CASE PRODUCT EVALUATION & APPLICATION IN H.K.

by

Chu Lung Yan, Cliff
朱隆恩

Sung Sai Kit, Edmond
宋世傑

MBA Project Report

Presented to

The Graduate School

In Partial Fulfilment
of the Requirements for the Degree of
MASTER OF BUSINESS ADMINISTRATION

THREE-YEAR MBA PROGRAMME

THE CHINESE UNIVERSITY OF HONG KONG

APRIL 1991

Y. T. Fung

Dr. Y. T. Fung
Advisor

TABLE OF CONTENTS

Chapter

Chapter I

OVERVIEW OF STRUCTURED METHODOLOGY & CASE SOFTWARE

In 1990, Computer Aided Software Engineering (CASE) is no longer a concept. There are many success stories of implementing structured methodology & CASE tools in the organization. An example can be found in a credit company called Nykredit which is based in Denmark.

In 1989, Nykredit implemented the structured methodology from James Martin Associates (JMA). The JMA methodology was selected because of its ability to integrate fully all work from business planning to building operational systems.

The application systems designed with the JMA methodology were noted with great improvements over those using the traditional design methods. Much less errors were found after the system was put in production. It became easier to maintain, and the change of direction was also easy during the development stage. As a result, Nykredit ran ahead of the competition.

1.1 Introduction

1.1.1 Driving Force behind the CASE evolution

In recent years, the market demand on CASE is growing very fast. The following trends on system demands are also observed.

Programs are getting larger & more complicated. Even PC programs can be of 40,000 to 50,000 lines. Program maintenance becomes gradually out of control. It directly results in the rapid rising of maintaining cost.

With the introduction of microprocessor and the drop of hardware cost, software demand is growing rapidly. The supply of software cannot meet the demand and there are many backlogs in software production.

Producing software is very labour intensive. The use of structured methodology makes software development more like a building project in which the architect designs and produces a project blueprint. The building contractors then build according to the blueprint. The division of labour becomes easier.

Users are getting more away from proprietary standard of the computer vendor. Software development base on CASE techniques make the development more independent of the hardware platform & operating system.

All the above trends lead to the rapid demand on CASE tools.

1.1.2 Short Review of the CASE technology

In 1987 - 1988, CASE was still more a theory than a commercial available tool. The structured approach has been around for more than 15 years. But without the right tool, most software shops found the implementation of structured methodology too tedious.

1.1.3 Benefits of CASE Tools

If the introduction of CASE tools and methodology is well planned ahead so that they can fit into the organization needs, CASE implementation can provide numerous benefits to the organization.

It can provide complete requirement specification. Analysis tools will guide the analyst what are required in the requirement analysis phase. So it ensures the specification is complete.

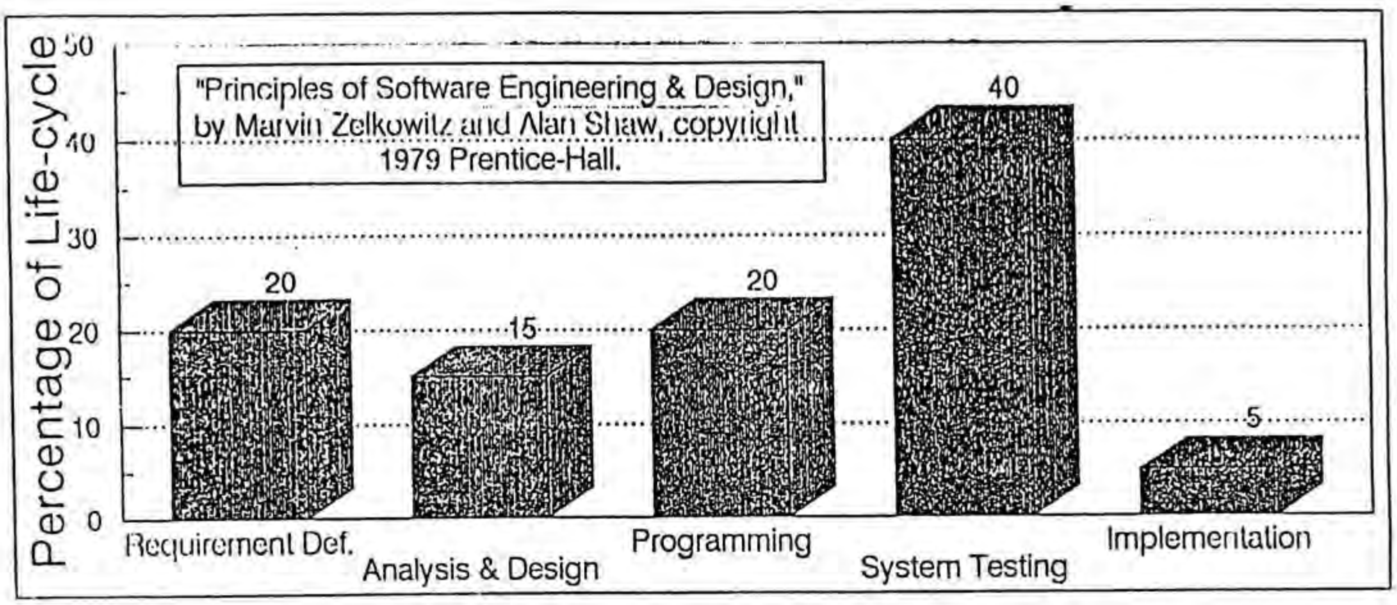
CASE design tools can help to maintain synchronization with the code implementation. Some CASE tools can generate code automatically.

Once the analysts get used to the CASE tools, the overall development time will be reduced. Structured Methodology & CASE concepts emphasis 'Analysis & Design' phase over 'Implementation' phase. During some CASE pilot projects, more time was spent in the project as a whole. It is because the reduction of time spent in implementation was offset by more time spent in Analysis. But as the analyst learns and gets familiar with the techniques in CASE tools, the overall project development time should be reduced. Figure 1.1 and 1.2 highlights the differences between the software life cycle with and without using structured techniques.

For the software maintainability, structured methodology improves quite a lot in this area.

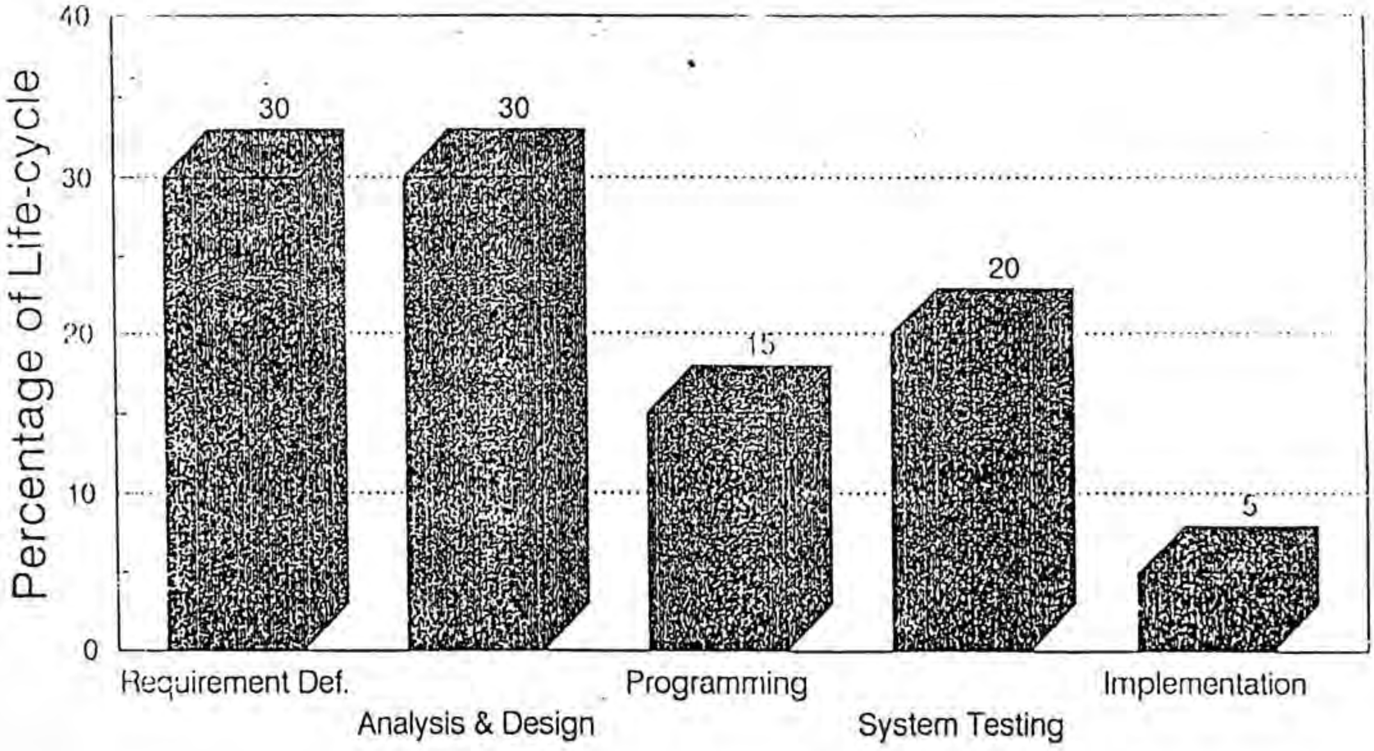
The last but not least benefit of using structured methodology is the compatibility among different software projects is enhanced and ensured. When all projects are based on one common dictionary, the data definition in different projects would be the same & compatible.

Fig. 1.1: The Traditional Software Life Cycle



"Principles of Software Engineering & Design," by Marvin Zelkowitz and Alan Shaw, copyright 1979 Prentice-Hall.

Fig. 1.2: The Traditional Software Life Cycle with Structured Techniques



1.1.4 CASE products

In the beginning, CASE products are mainly started by small entrepreneurial companies. Those CASE products are usually specific product for a specific task in the development life cycle.

In recent years, IBM & DEC started to go into the CASE market. Both promote a standard platform which covers the entire development life cycle.

1.2 Structured Methodology

In this section, the common techniques used & intermediate deliverables of the structured approach are described briefly.

CASE (Computer Aided Software Engineering) automates a large variety of tasks in the entire software development process. Before the emergence of the CASE technology, a number of different structured methodologies have been used in the software industry. CASE helps to automate & integrate these methodologies. So a review of the software system life-cycle and the various methodologies provides the basis for the readers to appreciate the benefits of CASE.

1.2.1 Software System Life Cycle

Traditionally, the life cycle can be divided into five phases:
Requirement Analysis, System Design, Implementation & Coding, Testing and Production & Maintenance.

1.2.2 Structured Analysis methodology

This approach was first developed by Yourdon & Demarco. It applies to the requirement analysis phase and system design phase. The outcome of structured analysis is structured specifications. The structured analysis covers three main areas:

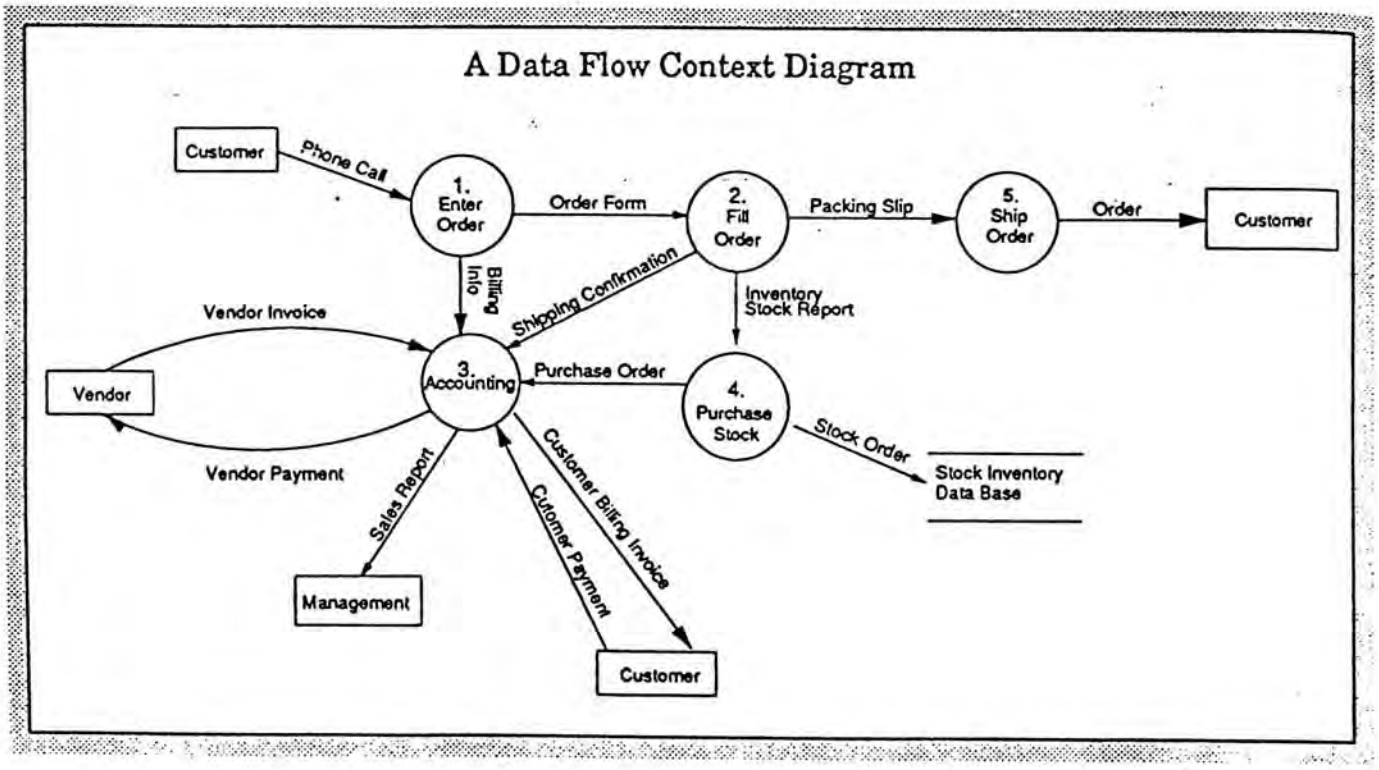
Functional design, Data model, and Control specification.

It is basically a set of if-then rules. It describes how the system responses to the outside stimulation.

1.2.2.1 Functional Design

Systems are made up of functions & processes which process data. The usual way to describe the functional design is the 'Data Flow Diagram'. An example of Data Flow Diagram is shown in Figure 1.3.

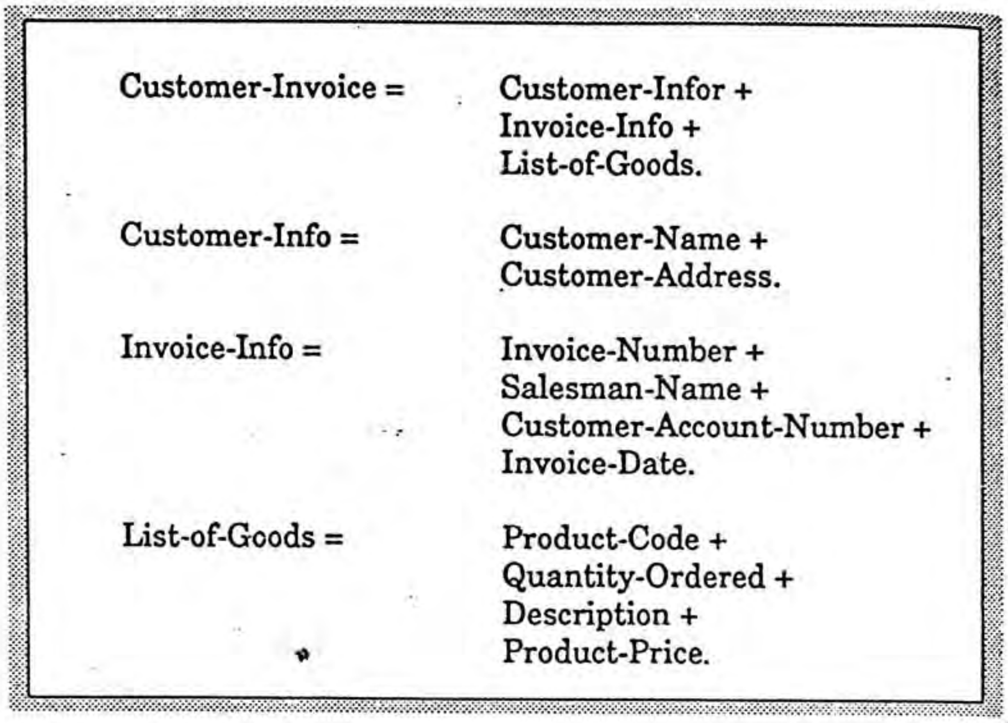
Fig. 1.3: Example of a Data Flow Context Diagram



1.2.2.2 Data Model

All the data elements found in the Data Flow Diagram are catalogued in the 'Data Dictionaries'. It is a database holding all the definitions of data and their interrelationship. Figure 1.4 illustrates an example.

