THE USE OF INFORMATION ENGINEERING AS A FRAMEWORK FOR
ANALYZING RECORDS IN ELECTRONIC FORM

by

Jayne Bellyk
B. G. S., Simon Fraser University, 1981

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF ARCHIVAL STUDIES
in
THE FACULTY OF GRADUATE STUDIES
School of Library, Archival and Information Studies

We accept this thesis as conforming
to the required standard

THE UNIVERSITY OF BRITISH COLUMBIA
April 1995
© Jayne Bellyk, 1995
In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

School
Department of Library, Archival and Information Studies
The University of British Columbia
Vancouver, Canada

Date April 25, 1995
ABSTRACT

This thesis examines an approach and a methodology used by information technology professionals to develop information systems. Information engineering is a methodology for developing information systems following a specific process. It does not set out to create or manage records, yet it does have significance to archivists as a framework for analyzing information and records in electronic form.

The framework that information engineering extends to archivists is one that links administrative goals and business functions to individual activities and acts. The analysis of system documentation from an actual development project reveals how these relationships are created and maintained. It also describes and presents the lexicon and graphic representations of information management as it follows a particular action through the phases of development. Finally, it reveals the fundamental difference between traditional paper records and data in electronic form. The separation of content from structure is a result of the demands of database technology for storage and retrieval, and is a significant issue to be faced in developing methods for the management and preservation of electronic records.

The analysis also brings to light the importance of possessing this knowledge before establishing appraisal or description approaches or electronic record management program strategies. The study concludes that information engineering provides a useful
framework for archivists in the analysis of information and records in electronic form, and that this understanding is essential to the evolution of archival knowledge and the responsibility of archivists to ensure the reliability, authenticity, availability and integrity of records in an electronic environment.
TABLE OF CONTENTS

ABSTRACT .................................................................................................................... ii

TABLE OF CONTENTS .............................................................................................. iv

LIST OF ILLUSTRATIONS ......................................................................................... vi

ACKNOWLEDGEMENT ............................................................................................... vii

Chapter

INTRODUCTION ........................................................................................................... 1

1. OVERVIEW OF INFORMATION ENGINEERING AND THE SYSTEM
   DEVELOPMENT PROCESS ....................................................................................... 5
   The Discipline of Information Engineering ...................................................... 5
   System Development Prior to the Introduction of Information Engineering .... 8
   The Emergence of Information Engineering .................................................. 11
   Difference Between Data File and Database ................................................. 15
   The Information Engineering Pyramid .......................................................... 17
   The Information Engineering Methodology ................................................. 20
   History of Diagramming Conventions ........................................................... 22
   System Development Tools ........................................................................... 24
   The Descriptive Framework: Representations, Descriptions and Models ....... 25
   The Knowledge Base ....................................................................................... 27
   Project Deliverables ....................................................................................... 29
   The Four Phases of Information Engineering ............................................ 31
     Phase 1: Planning ......................................................................................... 31
     Phase 2: Analysis ......................................................................................... 34
     Phase 3: Design .......................................................................................... 36
     Phase 4: Construction ................................................................................. 38
   Summary of Information Engineering Phases ............................................. 40

2. THE ANALYSIS OF DOCUMENTATION FROM AN INFORMATION
   ENGINEERING PROJECT ....................................................................................... 42
   Study Project Background .............................................................................. 45
   Planning Phase Documentation ..................................................................... 51
     Deliverable: Operating Plan ....................................................................... 51
     Work Product: Corporate Vision, Mission and Commitment ................. 51
     Work Product: Action Program Summary ............................................... 52
Analysis Phase Documentation ........................................................................... 54
Deliverable: Project Charter ............................................................................. 54
  Work Product: Goals and Objectives .............................................................. 54
  Work Product: Project Scope ......................................................................... 55
Deliverable: Business Area Information Model ............................................... 56
  Work Product: Business Area Context Diagram .......................................... 56
  Work Product: Process Decomposition Diagram .......................................... 57
  Work Product: Business Area Process Definition ......................................... 63
  Work Product: Elementary Process Definition ............................................ 63
  Work Product: Intra-Model Association Diagram ......................................... 65
  Work Product: Business Area Entity-Relationship Diagram ......................... 66
Deliverable: Conceptual System Design .......................................................... 68
  Work Product: Business Transaction to Procedure Diagram ........................ 68
  Work Product: Application Structuring Diagram .......................................... 69
  Work Product: Procedure Descriptions ....................................................... 70
Design Phase Documentation ........................................................................... 74
Deliverable: Design Phase Project Charter ...................................................... 74
  Work Product: Design Phase Goals and Objectives .................................... 74
  Work Product: Design Phase Project Scope ................................................ 75
Deliverable: Business System Design .............................................................. 77
  Work Product: Screen Presentation Description .......................................... 77
  Work Product: Logical Database Design Diagram ....................................... 78
  Work Product: Screen Flow Diagram .......................................................... 81
  Work Product: Screen Specification Design ................................................. 83
Deliverable: Technical System Design ............................................................ 84
  Work Product: Program Structure Chart ...................................................... 84
Construction Phase Documentation ................................................................ 87
  Work Product: Module Action Diagram ....................................................... 87
Project Documentation Summary .................................................................... 89

3. THE SIGNIFICANCE OF INFORMATION ENGINEERING FOR RECORDKEEPING ........................................... 91

CONCLUSION ................................................................................................. 113

BIBLIOGRAPHY ............................................................................................. 117
# LIST OF ILLUSTRATIONS

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Information Engineering Pyramid and Phase Activities</td>
<td>19</td>
</tr>
<tr>
<td>2. CPAY Production System Menu Screens</td>
<td>49</td>
</tr>
<tr>
<td>3. Claims Payment System Information Architecture</td>
<td>50</td>
</tr>
<tr>
<td>5. Action Program Summary, Claims Payment System</td>
<td>53</td>
</tr>
<tr>
<td>6. Project Goals and Objectives [excerpt from]</td>
<td>55</td>
</tr>
<tr>
<td>7A. Analysis Phase Project Scope</td>
<td>58</td>
</tr>
<tr>
<td>7B. Analysis Phase Project Scope (continued)</td>
<td>59</td>
</tr>
<tr>
<td>8. Business Area Context Diagram</td>
<td>60</td>
</tr>
<tr>
<td>10. Claims Payment Definition [excerpt from]</td>
<td>63</td>
</tr>
<tr>
<td>11. Elementary Process Definition: Receive Cheques</td>
<td>64</td>
</tr>
<tr>
<td>12. Intra-Model Association Diagram [extract from]</td>
<td>66</td>
</tr>
<tr>
<td>13. Business Area Entity Relationship Diagram</td>
<td>67</td>
</tr>
<tr>
<td>14. Business Transaction to Procedure Diagram</td>
<td>71</td>
</tr>
<tr>
<td>15. Application Structuring Diagram</td>
<td>72</td>
</tr>
<tr>
<td>16. Procedure Descriptions [extracts from]</td>
<td>73</td>
</tr>
<tr>
<td>17. Design Phase Goals and Objectives CPAY 2B</td>
<td>75</td>
</tr>
<tr>
<td>18. Design Phase Project Scope CPAY 2B</td>
<td>76</td>
</tr>
<tr>
<td>19. Screen Presentation Description (Receive Cheque Series)</td>
<td>78</td>
</tr>
<tr>
<td>20. Logical Database Diagram - Claims Cheque Inventory</td>
<td>80</td>
</tr>
<tr>
<td>21. Screen Flow Diagram</td>
<td>82</td>
</tr>
<tr>
<td>22. Specification Screen Design (Receive Cheque Series)</td>
<td>83</td>
</tr>
<tr>
<td>23. Program Structure Chart: Receive Cheque Series (PY150X)</td>
<td>85</td>
</tr>
<tr>
<td>24. Program Structure Chart [detail]: Receive Cheque Series (PY150X)</td>
<td>85</td>
</tr>
<tr>
<td>25. Module Action Diagram (PY150X)</td>
<td>88</td>
</tr>
</tbody>
</table>
I would like to thank the following people for their advice and suggestions: my supervisor, Terry Eastwood, and my advisors, Mary Sue Stephenson and Charles Dollar. Thanks also to Jon Anderson, Kristen Seer and colleagues at the Insurance Corporation of British Columbia, especially: Silena Chow, Pat Mowatt, Larry Myrick, Keith Munro, Al Richer and Carol Anne Saxon for their support and technical expertise.
INTRODUCTION

Records in electronic form represent a new domain of inquiry for archivists. There are no easy answers in regards to the management, preservation, and dissemination of records in this form as their characteristics often prevent us from applying traditional archival practices. After a decade of writing and thinking about electronic records, there has been only modest progress made by the profession to solve the problems these records pose to the archival profession. Issues surrounding the management, preservation, and definition of electronic records remain largely unresolved, and there is still no agreement on what is required and how archival records management programs should be instituted.

The evolution of archival professional knowledge could benefit from knowledge of systems development and its documentary products in order to better understand the anatomy of electronic records and transactions. Valuable information about the administrative structure, business functions, activities and procedures is documented during an information engineering project that can be used by archivists to analyze information systems and the products they generate. Familiarity with the development framework is important in order to understand the nature of what is and what is not an electronic record, and how and when records are created. This level of comprehension
should represent the foundation for professional discourse about electronic record
management programs, appraisal and description.

James Martin said the "future of computing is a battle with complexity." Martin
was referring to the complexity of modern organizations and the complexity of
information processing in government and business. The activity of developing computer
systems able to meet the information requirements of large organizations is also a complex
undertaking. For example, a project with a duration of six to twelve months often has
400-500 identifiable tasks that require the close cooperation of many people with distinct
responsibilities, as well as extensive planning, management, and evaluation of all work
components.

Systems development, in the 1990s, is not immune to the worldwide movement to
improve quality. There is an effort to bring more structure to development endeavours
and to increase productivity and system quality. Computer-aided software engineering
technology has influenced current trends to make the system development process more
formal and rigorous. There have been many tools, techniques, and methods used in the
past to increase the productivity and quality of computer systems, however the limitations
of their scope, limited as well their benefits, which were at best incremental and piecemeal.
A more holistic approach was needed so that developers, designers, and analysts could
produce repeatable, predictable results.

