desired.

ISP will also develop **application system data models** according to the business processes groupings. The data attributes will be regrouped, based on the application or functional activities in the organisation. The next few tasks in ISP are the identification of data responsibility for each data entity, decision on the domain of automation and prioritisation of application systems according to the impact to the organisation and chances of success. An **information systems plan** could then be worked out and proposed to the user management.

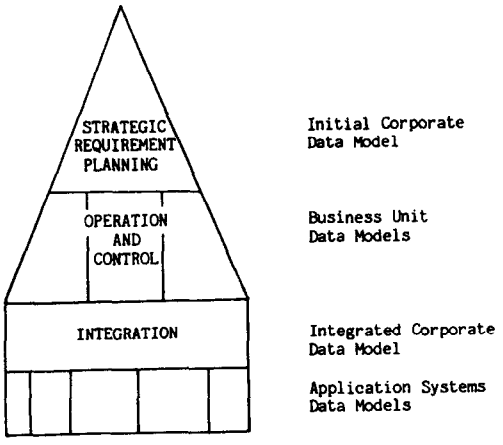


Figure 2

Structured Analysis and Logical Design (SALD) analyses the application systems that were identified in the ISP phase. An SALD study will be conducted for every application system identified. Tools used in SALD include data modelling tool, data flow diagram, mini-spec, and tree structure diagram.

Briefly, SALD comprises five main activities:

(i) Review and normalise the data entities identified for the application system data model, and subdivide it into business process data models.

(ii) Identify business transactions based on how data are being manipulated (add, delete, modify and read) by the operational procedures in the application system. This includes the specification of user queries, reports and input screen interfaces.

(iii) Conduct logical database design which computes the total usage frequencies of the paths between two data entities by all business transactions identified in (ii). This information would be very useful for deciding on the man-machine interface and the physical database design. It will be a very tedious step if computed manually.

(iv) Identify data administration requirement like data security, integrity and control requirements.

(v) Design application system using tree structure diagrams to specify the application system structure and program structures, if applicable.

DATADICT

The goal of DATADICT is to improve productivity for the information systems professionals. It was designed with the following subgoals:

a. To enable an Information Systems Plan to respond quickly to environmental changes by providing facilities to add new data attributes, data entities or modifying them.

b. To help maintain the integrity of large volumes of data by providing integrity checks automatically or by issuing warning messages when checks are not automatically done.

c. To aid data base design by providing cross-referencing information through reporting and query facilities.

d. To assist data administrator in maintaining a central data dictionary, keeping records of data definition, security, usage and integrity requirements.

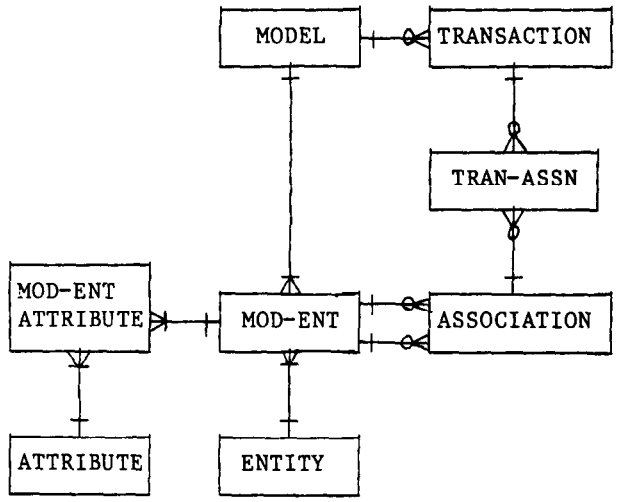
As a **documentation tool,** DATADICT serves as a central repository of all data definitions, data relationships, data security and data usages required to support the information needs in the organisation.

As an **analytical tool,** it helps our information systems professionals in data modelling by facilitating the integration of business unit data models through conflict resolution, and decomposition of data model by functional activities. It analyses the data entities stored to provide useful cross-referencing information, such as the occurrence of data attribute in the data entities, to aid data analysis. It does not automatically normalise the data but helps to carry out data normalisation.