Fig. 1.4: A Data dictionary example



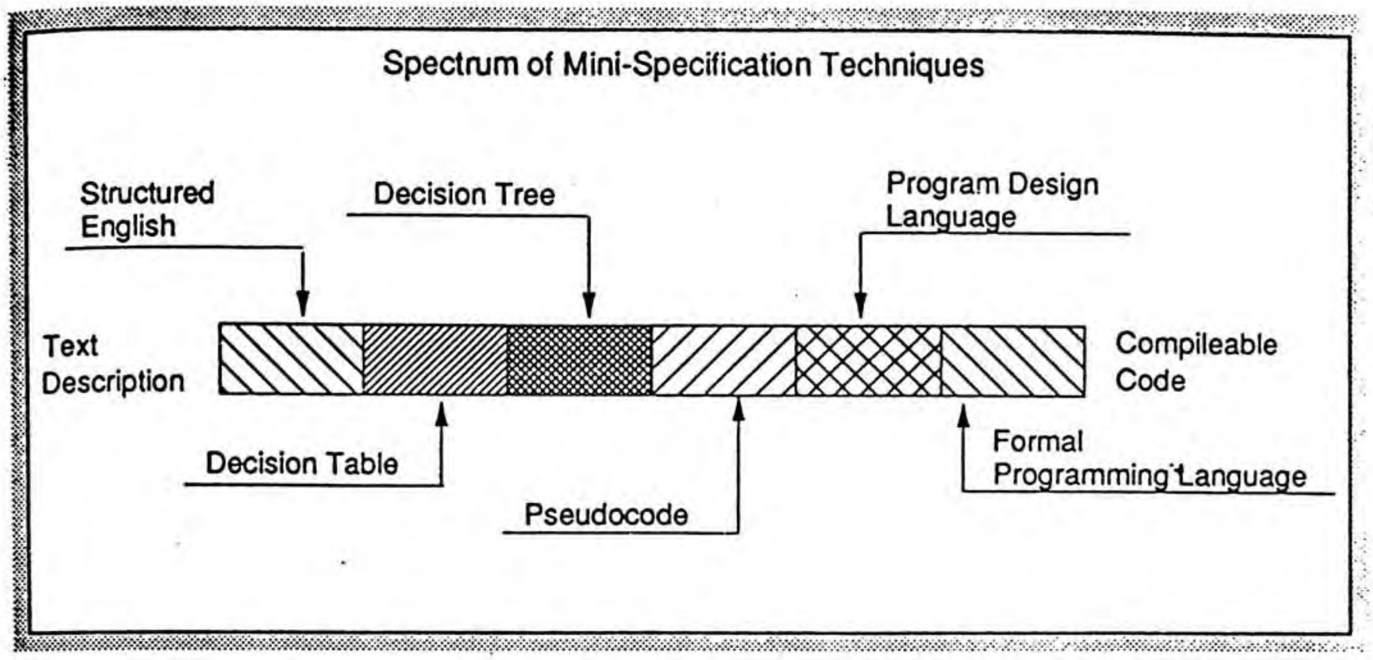
Customer-Invoice =	Customer-Infor + Invoice-Info + List-of-Goods.
Customer-Info =	Customer-Name + Customer-Address.
Invoice-Info =	Invoice-Number + Salesman-Name + Customer-Account-Number + Invoice-Date.
List-of-Goods =	Product-Code + Quantity-Ordered + Description + Product-Price.

1.2.2.3 Control Specification

Although control specification can exist in various forms, there are four common components among them, viz, process name, input data list, output data list and descriptions.

A variety of choices of implementing the process specification exists. There is a whole spectrum of methods, from text description to programming code alike. Refer to Figure 1.5 for more details.

Fig. 1.5: Spectrum of Mini-Specification Techniques



1.3 CASE software products

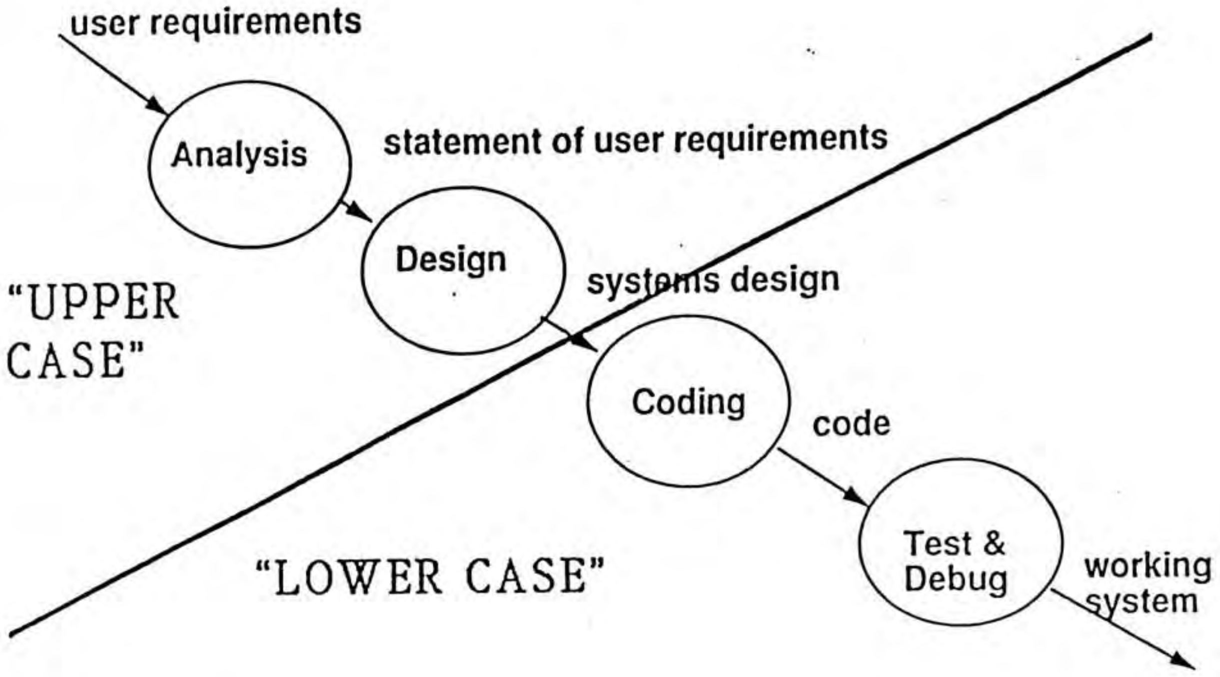
In this section, the objectives of CASE are discussed and the functions provided by CASE software are also described. According to James Martin, CASE can serve multiple purposes. Figure 1.6 lists all its purposes.

Fig. 1.6: Purposes of CASE

- improve software quality & reduce error
- improve productivity
- speed up software development process
- reduce software costs
- automate tedious manual tasks
- offer greater control of the development process
- enforce software engineering
- formalise and standardise software development
- promote software reusability sharing
- improve software portability

CASE software can be viewed as a set of tools that provides leverage at any point in the software project development life-cycle. As an automation tool to structured methodologies, CASE software can be classified broadly into 2 categories, according to their position in the development life-cycle. The "Upper CASE" or "Front End" tools help the analysis & design phase, while the "Lower CASE" or "Back End" tools concentrate on code generation & test automation.

Fig. 1.7: 'Upper CASE' & 'Lower CASE'



Since the application of CASE software relates closely with what structured methodology is being used, the evaluation of CASE tools in an organization is not merely on the CASE tool itself. Both the organization culture & software development methodology should be considered.

Chapter II

PROJECT FRAMEWORK & CASE EVALUATION FRAMEWORK

2.1 Project Objectives

Our objectives are two folds: Firstly, we want to suggest a framework for evaluating CASE in Hong Kong environment. Secondly, we want to compare and contrast the functionalities of two typical CASE environments, namely: IBM's AD/Cycle and DEC's Cohesion. Both of them are now available in the market.

2.2 Research Methodology

2.2.1 Literature Survey

Since CASE is a relatively new area, literature search using ABI/inform provides a quick update of the latest developments. A study of the technical manuals of IBM and DEC regarding these two products are also required.

2.2.2 Questionnaire

Questionnaires will be sent to the large computer-using companies in Hong Kong. The objective is to understand the general use on CASE products and their comments.

2.2.3 In-depth interview

Since there are not many companies in Hong Kong implementing CASE tools, we will interview those CASE users to get a more in-depth understanding in the way they use these tools.

2.2.4 Product demonstration

We will try to arrange with CASE vendors and users some CASE demonstrations to enhance our understandings on their functionalities and user interface.

2.3 CASE Evaluation Framework

This framework is made up of a set of criteria. Since each organization has its particular need, each has a different weighing on these criteria. It is suggested that the evaluator first prioritise these criteria according to the requirement. For important areas, there should be a minimum standard that the CASE tool must comply. These criteria are discussed in the following:

- How is the Life-cycle support of CASE tool?
  If we divide the entire life cycle into requirement analysis, design, code development, testing and implementation, the question to ask is what areas are automated.

- What methodologies are supported?
  Do these supported methodologies include the methodology chosen by the organization?

- Graphic interface
  The analyst use the CASE tool to construct various graphs during the modelling process. A checklist on the Diagramming interface (graphics) is in Fig. 2.1.

- Input & output
  It refers to keyboard, mouse, plotters, background printing capability, & page layout, etc.

Fig. 2.1: Diagramming interface of CASE system

Image primitives	1. Types of primitives supported
	2. Do these primitives meet requirements of chosen methodology?
	3. Can real-time modelling requirements be met by these primitives?
Graph creation	1. Effort required to create diagram
	2. Ease of locating and naming diagram objects
Graph editing	1. Effort required to modify diagram
	2. Trickle-down effect of deleting diagram component
	3. Relocation of diagram
	4. Volume of relocation
	5. Scope of relocation
	6. Renaming graph object
	7. UNDO for graph editing
	8. Scaling graph objects
Graph viewing	1. Rotation of diagram
	2. Zooming diagram
	3. Support of windowing
	4. Manually controlled paging
	5. Graph preview

- How is the Dictionary support & Integration?

- Screen Layout

  It refers to the design of I/P & O/P screens as well as report layouts.

- Analysis & report generation

  The CASE tool should automate the analysis of the model's quality.

- Integration & co-ordination for team use.
  It makes it possible for several analyst to do the same phase of a project together. Each work on his/her own part & then integrate afterwards.

- Interface to external database
  It is for the easy transition from the 'old' database into the CASE Repository.

- System response
  The tool should not be slow in responding to commands.

- Ease of introduction into the organization, emphasising areas like easy to learn, training, hardware, software and operating system compatibility and local support provided.

2.4 **Introducing CASE into the organization**

Since CASE is a new technology to most people in the current MIS organization, introducing CASE into the organization is not just going through the checklist & evaluation framework discussed in the previous section. A good way to start is having an individual in the organization gets educated on CASE first. He/she can be educated about the technology through attending classes & seminars. The following paragraphs suggest the sequence of introducing case tools.

- Individuals get familiarised with the CASE technology first.

- Information about a successful project using CASE tools in other companies are gathered. This helps the management approval.

- CASE vendors are invited to make presentation.

- Several CASE tools are selected for in-house tool evaluation. At this stage, there should not be any bias to favour one vendor over the rest. There is no 'best' CASE tool. (It depends very much on the user environments & needs.) Most people are still very unfamiliar with CASE tools. It is very unlikely that one CASE tool can be selected at this preliminary stage.

- Involve other programmers & developers in the organization. It helps to educate them. Feedback from them on these CASE tools are also useful.

- To prepare evaluation report for management.
  The general guidelines are:
  Selecting a CASE tool is more than selecting a software package/ product or a tool. Since the entire software life cycle is affected by adopting CASE, it is usually an all round vendor relationship. The main concern beside the CASE tool itself are:
    - Product support capability
    - Training
    - Consulting service

  The evaluator must prioritise different CASE features functionalities according to the organization need.
  The common CASE functionalities are:
    - methodology support
    - graphics
    - input/ output capability
    - dictionary support
    - analysis capability

  For these functional areas, the evaluator should first ask themselves two questions. The first one is can the tool meet the organization needs and the second one is can the tool be 'customised' if necessary?

One major benefit of CASE is more end user interaction in the initial requirement & analysis stage. The evaluator should check if the CASE tool provides sufficient interaction to leverage design & development effort.

- After a go-ahead is received, it is important to select the 'right' first CASE project which should meet the following criteria:
  a) It should be a new project about to start, not one already in progress.
  b) This project should not be too complex or risky. Small or moderate size projects are perfect.
  c) One should also consider how to measure the results due to CASE. Quantifiable data can help to justify further adoption of CASE technology in the organization.
  e.g. end-user feedback, no. of changes, time, manpower spent.

Chapter III

PRODUCT EVALUATION

The process of introducing structured methodology with CASE tools is described in the previous chapter. The CASE tools are mainly for the implementation of a structured methodology in the application software development. The prime objective of using CASE is to further computerise the whole software life cycle to increase the productivity and reduce cost.

The most important criteria of selecting the 'right' CASE tool is how it supports the methodology being used or planned to be used in the organization. The MIS department (user organization) is assumed to be evaluating CASE tools for the first time.

In this chapter, two CASE environments are described & compared:-
IBM's AD/Cycle and Digital's COHESION.
Both of them are made up of a set of CASE tools and an integrated repository. They both support, or at least working towards to support, the entire life cycle.

Our Evaluation Approach

Since AD/Cycle and Cohesion are two different products, a strict comparison of them using a solid framework will limit our discussion on these 2 products and the characteristics of each one may be hidden. We may then run into the risk of comparing apple and orange. But, without a framework, we cannot build a common ground to look at each of them. Therefore what we are going to do here is to describe and evaluate them one by one first. Then we will apply the framework described in Chapter II to evaluate and contrast the front-end analysis/design tool of each of them, namely,

a) EXCELERATOR in IBM's AD/Cycle environment

b) DECDesign in Digital's COHERENT environment

This CASE evaluation framework was suggested by S K Misra (Misra 1990). A checklist is constructed based on that framework. The two tools are compared according to that checklist. It gives an overall feature comparison.

3.1 Functional evaluation of AD/CYCLE

3.1.1 Functional Overview

AD/CYCLE, the shortened form of Application Development Life Cycle, is IBM's framework for application development. This framework consists of an integrated set of application development offerings that assist developers throughout the application development life cycle, from enterprise modelling, to code generation, to ongoing maintenance.

Conceptually, AD/Cycle is comprised of the following two elements:

a) A family of integrated tools that shares application development information throughout the entire life cycle, from the definition of requirements to production and ongoing maintenance.

b) An application development platform that provides the following services for the integration of those tools:

- Services for tools on a PS/2 workstation running under OS/2 EE, and connected to a host

- A consistent interface between users and workstation tools throughout the life cycle, conforming to IBM's System Application Architecture (SAA).

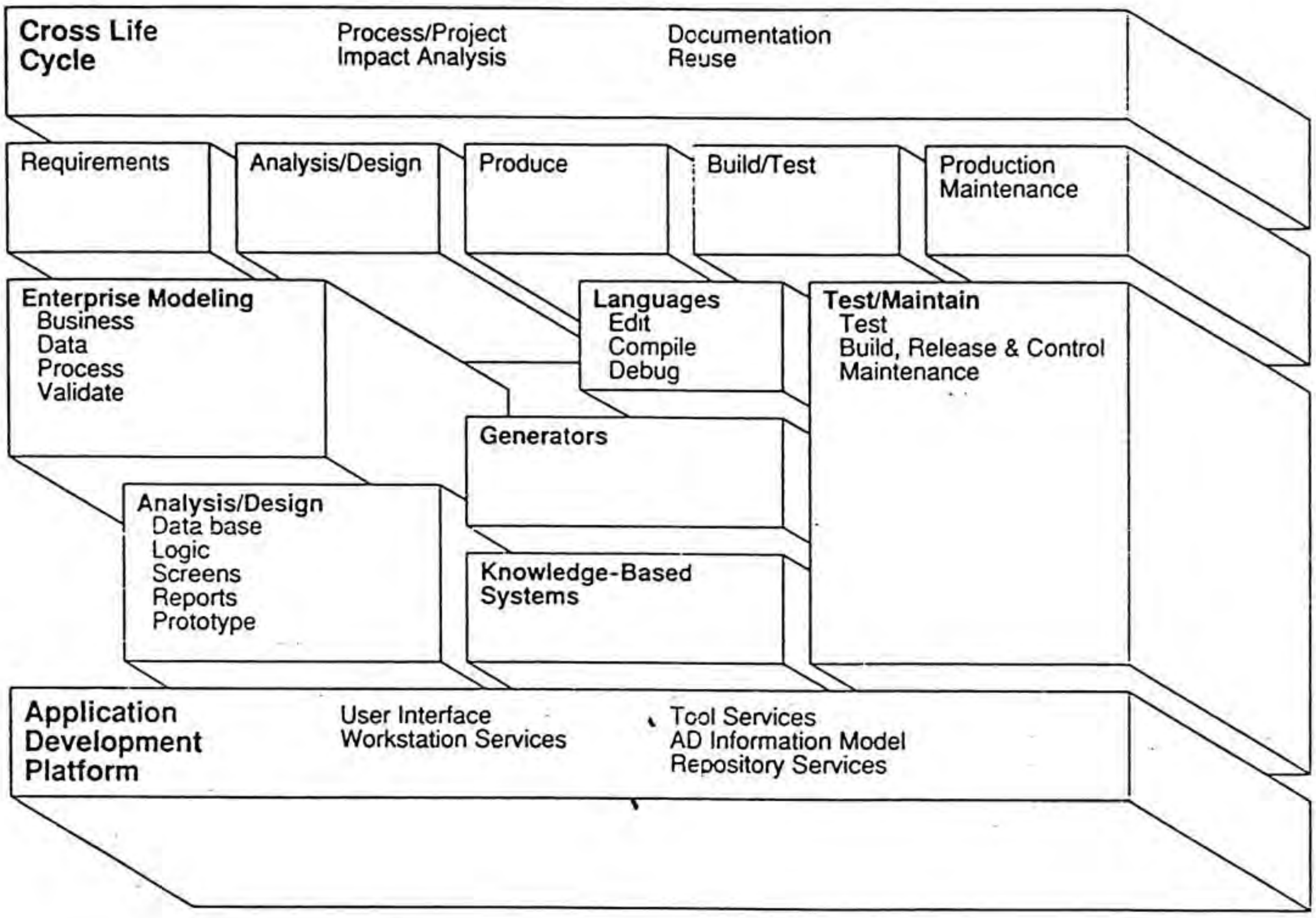
- A set of common tool services, providing library functions, administrative function, and standard operations such as copying, deleting and storing information

- An information model. It presents a common view of application development information

- A common repository and related services which provides a single point of control for defining and sharing access to application development information.