Systems development has become an engineering-like discipline of which
information engineering is just one approach. Information engineering is not a software
product or a technology tool. It is an approach to development that specifies, defines, and
supports a particular process for developing information systems which utilizes the accumulated experience of business people and developers. What is unique about information engineering as a discipline and methodology, is that it integrates both strategic planning and information management into its philosophy and practice, in addition to the best practices of existing systems development techniques. In other words, it marries the techniques of the computer scientist to the strategy and principles of the business executive by linking the mission of the enterprise, its factors for success, and the objectives of senior management, to the creation of its information systems. The entire enterprise, the sum of its functions and activities, is the context for the creation of information using this approach. It is one of the most comprehensive system development methodologies in use today because of the union that exists between strategic planning and system development. It is this connection to the larger administrative structure, and its extensive use of documentation in the development process, that make it suitable as a topic for interdisciplinary study.

It is the hypothesis of this thesis that understanding the scope and products of a business project that utilizes an information engineering methodology will enable archivists to understand the origin and relationships of electronic information to the administration, and to the business activities which created the system.

The thesis will take a baseline approach to investigating the relevance of the information engineering methodology to the analysis of electronic information. In the first chapter, an overview of the discipline and methodology of information engineering will be examined. In the second chapter, the application of the methodology will be presented in
the context of an actual system development project. Within this larger project, the thesis will trace a relatively simple manual activity through the process of automation. In the last chapter, the significance of information engineering will be discussed in relation to recordkeeping.

The thesis will not define electronic records because the investigation is not about records, but rather, it looks at the development of the systems that capture and manipulate data that can result in records, and examines these structures in the light of the traditional concept of a record. Information engineering is a construct and collection of practices from the best of strategic planning, information analysis, and design techniques and concepts, and represents, in this sense a framework that will enable archivists to better comprehend records in electronic form.

In order to explore the usefulness of information engineering as a analytical tool for archivists, issues and topics beyond its main concepts, such as specific technologies, platforms, techniques, and project management, will be considered outside the scope of this investigation.
This chapter discusses the concepts, principles, terminology and issues relating to the information engineering approach to systems development. The intent is to provide an overview of the phases and activities of the system development process as context for the analysis of project documentation in the next chapter. It will not discuss at length the practical side of information engineering that pertains to issues of technology, programming and coding, or project management, which are not germane to the matters this thesis examines.

The Discipline of Information Engineering

In the information engineering approach, the development of individual systems takes place within a framework of strategic business planning. Information Engineering fosters the development of information systems that are aligned to the strategies and goals of the enterprise. It endeavors to find new ways to improve the enterprise through the use of information technology. It is a response to the deficiencies in traditional data processing and structured analysis, that addresses the entire systems development cycle,

---

including strategic planning. Changes to the philosophy and direction of the information systems function during the 1980s and 90s resulted in a corresponding need for more rigorous planning and analysis. The emphasis on enterprise-wide information systems resulted in major changes to information system philosophy and planning to enable the implementation of complex and strategic systems. Information engineering removes some of the technical complexity of computer analysis and design by using existing management experience and business knowledge in the development process. In contrast to traditional application development that typically focused on the automation of circumscribed activities, information engineering represents a comprehensive and integrated set of strategic methods which are aimed at the sum of all enterprise activities and functions.

Information engineering also differs from traditional application development in that it is a discipline that applies formalized techniques and automated tools to the enterprise as a whole, or to a major segment of the enterprise, rather than to isolated

---

2 The information systems function (IS) involves the creation, supervision, and control of information residing in systems that support an organization’s business processes. IS functions include the following: systems planning and development and information management. IS can also mean IS infrastructure, IS management practices, or an actual IS organizational unit. Note: From hereon, terms high-lighted in bold type indicate that an explanation of the term will be provided in a footnote.

3 The terms enterprise and business as used in this thesis are intended to include organizations that are engaged in typical business activities, as well as universities, government, and other organizations.

projects. Information engineering values the users' expert business knowledge over technical computer experience. James Martin asserts that information engineering is an alternative to traditional data processing because it is formal, computerized, and consistent. He defines information engineering as "the application of an interlocking set of formal techniques for the planning, analysis, design, and construction of information systems on an enterprise-wide basis or across a major sector of the enterprise." With reference to automated design techniques, information engineering is defined as, "an interlocking set of formal techniques in which enterprise models, data models, and process models are built up in a comprehensive knowledge base and are used to create and maintain data processing systems."

Information engineering also aims to develop systems that are stable and easy to maintain. According to Steven Cheung, its main goals are:

- to create a shareable and integrated database that uses a data-centric approach rather than the traditional process-centric one;
- to apply an engineering rigour to systems development that emphasizes standardization and reusability of information objects;

---

5 A technique is a complete description of how to construct a work product from a specified set of inputs. Techniques are generally applicable for multiple tasks, but some tasks have multiple techniques which can be employed. Some critical development techniques are: application structuring and identification; business area data modeling; context diagramming; critical success factor analysis; elementary process identification and specification; entity life-cycle analysis; event analysis; procedure identification and specification; prototyping; and subject area identification.


7 ibid., 1.

8 ibid., vii.

9 The use of the word rigour as a characteristic of information engineering represents an attempt to associate the rigour of the physical sciences to the development of information systems in modern business environments.
• to emphasize the importance of the modeling technique and its role in capturing business rules and other strategic information;
• to combine the power of computer-aided system development with automated diagramming tools;
• to create a knowledge base comprised of consistent and reusable information objects.\[10\]

Champions of the information engineering approach believe that the rigorous management of the system development process can reduce the time needed to build systems without sacrificing their quality.

**System Development Prior to the Introduction of Information Engineering**

Historically, objectives for the development of information systems were often narrowly focused, or so obscured during the development process, that the end results often did not meet the user's needs or coincide with the organization's mission. It has been estimated that a major portion of the cost of a system is incurred in maintenance after the system becomes operational. Prior to the acceptance of the system development life cycle concept by developers, and the advent of information engineering in particular, a large portion of computing resources went directly toward the programming activity which was labour intensive, error-prone, and extremely slow.\[11\] Therefore, most efforts to

---


\[11\] The concept of the system development life cycle predates information engineering. Most methodologies use this concept which divides the process into stages to support development activities. These stages typically are: initiation, analysis, design, construction, with some variation depending on the particular methodology in use.
improve system development centered on increasing the productivity of the programmers by continually refining their working tools. High level languages (also called third generation) had some positive effects on productivity in the 1960s because they more closely resembled natural language and were easier to use and understand than machine or assembler languages, which tended to be machine and model specific.

However, it was found that additional technology was required to improve system development performance due to the increasing complexity of computer applications. In the 1970s, punch card coding was replaced by more sophisticated software that allowed project code to be managed in a format that could be shared. Teleprocessing systems that used new platforms and gateways to communicate across old technological boundaries of hardware and software replaced simple protocols. Direct programming of data files had given way to the use of powerful file management systems that were later replaced by database management software. By the 1970s, programmers had computer terminals and new software tools, all aimed at reducing the time required to develop a system.

The increased sophistication of both hardware and software, combined with the growing complexity of business, and the importance of managing information in order to compete in the marketplace, brought more specialization to the computing professions. The definition, roles, and responsibilities of the data administrator, the database administrator, and the programmer grew more distinct from each other. These distinctions became necessary to the system development process and the ability to manage and operate the new information systems environment.
Techniques continued to improve yet the backlog of systems queued for development grew larger. As was already stated, an emphasis on high-level programming languages had some advantages, but there was still a tremendous and increasing pressure to provide users with complete and effective information systems as soon as a need was identified. Modern organizations require new information systems quickly in order to support changing business strategies. Trends in hardware and software development performance followed two very different curves throughout the 1970s and 1980s. Relative to software performance, hardware price/performance increased exponentially during this period, when it became evident that the technology of the database was improved to become more efficient. By the mid-eighties, it was evident that technology alone could not solve the problems of the management of information. It began to become clear that, if little or no attention was paid to managing the process that creates the system, the costs of maintaining and reworking major features of the system would continue to escalate. The idea of using development methodologies was gradually considered to be a cost-effective measure for the enterprise. If the use of methodology could improve quality and if properly designed systems are less expensive to maintain, then there would be a corresponding reduction in maintenance costs. Moreover, there

---

12 C.J. Date, *An Introduction to Database Systems* (New York: Addison Wesley, 1982) 6, explains that fourth generation languages (4GL) were introduced by database management system DBMS vendors in the late 1980s because the third generations languages like FORTRAN and COBOL did not function well with the database management software. 4GL offered a reduction of five to ten times the number of lines of code needed to write a program. This enabled developers to deliver prototypes to users in weeks rather than months, which users liked because they got to see what the application would look like before it was fully constructed.
was a growing consensus that these methodologies are necessary to ensure that new systems meet the needs of the organization.

The advent of information engineering was an evolution, rather than a revolution. It came about as a response to the overwhelming information processing needs of modern organizations. It also responded to the failure of traditional data processing methods to satisfy business requirements. In the 1990s, the benefits from aligning information processing and technology goals with the goals of the enterprise are demonstrable. Costs of developing new systems are now treated like any other type of investment, because the resulting system is recognized to be an tangible asset of the enterprise. When information is viewed as a corporate asset, the importance of managing data, process and technology as resources become evident. The ideal, as Cheung stated, is to have enterprise data that is shareable and software modules that are reusable. This is the challenge for most modern organizations -- to integrate business strategies and information systems.

The Emergence of Information Engineering

Computer technology has been around now for 40-50 years. Technical advances have played an important role in the system development process. The names used to describe the function of the computing department over time evidence another change. The data processing departments of the 1970s became the system development departments of the 1980s, and the information services departments of the 1990s. Inherent in these name changes is the shift of focus from process to data to a middle ground called information that combines the two. Understanding the distinction between data and information is important. **Data** is the direct result of observation or measurement, whereas **information**
is data that has been processed into a form that is meaningful to a user. Data alone is meaningless.

