TITLE FROM BUSINESS ACTIVITIES TO ONLINE APPLICATION DESIGN

Theo Gielens
Database Consultants Europe B.V.
Amsterdam, Netherlands

ABSTRACT

All Design begins with Analysis and, in this paper, we look at how the results of a top-down, structured analysis of activities, using functional decomposition, can be input into a structured online application design.

The applications, when developed, should support the natural flow of operational and administrative activities and, further, should ensure that the organisation's data is kept consistent, up-to-date and accurate.

The following topics are covered:

- Activity Analysis
- Online Application Systems Design.

1. INTRODUCTION

Figure 1 shows the System Development Cycle required in a Shared Data Interactive environment. The left side of the diagram is concerned with the analysis and design of the data ie. Data Analysis, Logical Database Design and Physical Database Design.
The right side is concerned with the analysis and design of the activities ie. Activity Analysis, Application System Design and Structured Program Design.
The centre of the diagram covers the steps required to ensure that the interactions between data resources and business activities are fully catered for during the analysis, design and eventual implementation.

This paper covers the following areas:

1. Activity Analysis
   A method used to understand and record the activities and the flow of data between them.

2. Application Systems Design
   A method to transform the decomposed activities from the activity analysis phase into structured and clearly defined online or batch programs.

It is evident that the participation of the user in both areas is of vital importance. This implies that the techniques used by the analysts and the designers should provide them with simple, diagrammatic tools that enable effective communication with the users.

The business area "COURSE ADMINISTRATION" has been chosen for illustration.