The following diagram shows a general representation of the AD/CYCLE framework.

Fig. 3.1: AD/CYCLE framework



As shown in the above diagram, AD/CYCLE groups the tools into sets that focus on the following types of application development activities:

Cross cycle functions
Enterprise modelling
Analysis and design
Producing applications using third-generation Languages
Producing applications using an application generator
Producing applications using knowledge-based products
Testing, building, and maintaining applications

The actual program products under the AD/CYCLE umbrella are listed below in categories:

A. Cross Life Cycle

1. Application Development Project Support
2. AS/400 Application Development Tools

B. Requirements

1. DevelopMate
2. Index Technology's Excelerator Series
3. KnowledgeWare's Information Engineering Workbench

C. Analysis/Design

1. BACHMAN's Re-Engineering Product Set
2. DevelopMate
3. Index Technology's Excelerator Series
4. KnowledgeWare's Information Workbench

D. Produce

1. C Programming Language
2. COBOL
3. Cross System Product
4. INSPECT for C/370 and PL/I
5. PL/I
6. Query Management Facility
7. REXX Compiler and Library
8. RPG/400
9. SAA Procedures language

E. Build/Test

1. BACHMAN's Re-Engineering Product Set
2. C Programming Language
3. COBOL
4. COBOL and CICS/VS Command-Level Conversion Aid
5. COBOL Structuring Facility
6. INSPECT for C/370 and PL/I
7. ISPF/PDF
8. PL/I
9. Query Management Facility
10. REXX Compiler and Library
11. RPG/400
12. SAA Procedures Language
13. Workstation Interactive Test Tool
14. Software Analysis Test Tool

F. Production Maintenance

1. BACHMAN's Re-Engineering Product Set
2. COBOL Structuring Facility
3. Cross System Product
4. ISPF/PDF
5. Software Analysis Test Tool
6. Workstation Interactive Test Tool

G. AD/Platform

1. Repository Manager/MVS

3.1.2 Functional Description of Selected Products of AD/CYCLE

Many of the products that are currently running in the market are repackaged under the umbrella of AD/CYCLE. The writer wants to pick up some of them which worth special attentions. They are

a. BACHMAN's Re-Engineering Product Set,

b. Index Technology's Excelerator Series,

c. KnowledgeWare's Information Engineering Workbench,

d. IBM's Repository Manager/MVS.

e. IBM's DevelopMate

f. IBM's Cross System Product

g. IBM's COBOL Structuring Facility

They are discussed in details in Appendix A.

3.1.3. How well does AD/CYCLE fit in the System Development Cycle?

The traditional flow of system development cycle with a data repository is shown in the following diagram. In this section, We will try to match the AD/Cycle functionalities with each development phase. Some comments on how well does AD/Cycle perform in each are put forth by the writers.

Fig. 3.2: Flow of System Development

```
------------     -----------     ----------     -------------
|Enterprise|     |Analysis/|     |Produce |     |Test/      |
|Modelling |     |Design   |     ----------     |Maintenance|
------------     |for      |              |     -------------
      |          |Process &|              |           |
      |          |Data     |              |           |
      |          -----------              |           |
      |               |                   |           |
      |               |                   |           |
      |               |     |             |   |       |
      |               |     |             |   |       |
      |               -->|                |<--        |
      --------------->|  Data             |<-----------
                      |  Repository       |
                      |                   |
                      |                   |
```

Each system phase is discussed in the following sections.

3.1.3.1. Enterprise Modelling

In general, enterprise modelling involves analyzing and documenting three facets of the business: the processes that the business performs; the information that these processes use; and the business rules, polices, and practices that control these processes.

AD/CYCLE will support capturing, modelling and analyzing information about

- the organization of the enterprise and its use of information
- high-level business processes
- high-level data requirements
- information obtained from this process will be retained in the Repository Manager. It will be referred or modified in the subsequent development steps like analysis and design.

Comments:

a) The tools in AD/CYCLE which assist in the actual construction of the Enterprise modelling are: DevelopMate, Excelerator and Information Engineering Workbench (IEW). All of them provide graphic interface to assist the user to do the design work.

b) The tool in AD/CYCLE which assists in making decision on the priorities of data and process in the enterprise model is PC PRISM, one of the component of Excelerator. Through the matrix provided by PC PRISM, user can give priority to different processes, according to their significance to the organization. A list of processes in priority sequence will finally come up on which the user can make decision to automate which process first.

c) Overall speaking, AD/Cycle performs well in this area.

3.1.3.2 Analyzing and Designing the application

In adopting the structured analysis and design methodology, we usually have two sets of activities: Process design and data design. AD/CYCLE as a structured framework for application development, also provides these 2 functions.

a. Process Design

Process design includes the design of:

a) Context Diagram - the highest level of data flow diagram

b) Data Flow Diagram - the detailed flow of the application system

c) Structured chart - Based on DFD, analyst will derive structured chart which shows the hierarchy of functional modules.

AD/CYCLE provides:

- Analysis of the user requirement
- Tailoring of diagramming techniques
- Logically complete and validated specification
- Specifications structured for input to application generators

Our comments:

- the AD/CYCLE tools responsible for the above processes are BACHMAN (the new version), Excelerator and Information Engineering Workbench.
- the natural flow of the 'process design' is a stepwise refinement process. When using the structured approach, analyst first defines the top level context diagram and then break it down into detailed data flow diagrams. BACHMAN, Excelerator and IEW provides similar flow of functions on which analysts can design the top level context diagram using the user-friendly graphic

interfaces.
The information is then inherited to the subsequent steps to build the data flow diagrams. Data can thus be passed from top level downwards and data consistency can be guaranteed.

- AD/CYCLE assists the user to build DFD from Context Diagram. However there is no direct way to build Structured Chart from DFD. This is one of the major limitation.

b. Data Design

Data Design, or Business Data Modelling, as some people call it, involves the following steps:

a) Perform the Entity-Relationship Diagrams construction

b) Perform the Conceptual Data Design

c) Perform the Logical Data Design

d) Perform the Physical Data Design

AD/CYCLE provides functions of:

- Graphic interface
- Rule-based design guidance
- Generation of data definition language (DDL) in the physical data design stage.

Our comments:

a) The AD/CYCLE tools responsible for data design are the same as that for process design, viz, BACHMAN, Excelerator and IEW.

b) The four data design stage: E-R diagrams, Conceptual data design, logical data design and physical data design are integrated. Information from one stage is inherited to the subsequent one and data consistency is thus ensured.

c. General comments on process and data design in AD/CYCLE

Since all the three tools, BACHMAN, Excelerator and IEW perform both the process and data design, data can be passed among all these processes. As a result, data design and process design can be cross-checked to ensure data consistency. For instance, all the data appear in the data flow diagram will have corresponding data attribute assigned to it in the data design process. Nothing will be missing out. This is one of the very strong point in AD/CYCLE, which is hardly achieved by manual process when the application is large and complicated.

3.1.3.3 Producing the Application

AD/CYCLE provides the Cross System Product to help generating application & a Knowledge based system.
These applications will be able to mix knowledge-based and high-level language or generator components. Knowledge-based system application components will be stored in the repository and shared throughout the life cycle.

AD/CYCLE will assist the system developers in the following ways:

- Data structure definitions will be generated from data structure design information in the repository for inclusion into source programs and generator specifications.

- Database and file definitions needed in the operational environment will be generated from design information in the repository.

- Panel specifications and generation tools will be provided to assist in the creation of interactive applications.

Our comments:

a) Information can be inherited from the design process to the code generation process via the repository. The code generation period can thus be reduced but quality improved.

b) With the help of code generator, analyst can put more time to the more important phases like analysis and design. Overall productivity can be enhanced.

3.1.3.4 Testing, and Maintaining Applications

AD/CYCLE test tools for interactive applications will capture the activity of test sessions, allowing the same tests to be rerun on for retesting of new applications.
Tools will also provide test coverage measurement and analysis, including an animated display of test execution.

Maintenance of applications developed within AD/CYCLE is simplified. It will support maintenance by allowing designers and developers to use the information in the repository to perform impact analysis, and then use the same tools used for a new application to apply changes to existing application components.

Our comments:

AD/CYCLE cannot support reverse-engineering of process. Over time when it becomes possible, we can extract information from the existing program to build structured charts and then from Structured charts to build data flow diagrams. We can then modify these existing data flow diagrams to develop new DFDs, which will be forward-engineered to Structured charts which act as the input to the code generation. Since this flow is not possible right now, The maintenance of old or existing applications will be a big headache to the organization.

3.1.3.5 Data Repository for the whole development cycle

Over time, Data Repository will integrate all the tools supported under AD/CYCLE by providing a common data interface to which each tool can import and export data.

Our comments:

- Today the applicability of this data repository is still very limited, as most of the data interface to other tools are still under development. For instance, IBM with both Bachman and Index Technology (Excelerator's mother company) form a joint task force to develop the common interface to the data repository. It is still a statement of direction for Repository to integrate the whole AD/CYCLE products under the same umbrella.

- Right now, each product has its own data dictionary (e.g. Bachman, Excelerator or IEW), the complete exchange of information among them is impossible.

3.2 Technological evaluation of AD/CYCLE

Regarding the technological aspects of CASE software, there are a few hot spots to which both researchers and software vendors pay much attention. For instance, nearly all CASE software vendors claim their products to be able or going to be able to support concepts like data and program reuse, integrated data repository or object-oriented programming.

What are the real meanings of these terms? What is the position of Ad/Cycle regarding these technologies? In this section. the writers will try to summarize their understandings on AD/Cycle and present its functions from the technological point of view.

1. Process Management

The process model in AD/CYCLE guides the analyst through the structured development methodology. Such an approach provides a greater degree of consistency in the development process and can enhance the overall quality of application systems.

2. Project Management

There are quite a number of project management tools in the market like lotus-123 or others which keep track of information like work items, schedules, people and cost, etc. However there is no well established interface right now between these tools with the AD/CYCLE architecture.

3. Impact Management

Impact analysis information will enable the analyst to evaluate the effect of specific changes to the components of the application system before the change really occurs.

Right now, the AD/CYCLE repository only provides information of the relationship among the program components. Neither the impact analysis tools nor the interface to this repository is created. These tools have the potential for dramatically improving the change management capability and reducing the maintenance costs.

4. Documentation

The development of applications often includes design, specification, and procedural documentation. It is still a statement of direction for AD/CYCLE to generate documentation using existing application development information from the repository. Eventually the documentation will also be stored in the repository.

5. Reuse

AD/CYCLE supports reuse of programs by re-engineering and reuse of data by reverse-engineering.

AD/CYCLE provides the re-engineering mechanism for the program components by building Structured Charts based on the information of program itself. Analyst then redesigns the structure chart and build the new program. Program reverse-engineering is still not yet available because data flow diagrams cannot be abstracted directly from structured charts. The following diagram shows the difference between reverse engineering and re-engineering.

As a result of this limitation, only program modules which conform to the structured design can be reused without great problems. If we want to modify the structure of a system of modules which involves the redesign of data flow diagrams, AD/CYCLE cannot support it.

Besides program reverse-engineering is not supported by AD/CYCLE, another limitation of Ad/CYCLE in this area is that program re-engineering applies only to 3rd generation language. No assembler programs can be re-engineered.

Data reverse-engineering is supported by AD/CYCLE. Analyst can rebuild the enterprise model based on the existing database specifications and then massage the data model and construct the new data specification.

Fig. 3.3: Forward & Reverse Engineering

```
                    | Enterprise|Information |Technology|Detailed
                    | Model     |System Model|Model     |Spec.
--------------------|-----------|------------|----------|--------
                    |           |            |          |
                    |           |            |          |
Forward Engineering | --------->|----------->|--------->|------>
                    |           |            |          |
Reverse Engineering | --------->|----------->|--------->|------>
                    | <---------|<-----------|<---------|<------
                    |           |            |          |
Re-Engineering      |           |            |          |------>
                    |           |            |          |<------
```

Data reverse-engineering capability is provided by Bachman and process re-engineering, is provided by COBOL Structuring Facility.

6. User Interface

AD/CYCLE is doing a good job in this aspect. Both IBM- and vendor-supplied workstation based tools will conform to IBM's SAA (System Network Architecture) CUA (Common User Access). Future tool development will also base on CUA as the user interface standard.

7. Repository Services

This is the most important deliverable in AD/CYCLE architecture. In fact. many AD/CYCLE tools are just the ready-made tools available in the market long before the emergence of the AD/CYCLE concept. What IBM adds her value in it is to use Repository to try to integrate all the different tools to form a complete platform to assist user from the very beginning of user requirements down to the program testing and maintenance.

AD/CYCLE repository contains the information about application development activities and components, & the information of the components themselves.

AD/CYCLE repository helps to define, control and maintain data described in E-R format. It also provides access control to application objects.

The repository has already developed by IBM and is now being marketed. But the functions mentioned above are still limited to one or two tools. For instance, Bachman and Excelerator still cannot import and export data directly to Repository without doing data format conversion. What is more is that these tools have their own data dictionary which make the concern of information storage even more complicated.

We feel that unless the interface to these common tools are enhanced, the applicability of Repository is still very limited, despised that the concept is brilliant.

8. The E-R Concept

This technique provides for the definition of entities (data elements, programmers, parts of a business model, etc.) the relationships among them, and constraints on the information described by these entities and relationships.

The ER modelling technique is well supported by AD/CYCLE and the information output can be used as input to subsequent data design like conceptual data design.

9. The Object Concept

Conceptually, an object contains both data and process. Objects provide the means for representing both the information that tools share, and the functions that tools can perform against that information. Since it has both data and process defined within the same module, the interface to other modules may be kept to a minimum and we can expect that object-oriented programming can greatly enhance the reusability of programs.

The AD/CYCLE product, Bachman supports Object-oriented modelling and repository supports the storage of objects. But object-oriented programming language like ADA, C++, or SmallTalk are still not supported under AD/CYCLE. Furthermore DB2, which Bachman interfaces to, is not an OOP product. We are expecting to see a dramatic increase in program reusability when OOP is supported under AD/Cycle.

10. AD/CYCLE's Scope

AD/CYCLE only provides process design and data design. This is enough for the automation of one particular department. But for some cases which involve the design of applications across several countries, like the implementation of a new marketing information system for a multi-national company, issues like network design, distributed database design, and security must be considered during the application design.

Currently there are no products in AD/CYCLE which can handle network design and distributed database design. The automation provided by AD/Cycle to those large application development project is still limited.

3.3 Functional evaluation of COHESION

The CASE environment introduced by Digital is called COHESION. Like AD/ Cycle, its goal is to provide a wide spectrum of CASE tools for the software developers throughout the entire life cycle.

3.3.1 DIGITAL's CASE integration framework

a) The objective is to integrate Digital and independent CASE vendor tools. On the back end side of the development process, information can be shared among different tools. On the front end side, Digital's environment provides a common user interface independent of the CASE tool.

b) Network Application Support (NAS) is the network platform. The Digital CASE tools are primarily running on Digital's operating system - VMS.

c) The Digital CASE environment addresses three types of integration needs:

- Presentation Integration

  DEC Window provides the common user interface allowing the software developers to interact with different tools in the same way. DEC Window is based on X-Window.

- Data Integration

  It allows different CASE tools to share data. So all developments are able to utilize a common set of data definition through Digital's repository - CDD/Plus.

- Control Integration

  It regulates how CASE tools interact with each other. It addresses areas like: system reconstruction, version control (e.g. prepare a new version for a released software), change control & discrepancies among the deliverables.

d) CDD/Plus, the distributed CASE repository, provides both Data Integration & Control Integration.

   An example of data integration through CDD/Plus:
   The data model is first built using DEC Design (an analysis & design CASE tool). The same data definition is used in the relational database e.g. VAX Rdb VMS. The application software is generated using the same data definitions. So the whole development process uses the same data definitions.

e) ATIS (A Tools Integration Standard) is an object-oriented interface allowing saving, accessing & managing information in a common repository. This information includes list, current status and interrelationship of all data objects of a software project. CDD/Plus will support ATIS in its next major release.

3.3.2 Development life cycle support

Digital also emphases the entire life cycle support. It groups the CASE or related products into two main phases:

a) Requirement & Analysis
It is the front end of the development life cycle which includes:-
Requirement analysis; Requirement definition;
and Analysis & design.

b) Implementation
It covers the back end processes like:
code generation; 4GL; languages; code management;
& system builder.