It is a basic precept of information engineering that the types of data used in an enterprise do not change very much over time. Therefore, the logical representation of data is relatively stable and can be used as a keystone supporting other elements in the information engineering process. A properly designed data model makes data accessible to constantly changing application requirements. While the data model can be relatively stable the procedures that use the model change frequently. Information engineering considers process and data jointly in the systems development process. It is desirable that processes be easily changed so that a business can be dynamic and change its procedures quickly in order to compete.

Data and process are two essential elements of an information system. Processes act on data and enable data to be transformed into information. This information conveys

---

13 Data will be used in this thesis to refer to stored values that have no meaning on their own. Information will be used to refer to data that has been interpreted in some way and has significance in relation to other data. For example, the letter “a” is meaningless, and so is “neaJy” but “Jayne” can be a useful fact (information) depending on the rules of interpretation attributed to the fact, which is this case is knowledge that Jayne is a name in the English language.


15 ibid., 58.

16 A data model is considered stable when it is fully “normalized”. Normalization is defined as “the simplification of more complex data structures according to E.F. Codd’s rules [relational theory] which are designed to produce simpler, more stable structures. Third normal form is usually adequate for stable data structures.” James Martin, Information Engineering, Book III: Design and Construction (Englewood Cliffs, New Jersey: Prentice-Hall Inc., 1990), 18.

17 Process is used as a synonym for activity, that can be decomposed into other processes or elementary processes.
a specific context and meaning to the user. In traditional application development, data
was treated as secondary to process. Data was not managed separately until it was
understood that it was more stable than process, and was more appropriate to serve as the
foundation for information systems building. It was recognized that if the foundation is
not stable then all systems and interfaces are likely to be unstable too, leading to user
confusion, increased need for reprogramming, and increased chance of error.\(^{18}\) If systems
are built on a foundation of data (what) rather than process (how it is done), then systems
could be useful for longer periods of time. For example, the Insurance Corporation of
British Columbia collects premiums and pays out claims. At this level, the functions are
relatively stable but how they can be achieved is very fluid, and can take many shapes in
the form of the systems built to facilitate the mission of the business. The
acknowledgment by organizations of the independent existence and value of data, separate
from the application programs that use the data, coincides with the adoption of the
database approach to information management and administration. The founders of the
information engineering method believed that both data and process can be stable, and that
both have equal importance in the system development process, and must be jointly
managed in order to create stable and integrated systems. It is this relationship between
data and process that is considered to be a hallmark of information engineering.

Based on an idea developed by James Martin and Clive Finklestein, information
engineering describes a method that incorporates the best features of several previous
approaches into a method with formal rigour and with more data-centric concepts and

\(^{18}\) Date, *An Introduction to Database Systems*, 473.
techniques than traditional methods used. As a concept, information engineering has its roots in database technology and the relational theory of E.F. Codd. It put together the power of relational theory with the strength of emerging strategic planning concepts to create a new business technique. Over the last decade or so, there has been a growing emphasis on the development of enterprise-wide information systems as well as a corresponding demand for information integration and distribution. Through this integration, a mechanism is established which can eliminate or minimize the development and implementation of stand-alone application systems that do not share common data or support information flows within the organization.

Information engineering links the business goals of the enterprise to the design of the information system. It views the information system as a business subsystem that fits into the larger system of knowledge that the business represents. In other words, the business-oriented approach encourages the examination of each automation effort in the context of the whole enterprise, and does not recognize the boundaries of a system as understood in the computing environments of the 1960s and 1970s. Because of the close relationship between business and its information systems, the information system supports

---

19 Even though James Martin is most often credited with the genesis of information engineering, this is as often disputed. In light of the sources reviewed and listed in the bibliography, I think that it is more accurate to say that Martin popularized the concept along with Ed Yourdon, Chris Gane, and Clive Finklestein. The origin of information engineering however, appears to have gestated at IBM during the period 1976-82 with the greatest advances made between 1983-88 when all these individuals were in attendance.

20 Finklestein, 16.

21 ibid.
the purpose, goals, objectives, and policies of the business. In essence, what Martin, Finklestein, and other information engineers felt they had uncovered was the secret of developing good, usable systems tied as much to business knowledge as to new technology.

Difference Between Data File and Database

The distinction between the data file and the database is essential in understanding the intent and purpose of information engineering. Database technology was created so that an enterprise could maintain and use its data as an integrated whole instead of as separate data files. A database functions as a community of data for a community of users.

Prior to the development of databases, data files were designed for, and were viewed as belonging to, one or more closely related application programs. In an environment that uses traditional data file structures, the data is maintained by applications which are usually designed to meet the needs of a single user or client department. Traditional files are typically simple sequences of records. These older file structures could not cope with complex interrelationships among business processes and the data that support them. By contrast, the database concept separates data requirements from application requirements, so that data can be integrated and shared among different programs. Although each data record and item is stored only once in the database, data

---

may be shared by many information systems and users involved in quite different but often interconnected parts of an organization. Information systems which are file-based (also known as flat files), generally cannot share data with other systems. Clive Finklestein describes traditional application development as individually designed and coded programs which make future maintenance wholly dependent on the adequacy and accuracy of the documentation of the original designer.23

File-based environments emphasize the computer system and its procedures. As systems are developed, customized data files are built around separate applications or procedures. This is why it is called “data-processing”, but perhaps a better name may have been just “processing” since controlling the process was more important than managing the data the process was manipulating.24 This style of processing resulted in program maintenance problems, piecemeal development efforts, and duplication of data. Duplication of data is normally cited as the principle disadvantage of data file technology because it results in duplicate input, duplicate maintenance, duplicate storage, and can cause integrity errors. In a database environment system, changes to storage or to the access structure (due to the addition of new functional requirements or performance considerations) do not necessitate corresponding changes to the application code.

A database is an organized, interrelated and stored collection of logically similar computer records of importance to the enterprise and intended for use by multiple users

23 Finklestein, 4.

24 Traditional development methodologies are considered procedure driven methods and include: systems analysis, structured analysis, structured design, and business systems planning. See: Finklestein, 4.
for various purposes. These records contain data elements (on a screen these are fields, in the database they are columns) which are collected and stored in the database. Data elements are usually organized in a structure that retains their relationships for processing by different applications. This particular organization or physical structure is the way data is stored by the database management system.

A database management system is software that supports the shared database concept and can be used to reduce the amount of redundant data stored. It supports numerous application programs by allowing each program to have a unique view of the database. In the next chapter we will see that this view is called the logical database design, which enables an application program to remain independent of the physical details relating to how the data is stored in the database. The database management system serves as an intermediary between the data and the application. Access to the database can be made according to the database design structure which is controlled by the database management system. The use of this software changes how data is created, processed and retrieved. Each request for data made by an application program is analyzed and retrieved by the database management system, and transmitted by this software to the application program.

The Information Engineering Pyramid

The schema of the four-level pyramid is the most typical description of information engineering development phases and is a useful way to describe and view them (see: Fig. 1). In the pyramid arrangement, the planning activity is at the top and represents the strategic plans of the business. The next level down refers to the analysis phase in which
a model is built to represent the fundamental data and processes needed to operate the enterprise. The first two of these four levels creates a framework within which different teams can build different systems at different times.25 From this analysis, the need for systems is determined. The third, or design level, produces designs expressed in rigorous and complete detail for construction of the database, which constitutes the fourth level.

One side of the pyramid represents data and the other side represents activities. Both data and activities (process) are managed from a high-level management-oriented view at the top of the pyramid through to construction at the bottom.26 The third side of the pyramid represents technology. Technology must be taken into account in determining how information engineering will be implemented. At the first level of the pyramid there must be a strategy relating to future technology and how it could affect the business, its products or services, and its goals and critical success factors. This is important because technology changes so quickly. No enterprise is untouched by the growing power of technology; indeed, some organizations and industries will be changed drastically by it. The technology side of the pyramid also must have strategies for the deployment and management of information engineering, and of corporate communication networks, both of which are closely tied to the availability and adoption of new technologies.27


26 James Martin, Information Engineering, Book I, 4.

27 ibid., 5.
Figure 1: Information Engineering Pyramid and Phase Activities

<table>
<thead>
<tr>
<th></th>
<th>DATA</th>
<th>ACTIVITIES</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLANNING</td>
<td>strategic overview of the</td>
<td>strategic overview of how technology can be used to improve the enterprise (its</td>
</tr>
<tr>
<td></td>
<td>information needed to run an</td>
<td>functions and goals)</td>
</tr>
<tr>
<td></td>
<td>enterprise as effectively as</td>
<td></td>
</tr>
<tr>
<td></td>
<td>possible</td>
<td></td>
</tr>
<tr>
<td>ANALYSIS</td>
<td>logical data model</td>
<td>processes and their relationships needed to operate the enterprise are</td>
</tr>
<tr>
<td></td>
<td></td>
<td>identified</td>
</tr>
<tr>
<td>DESIGN</td>
<td>design of the records used</td>
<td>design of procedures for executing specified processes</td>
</tr>
<tr>
<td></td>
<td>by specific procedures</td>
<td></td>
</tr>
<tr>
<td>CONSTRUCTION</td>
<td>application program view of</td>
<td>design of detailed program logic or input to a code generator</td>
</tr>
<tr>
<td></td>
<td>data</td>
<td></td>
</tr>
</tbody>
</table>
The information engineering pyramid on page 19 illustrates the complex associations between multiple corporate and technological factors that must be identified, planned, integrated, and maintained in order for management information systems to contribute to corporate objectives efficiently and effectively.\(^{28}\)

The Information Engineering Methodology

Information engineering is both a discipline and a methodology. On one hand, the discipline consists of the underlying concepts and principles that refer to all systems development techniques and to a generic class of methods employed by the discipline in the context of building information systems. On the other hand, an information engineering methodology is a particular selection of techniques used to create an information system following a specific process as set out by the methodology.\(^{29}\)

In order to achieve quality systems and increase productivity for the enterprise, development activities must be planned, managed and evaluated. The methodology used must be comprehensive and cover all aspects of the systems development life cycle. It should cover the complete breadth of activities needed during a development effort and guide the project to completion. "The value of the methodology is that it provides predictability and repeatability, encourages adherence to organizational standards for

\(^{28}\) ibid., 5-7.