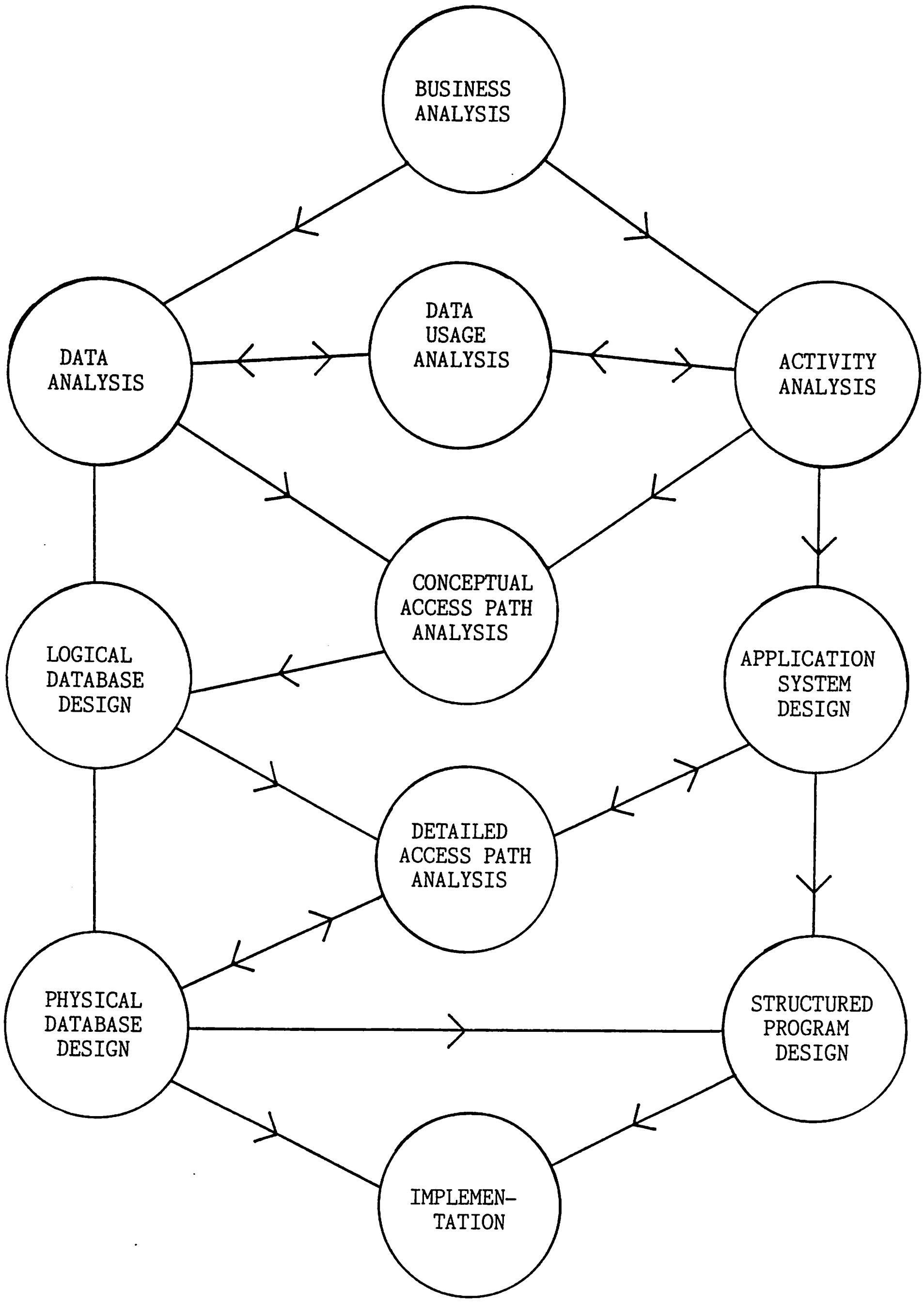


FIGURE 1 - The Systems Development Cycle in a Shared-Data, Interactive Environment.

2. ACTIVITY ANALYSIS

Activity analysis is a step-by-step, diagrammatic method used to analyse business activities by starting at a high overview of business areas and progressively decompose them, through lower levels of activity, into greater detail.

2.1. Deliverables

The output from activity analysis comprises Data Flow Diagrams and Activity Decomposition Diagrams supported by appropriate documentation. The DFD shows the activities, concerned with a business area, and the flow of data into and out of them.
The ADD provides an overview of the activities showing commonality, trigger events and the sequence in which the activities are performed (top-to-bottom and left-to-right).

2.2. Tasks for Activity Analysis

Activity Analysis consists of the following tasks:

2.2.1. Decompose the activities

The first objective is to clearly identify the major activities within the business area and the flow of data concerned with these activities (see Figure 2).
The next step is to break these activities down into smaller activities. The information is gathered using a number of sources as follows:

- interviews
- documentation of current systems
- documents currently in use
- etc.

2.2.2. Identify Elementary Activities

The decomposition of activities stops when a level of Elementary Activities has been reached (see Figure 3):

An elementary activity is one which cannot be decomposed further without either destroying the consistency of the data it uses and outputs or (in case of an enquiry type activity) ceases to provide all the output necessary to the objective of the activity.