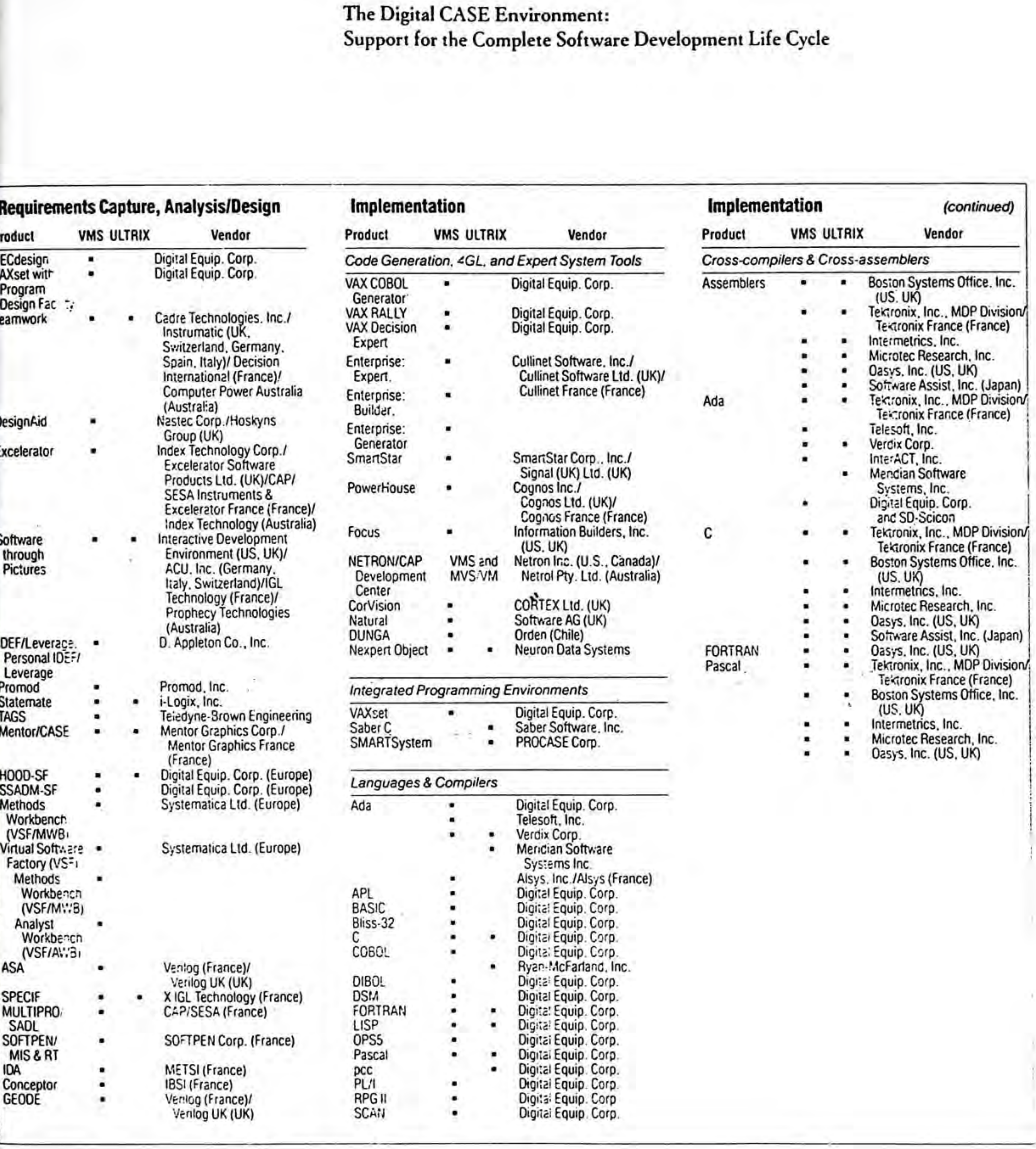
Fig. 3.4 on the next page lists the available products in the COHESION environment according to differing needs in the development life cycle.

In the figure, it appears that there are many CASE products in the COHESION environment. But many of them exist before COHESION emerges. They are bundled into the COHERENT environment (e.g. programming languages, project management). The two CASE products by Digital are DECDesign & CDD/Plus. In the product description section, they are described in details.

Digital also integrates third party CASE tools into the environment: e.g. TEAMWORK, Excelerator.

Excelerator can be bridged to CDD/Plus. It is available under the VMS operating system.

Fig. 3.4: Products in Cohesion



The Digital CASE Environment:
Support for the Complete Software Development Life Cycle

Requirements Capture, Analysis/Design

roduct	VMS	ULTRIX	Vendor
ECdesign	●		Digital Equip. Corp.
AXset with Program Design Facility	●		Digital Equip. Corp.
eamwork	●	●	Cadre Technologies. Inc./ Instrumatic (UK, Switzerland, Germany, Spain, Italy)/ Decision International (France)/ Computer Power Australia (Australia)
esignAid	●		Nastec Corp./Hoskyns Group (UK)
xcelerator	●		Index Technology Corp./ Excelerator Software Products Ltd. (UK)/CAP/ SESA Instruments & Excelerator France (France)/ Index Technology (Australia)
oftware through Pictures	●	●	Interactive Development Environment (US, UK)/ ACU. Inc. (Germany, Italy, Switzerland)/IGL Technology (France)/ Prophecy Technologies (Australia)
DEF/Leverage. Personal IDEF/ Leverage	●		D. Appleton Co., Inc.
Promod	●		Promod, Inc.
Statemate	●	●	i-Logix, Inc.
TAGS	●		Teledyne-Brown Engineering
Mentor/CASE	●	●	Mentor Graphics Corp./ Mentor Graphics France (France)
HOOD-SF	●	●	Digital Equip. Corp. (Europe)
SSADM-SF	●		Digital Equip. Corp. (Europe)
Methods Workbench (VSF/MWB)	●		Systematica Ltd. (Europe)
Virtual Software Factory (VSF)	●		Systematica Ltd. (Europe)
Methods Workbench (VSF/MWB)	●		
Analyst Workbench (VSF/AWB)	●		
ASA	●		Venlog (France)/ Venlog UK (UK)
SPECIF	●	●	X IGL Technology (France)
MULTIPRO/ SADL	●		CAP/SESA (France)
SOFTPEN/ MIS & RT	●		SOFTPEN Corp. (France)
IDA	●		METSI (France)
Conceptor	●		IBSI (France)
GEODE	●		Venlog (France)/ Venlog UK (UK)

Implementation

Product	VMS	ULTRIX	Vendor
Code Generation, 4GL, and Expert System Tools			
VAX COBOL Generator	●		Digital Equip. Corp.
VAX RALLY	●		Digital Equip. Corp.
VAX Decision Expert	●		Digital Equip. Corp.
Enterprise: Expert.	●		Cullinet Software, Inc./ Cullinet Software Ltd. (UK)/ Cullinet France (France)
Enterprise: Builder.	●		
Enterprise: Generator	●		
SmartStar	●		SmartStar Corp., Inc./ Signal (UK) Ltd. (UK)
PowerHouse	●		Cognos Inc./ Cognos Ltd. (UK)/ Cognos France (France)
Focus	●		Information Builders, Inc. (US, UK)
NETRON/CAP Development Center	VMS and MVS/VM		Netron Inc. (U.S., Canada)/ Netrol Pty. Ltd. (Australia)
CorVision	●		CORTEX Ltd. (UK)
Natural	●		Software AG (UK)
DUNGA	●		Orden (Chile)
Nexpert Object	●	●	Neuron Data Systems
Integrated Programming Environments			
VAXset	●		Digital Equip. Corp.
Saber C		●	Saber Software, Inc.
SMARTSystem		●	PROCASE Corp.
Languages & Compilers			
Ada	●		Digital Equip. Corp.
	●		Telesoft, Inc.
	●	●	Verdix Corp.
		●	Meridian Software Systems Inc.
	●		Alsys, Inc./Alsys (France)
APL	●		Digital Equip. Corp.
BASIC	●		Digital Equip. Corp.
Bliss-32	●		Digital Equip. Corp.
C	●	●	Digital Equip. Corp.
COBOL	●		Digital Equip. Corp.
		●	Ryan-McFarland, Inc.
DIBOL	●		Digital Equip. Corp.
DSM	●		Digital Equip. Corp.
FORTRAN	●	●	Digital Equip. Corp.
LISP	●	●	Digital Equip. Corp.
OPS5	●		Digital Equip. Corp.
Pascal	●	●	Digital Equip. Corp.
pcc		●	Digital Equip. Corp.
PL/I	●		Digital Equip. Corp.
RPG II	●		Digital Equip. Corp
SCAN	●		Digital Equip. Corp.

Implementation *(continued)*

Product	VMS	ULTRIX	Vendor
Cross-compilers & Cross-assemblers			
Assemblers	●	●	Boston Systems Office, Inc. (US, UK)
	●	●	Tektronix, Inc., MDP Division/ Tektronix France (France)
	●	●	Intermetrics, Inc.
	●	●	Microtec Research, Inc.
	●	●	Oasys, Inc. (US, UK)
	●	●	Software Assist, Inc. (Japan)
Ada	●	●	Tektronix, Inc., MDP Division/ Tektronix France (France)
	●		Telesoft, Inc.
	●	●	Verdix Corp.
	●		InterACT, Inc.
		●	Meridian Software Systems, Inc.
	●		Digital Equip. Corp. and SD-Scicon
C	●	●	Tektronix, Inc., MDP Division/ Tektronix France (France)
	●	●	Boston Systems Office, Inc. (US, UK)
	●	●	Intermetrics, Inc.
	●	●	Microtec Research, Inc.
	●	●	Oasys, Inc. (US, UK)
	●	●	Software Assist, Inc. (Japan)
FORTRAN	●	●	Oasys, Inc. (US, UK)
Pascal	●	●	Tektronix, Inc., MDP Division/ Tektronix France (France)
	●	●	Boston Systems Office, Inc. (US, UK)
	●	●	Intermetrics, Inc.
	●	●	Microtec Research, Inc.
	●	●	Oasys, Inc. (US, UK)

Implementation *(continued)*

Product	VMS	ULTRIX	Vendor
Cross-development Tools			
VAXELN Ada Toolkit	VAXELN		Digital Equip. Corp.
Debugging Tools			
VAX Debug	•		Digital Equip. Corp.
dxdb		•	Digital Equip. Corp.
Editors			
VAX Language-Sensitive Editor (VAXset)	•		Digital Equip. Corp.
dxnotepad		•	Digital Equip. Corp.
xedit		•	Digital Equip. Corp.
vi		•	Digital Equip. Corp.
Source Code Analysis			
VAX Source Code Analyzer (VAXset)	•		Digital Equip. Corp.
cxref, lint, ctrace		•	Digital Equip. Corp.
Code/Configuration Management			
VAX DEC/Code Management System (VAXset)	•		Digital Equip. Corp.
sccs		•	Digital Equip. Corp.
RCS		•	Rochester Institute of Technology
CCC	•	•	Softool/ K3 Software Srvs. Ltd. (UK)
Lifespan	•		Yard Software Systems (UK)
System Builders			
VAX DEC/Module Management System (VAXset)	•		Digital Equip. Corp.
make, S5make		•	Digital Equip. Corp.

Integration and Testing

Product	VMS	ULTRIX	Vendor
VAX DEC/Test Manager (VAXset)	•		Digital Equip. Corp.
VAX Performance and Coverage Analyzer (VAXset)	•		Digital Equip. Corp.
cflow, perfmon, grep		•	Digital Equip. Corp.
dxdiff		•	Digital Equip. Corp.

Project Management

Product	VMS	ULTRIX	Vendor
VAX Software Project Manager	•		Digital Equip. Corp.
Project2	•		PSDI, Inc./ PSDI (UK) Ltd. (UK)
QUICKNET Professional	•		PSDI, Inc./ PSDI (UK) Ltd. (UK)
Artemis	•		Metier, Inc./ Metier Management Systems Ltd. (UK)/ Metier Systems (France)
Planner	•		Productivity Solutions, Inc./ Boston Systems Office (UK)
Metratech		•	Yard Software Systems (UK)
Acuity		•	Computer Cognition, Inc.
APECS/8000		•	ADP Network Services, Inc.
VUE Project Manager		•	National Information Systems, Inc.

Documentation

Product	VMS	ULTRIX	Vendor
DECwrite	•	•	Digital Equip. Corp.
VAX DOCUMENT	•		Digital Equip. Corp.
Interleaf TPS	•	•	Interleaf, Inc./ Interleaf (UK) Ltd. (UK)
nroff, troff, spell, diction style		•	Digital Equip. Corp.
Uniplex		•	Uniplex Ltd. (UK)

Communication

Product	VMS	ULTRIX	Vendor
VAX Notes	•		Digital Equip. Corp.
Mail	•	•	Digital Equip. Corp.
VAX VTX Videotex	•		Digital Equip. Corp.

Maintenance and Support

All the above products can be used when modifying or enhancing your application. From bug-fixes to major new releases, the Digital CASE Environment helps you maintain software better and faster.

Tools for Multiplatform Applications

Product	VMS	ULTRIX	MS-DOS	Macintosh	Vendor
DECwindows	•	•	•	•	Digital Equipment Corporation

Tools for Device-independent Applications

Product	VMS	ULTRIX	Vendor
DEC GKS	•	•	Digital Equipment Corporation
DEC PHIGS	•	•	Digital Equipment Corporation
DECforms	•		Digital Equipment Corporation

Additional Tools for Information Management Applications

To assist in building and maintaining data-centered application systems, the Digital CASE Environment provides the tools listed in Table A — including core CASE productivity tools, code generation capabilities, and fourth-generation languages — as well as the following database and information management tools.

Product	VMS	Vendor
DECintact	•	Digital Equipment Corporation
Rdb/VMS with VAX SQL	•	Digital Equipment Corporation
VAX ACMS	•	Digital Equipment Corporation
VAX CDD/Plus	•	Digital Equipment Corporation
VAX Data Distributor	•	Digital Equipment Corporation
VAX DATATRIEVE	•	Digital Equipment Corporation
VAX DBMS	•	Digital Equipment Corporation
VAX TEAMDATA	•	Digital Equipment Corporation
VAXlink	•	Digital Equipment Corporation
VIDA with IDMS/R	•	Digital Equipment Corporation

3.3.3 **Function description on selected products**

Five Digital products in the COHESION environment are described. They address different stages in the development life cycle:

Stage in Life Cycle	Product
1) Cross Life Cycle	CDD/Plus
2) Requirement, Analysis & Design	DEC Design
3) Coding (Editor)	LSE
4) Code Management	CMS
5) System Builder	MMS

3.3.3.1 CDD/Plus

VAX Common Data Dictionary/ Plus (VAX CDD/Plus) is the common repository in Digital's CASE environment.

a. INTEGRATION

It supports tools throughout the life cycle from Analysis to Coding. Examples of tools that integrate with CDD/Plus are listed below:

1) Analysis & Design
   - DEC Design
   - Excelerator
   - IDEF/ Leverage,
     an automated database design tool for VAX Rdb.

2) 4GL
   - VAX RALLY
   - Smart Star (SQL based)
   - FOCUS

3) Code Generator Tools

- VAX COBOL Generator
- Foundation,
  facilities to automate & support the development of Rdb applications.
- NETRON/CAP Development Center,
  for VMS or IBM's MVS, CICS % DB2.
- Powerhouse

4) Languages

- Ada, BASIC, C, COBOL, FORTRAN, PASCAL,
  PL/I, RPG II. (all on VAX)

How they integrates with CDD/Plus:
A data definition in CDD/Plus can be included in a program at compile time. In the program source file, there are lines:-

```
COPY "xxxxx" FROM DICTIONARY
```

5) Others

- VAX Rdb
- VAX DBMS

b. LOGICAL STRUCTURE

In CDD/Plus, data definitions consist of field definitions & record definitions. A record consists of a grouping of fields.
The dictionary can be organised by:

- application
- organizational entity
- individual user
- or a combination of the above

All CDD/Plus dictionaries are organised by path. It is similar to a VMS file specification.
e.g. EMPLOYEES.SALARIED.EMPLOYEE_REC
represent the path that leads to the record definition EMPLOYEE_REC

c. NAVIGATION

There is no direct graphical support in CDD/Plus. (Although it can be linked to a graphical analysis/design tool like DEC Design.)

There is a screen editor in CDD/Plus. User can use EDIT FIELD or EDIT RECORD. The editor can also display all the current definitions in the current directory.

The editor is useful when the user is not yet familiar with the syntax.

d. OTHERS

Since CDD/Plus only deals with the dictionary, the criteria related to process are not applicable. It depends on the Analysis & Design CASE tool being used.

CDD/Plus keeps track of dictionary usages. It can also check & monitor dictionary integrity. e.g. VERIFY in CDD/Plus gives a report of any corrupted definition & condition of the dictionary.

3.3.3.2 DECDesign

DECDesign provides a DECwindow based graphic environment for the Analysis & Design phase of the development life cycle. It supports process and data modelling with integrated techniques. It can interface with CDD/Plus. It runs on the VMS operating system.
Detail feature description of DECdesign is in the last section on feature comparison.

3.3.3.3 LSC/SCA

LSC/SCA stands for Language Sensitive Editor/ Source Code Analyzer.

LSC is an editor that has knowledge of the syntax of Digital's programming languages. It helps to ensure that the right syntax and punctuation are used.

Features of LSC include:

- Source Code template
- support Pseudo Code entry
  It can mix free text with the programming language.
- viewing code at various levels of detail
- interface to VAX CMS (Code Management System)

SCA allows interactive inquiries about program structure, including cross-reference information, calling structure, and where and how often different program elements are used.
LSC/SCA supports most programming languages on VAX VMS. e.g. BASIC, COBOL, Ada, Pascal, etc.
Although LSC/SCA is not a Code Generator, it is a useful tool for programmers in the coding phase.