As a **logical design tool,** it aids in the design of logical database by computing the total usage frequencies of each path connecting the data entities. Missing paths that are required in the transactions and unused paths are highlighted.

To make DATADICT user friendly a **menu-driven** approach was taken. Instructions and messages are displayed to guide the user as he navigates through the system. Both input and output screens are designed to give the user clear and meaningful views of the data models and their associated transactions.

An illustration of DATADICT design is shown in the data model below:



The DATADICT Automated Data Dictionary System comprises 2 main modules, namely **DATADICT-ISP** and **DATADICT-SALD** to automate the various tasks to be performed in ISP and SALD respectively as shown in Figure 3.

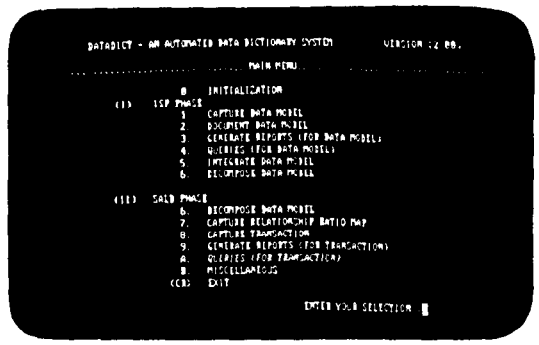


Figure 3

DATADICT-ISP

DATADICT-ISP allows the user to create **business unit data models** which may be integrated to form the **corporate data model** (CDM) and subsequently decomposed by applications into **application system data models** (ASDM).

. Defining a Data Model

Data should be organised in terms of entities, keys and attributes, and are captured as such (Figure 4). There is no limit to the number of entities within the data model, and the number of keys and attributes within the entity.

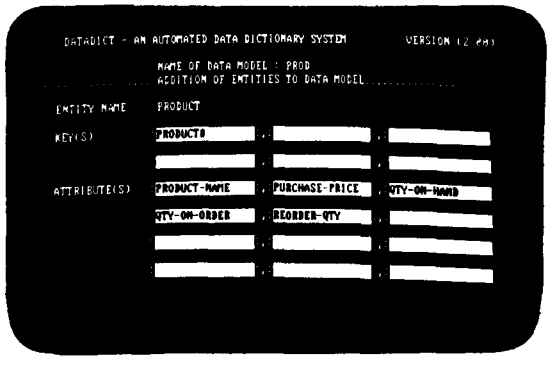


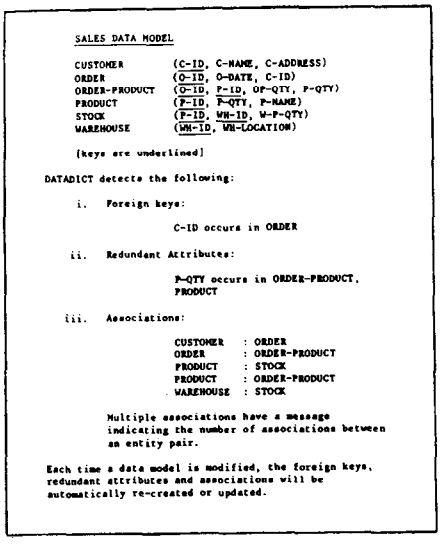
Figure 4

Data that is captured may be modified. To maintain integrity of data, modifications made to entities and attributes at higher level data models (parent data models) will be automatically reflected at lower levels (decomposed data models).

DATADICT analyses the data attributes to determine whether they are plain Attributes, Redundant Attributes or Foreign Keys. The occurrence of attributes which are not indicated as keys in more than one data entity suggests that they are Redundant Attributes and hence the data entities are not in the Third Normal Form. An attribute which appears in one data entity and has been indicated as key in another data entity is detected as a Foreign Key in the former data entity. DATADICT is able to provide a list of possible associations between entities within the data model. Associations may arise from the following occurrences :

- a Primary Key (Compound Key) of one entity is also the Primary Key (Compound Key) of another
- a Primary Key or Compound Key of one entity is part of the Compound Key of another
- the Primary Key or Compound Key of one entity occurs as attribute(s) (Foreign Key) in another

An example of a simplified data model will illustrate the analytical abilities of DATADICT (Figure 5).