Elementary activities become candidate transactions

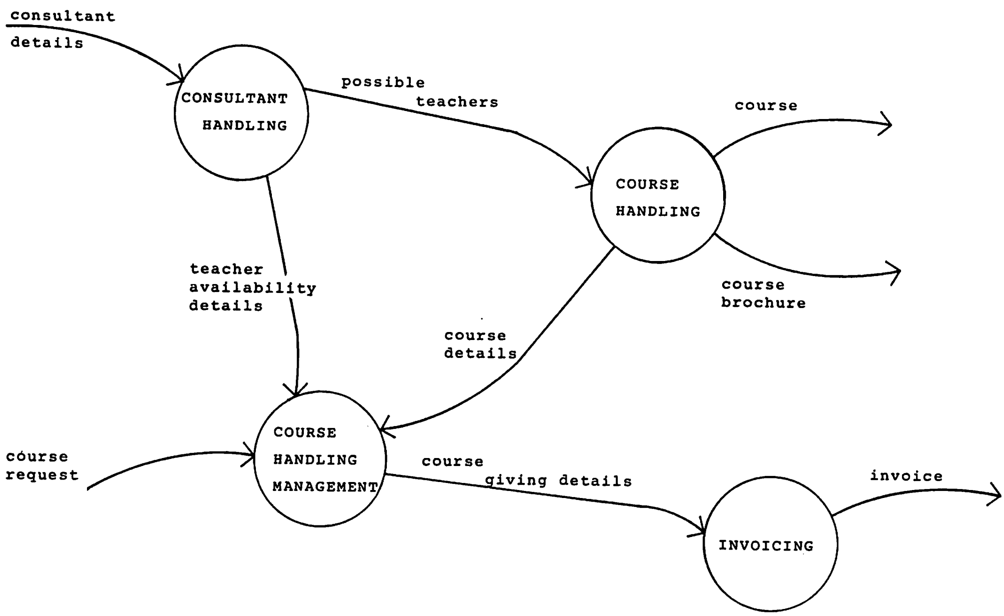
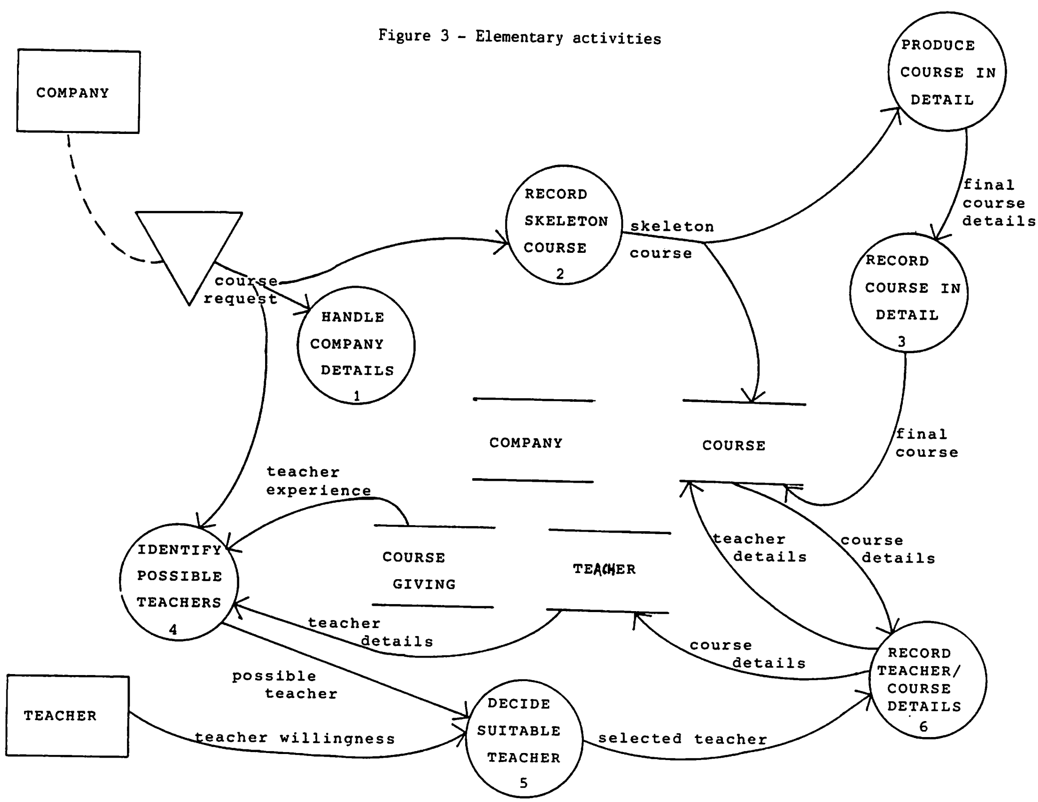


Figure 2 - Major activities in a course administration

Figure 3 - Elementary activities



2.2.3. Document low level DFD's

Since the DFD's are input to both the System Application Design and Conceptual Access Path Analysis phase it is essential to document them showing:

- description of the activity and the data flows.
- volumes
- where the activity is performed

2.2.4. Describe the elementary activity in detail

In order to avoid misconception and misunderstanding it is essential to describe the elementary activity in detail. A very useful technique to employ is structured text. This technique also provides a means of identifying commonality during System Application Design.

2.2.5. Create the ADD

The ADD is a hierarchical representation of the business activities showing how they decompose from the highest level activities down to the elementary activities. The sequence in which the activities are being performed (implied within the DFD) is also shown (see Figure 4).

2.3. Activity analysis and its interaction towards the data

The activities that need to be performed on the data will normally influence how the database is organised. From the elementary activities the entry points, navigation paths and access sequences are identified during a process called Conceptual Access Path Analysis (see Figure 1). The resulting output comprises:

- Activity Logic Diagrams
  diagrams showing what entities are accessed/created/deleted and the sequence that is required.

- Activity Usage Figures
  figures showing the frequency of logical accesses on entities.