3.3.3.4 CODE MANAGEMENT SYSTEM (CMS)

It is an automated file librarian that tracks revisions of source code, documentation, data files, test files, system build descriptions, & requirement documents. It solves a portion of the complex coordination problem by keeping track of everything happened to project files during development.

It automatically records every change, who made it, and when.
As a result, developers are always working with the most up-to-date files.

3.3.3.5 Module Management System (MMS)

It uses a script to automatically build systems and documentation using the latest versions of components. When programmers make changes, it can rebuild the system automatically.

3.4 FEATURE COMPARISON BETWEEN EXCELERATOR (AD/CYCLE) AND DECDESIGN (COHESION)

A number of areas are selected as the basis of comparing Analysis & Design tools in an integrated environment. It is based on the framework suggested by S K Misra.

The following sections describe its features according to the following framework:

- Methodology(technique) support
- Integration with Repository
- Integration with other tools
- User interface
- Graphics
- Validation/ Analysis
- Ease of use, Help
- Customization

3.4.1 DECDESIGN UNDER COHESION

a. METHODOLOGY SUPPORT

It supports the Yourdon method and the Gane & Sarson method. It also supports the Ward/ Mellor extension for real-time modelling.

b. INTEGRATION WITH REPOSITORY

In the DECdesign environment, it is possible to update to or from CDD/Plus repository. (This feature is useful when changes occur either in DECdesign or CDD/Plus.

There is a CDD/Plus submenu in DECdesign. It supports the following functions:

- Validate Link,
  check between the aggregate DDE (in DECdesign) and a public Dictionary record (in CDD/Plus) with the same name.

- Export & Link,
  export data from a DDE to a field/ record in the public dictionary.

- Import & Link,
  similar to Export but reversed.

- Examine CDD Definition

- Delete Link

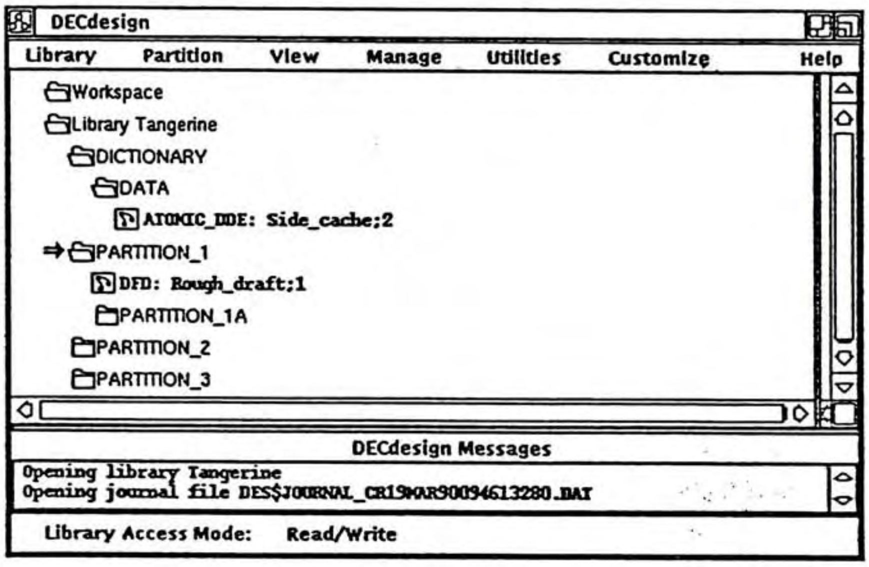
c. INTEGRATION WITH OTHER TOOLS

DECdesign is integrated with LSE/SCA and the VAX COBOL Generator.

d. USER INTERFACE & GRAPHICS

The user interface is organised on partitions, views & library. Fig. 3.5 shows an example of their organizations.
Partitions & views are icon driven. A list of different possible icons is shown in Fig. 3.6.

Fig. 3.5: Hierarchy of Partitions and Views



```
Workspace
Library Tangerine
  DICTIONARY
    DATA
      ATOMIC_DDE: Side_cache;2
⇒ PARTITION_1
    DFD: Rough_draft;1
    PARTITION_1A
  PARTITION_2
  PARTITION_3
```

```
Opening library Tangerine
Opening journal file DES$JOURNAL_CR19MAR90094613280.DAT
```

Library Access Mode: Read/Write

Fig. 3.6

Understanding Icons

The DECdesign session manager window and the user registry dialog box contain icons that provide information about the views and partitions in a library. The following list describes the icons:

Partition icon—An open file folder
Indicates an expanded partition. When highlighted, this icon indicates that another user made a change that affects the partition or its contents.

Partition icon—A closed file folder
Indicates a contracted partition. When highlighted, this icon indicates that another user made a change that affects the partition or its contents.

⇒ *Default partition icon—An arrow pointing to the right*
Indicates the default partition. (Appears in the library area.)

View icon—A miniature graphics window
Indicates a view (not necessarily a graphics view).

Modified view icon—A view icon with a paper clip symbol
A view icon with the addition of the paper clip symbol indicates that you have modified the view. (Appears in the workspace area.) When highlighted, this icon indicates that another user made a modification to the view and that the modification is not contained in your version.

Deleted view icon—A view icon with a diagonal line
A view in the library that has been deleted. (Appears in the library area.)

✓ *Fetched view icon—A checkmark*
A checkmark beside a view in the library indicates that you fetched the view to the workspace. The checkmark appears beside the version of the view you fetched. For example, if you fetch version 2 of a view that has five versions, and you contract the display of the view, you will not see the checkmark beside version 5; you must expand the display to see the checkmark beside version 2.

✓ *Open edit window icon—A checkmark*
A checkmark beside a view in the workspace indicates that you have opened the view's edit window.

Read-only view icon—Reading spectacles
A view in the workspace that you cannot modify. (Appears in the workspace area.) Also, indicates a read-only user account in the user registry.

Enabled account icon—Key
An enabled user account in the user registry. (Appears in the user registry dialog box.)

Deleted account icon—Deleted key
A disabled (deleted) user account in the user registry. (Appears in the user registry dialog box.)

A checklist summarising the diagramming interface of DECdesign is shown in the following table (The scores are based on our understanding of the product & confirmation with the supplier):

<u>Diagramming interface score of DECdesign</u>

	Score
Image primitives:	
1. Types of primitives supported	Good
2. Meet common methodology requirement?	Yes
3. Meet real-time modelling requirement?	Yes
Graph creation:	
1. Easy to create diagram?	very easy
2. Ease of locating/naming diagram objects?	very easy
Graph editing:	
1. Easy to modify diagram?	very easy
2. relocation of diagram?	very easy
3. Support relocation of a whole scope?	Yes
4. Can rename graph object?	Yes
5. UNDO for graph editing possible?	Yes
6. Scaling graph objects?	Yes
Graph viewing:	
1. Rotation of diagram?	No
2. Zooming diagram?	Yes
3. Support of windowing?	Yes
4. Manually controlled paging?	Yes
5. Graph preview?	Yes

DECdesign meets most of these requirements.

e. VALIDATION/ ANALYSIS

It can automatically check syntactical and semantical design errors.

f. EASE OF USE, HELP

Interactive on-line help is available.

g. CUSTOMIZATION

Not very much. Customization is restricted to Digital's selected user sites.

h. OTHERS

DECdesign architecture is based upon two technologies: object-oriented and hyper-information. The object-oriented technology provides a better support for different modelling techniques. The hyper-information provides navigation facilities.

3.4.2 EXCELERATOR UNDER AD/CYCLE

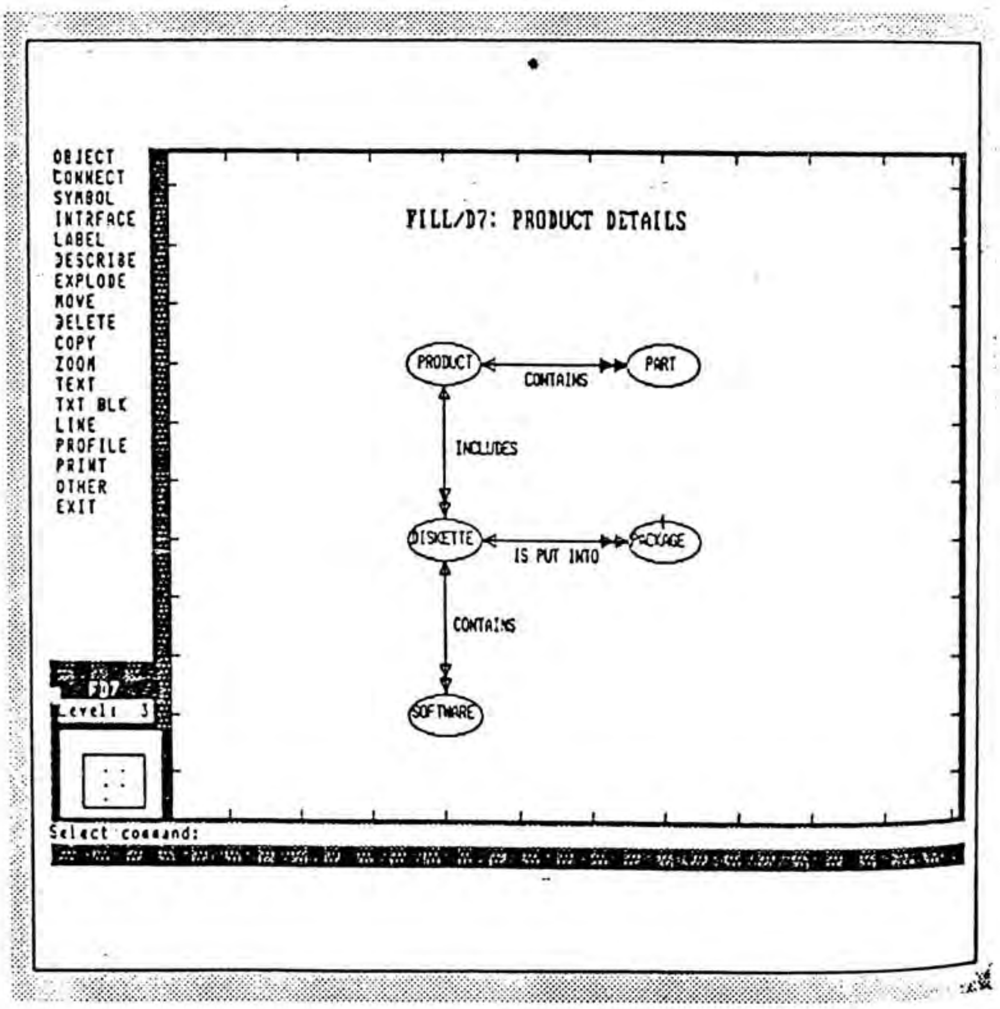
a. METHODOLOGY SUPPORT

Excelerator does not specify the methodology it supports. The product was not designed for any specific methodology.

It has its pros & cons:
The advantage is that the product is very flexible, the disadvantage is it would be difficult for the product to provide verification & checking according to the methodology used by the organization.

For example, it supports all the graphic primitives of the Yourdon method & the G&S method.
An Excelerator Data Flow Diagram is shown in Fig 3.7.

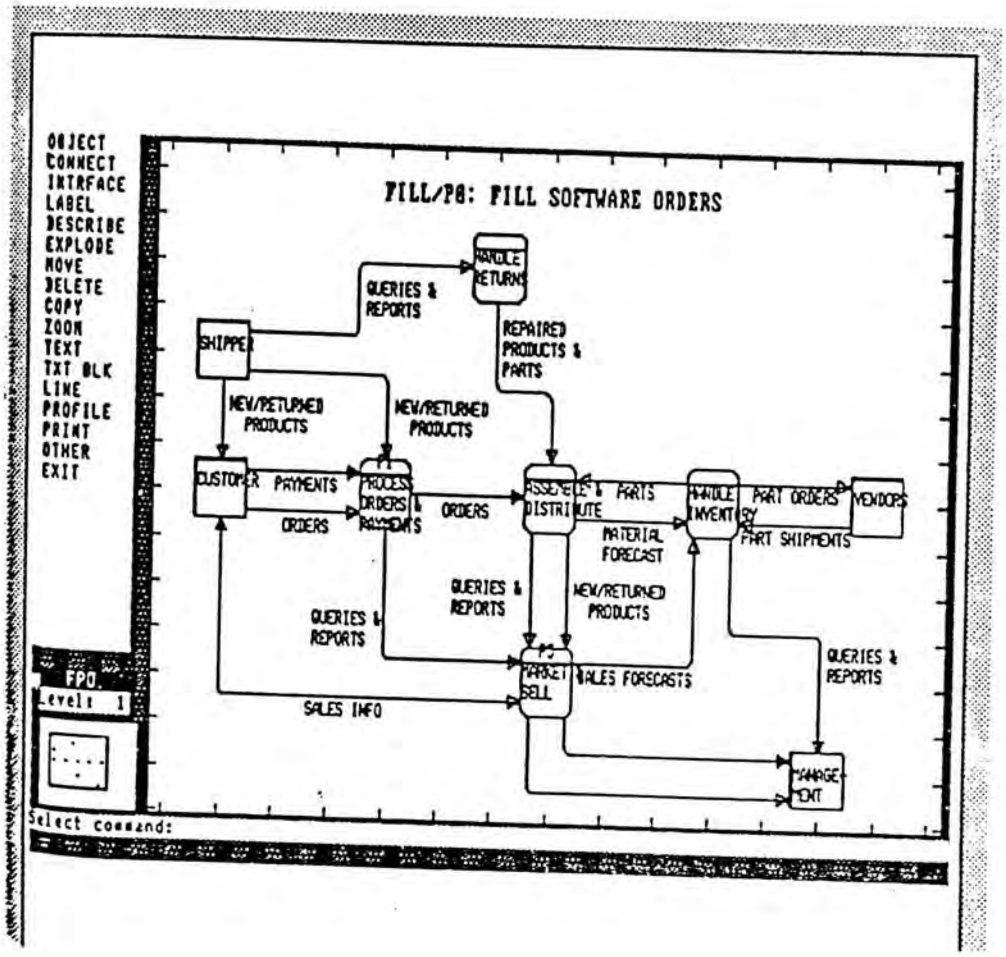
Fig. 3.7: An Excelerator Data Flow Diagram



Select command:

Fig 3.8 shows a Data Model diagram in G&S notation.

Fig. 3.8



The user can use CUSTOMISER to create additional primitives. The standard Excelerator product model is EX/IS. There is an enhanced model for Real-time modelling called EX/RTS.

b. INTEGRATION WITH REPOSITORY

Although Excelerator is running on DOS, it is integrated with the IBM AD/Cycle Repository environment.

c. INTEGRATION WITH OTHER TOOLS

There are two examples:

a) Integration with IBM DB2,
the designer can transform Excelerator logical design into physical design data.

b) Integration with a 4GL - CSP.

d. Operating system (O.S.) & HARDWARE PLATFORM

Excelerator current version is DOS based. An OS/2 version is planned latter this year.
Although the product can only support single user now, a LAN version is planned.

The application software produced is not restricted by the fact that Excelerator runs on DOS. For example, Excelerator can be used to design programs that are to be run on MVS.

e. USER INTERFACE & GRAPHIC

The input devices can be keyboard, mouse.
Plotters & printers are supported as its output device.

The checklist summarising the diagramming interface of EXCELERATOR is shown in the following table:

Diagramming interface score of EXCELERATOR

	Score
Image primitives:	
1. Types of primitives supported	Good
2. Meet common methodology requirement?	Yes
3. Meet real-time modelling requirement?	Yes
Graph creation:	
1. Easy to create diagram?	O.K.*
2. Ease of locating/naming diagram objects?	mediate
Graph editing:	
1. Easy to modify diagram?	mediate
2. relocation of diagram?	very easy
3. Support relocation of a whole scope?	No #
4. Can rename graph object?	Yes
5. UNDO for graph editing possible?	Yes
6. Scaling graph objects?	Yes
Graph viewing:	
1. Rotation of diagram?	No
2. Zooming diagram?	Yes
3. Support of windowing?	Limited+
4. Manually controlled paging?	Yes
5. Graph preview?	No

* Icon & pull-down menu are not provided in the current DOS version although they are promised in the future OS/2 version.

# In editing the graph, it is not possible to move a set of graphical elements as one object.

+ It is not a window based product. So one cannot expect windowing capabilities similar to WINDOW 3.0 or X-WINDOW.

f. VALIDATION

A large part of the checking still relies on the analyst/ programmer. So the programmer is assumed to be well trained in the methodology used.
Excelerator can generate some reports, e.g. Level Balancing report, to help the programmer to check the final design. But there is no check during the design process. So a novice programmer may produced a lot of errors without any prompt or warning from Excelerator.

g. EASE OF USE, HELP

According to user feedback, Excelerator is easy to learn. On-line help is available.

h. CUSTOMIZATION

Customization is available in CUSUMISER of Excelerator. Customization capability helps to adopt tools to the organization's own unique requirements.

3.4.3 COMPARISON SUMMARY

Since Excelerator was first developed before DECdesign, the graphical features & ease of use are not so strong as the DECdesign.

In methodology support, each has its own approach:
Excelerator is very flexible but there is not much validation according to a methodology;
DECdesign provides more checking according to the methodology - e.g. Yourdon, G&S.

Hardware platform:
Excelerator runs on a normal P.C. while DECdesign has to run on a powerful DEC-window machine (e.g. RISC) to give reasonable response time.

It is not fair to say which tool is better. It depends what the user require. Also, if the department is already a pure 'IBM shop', it would be not possible to use the COHERENT environment & vice versa.