\(^{29}\) Jessica Keyes, (1992 January), "How software is developed undergoing basic changes; with GIUs, servers, objects and parallelism, the question remains: How best to write software?" in Software Magazine 12 [CD ROM], 39(9). Available: Miller Freeman Publications. File: Computer Select Item 55566. In this article, Keyes says that a large part of the [computer] industry feels that the information engineering approach to systems development is the single most effective methodology because it is equally applicable to batch systems, on line systems, and object oriented systems.
quality, and uses the accumulated experience of the developers.\textsuperscript{30} It provides the framework for the logical and smooth progression of work from one activity to another, and eventual delivery of the final system to the end user.\textsuperscript{31}

Methodology, technique and tool are interrelated concepts. If a methodology consists of guidelines or procedures to do work, then a technique is a method for performing the work, and a tool the means of carrying out a technique. Phases in the life cycle of the system development methodology are represented as modular components. Procedures, tools, and techniques are used to create, refine, and control the work products that are produced in each life cycle phase. Organizational roles and responsibilities are also defined and associated with the various tasks performed during the life cycle. An organization’s methodology should be formalized and documented to provide consistent guidance to all staff at all levels of skill and experience. The integration of these components introduces structure and discipline into the development process.

The use of a particular methodology by an organization can offer additional opportunities and benefits, such as: the ability to capture business knowledge and experience in a form that is accessible to all practitioners of the methodology (a common language of representation and terminology); the provision of a tool for establishing a framework for communicating the organization’s best practices (superior performance); and a means of simplifying the understanding of previous work (without a standard


\textsuperscript{31} ibid.
process, how can we interpret the work of others and be certain of what it represents?}).
Communication between the members of the project team is facilitated by the highly evolved and precise language of representation used in the system development process.

History of Diagramming Conventions

Beginning in the early 1970s, diagrams and models of data and process offered for the first time a "language" that could be used intuitively by both analysts and users. Provided that the meaning of the picture and the rules for drawing it were well understood, considerable rigour could be built into a good diagramming process. As a result, a number of diagram based approaches became well established and began to be used on major development projects. It was at this point however, that a few practical problems became apparent. Relationships among these diagrams were complex. A change in one diagram often had impacts on other diagrams. Because of the difficulty of incorporating revisions, users of the diagramming techniques tended to work toward further changes, rather than stabilizing the information in the diagrams. Nevertheless, the manual approach to structured diagramming was a big step forward. It proved that diagrams work, that attempting to incorporate rigour was actually worthwhile, and that modeling only business processes without their counterparts in data was inadequate.

The design of database management systems forced some developers to look at ways of modeling information systems in terms of structural representation rather than just in description of use.\textsuperscript{32} Once again, diagramming was adopted as a means of simplifying

and presenting the results of data analysis in a graphic form. **Data structure** and **entity relationship diagrams** with a variety of notations were introduced and refined for use in the structured analysis and design processes.\(^{33}\) By comparing the models of how data was to be processed with the model of data relationships, it was then possible to incorporate an element of rigour into specification. It was also possible to provide a high level of consistency and completeness-checking that had not been practiced prior to the use of structured methods.\(^{34}\)

During the 1980s development requests were backlogged; users were turning to microcomputers and local area networks for short-term processing solutions. Senior management was reluctant to continue investing in new information technology. Every time a possible solution appeared, either the technology failed or a new set of problems developed that required more investment. The transition from manual flowcharting to computerized modeling took about fifteen years. Experiences in interactive systems development had demonstrated that formal specification procedures, although difficult to learn, improved the quality of the delivered application system. Business and data modeling led to a number of quantifiable benefits. It helped many users understand their own organizations better. Modeling also helped business analysts identify potential issues

\(^{33}\) **A data structure** is a particular arrangement of data elements that have meaning to users, such as a report, panel, of the application. Data elements are combined to form data structures and are the smallest unit of data that has any meaning (e.g. customer name) to users, management and information professionals. **An entity-relationship diagram** is a model of entities, their attributes, and the relationships among the entities, and their complete descriptions.

\(^{34}\) Information engineering evolved from earlier methods and brought with it the best practices from each. These methods were: structured programming; database design; structured analysis; software engineering; information strategy planning; and data modeling.
in business functions and organization structures that could not be seen easily any other way. Data modeling was a catalyst for many business development processes and formed a stable platform for a much more cost-effective implementation of information systems. Methods of analyzes that performed well in a manual environment had elements that could be automated, but the overall structure and approach was based on certain assumptions that were no longer completely valid once the design process was automated by computer-aided software engineering design tools.

System Development Tools

Design tools, as used in systems development activities, can be defined as software that aids in work performance and supports procedures, techniques and management of the development process. This definition refers to a special category of tools called computer-aided software engineering technology. The management and control of the system development process is supported by the increasing acceptance of this technology. In today's development environment, computer-aided software engineering tools are the principal means for building and maintaining models for both process and data. Modern development tools can help produce an information system faster, and can enhance the maintenance of the final system. These tools eliminate the drudgery and relative inefficiency associated with manual methods. The software has built-in controls to assist in the enforcement of discipline and standardization across life cycle tasks and phases, increase productivity, and facilitate accuracy and consistency. Such tools impose formality, increase the speed at which systems can be built and modified and, most importantly, can coordinate the vast amount of knowledge that must be collected and
updated across each of the information engineering phases. Models therefore can be kept current as changes in requirements occur.

Computer-aided software engineering is an important development in information technology and in the practice of information engineering. These design tools automate the encyclopedia function that is the central repository of knowledge about the enterprise.\textsuperscript{35} Data models and process models are stored in the encyclopedia, as well as facts about and policies governing the enterprise and its systems.\textsuperscript{36} It is important to note that the computer-aided software engineering tool is not the same as the methodology and should be considered separately. It acts solely as a technology aid to assist in the practice of information engineering. The type of design tool in use can effect the model's form or mode of presentation. No uniform and consistent notation, set of symbols, or conventions prevail for the forms of presentation. Models vary greatly in the symbols they use. Different methodologies favor the use of one kind of symbol over another. This often makes it appear that models and methodologies differ fundamentally when they do not.

The Descriptive Framework: Representations, Descriptions and Models

Information engineering maximizes the use of representations, description and models to make communication easier and faster among all involved parties: developers; project team; end users; and management. The use of representations and descriptions

\textsuperscript{35} Encyclopedia refers to centralized data storage of information automatically collected and cross referenced by the design tool during the system development process.

\textsuperscript{36} A data model is all information stored in the knowledge base about both logical and physical data. A process model is a representation of one of more processes and their associations.
provides the basis for an information architecture for data and applications. John Zachman is credited with delineating the relationship between descriptions, representations and models. For every different type of description there are different models that address the points of interest for each system life cycle audience or participant.37

Representations can be defined as the collection of deliverables or work products that illustrate, at a point in time, the particular perspective of a participant in the application system development process. A representation is a model presented in a meaningful form and intended for and understood by a particular audience or party involved in the application development process.

Descriptions are a specific kind or type of representation that present a specific aspect, view, or part of the system being developed or maintained. A description depicts a particular component of an information system. Types of descriptions (e.g., data and process) for various representations are produced during the successive phases of the systems life cycle. Each description communicates, confirms, and specifies to a particular audience certain conditions and aspects of the eventual system product. Descriptions serve to form a pattern from which the information system is developed and implemented. The two main types of descriptions are: process and data.

There are significant differences in the representations used during the different phases of the systems development life cycle. Each representation is different, and there is a refinement in terms of scope, system functionality, and purpose from one representation

---

to the next. Each representation has its own distinct nature. Level of detail cannot be used to distinguish between representations. It is incorrect to conclude that a representation from an earlier stage of the life cycle has less detail than one prepared during later stages.

A model is the result of a series of planned efforts and tasks outlined in a system development life cycle methodology. These efforts and tasks culminate in the delivery of conceptual, logical, and physical models that describe either data or processes that act on the data. Objects depicted in these various models are maintained in an encyclopedia, which serves as the database for systems developers during the system development life cycle and allows this information to be shared by development teams and other individuals. During the modeling process, developers translate a part of the overall enterprise model from one phase to another. At the end of each transition the methodology describes how to check that the model is: still within the original scope of the project, potentially complete, and internally consistent. This characteristic of information engineering is referred to as "iterative development". Once these factors are confirmed, the model becomes a sound basis for translation into the next phase.

The Knowledge Base

A modern enterprise must comprehend and manipulate significant amounts of data about itself, including detailed models, organizational and structural details, strategies and

---

38 Translation is used to refer to points in the development process (between logical and physical domains) when project products are interpreted by information specialists. The enterprise model is the overall high-level model of information processing by the enterprise. It is comprised of models of strategy, data, process and organization at the enterprise level.
plans. It is a fundamental tenet of the information engineering discipline that this information be organized and preserved so that it can be shared across applications by the entire enterprise. Information engineering refers to this as the knowledge base, and considers its use to be the key to maintaining a rigorous approach to systems development.  

The knowledge base defines how knowledge is acquired and used during the systems development process. It is built up incrementally as the data is collected during the course of planning, analyzing, designing, constructing and implementing the information system. The information collected is in the form of models that have evolved through the many stages of each phase of the project. The knowledge base can include goals, entities, organizations, processes, procedures, applications, and other information systems. The knowledge base contributes to the achievement of project objectives by organizing project information and documenting project results. 

The knowledge base also serves as the initial foundation for the development of any information system within an enterprise. The value of the knowledge base increases as its level of detail and complexity increases. At the most detailed level, the knowledge base can be implemented at the global level as the enterprise repository or at the local level as project encyclopedias. When a project is divided into sub-projects, the number of encyclopedias increase. Serious control and management issues may surface regarding different perspectives or collections of knowledge about a single or group of activities. A knowledge base administrator is often assigned to synchronize the knowledge base between multiple projects. Modeling and naming standards become very crucial in such an environment. A knowledge base can be stored in an automated encyclopedia, repository, spreadsheet, word processor, handwritten form or combination of these.