SALES DATA MODEL

CUSTOMER	(C-ID, C-NAME, C-ADDRESS)
ORDER	(O-ID, O-DATE, C-ID)
ORDER-PRODUCT	(O-ID, P-ID, OP-QTY, P-QTY)
PRODUCT	(P-ID, P-QTY, P-NAME)
STOCK	(P-ID, WH-ID, W-P-QTY)
WAREHOUSE	(WH-ID, WH-LOCATION)

[keys are underlined]

DATADICT detects the following:

i. Foreign keys:

C-ID occurs in ORDER

ii. Redundant Attributes:

P-QTY occurs in ORDER-PRODUCT, PRODUCT

iii. Associations:

CUSTOMER	: ORDER
ORDER	: ORDER-PRODUCT
PRODUCT	: STOCK
PRODUCT	: ORDER-PRODUCT
WAREHOUSE	: STOCK

Multiple associations have a message indicating the number of associations between an entity pair.

Each time a data model is modified, the foreign keys, redundant attributes and associations will be automatically re-created or updated.

Figure 5

Integrating Data Models

More than one data model may be integrated to produce a corporate data model which provides the user with a global view of the data from the entire organisation (Figure 6). With this, the user may be able to determine how data flowing among various business units affect one another, which may not be possible when they are viewed individually. DATADICT keeps track of the data models involved in each integration process.

In the process of integrating the data models, several instances have to be resolved. In cases where data entities with same entity names and same keys, these data entities will be combined into one. In cases where data entities with same entity names but different keys, a different digit will be appended to each entity so that the names are different. (eg. ORDER — ORDER1). The user may change the name (ORDER1) later on to avoid confusion. In cases where data entities with different names but same keys, the entities will be highlighted as vertical or horizontal partitioning entities. These entities, however, will not be integrated. For data entities with different names and keys, these entities will remain unchanged.

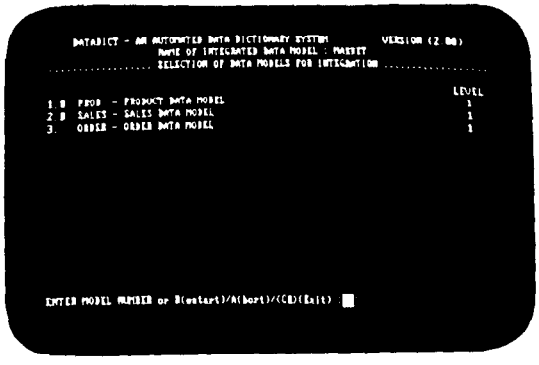


Figure 6

. Decomposing a Data Model

Just as data models may be integrated, they may also be decomposed or repartitioned according to application or functional areas for ease of implementation (Figure 7).

The family of parent-to-child relationships created between data models may extend down several generations ie. data pertaining to a particular functional area may be further decomposed according to activities within the functional area.

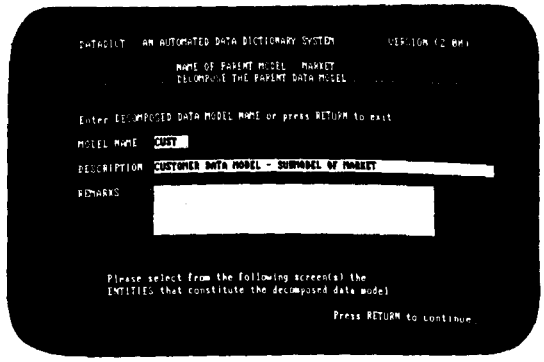


Figure 7

A data model is decomposed by entities. (Figure 8). For each entity selected, the user may further select only part of the attributes originally defined in the data entity. A decomposed data model must be a subset of the parent model. New entities to be added to the decomposed data model is possible only if these entities are added to the parent model first.