Chapter IV

APPLICATION OF CASE TECHNOLOGY IN HONG KONG

In the previous chapters, the latest CASE technology is reviewed. In this chapter, we will examine the adoption of the CASE technology in Hong Kong. Since CASE is a pretty new area here, the secondary data available is very limited. So our discussion is mainly based on the results of a questionnaire survey and interviews.

4.1 Mailing Survey

The purposes of this mailing survey is to find out:

a) the awareness and usage of CASE tools in Hong Kong, and
b) the feedback from the CASE users about these CASE tools

Questionnaires were mailed to the analyst programmers of large MIS departments in Hong Kong. We sent out 85 questionnaires and 20 were collected. Please refer to appendix B for the actual result.

4.1.1 Survey Result Summary

We understand that a size of 20 respondents cannot be considered as a representative sampling. But to a certain extent, it gives us hints on the current situations of CASE application in Hong Kong. The interpretation of the questionnaire raw data is presented below.

Before installation of CASE products, we find that over 50% of respondents said that they have problems in Program maintenance. This matches the general understanding that maintenance effort, without the help of any case tools, would cost quite a large portion of the total system development time.

When they are asked about the area to automate, 75% of respondents pick up the system design phase. This means they understand the importance of system design and their current design can be further improved. We also believe most of them understand the benefits of a structured methodology. It is only the work that required to produce those intermediate deliverables like Data Flow Diagram, Structured Chart without the help of a CASE tool is so tedious that stop them from implementing such structured methodology.

After installation of case tools, we find that only 35% of respondents find improvements. This is not surprising at all as benefits may not be immediately observed due to the learning curve effect. One written comment said that it is because they have had a too high expectation beforehand which make them frustrate after the installation.If an EDP shop aims at some long term benefits like program maintainability, reusability and better documentation, they will find it easier to justify the adoption of CASE tools.

Some of the written comments from the collected questionnaires are interesting and are quite representing the prevailing thinking of CASE usage in Hong Kong. Therefore we extract them out and list below:

- "The processing speed (of CASE tool) is slow"
- "People expect CASE products to give a magic solution"
- "Case products consume a lot of CPU resource."

Based on the 20 respondents, we find that the EDP shops that are using CASE products have an average size of 85 staff. To a certain extent, it brings out a point that medium to large EDP shops tend to adopt the CASE technology to assist their system development work.

4.2 Interview Result Summary

People related to the CASE technology in different aspects were interviewed. So we can investigate the CASE situation from different perspectives. There are users' views as well as consultants' view and we hope this will give us a more balanced picture. We summarise the four interview results below. For detailed interview reports, please refer to Appendix D.

4.2.1 Interview IBM, as a CASE user - Result summary

The EDP shop in IBM Hong Kong has about 35 application programmers and analysts. They are using MVS/XA under 3090-400E. They use IMS DB/DC as their database and online interface. Their system analyst William Yung told us that many applications are developed in other countries and tailored in Hong Kong. As for some locally developed applications, they will involve end users in formulating the user requirements.

They started the CASE pilot project last year by forming a task force to study Excelerator. After five months' intensive study, tests and discussions with end users, Excelerator was found to satisfy most of the testing criteria. It was then put into production this year.

William said to us that they considered user-friendliness as the most critical factor in selection. Since Excelerator is relatively simple to use and it provides sufficient online help information, IBM chose it as their first CASE tool.

Following the introduction of its first CASE tool, IBM realigned its supporting structure by appointing one analyst to be the overall coordinator for the implementation of Excelerator. His main duty is to ensure the smooth transition and to provide technical support to other analyst programmers. He will also conduct internal training sessions to other application programmers and analysts.

While the implementation of CASE tool is going on smoothly, IBM plans to start another project soon. The new-versioned Bachman will become available at late 1991 which provides the function of process design on top of its existing data design. This time William said that they are more comfortable to do this pilot test despite the fact that Bachman is a more complicated product than Excelerator. Here we can see that a step-by-step implementation approach is a key to success CASE installation.

4.2.2 Interview DEC - as a CASE user user

In Digital Equipment Corporation Hong Kong (DEC), DEC Repository (CCD/Plus) is the only used CASE tool. Since most of the application programs have been developed before the CASE tools is available, their analyst programmer Adam Wong told us that switching to CASE is really not an easy task.

Summary:

There are about 20 application programmers and analysts in DEC Hong Kong. All applications are running on DEC VMS. Many existing application programs were developed in 1983-84. During which CASE tool is not available.

For the automation strategy, the DEC Repository (CDD/Plus) is being used in a way to help co-ordinate all the programs. It is a data dictionary containing all the data definitions used in programs. No Front End CASE tools like those help system analysis and design are used in Hong Kong. In the Pacific region, only Singapore is using DECdesign in a pilot program.

DEC Hong Kong does not have any explicit selection criteria on CASE tools. DEC repository is used only because it is available at no cost. Other third party CASE tools are not considered because they are expensive.

About their supporting structure, there is not any person who is dedicated to supporting CASE. At the moment, there is not any pilot CASE project in Hong Kong yet. Repository is just an as-is product and it is up to individual analyst programmer to decide whether to use it or not in their programming work. According to Adam, the use of Repository do really result in the improvement in co-ordination among many different programs and easier software maintenance.

4.2.3 Interview Hong Kong Bank

With the top management support, HongKong Bank gets many benefits in using CASE. Front End (Analysis & design) as well as Back End (Code Generator) CASE tools are tested and used. Most programmers welcome and support this change.

Background:

Top management felt that CASE technology & Structured Methodology would benefit the bank. They initiated to start a CASE project team to introduce CASE into the organization. The CASE project team was started 2 years ago to co-ordinate all CASE activities in the bank. The approach is to automate 'area-by-area'. Instead of an quantum change, the bank preferred a gradual change.

Initially, CASE automation is used in two areas:

- Analysis & Design, which is referred as the Front End CASE. Excelerator is chosen as the tool.

- Code generation, which is referred as the Back End CASE. APS is used.

Front End CASE is used on project basis. It is up to the Project leader to decide whether to adopt CASE in his/her project or not. Code Generator is a tool available to all programmers. They can use it wherever they feel appropriate.

Throughout the whole automation process, the structured methodology is emphasised. The CASE team regards Structured Methodology as important as the CASE tools.

The team highlighted several criteria in evaluating CASE products. They are:

- ease to use
- easy to learn
- Common design and analysis methodology should be supported
- hardware requirement should be reasonable
- it should be compatible with the bank's operating system, programming languages and data base system.

In order to make sure end users accept this change, they adopted the user-involvment approach. All potential users of CASE were invited by the CASE team in the CASE product evaluation and testing.

The bank has now formed a five members team and their responsibilities are product evaluation, CASE promotion to project leaders, in-house support to CASE tools, and conduct internal training on Structured methodology & CASE tools.

Up to now, the result of the CASE pilot project is encouraging. Regarding the Front end CASE, four project have used it and the actual results were satisfactory. The outputs have better quality and maintainability. As for the effort saved, the bank said that there it is still hard to see any improvement here after using CASE. The same result was also revealed by the questionnaire. One possible reason is that they are on the rising slope of the learning curve.

Besides the Front End CASE project teams, the bank has another project team looking into the Back End CASE tools. According to the bank, the result was very promising. Productivity was increased by 20% and more than 50% of the programmers used this tool regularly.

Though the bank is adopting a step-by-step automation approach, it is by no mean they are not aggressive in their automation strategy. Hong Kong bank will automate other areas of the development life cycle like program testing. They also plan to integrate their Front End CASE (Excelerator) with the Back End CASE (APS) by sharing a common data interface. This will help further improve their design quality and reduce the human effort required.

4.2.4 Interview IBM CASE Consultant

According to Samuel Lo, the IBM System Specialist on DB2, the situation in Hong Kong is quite different from the U.S. In Hong Kong, the first step which is also the most important step of going automation is to choose the right structured methodology. Followings are the summary of Samuel's opinions on CASE applications.

The consultant's opinions:

Samuel said that the reasons why Excelerator fits Hong Kong Bank's requirement is mainly because of its flexibility. The flexibility makes it easily compatible with their existing software in the bank.

Generally speaking, a Front End CASE tool with more powerful analysis and checking is more suitable in Hong Kong. It is because the programmers are still very new to the structured techniques. More checking would prevent unnecessary mistakes.

He also mentioned that when a shop decides to go for CASE, it must carefully choose the right structured methodology first. CASE tool is only for implementing that structured methodology.

Most CASE tools nowadays have a lot of features. Many organizations do not need all these features. When they select a CASE tool, they have to make sure the features that are relevant to their methodology are there.

The DP environment in Hong Kong is quite special. In most cases, the standard structured methodology supported in the CASE tool package does not fit the requirement. Some local modifications is therefore a must.

4.3 CONCLUSION

Based on the above questionnaire results and interview results, we find that the CASE introduction process as mentioned in Chapter 2 is also applicable in Hong Kong. For instance, the starting of CASE implementation in Hong Kong Bank is very similar to the scheme we suggested in Chapter 2.

We also identified six critical factors for successful CASE implementation.

a) Management commitment is very important.

b) Support/ training capability is also very important. For example, in Hong Kong, the local support from CASE vendor is not strong. So an organization should allocate a specialist or team for internal support.

c) User involvement (the application programmer) in the CASE product evaluation is a major factor.

d) For shops which are new to structured methodology, a step-by-step implementation is suggested. i.e. automate some phase of the life-cycle first.
E.g. Both Hong Kong Bank & IBM-Hong Kong adopt a similar approach.

e) The importance of structured methodology (technique) should always be emphasised; CASE tool is regarded as the automation tool.

f) Organizations should not aim at immediate increase in productivity after CASE implementation. Many benefits of CASE are long term and related to quality improvement which is somewhat intangible and difficult to quantify.

BIBLIOGRAPHY

Books

[DEC 90] D.E.C. The Digital COHESION Environment for CASE

D.E.C. Digital CASE environment: Comprehensive Productivity Solutions

[DEC 88] D.E.C. VAX/VMS Software: Information Management Handbook

[IBM 89] I.B.M. Systems Application Architecture: AD/Cycle Concepts GC26-4531-0

[IBM 90] I.B.M. AD/Cycle Digest - a guide to IBM's SAA application development framework Summer/ 1990

[IND 89] Index Technology: EXCELERATOR/IS Application Guide Rel. 1.9

[FISHER 88] Fisher, A.S. CASE - using software development tools Wiley, 1988.

Periodicals

[MERC 90] Mercurio, V.J. "AD/Cycle strategy & architecture" IBM Systems Journal Vol 29 No 2, 1990.

[MISRA 90] Misra, S.K. "Analysing CASE system characteristics: evaluation framework." Information & software technology Vol 32 no 6, July/Aug. 1990.

[MISTRY 89] Misry, K. "Implementation of Structured Methodology into Credit Institution" Information & Software Technology (UK) Vol 31 Iss 10, Dec 89.

[OMAN 90] Oman, P.W. "CASE Analysis & design tools" IEEE Software May 1990.

Appendix A:

FUNCTIONAL DESCRIPTION ON SELECTED PRODUCTS OF AD/CYCLE

APPENDIX A.

Function Descriiptions on Selected Products of AD/Cycle

1. BACHMAN/Re-Engineering Product Set

a. Programming Environment
MVS, OS2

b. Corresponding Development Phase
Analysis/Design, Build/Test, Production Maintenance

c. Function Description
It includes tools for data modeling and knowledge-based DB2 database design. Both new application development and re-engineering of existing data structures are supported. It provides a graphic-based work space that make it easier to say, analyze and manipulate database designs.

d. Tool Description

i. BACHMAN/DBA (DB2) Catalog Extract

It assists the system designers to copy information from the DB2 catalog to enable re-engineering of DB2 designs and the synchronization of new DB2 designs with the existing DB2 catalog. This tool extracts Data Definition Language (DDL) statements from the DB2 catalog in the form of files containing CREATE and ALTER statements, as well as statements from object' descriptions in the DB2 catalog tables.

ii. BACHMAN/Data Analyst

It aids in information modeling and analysis by providing analysts with tools for information modeling, data analysis and business information resource management. In addition, it gives developers import and export utilities that allow it to collaborate with other CASE tools. It gives a picture of business information in the form of Bachman entity-relationshop diagrams.

BACHMAN/DA departs from the top-down-only approach to informatin modeling and accommodates almost any modeling approach. It can populate the ER diagram with objects captured directly from the existing database designs.

iii. BACHMAN/Database Administrator (DB2)

It helps designers manage the physical aspects of database design, maintenance, enhancement, and documentation tasks. It also assists them in making such design choices as where to locate keys, how space should be allocated and how much free space is required.

iv. BACHMAN/Database Administrator (IDMS)

This is the key product to initiate the migration of IDMS data designs to DB2.

The online Expert Advisor assists in design decisions regarding index sizing, area sizing, page sizing, record clustering, and also validates record and set options for completeness.

The BACHMAN/Database Administrator (IDMS) allows designers to capture existing IDMS data descriptions and display them as Bachman Entity-relationship diagrams. The information can then be edited to meet maintenance requirements, or reverse-engineered to the BACHMAN/Data Analyst for further modification. The new or revised logical design can then be forward-engineered back to the BACHMAN/Database Administrator for efficient migration to IBM's DB2.

v. BACHMAN/DA Capture

BACHMAN/DA Capture (IMS) and BACHMAN/DA Capture (Files) help to build new DB2 designs. Working together, they read existing IMS data base descriptions (DBDs) and COBOL data descriptions to capture database designs from IMS applications into the BACHMAN/Data Analyst. The captured structure can then be converted to Bachman entity-relationship diagrams and used to upgrade existing models, or they can be forward-engineered into DB2 designs.

Additionally, BACHMAN/DA Capture (IMS), in conjunction with the BACHMAN/Data Analyst, can create a graphic picture of an existing application - creating a picture of a system's logical structure, generated directly from captured data description code and assuring a complete understanding of the underlying logic.

A seperate product, BACHMAN/DA Capture(Files), enables systems designers to capture and reverse-engineer existing file designs into the BACHMAN/Data Analyst in the form of COBOL data descriptions. These structures can then be displayed as Bachman entity-relationship disgrams for use in building enterprise-wide data models.

2. INDEX TECHNOLOGY EXCELERATOR SERIES

a. Programming Environment
OS2

b. Corresponding Development Phase
Requirements, Analysis/Design

c. Function Description

As a system analysis and design tool, excelerator automates the building of complete data and process models, validates and cross references design information, prototupes screens and reports, and generates system documentation.

d. Tool Descriptions

i. Excelerator Series

Index Technology's Excelerator Series provide a totally integrated set of automated tools that assist in the design and documentation of information systems.

Facilities include a mouse-driven color graphics editor, integrated dictionary, powerful process and data analysis capabilities, and a report and screen design facility.

Excelerator's graphics capabilities provide the tools and symbols necessary to crẹate and manipulate data flow diagrams, structure charts, structure diagrams, logical data models, entity-relationship data models and presentation graphs.

Individual objects and data flows can be exploded to greater levels of detail to create models tht include data record description and more intricate data model diagrams. Presentation graphs are useful in defining user requirements and establishing the feasibility of new or renovated systems.

An interactive dictionary, XLDictionary, maintains all system information, including data structures, process logic, screen definitions, report layouts, and system design diagrams in one facility. Analysts can quickly review and update system documentation, list items on-screen, or create or update dictionary entries.

The facilities assisting in the design aspect are listed below:

- Screen Design Facility

It allows the user to develop design prototypes of the system's frontend screens and user-interface menus.

- Report Design Facility

It helps analysts work closely with end-users to develop more constructive and informative reports.

- Documentation facility

It allows the user to pull information from the dictionary and word processing files and integrate it into a final system specification document.

- Dictionary Interface Facility

It allows sharing of information between projects as well as with host-based dictionaries.

- Housekeeping Facility

It provides a set of functions for backing up and restoring files, changing passwords and access privileges and setting the system configuration.

ii. Customizer

It helps the user create a custom development environment designed to work the same way the organization works. Within this specialized environment, analysts can use Customizer to tailor the XLDictionary and user interface to work with an organization's unique approaches to paplication'development. In addition, it provides the ability to integrate other facilities used in the development life cycle into the design environment.

iii. PC Prism

It is an analysis tool designed for stragtegic and information systems planning and enterprise modeling. It assists developers in aligning information system goals with corporate objectives and targets the projects that can best achieve them.

- Prism Define

It allows the user to begin building the planning model using his own logic and vocabulary to define categories and their relationships.