An entity is a person, place, thing, event or concept that is of some interest to the enterprise and about which it wishes to store data.
base contains hundreds of objects and thousands of associations. The term knowledge base is often used interchangeably with the terms encyclopedia, repository or data dictionary, although they do not mean precisely the same thing. Information engineering uses the iterative performance of tasks and activities to refine the information in the knowledge base and enhance understanding among users and members of the project team. Iterative development is another tenet of the methodology which states that all models and deliverables should be developed according to a controlled cycle of data gathering, analysis and review steps. As the process of iteration and refinement progresses throughout a project, more information can be collected and retained in the knowledge base. The benefit of this approach is that the chance of error through misunderstanding is reduced since each deliverable requires that a given level of detail must be achieved before further refinement can occur. Early successive iteration results in the improved accuracy of the work products (deliverables) from the previous phase or stage.

Project Deliverables

Systems development procedures are the guidelines and rules for carrying out the modules of work that comprise the methodology. The products of project work are called

\[\text{\footnotesize 41 In computer literature, these terms are often used as synonyms. For the sake of clarifying how they will be used in this paper, the following distinctions will apply. The encyclopedia represents a body of knowledge which includes all associations between information objects. A data dictionary on the other hand, defines single objects and does not include details of their associations to other objects. A repository is the physical implementation of the knowledge base, that is to say, it is an enterprise data model or global model that includes all data used in the organization regardless of the platform, for the purpose of sharing data objects within the organization.}\]
deliverables and work products. Deliverables are defined by the methodology as an aggregate of interrelated diagrams and models (work products). Deliverables are developed iteratively from knowledge gathered during the development phases of the project. This knowledge base becomes the basis for project estimation, workplan definition, and measurement of project results. During the initial structuring of the project, the scope statement identifies which deliverables will be required to meet project goals and objectives as well as establishing the review and approval cycle for work products. A complete list of project deliverables is essential to delivering a quality information system.

Project deliverables and the work products they contain serve as the primary method of communication to management, clients and developers, during the system development process. The project knowledge accumulated in the deliverables reveals the strengths and weaknesses of the system before the system is built. They are also evidence of decisions made during the process and can reassure management that progress is being made.

Accurate and comprehensive documentation should be important to any systems development effort, not only information engineering. Consequences of poor documentation in complex and expensive projects can be far-reaching. In addition to guiding the process of development, the deliverables represent an understanding of the business objectives that were the stimulus for the initiation of the system in the first place. After the system is built, system deliverables are still an important source of information for anyone who must use, operate, maintain, or understand the system.
The Four Phases of Information Engineering

A methodology decomposes a problem into smaller segments of work. It starts at the top with a management view of the enterprise and progresses downward to greater detail. The following discussion of the phases represents an ideal and generic development framework. While the phased approach is not an invention of information engineering, it is one of the established precepts that evolved from past methods, and that it embodies. The phases represent clearly separated modules of work.

Phase 1: Planning

Systems planning encompasses information gathering, business modeling, and decision making about both an organization's business and the information systems needed to support the enterprise. A substantial effort is placed on obtaining, structuring and analyzing information about the enterprise, its mission, goals, objectives, critical success factors, and its information systems environment. Understanding the relationships between major business functions and related processes, organizational structure, and the use of technology is essential in understanding the information needs required to conduct business. The result of the planning effort is the definition of the information requirements of the enterprise, a strategy for developing the information systems needed to guide the acquisition and development of information technology.

---

42 James Martin, Information Engineering, Book I, 10.
Information engineering begins by conducting an information strategy plan. It is accomplished by a small team that studies the enterprise and interviews its management. The plan produces a high level model of the enterprise and its data, linking information system requirements to senior management's strategic plan. The results of an information strategy plan process are usually stimulating because they provide clearly diagrammed representations of the enterprise and challenge management to rethink what it is doing. The business plan that is used to set targets, goals and strategies for the enterprise must be reflected in the information strategy plan. The plan also forces management to see the connection between technology and business (though today most corporate managers are well aware of the importance of information technology). Organizational and operational problems identified during system development are often solved by reorganizing or reengineering the enterprise.

The plan is updated continually as part of the business planning cycle. Concise, understandable representations of the planning phase analysis (typically diagrams and

43 Finklestein, 11.

44 Often in application development the analysts and programmers have no alternative but to interpret policies themselves with members from operational management. After policies are interpreted into code they are effectively set in stone. In this way, the system is a view of the present and not the future because the data needed for a quality system should be based on strategic plans set by top management. Only they can say what data will be relevant to the goals and strategies associated to it. Finklestein, 11.

45 James Martin, Information Engineering, Book II, 19.

46 ibid., 17.

47 ibid., 202.

48 ibid., 24.
matrices) make it easy to review goals and critical success factors. It is important that the information strategy plan be understandable not only to information professionals but to both senior management and end users. The development of a information strategy plan has proven to be a valuable exercise, even when other phases of information engineering have not been completed. The two overall purposes of the plan are to link information technology planning to strategic business planning, and to create an information architecture that accommodates further analysis and design, so that separately developed systems will work together.49

Information strategic planning has been performed differently by different corporations and with a variety of tools. However, it typically consists of six phases: 1) develop an entity model; 2) define business areas and goals; 3) establish critical success factors; 4) analyze current systems; 5) set priorities for information systems development, and; 6) develop an enterprise model.50

As the progression into detail occurs, decisions must be made about which business areas should be analyzed and which systems should be designed. The strategic business plan directs the management information system development plan based on the relative priority of projects.51 According to James Martin, information engineering perceives information to be a corporate resource that should be planned for "on a

49 ibid., 18.

50 An entity model refers to what is being represented (data). However, it is often used synonymously with entity-relationship diagram which refers more to the diagramming technique used to create the model.

51 ibid., 23.
corporate-wide basis regardless of the fact that it is used in many different computers and departments. The information needs often remain the same when the corporation itself is reorganized." This means that "the information architecture should therefore be designed independently of the corporate organization. Implementation of the architecture will reflect the current organization and its concerns. These will affect the choice of which modules of the architecture are implemented first."^52

Phase 2: Analysis

From the information strategy plan, a business area is selected for analysis. 53 The overall objective of the business area analysis is to understand what processes are needed to run a particular business area, how these processes interrelate, and what data is necessary to meet enterprise goals.54 This business area analysis establishes a detailed framework for building an information based enterprise by creating a foundation of data and process models.55

During the business area analysis stage the following activities occur: a data model is produced (which represents the complex interactions within a business area); functions are decomposed into processes; a data flow diagram is created that shows how the processes interrelate; and a matrix is built to show what data entities are used, updated,

---

52 James Martin, Information Engineering, Book II, 15.

53 A business area is that part of the enterprise that is included within the scope of the analysis project.

54 James Martin, Information Engineering, Book II, 97.

55 ibid., 197.
and created through what processes.\textsuperscript{56} The business area analysis should also be easy to understand so that it is useful as a communication device between users and developers. It should not be too complex since its goal is not system design. Its function is to understand and model the processes and data required to run the business and to create an information framework into which systems can fit.\textsuperscript{57} In other words, the business area analysis establishes what data and what processes are required to run the enterprise, not how individual procedures operate or how a specific system should function. It is important not to confuse this type of analysis with requirement planning for the design of a specific system.\textsuperscript{58}

Business area analysis is done several times for different business areas. A number of studies for different business areas may be done by several teams simultaneously. As already stated, the business area analysis does not attempt to design systems; both it and information strategy planning are conducted independently of technological considerations. This type of analysis will facilitate the later design of systems and ensure that the systems work together after they are built.\textsuperscript{59}

In summary, business area analysis has the following characteristics: it is conducted separately for each business area; it requires intensive user involvement; it remains independent of technology considerations; it often causes a rethinking of systems

\textsuperscript{56} A data flow diagram shows the flow and transformation of data through a set of processes within a system or business area.

\textsuperscript{57} ibid., 205.

\textsuperscript{58} ibid., 201.

\textsuperscript{59} ibid., 197.
and procedures in the organization; it identifies areas for system design; it creates a detailed data model for the business area and; it creates a detailed process model and links it to the data model.

**Phase 3: Design**

System design transforms selected processes into procedures that can be implemented into the physical domain. In the design phase, a complete and detailed specification of the system which is required to support a defined business area is produced. This occurs with the aid of automated design tools which use the information stored in the encyclopedia gathered during the information strategy planning and business area analysis processes. It is important to note that system building does not always wait until the framework of the first two stages is completely finished. It is an objective of information engineering to develop systems quickly in order to aid the modern corporation to compete in business by being able to implement new procedures as quickly as possible.

System design has dramatically changed with the use of computer-aided software engineering tools. With these intelligent tools, design work is accelerated because the design is created on a computer screen rather than on a drawing board with pencils and plastic templates. The encyclopedia is at the heart of all system design technology tools. Such intelligent software requires an architecture in which all tools and all developers

---


61 ibid., 18.
share an encyclopedia with computerized intelligence to coordinate the knowledge from all parts of the development effort in order to generate code and build the system.\textsuperscript{62}

The involvement of end users is particularly important during the system design stage. Requirements planning and design should be done in workshops, either jointly or separately, with end users.\textsuperscript{63} A joint application design session is a strategy for rapid application development that uses structured group sessions to build a system design through cooperative discussion.\textsuperscript{64} Prototyping (which is a representation of the system that simulates the main user interfaces of the new system) is also used in conjunction with joint application design to facilitate discussion between end users and system professionals.\textsuperscript{65} Joint application design has proven to be a highly successful technique because it is an organized, controlled and structured process that harnesses the expertise of users to create implemented solutions and easy to maintain designs.\textsuperscript{66}

In summary, the objectives of the system design stage are as follows: involve the end users fully in the design process; speed up design and implementation; make systems flexible and easy to change; automate design, documentation, and maintenance; link design automation to code generation or fourth generation languages and; create and

\textsuperscript{62} ibid., 22.
\textsuperscript{63} ibid., 143.
\textsuperscript{64} ibid.
\textsuperscript{65} ibid., 593.
\textsuperscript{66} ibid., 167.
evolve prototypes. It is important that the diagrams and system representations used during this phase are easy for end users to understand.

Phase 4: Construction

During the construction stage of information engineering the system is coded according to the detailed design specifications produced during the design phase. Once the designs have been developed, the computer can construct the system (which is the implementation of the procedures designed by users and developers in the last stage) with the help of code generators coupled to an automated design tool.