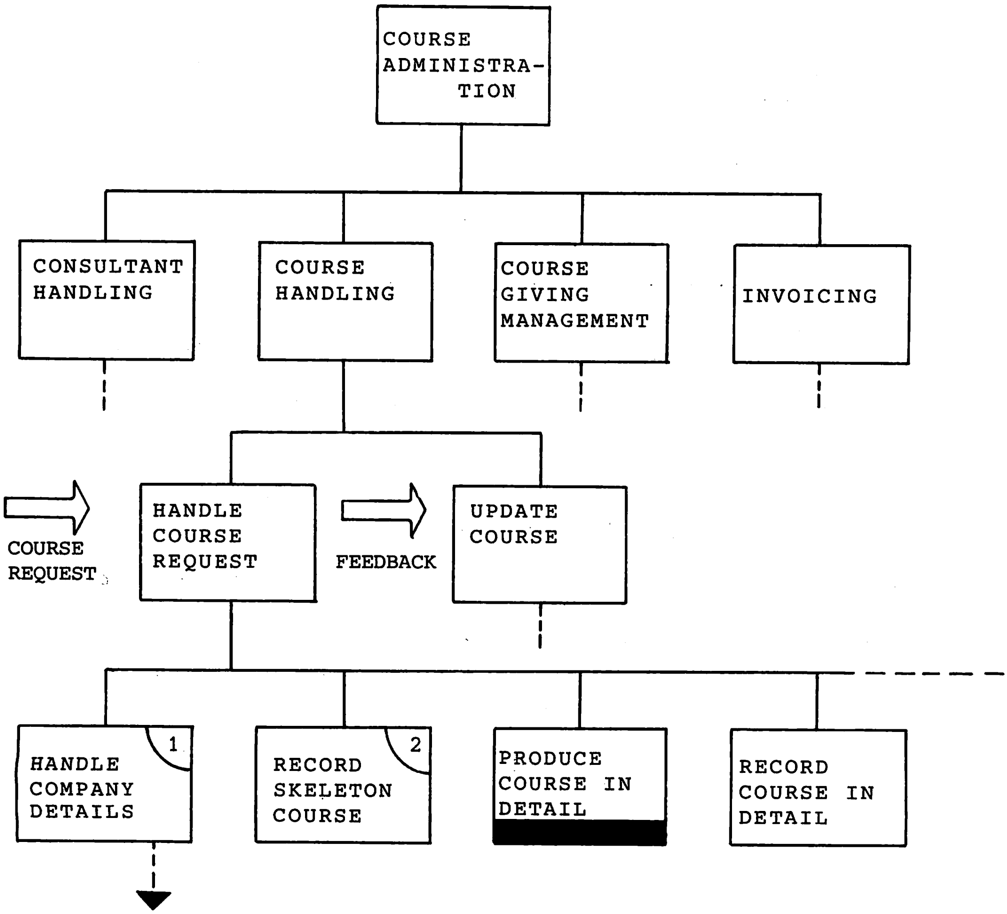
3. ONLINE APPLICATION SYSTEM DESIGN

Application System Design is a method of further decomposing the elementary activities into processes that can be implemented as programs, modules or sub-routines, providing, therefore, a technique to design the structure of the online system.

3.1. Tasks for Online Application System Design

3.1.1. Identify common processes

In order to design an optimum systems network and to avoid repetitive design and development of those processes which may be common to more



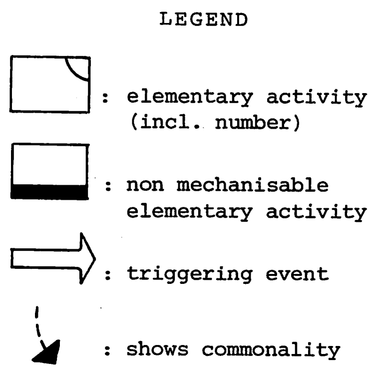


Figure 4 - Activity Decomposition Diagram

than one activity, it is necessary to identify commonality as the first step in Online Application System Design.
Commonality can exist at three levels, as follows:

- Elementary activities
- Components: action on an entity type and its attributes
- Primitives: action on an attribute type or its attribute values, or action on more than one attribute type in combination

Normally commonality of elementary activities would have been clearly identified, already, during the construction of the ADD. However, commonality of Components and Primitives can only be identified by a thorough examination of the structured text produced earlier.

COMPONENTS become candidate modules

PRIMITIVES become candidate modules or sub-routines

Both components and primitives are added to the ADD and, where they are common across more than one elementary activity, this is shown.