- Worksheet Facility

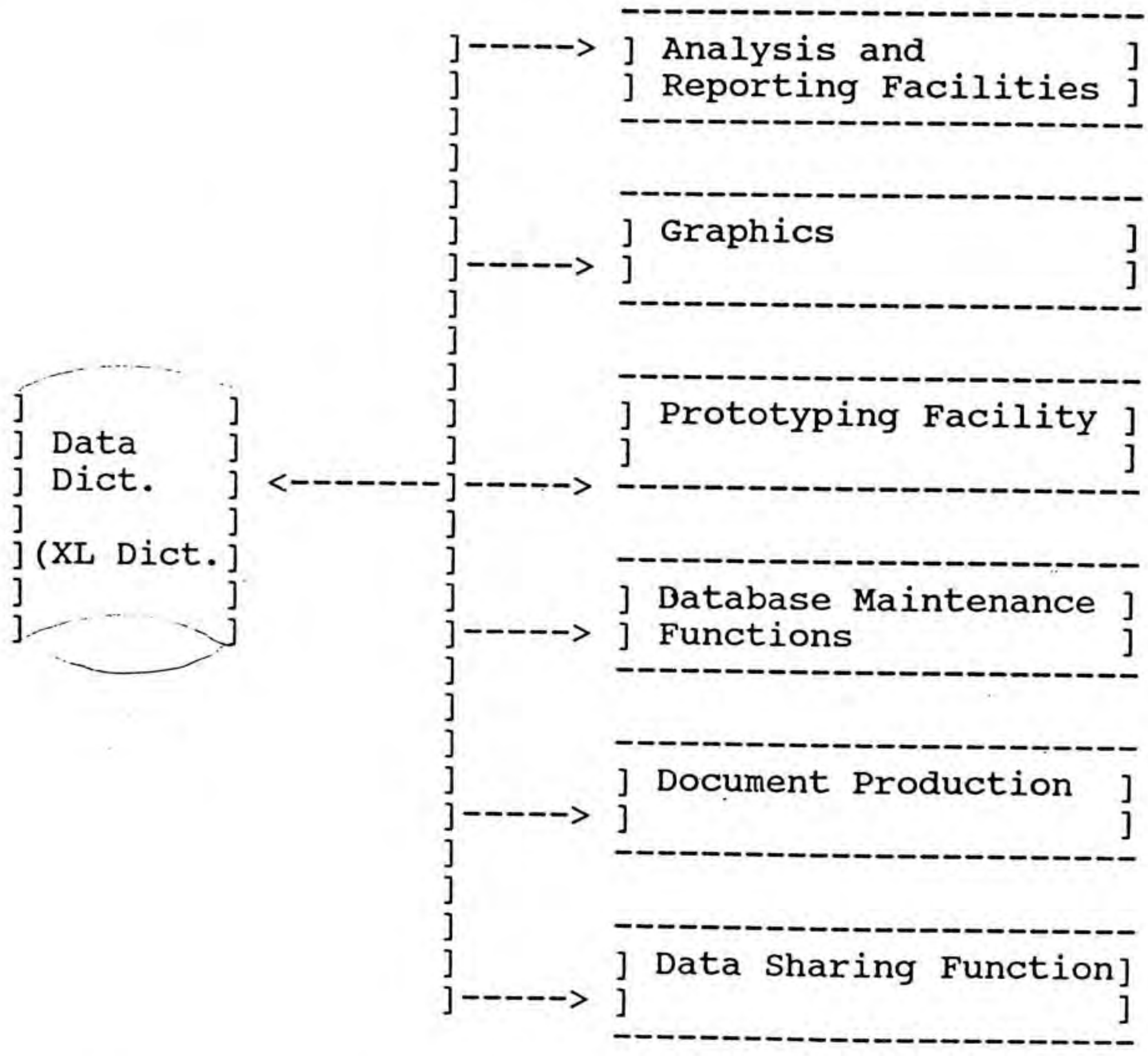
It provides a set of matrices that enable you to detail, validate, and prioritize the categories you established using Prism Define. Developers use the matrices to establish a databse that illustrates the relationshiops between categories and related details in their proper organizational context.

- Extensive Reports Facility

It lets you create both matrix and textual reports by selectively drawing data from the databse established with the worksheet facility.

iv. Excelerator for DB2

It provides analysts and developers with an effieient link between Index Technology's Excelerator development workbench and IBM's relational database for AD/Cycle, DB2.



3 KNOWLEDGEWARE INFORMATION ENGINEERING WORKBENCH

a. Programming Environment
OS2

b. Correspoonding development Phase
Requirement, Analysis/Design

c. Function Description

IEW/Planning workstation may be used for capturing, modeling and analyzing data about an organization and its use of information.

IEW/Analysis Workstation can be used for refining and analyzing end-user requirements and creating information systems specifications.

IEW/Design Workstation performs the subsequent design of information systems through the creation of physical specifications for both data and processes.

d. Tools Description

i. IEW/Planning Workstation

It provides the application developer with the functions necessary to manage and analyse planning data in far greater details than manual methods permit.

It includes an intelligent encyclopedia that serves as a knowledge base for the entire tool set. A knowledge coördinator ensures the consistency of information collected in the encycloipedia and provides realtime error checking to guarantee the quality of the work.

It's open architecture and import/export capabilities allow one to interface to other database management system, data dictionariès and 4GL.

Planning Workstation also provides a variety of reporting and analysis capabilities, including

- Affinity Analysis for grouping planning objects into clusters based on common information

- Encyclopedia Summary to inventory all the objects and associations in the encyclopedia

- Exception Analysis that verifies the completeness of the model and highlights problems

- Hierarchy Report for queryng the encyclopedia to see all the associations of an object at the same time

ii. IEW/Analysis Workstation

It helps to produce system specifications.

It describes the data needs of a business, identify what processesa re required to meet end-user needs, and identify how they interrelate, through:

- Decomposition Diagrams that helps to define the partition processes, subject areas of data, and organizational units into greater levels of detail.

- Data Flow Diagram that illustrate the input and exit of data in the system, how processes transform data, and how data stores and external agents are accessed.

- Entity-relatiobship Diagrams for identifying the data requirements of an organization to allow the developement of seperate data models specifying the data needed to support individual processes, subvject area of data, data stores and external agents.

- Action Diagrams to specify the procedural logic, or mini-specs, for the lowest level processes in the analysis.

iii. IEW/Design Workstation

It allows developers to move from a model of what a given system will do into the physical specifications of how the application will actually be implemented.

Diagramming functions include:

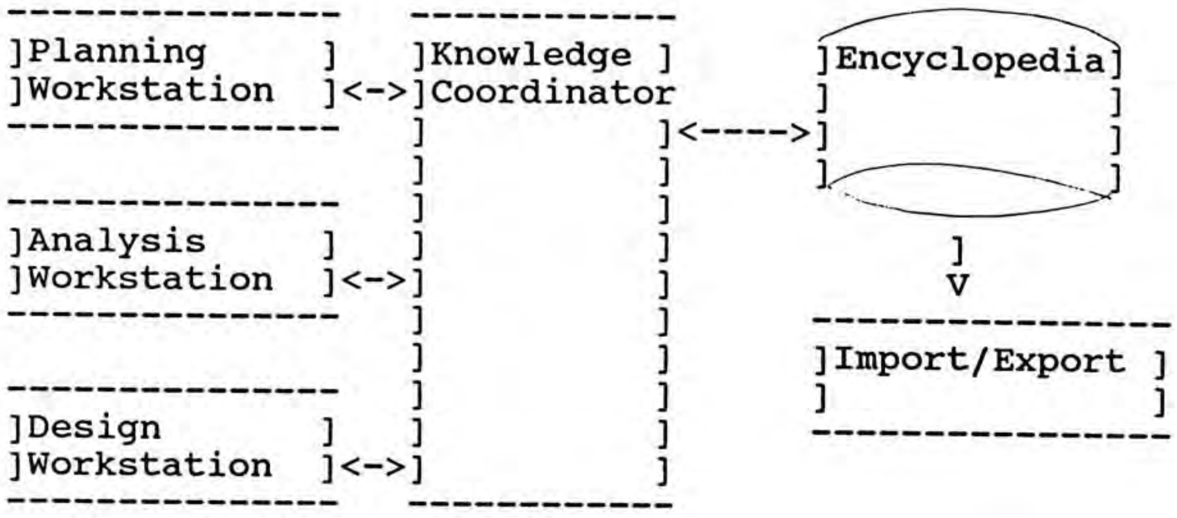
- Structured Charts display a hierarchy of modules that call or contain subordinate modules, as well as data passed between them.

- Action Diagrams spècify detailed procedural logic that ties together the individual pieces of a design. In addition to detailed module logic, Action Diagrams let developers refer to 'design objects such as screens and data structures.

- Presentation diagrams for screen layouts simplify and speed screen design.

- Database Diagrams help you represent the physical database designs for relational, hierarchical, and flat-file database implementations.

- Data structure diagrams describe the data elements involved in individual relations, segments, records and screens.



4. REPOSITORY MANAGER/MVS

a. Programming Environemnt
   MVS

b. Coresponding Development Phase
   Application Development Platform

c. Function Description
   It provides a single point of control for the sharing and exchanging of application development information across the application development life cycle. Information is stored in entity-relationship modeling format.

d. Tool Description

   i. IBM's Repository Manager/MVS provides the means to manage data processing through a systematic approach and structure to help the enterprise achieve accurate and accessible information. That means comprehensive, complete information about:

   -the enterprise's business activities, organization and goals.

   -the data needed to run the business

   -the applications that support the business

   -the meaningful relationships among all these

   Repository Manager is designed to maintain an organized, shared collection of information that supports application development, business modeling and data processing activities.

It stores the meaning the user give to the data as an by defining entities as identifiable persons, places things, concepts or events that can be described with information to be stored, and relationships as associations between two entities, two relationships or an entity and a relationship.

ii. Repository Manager orchestrates three points of view of information and function:

- conceptual view is a global perspective representing information that is common across all tools and functions within Repository Manager.

- The logical view is a database of application or tool-function specifications. It identifies information to which tools have access and specifies the way tools process information.

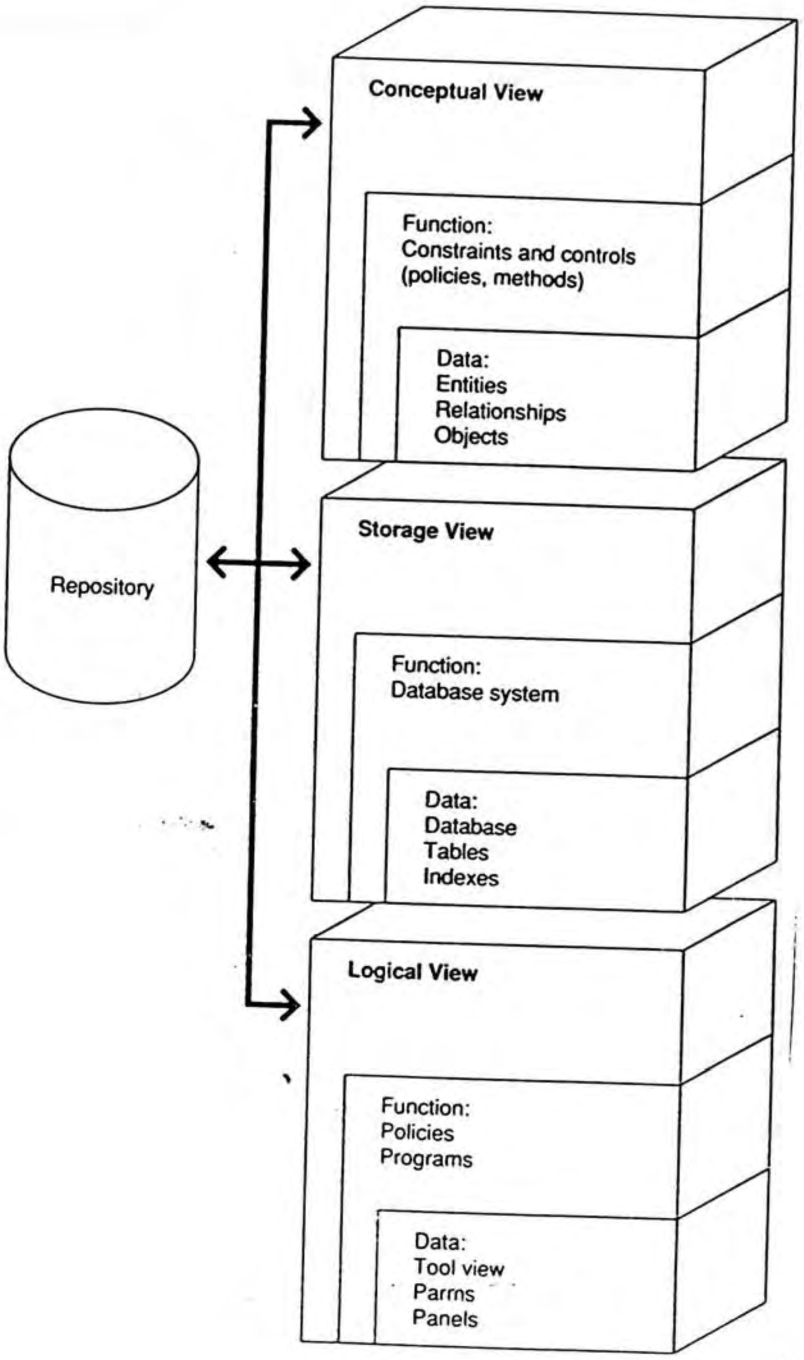
- The storage view, the physical storage perspective, defines how Repository Manager information is physically represented for use with DB2.

Because the 3 views are independent, changes can be made in one view without influencing the others. That means we can add new ER definitions to the conceptual view and map them to the storage view with no need to change any tools that don't need the new information.

iii. It can handle OBJECTs

Repository Manager provides a set of object services. They support controlled access and common information about collctions of ER model information called object data.

All Repository Manager-based tools manipulate objects or access object data by calling Repository Manager and providing the name of the object and the method to be performed.



Repository Manager allows you to specify both data and function from three views: Conceptual, a global perspective; Storage, a physical storage perspective; and Logical, an application or "tool" perspective. Each view's independence from the others translates to a high degree of flexibility in actual use because you can change data definitions in one view without necessarily affecting the others—Repository Manager automatically converts the data as required.

5. DEVELOPMATE

a. Programming Environment
MVS, OS/2

b. Corresponding Development Phase
Requirements, Analysis/Design

c. Function Description

i. It allows the definition of enterprise models in entity-relationship terms.

ii. It provides interactive prototyping of business processes against sample data to validate the model and user requirements.

iii. It generates reports about the model and the prototypes.

6. Cross System Product/Application Development

a. Programming Requirements
MVS, VM

b. Corresponding Development Phase
Produce, Production Maintenance

c. Function Description

i. It performs interactive definition, test and execution of applications programs.

ii. It produces formatted documentation

iii. It provides screen and application prototyping

7. COBOL Structuring Facility

a. Programming Environment
MVS, VM

b. Corresponding Development Phase
Build/Test, Production Management

c. Function Description

i. It helps to convert unstructured program to structured program, i.e. it helps to perform the re-engineering of old programs.

ii. Documentation will be generated automatically during the conversion. The Structured Chart gives a clear display of the structure of the output program.

Appendix B:

SURVEY RESULT

SURVEY RESULT SUMMARY

	Getting User Req't	System Analysis	System Design	Coding	Testing	Imple'n	Maint'ce
Before installation of CASE products, you found problems in this area.	35%	35%	35%	20%	20%	0%	55%
You found an urgent need to automate this area.	35%	55%	75%	55%	35%	0%	35%
Your CASE tools help in this area.							
Name of the tool							
After installation of the Case products, you found improvements in this area.	35%	20%	35%	35%	20%	0%	20%
After installation of the Case products, you still found problems in this area.							

Three rows are intentionally left blank because we did not get any response or only very few response.

QUESTIONNAIRE ON CASE PRODUCTS IN HONG KONG

We are the CU MBA Year 3 students and are now conducting a survey on the advantages and shortcomings of the CASE products available in Hong Kong. Your knowledge and work experience will be of vital importance to the success of our survey. Please help to spend 10 minutes to fill up the attached questionnaire. A stamped and self-addressed envelop is enclosed to ease your mailing back to us. As we have to submit the project to the MBA department by early March, Your early reply before Feb 7 will be greatly appreciated.

Thank you very much...

Cliff Chu
Edmond Sung

1. Please name the CASE (Computer Aided Software Engineering) products that your company is using and their functions.

   (Note : All those products that assist in getting user requirements, system analysis and design, coding, testing, implementation and maintenance belong to this category.)

   CASE Product   Functions

   ____________   ____

   ____________   ____

   ____________   ____

   ____________   ____

2. Please name all the CASE products you know that are available in Hong Kong, other than those mentioned above.

   CASE Product   Functions

   ____________   ____

   ____________   ____

   ____________   ____

   ____________   ____

   ____________   ____

   ____________   ____

   ____________   ____

   ____________   ____

3. The following is a matrix of questions. Please put Y for Yes, N for No, or leave it blank if it does not apply to you.

	Getting User Req't	System Analysis	System Design	Coding	Testing	Imple'n	Maint'ce
Before installation of CASE products, you found problems in this area.							
You found an urgent need to automate this area.							
Your CASE tools help in this area.							
Name of the tool							
After installation of the Case products, you found improvements in this area.							
After installation of the Case products, you still found problems in this area.							

4. What are the deficiencies of your CASE products?

CASE Product ____________ Deficiency ____________________________________

____________ ____________________________________

____________ ____________________________________

____________ ____________________________________

5. After considering the implementation cost and benefits, do you think it is worthwhile to implement the CASE product, and why?

____

6. COMPANY DATA

Your Name (Optional) ________________ Company Name ________________

No. of EDP Personnel ________________ Hardware used ________________

Appendix C:

INTERVIEW REPORTS

INTERVIEW IBM CASE CONCULTANT

The first part of the interview is about the functionalities of Excelerator. The detail results are already in the Product Evaluation section.

The second part which is his general comments about CASE in Hong Kong is as follow:

Q: What is the VALIDATION & AUTOMATIC ANALYSIS capabilities of Excelerator?

A: Excelerator was designed to provide a flexible tool to support many different techniques (methodology). So ther is not much automtaic checking on a particular methodology.

This aspect of Excelerator has its disadvantage & advantage:

Disadvantage:
For unskilled programmers who are not familiar with the methodology, errors may be missed until the final stage of development.