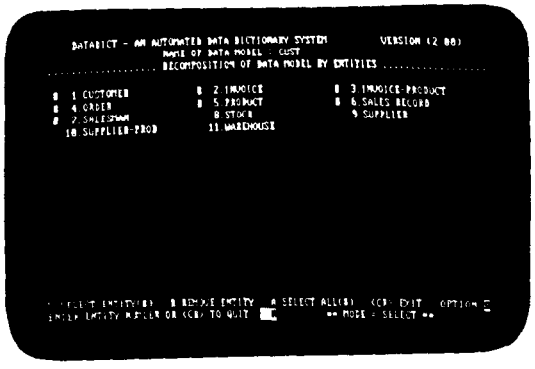


Figure 8

. Documenting a Data Model

Optionally, all entities, attributes and associations may be documented.
Documentation on entities may include such information as the Owner, Passwords, Capacity and Growth (Figure 9).
Documentation on attributes includes the Owner, Passwords, Type (eg. Fixed Point, Floating-Point etc), Field Length, Decimal

Places and Range. Associations may have Name, Type (eg. One:Many), Condition (eg. Mandatory:Optional) and Frequency. Lines of description may also be used to further explain these elements. Only one set of documentation is kept in the DATADICT database which may comprise any number of data models. With this, entities, attributes and associations occurring in more than one data model will not have more than one documentation. This avoids the problem of integrating the documentation of elements belonging to different data models when data models are integrated.

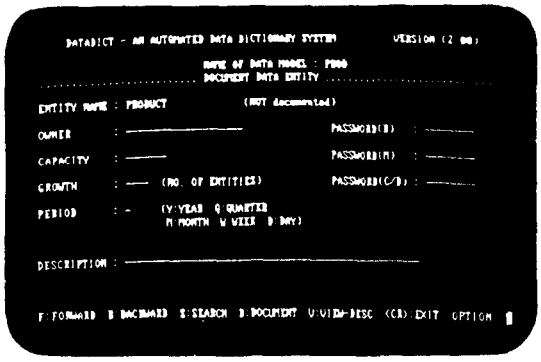


Figure 9

DATADICT-SALD

DATADICT-SALD also contains the decomposition function and allows further decomposition of the **application system data models** (ASDM)into **business process data models** (BPDM). At this stage, the user may create and update transactions within data models.

Creating a Transaction

Information required to create a transaction will include the name for the transaction, a unique transaction number. and the peak period frequency of the transaction. For each of the transactions created, the user may capture firstly, the access paths and their access types (Figure 10) and secondly, the minispecs for the transaction (Figure 11). Access paths are represented in terms of access from one entity to another entity within the data model. The frequency of access is represented by a frequency ratio or the relationship ratio (Figure 12).