3.1.2. Systems Network Design

The systems network is designed using information contained in the DFD's and ADD and, in addition, the following considerations:

- standard navigation paths (define function keys)
- user preference on menus:
  . based on trigger events
  . based around entities
  . based around responsibility
  . etc.
- frequency : a navigation path for an activity which is being used frequently should be as short as possible.

The result is a Systems Network Diagram showing :

- hierarchy of transactions.
- flow of control (sequence, selection and iteration)
- listings (output)
- passed parameters
- how control is passed

To illustrate how a DFD can impact on the eventual SND an example is provided showing how a "course request" could be handled in two different ways. Figure 5a shows one example resulting in a SND as shown in Figure 6a.
Figure 5b shows a similar "course request" containing one extra activity, "CHECK IF COMPANY EXISTS", resulting in a significantly different SND, as shown in Figure 6b.

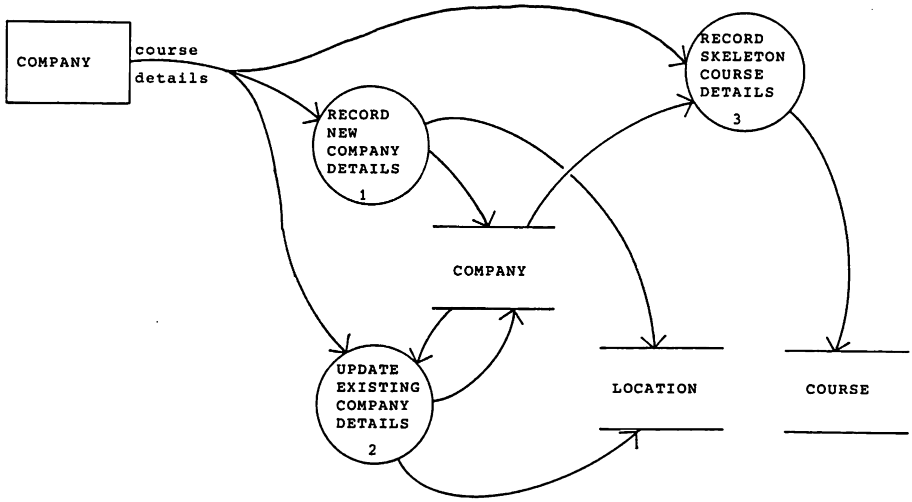


Figure 5a

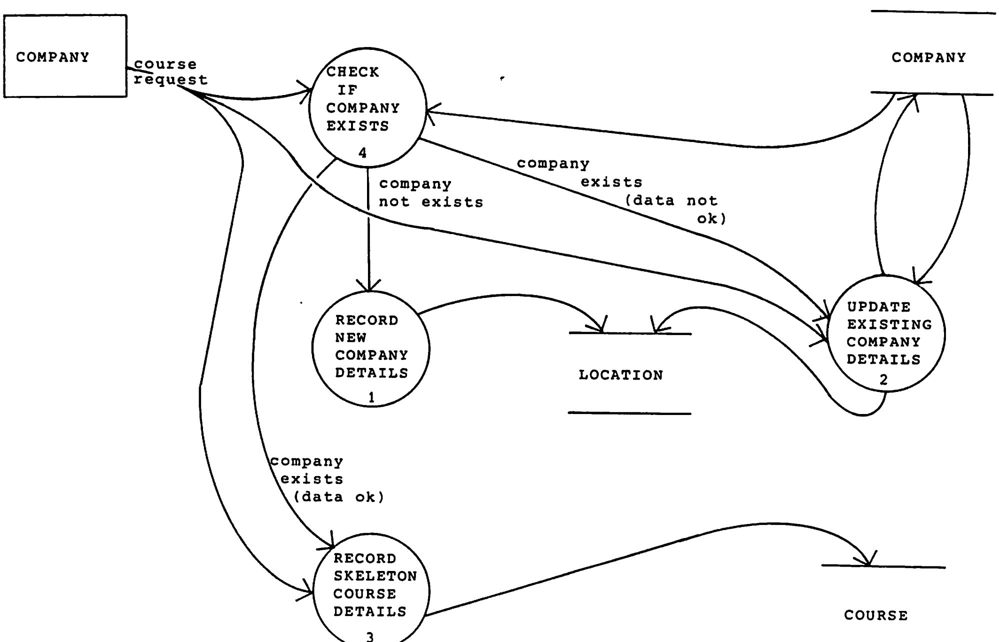
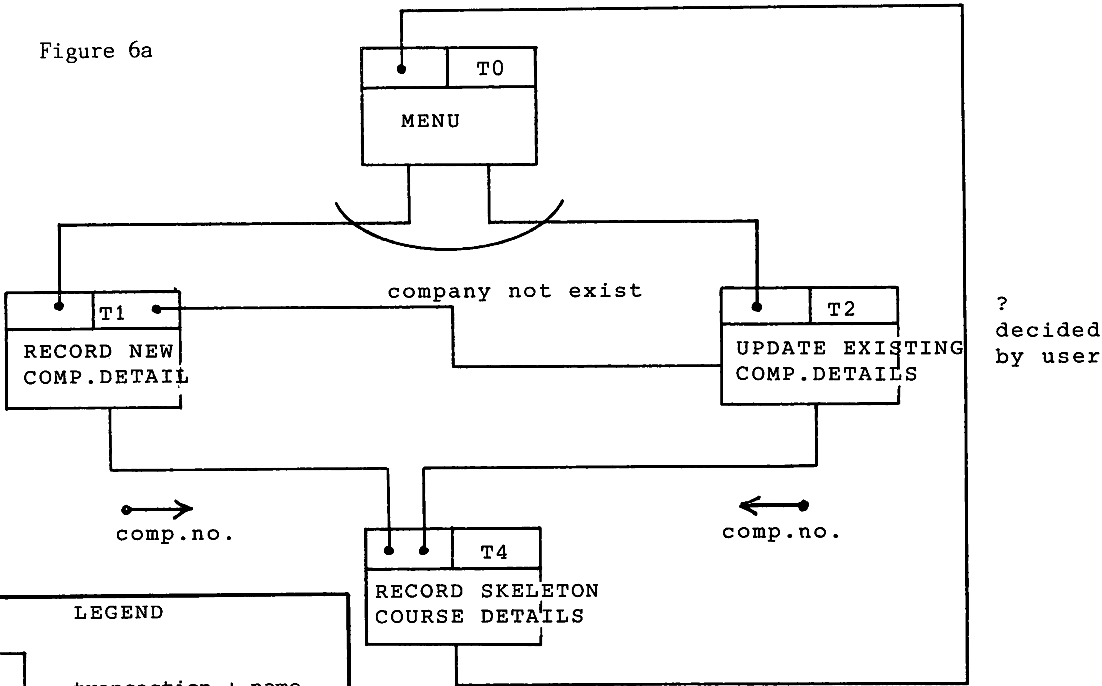
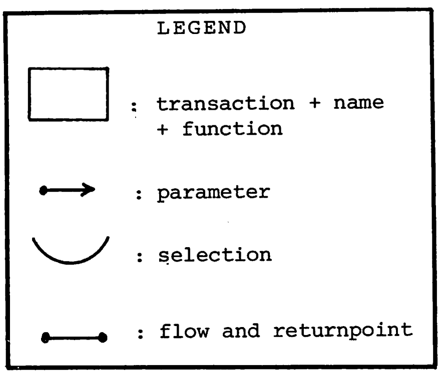