It is important to note that information engineering requires more time spent on planning and design than on execution of the design. The planning and analysis processes are critical to the success of the final system because they separate the information system into component parts for the purpose of identifying and evaluating problems, opportunities, constraints, and needs. The more work done initially results in less work required in the final phase, especially if a code generator and reusable code modules are used for construction. System design and construction can take weeks or years, depending on the complexity of the system. The information engineering commitment to technology speeds up the design and construction of systems by using automated design tools that are able to generate code directly from the encyclopedia.67

A key objective of information engineering is to impose rules on planning and design that are formal enough to direct the computer to write code, thus freeing the

information professional from the burden of coding. Martin says that, "perhaps the most revolutionary change in software development that workbench tools will bring is the enforcement of standards and the building of systems from reusable parts."\(^{68}\) By using data modeling techniques, information engineering makes possible a high degree of reusable design and code because it identifies common data and common processes across an enterprise.\(^{69}\) An objective of information engineering is to identify commonality in both data and processes and consequently minimize redundant systems development work.\(^{70}\)

The planning and rigour that characterize the information engineering discipline attack the problem of systems backlog by creating better systems that require less revision and maintenance and by the application of computing technology and power to code generation. In contrast to the first two levels of the pyramid, system design and construction are wholly dependent on the target technological environment (architecture).

Major products of the construction phase are: the coded and tested system; system documentation package; a training package; system operating instructions; an operational database; and the installed application system.


\(^{69}\) ibid., 40.

\(^{70}\) James Martin in his third book, *Information Engineering, Book III: Design and Construction* (Englewood Cliffs, New Jersey: Prentice-Hall Inc., 1990) 40-41, explains that "the same entity is used in many different applications across an enterprise. The same (normalized) data structure is associated with the entity type. The entity type represents a type of object which has a certain behavior, regardless of where it is used. These behaviors can be represented in reusable process modules. Two entity types may have a certain relationship. This relationship is the same throughout the enterprise. Certain behaviors are associated with the relationship, which is, again, represented in reusable process modules."
Summary of Information Engineering Phases

The planning phase project typically has a very wide scope, ideally the entire enterprise, in which the high-level models, the overall goals, and the mission of the enterprise are defined. The work products of the planning phase are periodically updated and reviewed to be current and representative of the enterprise and its activities. The objective of the planning phase is to provide a transition from systems planning to project-level activities that provide the foundation for meeting the overall goals of the enterprise.

During the analysis phase requirements for the system should be clearly defined. User requirements are defined in terms of business processes, information requirements, and the relationships between these processes and requirements. They are based on the perceived needs of the people who will use the system and on the job the system will accomplish. Many of the analysis activities are iterative in nature, with changes in one area sometimes necessitating a re-analysis of other areas. The analysis project takes a business area, or need for a system identified in the planning phase and delivers a detailed understanding of that area represented by models and diagrams and other knowledge objects. The data, process flows, and supporting descriptions, and the definition of the technology environment are organized in a logical manner without concern for how the system will ultimately be constructed. The system is still completely flexible at this point, since nothing has actually been designed.

During the design phase, the requirements are translated into a blueprint of the processing functions that the system will perform to meet users’ needs. The data and
process specifications are organized into a system that fits within the constraints of the available platform technology.

During construction the system is built. The programs are designed and coded, databases are created and populated, and the system is tested. As the system progresses through a series of test stages, it is migrated through a corresponding series of environments. Essentially, it is too late at this point to make any extensive amendments to the system.

In the next chapter, a selection of documentation from an actual information engineering project will be presented in order to illustrate how the concepts and principles already discussed, can be applied to a real business problem. The documentation selected will highlight the development of an individual action or act that forms part of a larger business transaction through the system development process, to examine how the process effects it as it moves from the logical to the physical domains, or rather, from analysis to design to construction.
CHAPTER TWO
THE ANALYSIS OF DOCUMENTATION FROM AN INFORMATION ENGINEERING PROJECT

This chapter will analyze the documentary products of an actual project to illustrate information engineering concepts, and the phases of system development. A selection of key representations, descriptions and models created during the planning, analysis, design and construction phases will be presented as a succession of work products that eventually result in the construction of an information system. Where information engineering was described in general and ideal terms in the first chapter, this chapter will present the application of the methodology to an actual business problem (or project). The intention of this chapter is not to exhibit the project in its entirety or to inventory all project forms and representations, but rather to present the documentation of a system development project in order of its temporal creation, to explain individual modes of representation, and trace the transformation of a "logical unit of action" through the phases of development. The impact of technology on the physical and logical unit will be discussed in the last chapter, when the qualities of its recordness in the electronic environment can be explored.

Before explaining what is meant by a logical unit of action, it should be noted that the goal of system development is to represent a portion of the business in a way that will
enable specific activities to be executed electronically. The system that results from the
development project will enable the automation of certain actions, as well as manage and
manipulate the data associated with them. The system is used to accomplish the actions
and also to capture and record information about them. It is the aim of this thesis to
examine the development of the system and role of the methodology in this process, and
then see in which ways the system is implicated in the production of records.

Project diagrams and documents (hereafter referred to as work products) are
organized within this chapter by: the order of phase; the project deliverable; and by their
approximate order of creation. The profiles of individual work product are intended to:
demonstrate the inter-relationships between specific documents and diagrams in the
context of a real project; provide a context for the specific illustration; explain the
purpose of individual diagram and document types; and describe how the logical unit of
action is changing as it moves through the system development process. The presentation
of documents in this chapter will preclude any detailed consideration given to project
management, design tool technology, or physical implementation issues as reflected in the
documents. While these are important practical components of any information
engineering project and comprise a significant portion of its documentation, they are
outside the scope of this study due to their marginal connection to the business activities
under investigation and the question of what records are left of them in the system.

1 The computer-aided software engineering technology products used at the Insurance
Corporation of British Columbia in the development of information systems are: Application
Development Work Bench (ADW) and Bachman. The use of a specific design technology may affect the
notation convention seen in documents produced from this software.
The specific methodology in use at the Insurance Corporation of British Columbia, is called the “Navigator System Series”. To clarify the relationship between the discipline of information engineering, and a specific methodology, it should be restated that: the discipline of information engineering is a generic class of methods that approaches systems development within the context of its larger business purpose; and that “Navigator” is the particular methodology used to develop systems at the Insurance Corporation of British Columbia. It should also be understood that the nature of the problem being automated, how the organization uses the methodology, and the means by which the project is controlled by the project team, are interpretations and judgments on the part of those using the methodology and affects the documentation of the project. Because information engineering is used as a mechanism to improve the productivity of the system development process itself, it may be used differently by different project teams. What the project documentation represents then, is one interpretation of one project, using one particular information engineering methodology.

Project Study Background

The project examined in this chapter is a system developed at the Insurance Corporation of British Columbia for the Claims Division. The claims payment system

---

2 Navigator System Series is a proprietary product of Ernst and Young International.

3 The stylistic idiosyncrasies of how Navigator presents deliverables is a matter of presentation or interpretation, and not of substance, which does not effect their usefulness in demonstrating the information engineering approach. It is also important to note that while it is the methodology that dictates the presentation style and content of project deliverables, the look and notation convention used in the diagrams is a consequence of the computer-aided software engineering (CASE) tool used to produce them.
project (hereafter known as CPAY) began in 1991 as a reengineering project to increase the capabilities of an existing system that was no longer supporting business objectives. The claims payment activity is a strategic and complex operational process that is tied directly to the primary business functions of Insurance Corporation of British Columbia, which are to provide insurance and settle claims. To understand the size and complexity of this project, it is noteworthy to add that the CPAY project is in its fourth year of development at a cost of approximately $1,130,000 or approximately 240 work months. The CPAY system serves roughly 2000 users and disburses $1.5 billion annually in claims payments.

Key objectives of the CPAY project were to: (1) provide an automated system to record payments and disburse claim settlement funds; (2) automate the payment process and eliminate manual cheque requisition; (3) improve service to claims customers by consolidating the processing of multiple or repetitive payments; (4) ensure the accuracy, legitimacy and validity of claims payments; (5) ensure that payments are made promptly and efficiently; (6) provide accurate financial information to management in a timely manner; (7) reduce the cost of maintaining the outdated technology of the older system; and (8) provide a system which can be integrated with existing and planned systems for the Claims division.

The claims cheque inventory function was automated during the second phase of the CPAY project. It was required to support the functionality of the main CPAY system described above and perform the following activities: (a) order new cheque stock; (b) assign and authorize cheque series; (c) confirm receipt of cheques by claims offices; (d)
print cheques at claims offices; (e) record spoiled cheques on-line, and; (f) access current US exchange rate on-line.

The cheque inventory function replaced several manual activities that pertained to the ordering and registration of blank cheque stock by a claims office. Blank cheque stock previously was ordered through an electronic mail request to the Banking Operations Department. There were three different types of cheques that could be ordered (manual, locally printed, and US). To replenish the supply of any type of cheque specific procedures were followed. A cheque register was previously used by the manager at a claims office to record: all cheques received from the Banking Operations Department; all spoiled cheques; and the return of all unused cheque books to the Banking Operations Department according to specific procedures.

As explained in the first chapter, the system development process is a complex activity that is, for the most part, represented graphically. This thesis will present a relatively simple and familiar action in order to manage the narrative explanation required to describe the effect of the system development process on the action's logical and

---

4 Manual procedures to order cheque stock were as follows: a) Claims office sent an electronic mail message to the Banking Operations department requesting the type of cheques required; b) Claims office kept original electronic mail message and made three additional copies of the message and forwards these to Banking Operations department; c) Banking Operations filled order; d) Banking Operations department retained copies 2 and 3 and returned 1st copy (as packing slip) with order of cheques to requester; e) Claims office would match packing slip with original to ensure order was filled accurately; f) Claims office would confirm receipt of cheques via electronic mail to Banking Operations department; g) Claims office would then enter the cheque book number into the Cheque Register.

5 Manual procedures to register cheques were as follows: (a) Entries into Cheque register were made; (b) Cheque register was sent to Banking Operations department with used cheque books and copies of spoiled cheques; (c) All cheques and cheque books were reconciled on a monthly basis to ensure all cheques were accounted for.
physical form. The logical unit of action selected for study is not, by itself, significant. What is important, is that it is understandable and easy to follow as the system development life cycle unfolds.