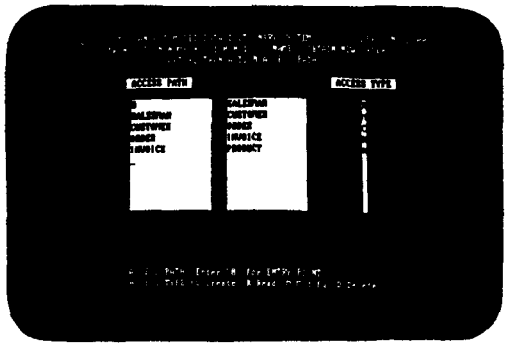


Figure 10

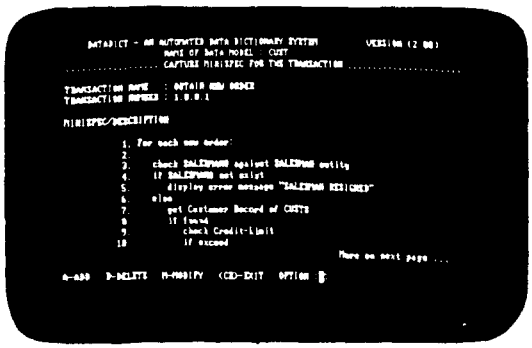


Figure 11

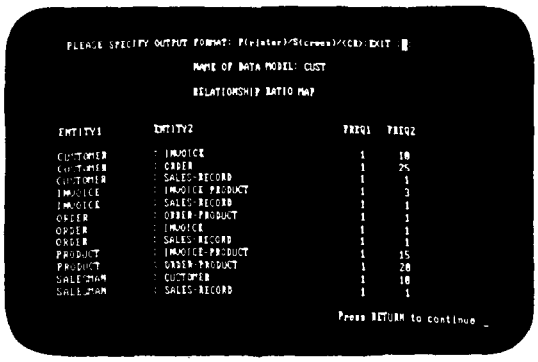


Figure 12

Documenting Transaction Hierarchy

Linkages between transactions within a data model may be created. When a transaction calls another transaction, the former will be a parent transaction and the latter a child transaction. The parent-to-child link may be classified as Sequential, Iterative or Optional. This document will facilitate the creation of Application System Structure Diagram during the preliminary design phase. DATADICT allows for an unlimited number of transaction levels in the hierarchy.

Designing Logical Data Base

It is always very tedious to perform logical data base design manually and as such this task is very often omitted during system development. With DATADICT-SALD, the task of performing logical data base design has become very easy by simply defining the transaction usage path, transaction frequency, and relationship ratio between the data entities. DATADICT-SALD will automatically generate a **transaction reference table** showing the number of logical reference per usage path per period per transaction as shown in Figure 13. It also provides a consolidated load matrix showing the computed peak load per period per access path as shown in Figure 14. These quantitative outputs produced from the logical data base design will serve as useful and important inputs for the physical data base design later on.

TRANSACTION REFERENCE TABLE
FOR CUST MODEL

NAME OF DATA MODEL : CUST
TRANSACTION NAME : OBTAIN NEW ORDER
TRANSACTION NUMBER : 1.0.0.1
PEAK-PERIOD FREQUENCY: 5 PER PEAK PERIOD
ACCESS TYPE : C-CREATE R-READ M-MODIFY D-DELETE

NO.	USAGE PATH SEGMENT		ACCESS EXIST	TYPE	RATIO	NO OF LOG REF PER TRANS	PERIOD
1.	* ENTRY POINT *	:SALESMAN	NO	R	1	1	5
2.	SALESMAN	:CUSTOMER	YES	M	10	10	50
3.	CUSTOMER	:ORDER	YES	C	25	250	1250
4.	ORDER	:INVOICE	YES	R	1	250	1250
5.	INVOICE	:PRODUCT	NO	R	5	1250	6250
TOTAL LOGICAL REFERENCES						1761	9805

Figure 13

ACCESS PATH VS ASSOCIATION
FOR CUST MODEL

PEAK LOAD ON ACCESS PATHS

* ENTRY POINT *	:CUSTOMER	6	**NOT EXIST**
* ENTRY POINT *	:SALESMAN	5	**NOT EXIST**
CUSTOMER	:ORDER	1400	EXIST
INVOICE	:INVOICE-PRODUCT	375	EXIST
INVOICE	:PRODUCT	6250	**NOT EXIST**
INVOICE-PRODUCT	:PRODUCT	375	EXIST
ORDER	:INVOICE	1250	EXIST
ORDER	:ORDER-PRODUCT	150	EXIST
ORDER-PRODUCT	:INVOICE	125	**NOT EXIST**
ORDER-PRODUCT	:PRODUCT	25	EXIST
SALESMAN	:CUSTOMER	30	EXIST
		10011	

Figure 14

In both the ISP and SALD Phases, DATADICT provides comprehensive reporting and query facilities.

OPERATIONAL ASPECTS

DATADICT is chosen to be implemented on a personal workstation for portability and is developed using the database management system called dBASE II, which runs on operating systems like CPM, DOS and CROMIX. The current version of DATADICT runs on the IBM PC or the IBM PC/XT, DOS 2.0 with a minimum memory size of 128K byte. Although DATADICT suffers from a slight drawback in performance, steps are already taken to address this problem.

CONCLUSION

It is our view that the physical data base design should be separated from the logical data base design which is hardware and software independant and hence can be carried out before the decision is made to acquire the hardware and software. DATADICT meets our objective of providing an automated tool to complement our in-house system development methodology so as to increase productivity for our information systems professionals and achieve quality for the system developed.