Figure 5b

Figure 6a





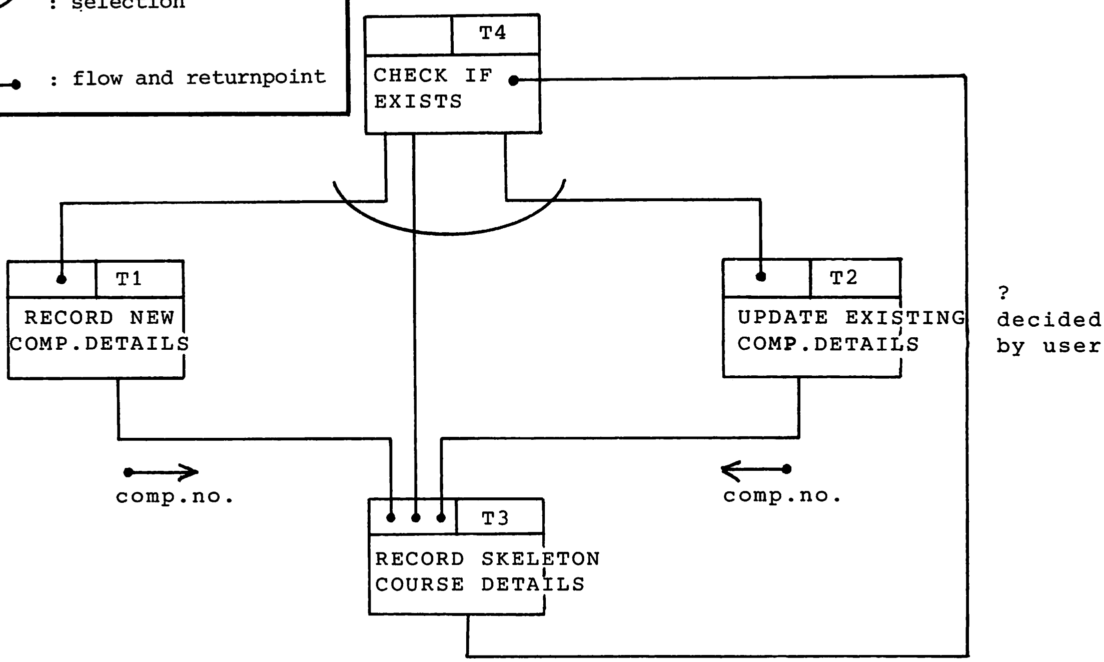


Figure 6b

Before the system network design is complete, it is necessary to check the following:

- Is every transaction available as required?
  Each user should be able to reach all activities he is entitled to use, subject to suitable authorisation.

- Is every common process fully integrated in the SND ?
  Where applicable, parameters should be added to the SND. Also all updating logic should be moved to "lower level" transactions in order to preserve the consistency.

3.1.3. Multiple implementation of Elementary Activities

It is possible that for some elementary activities, more than one transaction may need to be defined, for the following reasons:

- need for fallback systems

- different mechanisms for different users:
  - casual user version
  - experienced user version

- different installations (hardware/software)

For each transaction, however, the logic stays the same, although the mechanisms may be different. Whenever this occurs it affects the systems network:

- fall-back transactions and production transactions must never be put into the same network.
- transactions for different installations must never be put into the same network.
- transactions which provide alternative dialogues may or may not be put into the same network, depending on the users preference.

3.1.4. Design the dialogue(s)

The design is done together with the user. The design primarily consists of designing the user and computer actions for each transaction. The basic steps consist of:

1. determine the inputs and outputs
2. decide the technique of conversation
3. design the dialogue in broad outline
4. optimise and refine

3.1.4.1. Determine the input and output

From the structured text of each elementary activity, it can be determined what input (and when) is required to be able to perform the activity and what output is required to provide the user with what he requires.

3.1.4.2. Decide the technique of conversation

This depends highly on the users skills and preferences. Various possibilities exist, as follows:

- keyword (free format)
- question and answer
- form filling
- multiple choice (menu)
- etc.

3.1.4.3. Design the dialogue in broad outline

Again a diagram is used, which allows discussion with the user. The example (Figure 7) shows how new company details could be handled (refer to figure 6b activity T1). Note that the updates are performed in M06, with reference to the points discussed earlier regarding consistency for elementary activities.

3.1.4.4. Optimise and refine

Using prototyping tools the dialogue can be implemented. This could result in feedback from the user, as a result of which some steps may have to take place again. If no prototyping tools are provided, a desk check examining the dialogue should take place.

3.2. Interaction towards the data

Whereas, in the Conceptual Access Path Analysis (see Figure 1) the output is used to perform the Logical Database Design the outputs of Detailed Access Path Analysis provide the output that is used to either create a first pass physical database or optimise the physical database design.
The feedback from the database design could also enforce an alteration of some activities and possible the SND.

THEO GIELENS

Since the start of his data processing career in 1977, Theo Gielens has been particularly involved in the areas of Application Design and Activity Analysis. After joining DCE as a consultant, some two years ago, he extended his experience in the area of Data Analysis and he participated in the development of DCE's Application Design Course.