The identification of the logical unit of action that will be traced through the presentation of project documentation is based on what archivists would define as an act or action. During the analysis phase of information engineering it is referred to as an elementary process. An elementary process is defined by the methodology as the work that a business area does in reaction to an event. Both the logical unit of action and the elementary process are based on relationships to fact, event, and act. An elementary process is an action which forms part of a business transaction and can result in a record. Whether the unit selected for study by this thesis results in a record is determined by the existence of requirements articulated during the system development process and the translation of these requirements into design instructions.

The unit selected for analysis pertains to an activity that forms part of the Cheque Inventory Subsystem of the CPAY system. It is not defined until the analysis phase, when it is named in the system documentation as \textit{Manage Cheque Inventory}, and subsequently, in the design phase as \textit{Record Blank Cheques Received}. This process has two parts: the assignment of cheque series to a claims office; and the receipt of cheque series by a claims office. In the final implemented system, \textit{Record Blank Cheques Received} is represented as one of seven separate menu selections or applications (see: Fig. 2). As the presentation of work products will show, the purpose of \textit{Record Blank Cheques Received} is to update the inventory of blank cheques at a claims office. The action is initiated when Banking
Operations Department receives a cheque order from a claims office or when a claims office receives the blank stock from the Banking Operations Department. Upon receipt of an order of cheque stock the Banking Operations Department assigns a series of cheques to the claims office if stock is available. When a series of cheques is received, a claims office employee records *Cheque Order Receipt Date* on-line, which triggers several automatic system transactions, such as the updating of the inventory table to reflect the new quantity of cheques available for use.

The scope and complexity of CPAY requires that its presentation be confined in some way to facilitate analysis of its documentation for the purposes of this thesis. The selection of project documents was chosen to be representative of the entire systems life cycle (Planning, Analysis, Design, Construction). Excluded from the selection were views of the same process and data represented more than once by different techniques during iteration. Fig. 3 shows the work products that will be presented in this chapter arranged according to development phase and audience, and data or process classification.\(^6\)

\(^6\) This schema represents the information system architecture of John Zachman and is discussed in more detail in Chapter Three.
The logical unit of action under investigation is contained within the sixth application called **Claims Cheque Inventory Maintenance** and has five separate applications. The separate applications that comprise the Cheque Inventory Subsystem are: **Review Cheque Order**, **Maintain Re-order Information**, **Maintain Cheque Order**, **Receive Cheque Series**, and **Reset Next Cheque Number**. The logical unit of action is represented by the fourth menu selection **Receive Cheque Series** in the inventory subsystem of the implemented system.
<table>
<thead>
<tr>
<th>Planning (strategic planners, system architects and system analysts)</th>
<th>Data</th>
<th>Process</th>
<th>Strategy</th>
<th>Organization*</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>• Business Area Data Model</td>
<td>• Business Area Process Model</td>
<td>• Operating Plan</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Business Area Entity Relationship Diagram</td>
<td>• Business Area Context Diagram</td>
<td>• Corporate Vision, Mission and Commitment</td>
</tr>
<tr>
<td>Analysis (managers, knowledge workers, and system designers)</td>
<td></td>
<td>• Business Area Process Decomposition Diagram</td>
<td>• Action Program</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Elementary Process Definition</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Business Area Process Definition</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Intra-model Association Diagram</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Business Transaction to Procedure Diagram</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Application Structuring Diagram</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Procedure Description</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Design (system users and builders)</td>
<td>• Logical Database Design Diagram</td>
<td>• Screen Presentation Description</td>
<td>• Analysis Phase Charter</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Screen Flow Diagram</td>
<td>• Project Goals and Objectives</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Specification Screen Design</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Program Structured Chart</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Construct (builders and testers)</td>
<td></td>
<td>• Application System Report</td>
<td>• Design Phase Charter</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Module Action Diagram</td>
<td>• Design Phase Goals and Objectives</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 3: Claims Payment System (CPAY) Information Architecture

*The column of the framework called “Organization” is used to map business structure to business function. It is meant to show ownership and reveal the reporting structure in the hierarchy responsible for the project. It is empty in the case of CPAY, because this perspective is not required at ICBC because all projects typically have the same client or pertain to the same business function (Claims or Finance). In the case of CPAY, the project was a response to an existing function and organizational unit (Claims), therefore it was not necessary to connect the origin and relationship of the administrative hierarchy to the activities of concern to the project. In a dynamic organization with multiple functions, such as government, this type of representation would have greater practical use during system development than it does in this example.
Planning Phase Documentation

Planning Phase Deliverable: Operating Plan

The Operating Plan is not a deliverable of the methodology but rather a product of a cyclical business planning process already part of the established corporate culture at the Insurance Corporation of British Columbia (ICBC). The articulation and presentation of corporate-level goals and objectives generally varies from organization to organization. The illustrations used in the planning phase represent business goals that are linked to the CPAY project rather than to planning deliverables defined by the methodology. It is included here because the informational content is the same, even though the deliverable in this case deviates from that of the methodology.

The Operating Plan results from the annual corporate budget and planning process. It contains organization charts, a corporate mission statement and approved expense budgets for both the corporation as a whole, and for individual business areas (or corporate divisions).

Planning Phase Work Product: Corporate Vision, Mission and Commitment

The mission statement expresses the purpose, goals, objectives, and policies of the business. Goals are very general statements of aim that support the business mission, and are usually created at the executive level. Objectives are short term measurable targets that can be achieved in 10-12 months. The business mission must be supported by all new information systems.
The mission of the Insurance Corporation of British Columbia is “to provide protection from motor vehicle related loss to the citizens of British Columbia.” Fig. 4 is an excerpt that lists the specific corporate commitments (or goals) that link to the goals of the CPAY project.

- We will settle claims fairly, promptly, and in accordance with each claimant’s entitlement.
- We will emphasize productivity and spend our operating funds wisely
- We will take every opportunity to simplify and improve the way we do business for the benefit of our customers

Fig. 4: Corporate Vision, Mission, and Commitment [excerpt from]
(Operating Plan, 1993)

Planning Phase Work Product: Action Program Summary

The action program represents the support of corporate goals at the operational level of the enterprise. Corporate divisions translate business goals into programs of action which later evolve into projects. Each project then sets out to solve the problem or achieve the business opportunity which the action program described. In this way, the project is a response to undesirable conditions preventing the business from fully achieving its business purpose, goals, objectives, and policies.

Fig. 5 represents the approval required to authorize the development of a new information system, and to establish the project to do so. Relevant business commitments (goals) from the corporate mission statement are restated as “Corporate Direction/Commitment” and serve as the rationale or basis for approvals to initiate the CPAY project.
3.1.28

1992 ACTION PROGRAM SUMMARY

NAME: CLAIMS PAYMENT SYSTEM (CPAY) – PHASE I

OBJECTIVE:
To implement new business functions following implementation of the rewrite (Phase I) of the existing Claims Online Payment System (C.O.P.S.).

CORPORATE DIRECTION/COMMITMENT:
- We will emphasize productivity and spend our operating funds wisely.
- We will take every opportunity to simplify and improve the way we do business for the benefit of our customers.

COMMENTS:
The first phase of the Claims Payment System was a rewrite of the existing system utilizing new technology (TELON/DB2). This made the system compatible with our other operational systems and minimized system enhancement costs. This phase (II) will incorporate new system requirements to support and simplify the business process.
Examples:
- One payment for AB and BI settlement
- Automated repetitive payment option
- Alternative to payment requisition forms by providing the adjuster with easy to use on-line transactions
- One transaction for payment of work on the same claim done by more than one lawyer from the same firm

STRATEGY:

<table>
<thead>
<tr>
<th>Number</th>
<th>Description</th>
<th>Milestone Date</th>
<th>Cumulative Costs ($)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PHASE I (currently under way)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.</td>
<td>Business Requirements</td>
<td>Nov. 1991</td>
<td>195,000</td>
</tr>
<tr>
<td>4.</td>
<td>Implementation</td>
<td>Nov. 1992 (EST)</td>
<td>657,320</td>
</tr>
</tbody>
</table>

PROGRAM COMPLETION
Nov. 30/92 (EST) 657,320

POST-IMPLEMENTATION REVIEW
Dec. 31/92

N. Armstrong/J.T. Hunter  Sep. 6/91  H.G. Reid  Sep. 6/91
Manager Responsible  Date  Vice-President  Date

Fig. 5: Action Program Summary, Claims Payment System (Operating Plan, 1992)
Analysis Phase Documentation

Analysis Phase Deliverable: Project Charter

The project charter is a project management deliverable which defines the scope, goals, objectives, and overall approach of a project. It serves as a contract between the project team and the business. The charter defines what the project will deliver, the budget, time constraints, resources, and standards to which it must adhere. It also defines a discrete unit of work that can be independently staffed and managed. Comprehensive and accurate project documentation confirms and connects system development activities to aspects of project control essential to the quality of the project. The selection of deliverables is customized to meet one or more objectives of the project. Deliverables are developed as the project knowledge base is built up and refined during the process of iterative development.

Analysis Phase Work Product: Goals and Objectives

A project goal is a statement of direction for the development of the application project and is achieved when all project objectives have been satisfied. Objectives are derived from goals and represent a response to the problems that the project was established to address. They are measurable targets that the project team has set in order to meet its goals. Project objectives are short term targets with a duration of 12-14 months or less.

The definition of goals and objectives is typically an iterative process. In fact, goals and objectives are often adjusted a number of times throughout the course of a
project in response to user expectations, funding commitments, and changing business goals. The CPAY project addresses the goals (Fig. 6) of improving customer confidence, operational efficiency, and accuracy of payment information, which as seen in Fig. 2, represent areas of the business that are critical to achieving corporate goals.8

- Improve the payment process in the claims offices by allowing the adjuster to enter payment requests through a simple menu system. Elimination of the CL-288 form.
- Improve customer service and the efficiency of the payment process by providing the facility for repetitive payments.
- Improve customer service by providing the ability to produce one cheque for settlement of exposures on different files (accident benefit and bodily injury exposures), thereby improving the accuracy of information for accident benefits and bodily injury payments.