Advantage:
Flexible to fit any methodology.

For example, Hong Kong Bank has chosen Excelerator & find it fit for its use. One of the main factor is the flexibility of Excelerator.
Hong Kong Bank does not implement a particular standard methodology. Basically, there was no/ or very limited structured methodology before they use CASE. Structured methodology & CASE tool were introduced step-by-step to the Bank. So the flexibility & compatibility with the bank's existing standards is an important factor.

Q: In general, which approach - flexible Vs more analysis capability - is more suitable in Hong Kong?

A: Since Hong Kong is vey new to Structured Methodology & CASE, a Front End CASE tool with more analysis & checking capbility to huide the analyst during the development process should be more useful in Hong Kong.

Q: I understand you must have talked to many prospective CASE users in Hong Kong. In general, what are the important factors to these people?

A: The methodology, not the CASE tool.
A shop should select the appropriate methodology first. Then they select/ define some features that are required. The CASE tool product is evaluated base on whether it meet these requirements.

CASE tools are providing more & more features. In many situations, they are more thatn what a user need.

Hong Kong is still very primitive in understanding CASE.

CASE consultancy would be a very good business opportunity. Many Data Processing departments need guideline on how to implement CASE & Structured methodology.

The DP environment in Hong Kong is not the same as U.S. So, in many instances, the standard structured methodology from U.S. does not fit the local requirements.

INTERVIEW D.E.C. IN-HOUSE M.I.S. SYSTEM ANALYST

Q: What project development methodology is being used by DEC? Is there a structured methodology?

A: Yes. All projects must follow the DMR standard. DMR is not a software but a set of recommendations by a consultant company. It specifies the stages/ phases and what specifications/ documents are required after each stage. So DEC is adhering to a structured methodology for our in-house development.

Q: In DEC, are there already projects which make use of the CASE technology?

A: For most new start up projects, we make use of the CASE technology. Many of the current MIS softwares were developed in around 1983. At that time, CASE technology was still at a pilot stage. So, for these old projects, conventional programming tools were used.

Q: What CASE tools are used?

A: In Hong Kong, we only use the CDD/Plus for our central corporate dictionary. DECdesign is available. But it has not been actually used in projects.
Among the DEC offices in Far East, only the Singapore office is using CASE extensively for the whole project. i.e. DEC design for initial requirement analysis and design. For that project, I know that they are still discussing (arguing) on the proper way to model the corporate MIS model.

Q: Compared with conventional methods, what is the difference when you adopt CASE technology?

A: In adopting CCD/Plus, interface between programs & software maintenance are now better managed. Since records & fields must be drawn from the central CCD/Plus dictionary, there would not be two different meanings in the same data field. For example, if a field/ record is already defined or used by Program A. Program B, which is restricted by the central CCD/Plus dictionary, will adopt the same definition and valid values for that field/ record.

Software maintenance is also easier. Say, if there is a bug in one module. Since this may affect other modules, fixing this bug alone cannot solve the whole problem. CCD/Plus can point out other related modules. So the maintenance programmer can easily make the other related changes as well.

Q: What are the constraints that make it difficult to fully adopt CASE in your environment?

A: Hardware platform has to be upgraded. For DECdesign, which runs on DEC window, the programmers have to be provided with a powerful workstation e.g. RISC machine. I have tried to run DECdesign myself on a less powerful machine. The graphic screens and response time are not good enough for actual project use.

There are still areas in the development cycle that DEC uses third party products e.g. code generation.
In these cases, we would have to rely on third party products. But I estimate the software licence fee for so many DEC offices would be very expensive.

Q: What MIS environment or MIS project is most suitable for the adoption of the CASE technology?

A: MIS department of large corporation with a large programmer team.
CASE gives many benefits in projects where it involves the co-ordination among many team members. Program data are shared by many programs. i.e. The program is not small & standalone.
The CCD/Plus can help to check the consistency of data, fields among different programs, projects in an organization.

USAGE OF STRUCTURED METHODOLOGY IN YOUR ORGANIZATION – DEC.

Structured Analysis & Design Methodology	USAGE			
	Yes	No	Not sure	
1. Process Analysis				
- Data Flow Diagram	☑	□	□	
If yes, is the Diagram hierachical?	☑	□	□	
Is a mini-spec for each process also shown on the Data Flow Diagram?	□	□	☑	
2. Process Design				
- Design Specification				
a) Structured chart	□	☑	□	Note 2
b) Jackson Structured Design	□	☑	□	
- Mini Specification (process name, input, output, description)	□	□	☑	
Following is/are used to describe each process:				
a) Structured English	✓			
b) Decision Table	✓			
c) Decision Tree				
d) Pseudo code	✓			
e) Program Design Language				
f) Formal Prog./ computer Language				
3. Data Analysis				
- Data Structure Definition				
a) Warnier-Orr Diagram	□	□	□	
b) Entity Relationship (E-R) Diagram	☑	□	□	Note 2
4. Data Design				
a) File format specification	☑	□	□	
b) Data Dictionary	B	□	□	Note 1

Note 1: Before CASE, DEC adopts DMU
After CASE, use CDD/Plus.

2: Before CASE, too tedious to implement
After CASE, yes.

Key: Yes
✓ After CASE
B Both Before & After.

INTERVIEW IBM - AS A USER

Q: How is the EDP department (for in-house support) in IBM - Hong Kong organised?

A: There are around 35 programmers/ analysts. We use MVS running on 3090 mainframe. Database is IMS/DB & DB2.
Most applications are developed in other IBM sites & ported into the Hong Kong environment.

Q: Are there already projects which make use of CASE?

A: Last year, we formed a task force to study Excelerator for 5 months. It was found that Excelerator is easy to use.

So far, there were two pilot projects using CASE.

Q: What was the outcome of these pilot projects compared with projects using the 'old' techniques?

A: I think it is still very early for me to comment on the outcome of using CASE. For those pilot projects, we found that more time was spent. The programmers are not yet very familiar with the techniques. It will take some time for the programmers to be familiar with the CASE tools.

Q: What are the future plans?

A: We are now studying & evaluating the new version Bachman & Cross System Product (CSP). They provide both the process & data design capabilities.

For Excelerator, I heard that, since more analysts/ programmers are confident in the product, we are preparing Show Cases on Excelerator for demonstration to customers.

Q: What are the in-house support for CASE?

A: There is one analyst assigned to co-orginate the Excelerator support for all in-house projects. There have been in-house training sessions on CASE for all application analysts/ programmers.

Q: Is there any plan to go for a corporate/ universal dictionary supporting the entire development life cycle, like the Enterprise Model in AD/cycle?

A: No, there is not yet an Enterprise Model. In those pilot CASE projects, they started at the Conceptual Design level (e.g. specifying ownership).

USAGE OF STRUCTURED METHODOLOGY IN YOUR ORGANIZATION - IBM -

Structured Analysis & Design Methodology	USAGE		
	Yes	No	Not sure
1. Process Analysis			
- Data Flow Diagram	☑	□	
If yes, is the Diagram hierachical?	☑	□	□
Is a mini-spec for each process also shown on the Data Flow Diagram?	□	☑	
2. Process Design			
- Design Specification			
a) Structured chart	☑	□	
b) Jackson Structured Design	□	☑	□
- Mini Specification (process name, input, output, description)	B	□	□
Following is/are used to describe each process:			
a) Structured English	B		
b) Decision Table	B		
c) Decision Tree		✓	
d) Pseudo code		✓	
e) Program Design Language	B		
f) Formal Prog./ computer Language		✓	
3. Data Analysis			
- Data Structure Definition			
a) Warnier-Orr Diagram	□	☑	□
b) Entity Relationship (E-R) Diagram	B	□	□
4. Data Design			
a) File format specification	B	□	□
b) Data Dictionary	☑	□	□

KEY: Yes

✓ After CASE

B Both Before + After.

INTERVIEW HONG KONG BANK

Q: What is the background & function of this CASE team?

A: The CASE team was started two years ago. This team supports all the developments in the bank which require CASE. We concentrate on both the Tools & the technique (Structured Methodology). We have been mainly in two CASE tool areas:

a) Front End CASE,
which is for the initial phase of a project development. It is for Analysis & Design.

b) Back End CASE,
which we mean Code Generator. e.g. for COBOL

A summary of what we have introduced in our organization:

- Code Generator for COBOL in mainframe
- Front End CASE tool
- structured methodology

Q: How does your CASE team fit into the bank software development organization?

A: The CASE team belongs to the Technical Support Group. Although we are not directly involved in project development, we provide CASE support to, say, a development project on mainframe.
We also help to introduce structured methodology & procedures (which I refer as the Technique). e.g. Data Flow Diagram.

Q: What is the size of your organization? How about the average project team size?

A: Our software staffs can be grouped under three categories:

1) Mainframe - 250
2) Mid-range - more than 100
3) P.C. - less than 50

Since there are many different projects, it is very difficult to estimate the average team size.

Q: How did you start CASE in H.K. Bank?

A: It was started in Oct. 89. There were four pilot projects - three on mainframe, one on P.C. In some projects, we use Excelerator as the Front End CASE. APS, a code generator, was used in some projects.

Excelerator was chosen because we found it easy to learn. It can be run on a PC-LAN.

Q: How do your team promote CASE in the bank?

A: 1) Education
We provide training on Structured Methodology & CASE tools.

2) Support
Our team also provides daily support of CASE tools to encourage use.

The use of CASE is on a project by project basis. We talk to the project leaders about CASE but it is up to the leader to decide whether to adopt CASE in the project.

In the evaluation of a CASE tool, we always involve the users as well. It also helps.

Q: What was the outcome of these pilot projects compared with projects using conventional techniques?

A: To make it easier to understand, I will talk about the outcome/ effect of using Front End CASE tool first. (Code Generator project will be talked about later.)

It took about the same time (e.g. man-month) since more time was spent in the System Specification.
The user interface is better (i.e. better communication with end user) and, finally, better documentation.

In these pilot projects, we found that there was a lack of understanding - CASE & structured methodlogy emphasis more on the initial analysis & design. So more time would be spent in that phase but many people did not have that concept. Sometimes, the managers were not so familiar. When they did not see programs, codes coming out after a certain time, they got anxious.

Q: How did you introduce CASE Front End tool into these initial pilot projects?

A: We first interviewed the project managers. If it was decided CASE should be adopted in that project, the entire team will be trained on the CASE tool.

Q: What are your criteria in selecting a suitable project for CASE?

A: Our criteria:

1) New development (not just an enhancement) or a Major enhancement
2) The project should not be too large. But it cannot be too small - e.g. more than 10 man-months.

Q: What is the difficulty in introducing CASE?

A: It requires a change in culture - emphasis of Analysis & Design. That's why we also provide courses on Structured Methodlogy.

Q: How many people are there in your CASE team?

A: We started with 4 people. Now there are 5. But we involve other analysts/ programmers in CASE tool evaluation.

Q: What is the corporate strategy in introducing CASE in the organization?

A: The approach is top down. It started when some people in the top management feel that CASE can benefit the bank. There is a commitment from the top management. So a CASE task force/ team was set up.

Q: You told us the story of Front End CASE - outcomes, difficulties, etc - how about the Back End CASE (Code Generator)?

A: The approach for Code Generator is different. We first introduced APS for COBOL code generation. It is available to all the mainframe programmers so it is not restricted to a particular project.
We did an APS training in around Feb. - March, 90 to all mainframe programmers.
We did a survey to the senior managers in Dec 90. It indicated that more than 100 staffs were using the APS. It was more than 50% of the mainframe analysts/ programmers. The overall gain in productivity resulting from APS was 20%.

We now is introducing LANCER. It is a RPG generator on mid-range.

Q: How did you measure the 20% productivity gain?

A: In the survey, it was based on the subjective judgement from the programmers. They considered time spent & testing.

Q: What are your new plans?

A: Currently, we are planning, the introduction of several new tools:

1) Testing tools
   - tools for setting up testing cases & testing tasks.
2) Link
   We are looking into how we can link Excelerator, APS & LANCER together.
3) Bachman
   Since the new version supports both data & process, we are investigating the possibility of using Bachman in between Excelerator & our Code generators. This new version runs on OS/2. The previous version does not support the process side - Data Flow Diagram, structured chart, etc. Bachman has some data definition language which can facilitate data base design.

   Excelerator supports the IBM SSADM (Structured System Analysis & Design Method) which comprises of:
   - E-R diagram
   - conceptual design
   - logical design
   - physical design

   If Bachman also supports these, it can be used as a tool to interface between Excelerator & APS.

Q: How about Dictionary? Is there any plan to go for a corporate/ universal dictionary supporting CASE tools in different phases of the development life cycle?

A: At the moment, there is a dictionary associated with each project (in Excelerator). We will expand it into the Code Generation by having APS or LANCER to make use of the dictionary of Excelerator. Another direction is to integrate the individual project dictionaries to a common dictionary.

Q: Existing programs which were developed a few years ago base on conventional techniques are always a barrier to the use of CASE. What is your approach these exsiting programs?

A: For existing programs, we use an utility to convert existing programs to APS format. A number of programs have already been converted.
We plan to integrate APS back to Excelerator as well. There is an APS generator. The outputs from the APS generator, like structured chart, help to integrate back to Excelerator. It is similar to Data Reverse Engineering.

Q: What criteria you used in the evaluation of CASE tools?

A: Criteria:
- Ease of use
- Easy to learn
- techniques (methodologies) support
- hardware platform, networking
  e.g. LAN support

We also evaluate the features like:
graphic, dictionary, analysis & checking routine, etc.
We prioritise the features according to our needs first. We evaluate several CASE tool products on paper. We eliminate those products which do not meet the main requirements.

In Front End CASE tools, we did in-house evaluation on two CASE tools - Excelerator & IEW. We also involve the team leaders (our users) in the evaluation process. The in-house evaluation lasted for 2 to 3 months.
Reason for not taking IEW:
The main reason is that IEW can only interface to DB2 but not IDMS & VSAM. IEW is incompatable with IDMS which we use a lot in H.K. Bank.
The hardware requirement of Excelerator is very basic. In most cases, ordinary PC's are good enough. This feature of minimal hardware requirement means that we are not required to put in a lot of hardware investment in the pilot stage.

On the Back End side, we decided not to use 4GL. We considered that the change in adopting 4GL would be too drastic for the bank. 4GL has its own data base management system (DBMS) but it is incompatible with IBM IDMS. Other disadvantages:
- it uses system resources
- it concentrates on screen & reporting, but not sufficient for the applications in H.K. Bank.
- it is not good enough for batch applications.

So we decided to go for Code Generator.

We did in-house evaluation on four Code Generators. The evaluation period was 10 months. We also involve the users, application programmers in this case.

The main reason for choosing APS is because it can access all platforms that we use - VSAM, IBMS, CICS, DB2. It can also be extended to the PC & midrange (e.g. AS400) arenas.

Q: What are the platforms used in the bank?

A: Most of our computers are based on IBM - MVS, AS400, PC. There are also Tandem & DEC but they are not major.

(Then we give them the methodology questionnaire to fill in. We find that Structured Methodology was not used before the use of CASE tools. Drew told us that, when he was with Midland Bank in U.S., he tried to introduce structured methedology before CASE tools were available.)

Q: What was your experience in Midland Bank?

A: I tried to introduce structured methodology but the analysts/ programmers found it too tedious to do the drawings, data consistency check, etc. by hand.

Q: (Since Structured Methodology is not a standard procedure before the use of CASE), you have to introduce the structured approach as well as CASE to the organization.
Do your team provide in-house training courses on structured techniques?

A: Yes. These courses are open for the team leaders or programmers with at least 3 years experience.

Q: Is there any area that you think we have not asked?

A: An established support from the CASE tool vendor is also important. Previously, Index Technology supported our Excelerator through either Australia or US. Recently, they have appointed a local support agent.
We just heard that the company that manufactures APS is merged with Index Technology. It is a good news to us. Getting both Excelerator & APS from one company is good to us.

USAGE OF STRUCTURED METHODOLOGY IN YOUR ORGANIZATION - HKB

Structured Analysis & Design Methodology	USAGE		
	Yes	No	Not sure
1. Process Analysis			
- Data Flow Diagram	☑	□	□
If yes, is the Diagram hierachical?	□	☑	□
Is a mini-spec for each process also shown on the Data Flow Diagram?	□	☑	□
2. Process Design			
- Design Specification			
a) Structured chart	☑	□	□
b) Jackson Structured Design	□	☑	□
- Mini Specification (process name, input, output, description)	☑	□	□
Following is/are used to describe each process:			
a) Structured English	✓		
b) Decision Table	B		
c) Decision Tree		✓	
d) Pseudo code	B		
e) Program Design Language		✓	
f) Formal Prog./ computer Language		✓	
3. Data Analysis			
- Data Structure Definition			
a) Warnier-Orr Diagram	□	☑	□
b) Entity Relationship (E-R) Diagram	☑	□	□
4. Data Design			
a) File format specification	B	□	□
b) Data Dictionary	B	□	□

Key: Yes
✓ After CASE
B Both Before & After

Appendix D:

LIST OF INTERVIEWEES

LIST OF INTERVIEWEES

1. Interview IBM, as a CASE user

   Interviewee: William Yung
   System Engineer
   IBM

2. Interview DEC - as a CASE user

   Interviewee: Wong Ka Lok
   System Analyst
   MIS Dept.

3. Interview H.K. Bank - as a CASE user

   Interviewee: Drew McNichol
   Project Manager,
   CASE Development

4. Interview IBM CASE consultant

   Interviewee: Samnuel Lo
   System Consultant