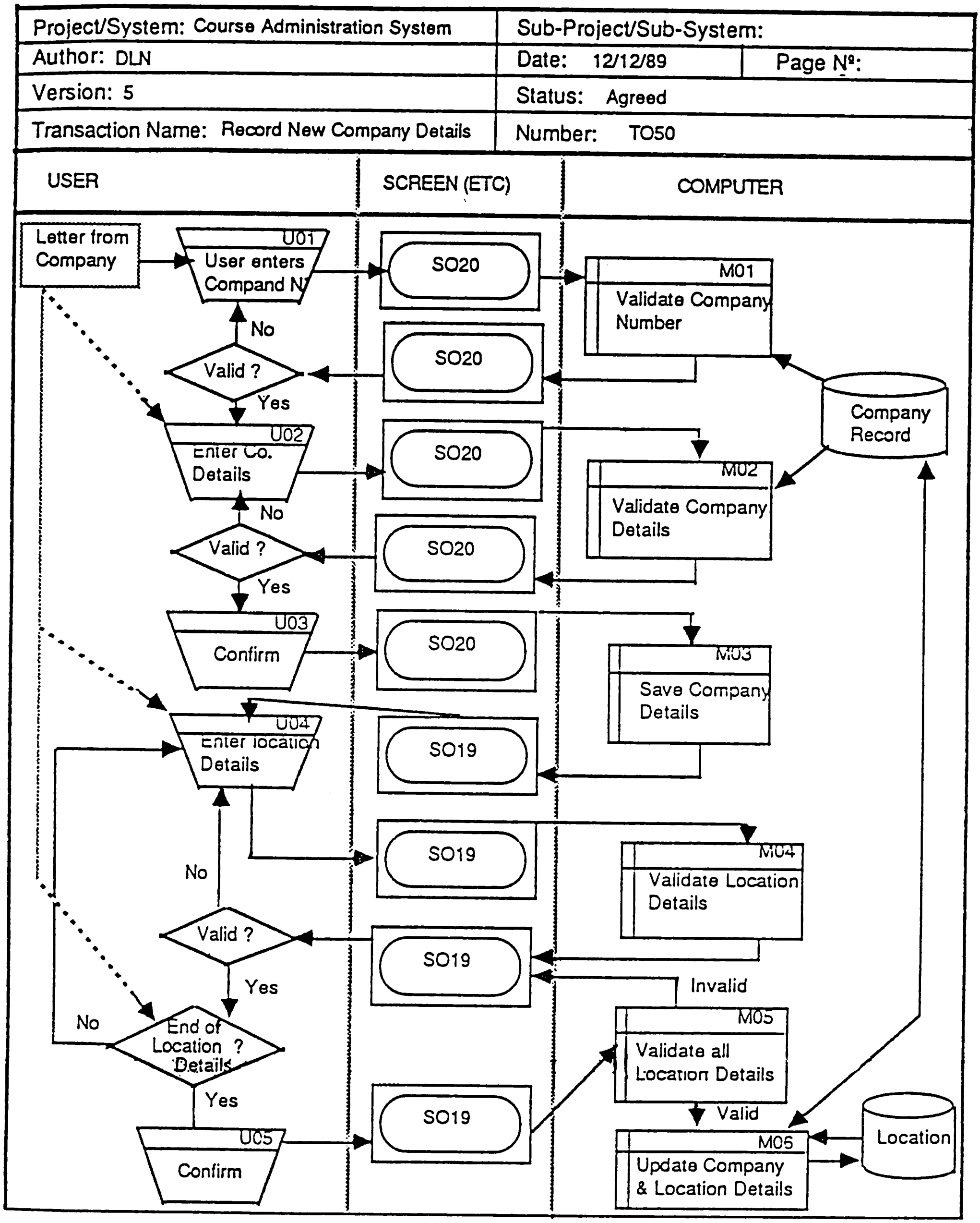

Project/System: Course Administration System
Sub-Project/Sub-System:
Author: DLN
Date: 12/12/89
Page Nº:
Version: 5
Status: Agreed
Transaction Name: Record New Company Details
Number: TO50
USER
SCREEN (ETC)
COMPUTER
Letter from Company
U01
User enters Compand N
SO20
M01
Validate Company Number
No
Valid ?
Yes
SO20
Company Record
U02
Enter Co. Details
SO20
M02
Validate Company Details
No
Valid ?
Yes
SO20
U03
Confirm
SO20
M03
Save Company Details
U04
Enter location Details
SO19
SO19
M04
Validate Location Details
No
Valid ?
Yes
SO19
Invalid
No
End of Location ? Details
M05
Validate all Location Details
Yes
SO19
Valid
U05
Confirm
M06
Update Company & Location Details
Location

Figure 7