Fig. 6: Project Goals and Objectives [excerpt from]
(CPAY Analysis Phase Project Charter PM1100)

Analysis Phase Work Product: Project Scope

The project scope defines the boundaries of the project by identifying the information and processes it requires to solve the business problem. This information includes business area data, process, and organization scope statements and models. The scope creates a boundary for applying and monitoring changes as the system development process evolves. It is reconciled with the project goals and objectives to verify that there is sufficient information to proceed. When the scope of a project is improperly or inadequately defined, there is the possibility that both the project and the system either will

8 However, the elementary process is a logical term that is only used during one phase of development. It is transformed in the design phase into a procedure that forms part of a transaction, then into a module of code.
not provide the functionality users and by management, or will provide extraneous functionality that is not required. Furthermore, a poorly defined scope will make revisions necessary to the project products resulting in a disruption to project plans and schedules. In the analysis phase, the objective of the project scope is to understand the problem in the context of the existing business environment (Fig. 7A+B).

Analysis Phase Deliverable: Business Area Information Model

The business area information model is a detailed subset of the enterprise, and includes models of data and process, their associations, and business area dependencies. It consists of three parts: the business area data model, the business area process model, and the business area behaviour model. These separate models are integrated later into the global model of the enterprise.

Analysis Phase Work Product: Business Area Context Diagram

The context diagram provides the most general picture of the information system. The system is depicted as a single process (business area process) or logical unit of work that is undertaken by the enterprise to achieve its mission. It is the starting point for a detailed analysis of all processes performed by the business area under investigation by the project. By analyzing who and what the process interacts with, the context diagram aids the project team define the boundaries of the business area.

The context diagram for CPAY supplements the more formal description of scope and can be used in conjunction with a data flow diagram to expand specific sections into
more detail. Fig. 8 shows what information is sent or received by all known external agents, databases, and high level data flows which interact with the claims payment process. The inward data flow represents an event to which the business area must respond and which causes an elementary process or instance of the action to occur. In this case, "request for cheque stock" is the event. The Banking Operations Department reviews "cheque stock detail" and determines how many cheques are needed to replenish the supply. An order for this number of cheques is then generated by the system.

**Analysis Phase Work Product: Process Decomposition Diagram**

At the highest level of the enterprise are its functions. A function may be defined as a group of business activities that together support an on-going aspect of the enterprise mission, such as the Purchasing or Payroll functions. Functions are not organizational units, and are independent of present and future organizational structures. It is useful to think of the very highest level functions as a broad subject area that can be decomposed into other functions, then into processes.

Process modeling describes what an enterprise does, how it carries out its work, and how information systems are used to support business processes. A process is defined as a series of repetitive actions that are well-defined in terms of input, transformation and output of entities, relationships and attributes. Processes can be decomposed into other processes called elementary processes, which are the basic units of a business area process model. They represent the most discrete component of that model and have a definable start and finish. Elementary processes are logical, which means that they are designed
PM1105 Project Scope

Project Scope

Business Area Scope

The scope of the Claims Payment System Analysis Project can be specified in terms of data, process, organization, and business strategy models.

Data Scope

The data scope of the Claims Payment System Project is restricted to the following entity as shown in the Data Scope - Entity Model and Data Scope - Data Architecture diagrams:

- Payable

This entity type represents the baseline data that will be modelled by this phase of the Claims Payment System Project. Additional subject areas, which have been included in the Entity Model, will be reviewed during the course of the Analysis Phase:

- Claim
- Customer Set
- Employee
- ICBC Insurance Policy
- Loss
- Organization Unit
- Receivable
- Supplier

Process Scope

In relation to the ICBC Functional Business Model, the process scope is restricted to the high level process of Claim Payment Administration, Claim Supplier Administration and Accounts Payable (refer to the Process Scope - ICBC Functional Business Model).
Within Claim Payment Administration, the process scope of the Claims Payment System Analysis Project is restricted to the following processes (refer to the Process Scope - Claim Payment Administration Model):

- Request No Fault Weekly Indemnity Payment
- Request Rehabilitation/No Fault Medical Payment
- Request Payment for Claim Expense
- Request Payment for Material Damage Claim
- Request Payment for Bodily Injury Claim
- Request Payment of Death Benefits
- Authorize Claim Payment Requisition
- Issue Claim Payment
- Reconcile Payment to Payment Authority

Within Claim Supplier Administration, the process scope of the Claims Payment System Analysis Project is restricted to the following processes (refer to the Process Scope - Claim Supplier Administration Model):

- Develop Directions for Payment to Supplier
- Identify Claim Supplier
- Prepare Payment Statement for Supplier

Within Accounts Payable, the process scope of the Claims Payment System Analysis Project is restricted to the following processes (refer to the Process Scope - Accounts Payable):

- Maintain Financial Data on Supplier
- Validate Payment Invoice/Request
- Reconcile Invoice/Payment Request Discrepancy
- Authorize Payment for Liabilities Incurred
- Calculate Payment Amount
- Issue Payment for Corporate Liability
- Reconcile Payment Issued to Payment Cleared

Additional lower level processes needed to satisfy the stated project objectives will be added to the model as they become identified.
Fig. 8: Business Area Context Diagram  
(Business Area Process Model B2000)
independently of physical or technological constraints.

During the Analysis phase, the functional decomposition technique is used primarily to display elementary processes identified through event analysis and other process modeling techniques. Event analysis concentrates on the stimuli to a business area which is in contrast with elementary process identification which determines what the business area does in response to such stimuli.

The main concepts of process modelling pertain to events, to elementary processes, and to business transactions. An event is something outside the business area to which the business area must react, and may only be initiated by an external agent or by the passage of time. As already mentioned, an elementary process is the work that a business area does in reaction to an event, or to a management decision to produce all the possible responses. A business transaction from the point of view of process modeling, is a description of how a user organization deals with an event. The description will later inform system analysts and programmers how the procedure can be executed in terms of technology, and shows how the transaction handles all inputs and outputs from one or more elementary processes.

Fig. 9 shows the logical unit of action as a process and its composite elementary processes. It is an extract from the process decomposition diagram and represents a hierarchical breakdown of the process and the actual work it carries out. The full diagram (not shown) describes the functions and activities of the claims payment business area that will have some interaction or direct involvement with the proposed system. The upper level processes are non-elementary processes that are created by consolidating elementary
processes into cohesive groupings. The process *Manage Cheque Inventory* is defined in the deliverable as “all elementary processes and data required to ensure that blank cheques are available when required by claims offices.” It decomposes into three elementary processes of which *Record Blank Cheques Received* is one. Each elementary process under the larger process must happen once to execute the *Manage Cheque Inventory* process.

![Process Decomposition Diagram](Business Area Process Model B2000)
Analysis Phase Work Product: Business Area Process Definition

The business area process definition is the starting point for analyzing the processes performed by the project. The boundary of what constitutes a business area is usually determined by the project scope. Fig. 10 describes the high level process represented in the business area context diagram (Fig. 8).

![Business Area Process Definition](image)

Fig. 10: Claims Payment Definition [excerpt from]
(Business Area Process Model B2000)

Analysis Phase Work Product: Elementary Process Definition

The definition of the elementary process describes the properties associated with the process. Defining the elementary process is the first step in articulating formal business rule specifications, which helps establish a concise way to communicate and understand the process. Every elementary process is governed by a rule, policy or practice of the business area and must have at least one activating event that produces a change in the data associated to it.

In Fig. 11, the definition identifies the entities involved in the elementary process by capitalizing them. It describes the work it performs, names the transaction of which it forms a part, the event which triggers it, the business rules that govern it, and explains how data security issues will be addressed.
Process: PR:RECEIVE CHEQUES

Définition

RECEIVE CHEQUES will allow the Claims Office to enter the receipt of CHEQUES sent by BANKING OPERATIONS.

BUSINESS TRANSACTION

RECORD BLANK CHEQUES RECEIVED

TRIGGER

Name
A Claims Office receives blank CHEQUE stock

OPERATIONAL MODE

Online - Update and Inquire

DETAILED INSTRUCTIONS

A Claims Employee will be prompted to provide details pertaining to the CHEQUES that were received. CHEQUE ORDER will be updated to confirm receipt of the specified series. The operators claim center will be used as the default as well as the most recent cheque order. The order date can be changed to reference prior orders. The Received Date and Received By fields will be for display purposes only. The Series Modification Flag may be used to either allow or disallow Banking Operations from updating the cheque series assigned to an order. The default will be to disallow Banking Operations from updating the cheque series.

SECURITY AND ACCESS

A Claims EMPLOYEE assigned to manage the CHEQUE INVENTORY within a Claims Office

Fig. 11: Elementary Process Definition: Receive Cheques
(Business Area Process Model B2000)
Analysis Phase Work Product: Intra-Model Association Diagram

The intra-model association diagram, or "CRUD" matrix as it often called, shows how the data will be modified by the processes that act on them. It is an overview which analyzes and documents the relationships between these two types of information. It is used to verify that the data and process models are complete in supporting each other, or rather, that all entities defined in the model are used by the processes identified by the project.

Each entity has a life cycle that begins with the creation of an occurrence of the entity, followed by a number of updates to the entity occurrence, and finally with the optional deletion of the entity. Each cell shows the type of action that the elementary process performs on the entity. These actions are either create (C), read (R), update (U), or, delete (D), hence CRUD matrix.

Fig. 12 shows how the elementary process Record Blank Cheque Received acts on specific entities from the claims payment business area. The create process implies that some new information has been created in the course of the process executing itself (output). For example, a table is created for each Cheque within a Cheque Series. Cheque Inventory updates the quantity of cheques on hand and updates the Cheque Order table. The update process implies that information has been created elsewhere that modifies the existing data (output from other system). The elementary process Record Blank Cheque Received reads, but does not change, the following entities: Position, Organizational Unit, Employment, Employee Position Assignment, Employee and Cheque Type.
Analysis Phase Work Product: Business Area Entity-Relationship Diagram

The purpose of the entity-relationship diagram is to develop, define, and improve the understanding of data by illustrating entities and relationships in the form of a data model. It is a logical map that represents the inherent properties of the data independently of software, hardware, or machine performance considerations. A data model is not just a schematic, it is also the underlying set of definitions and descriptions of the information objects and the rules that govern them. A data model is the set of definitions and descriptions of information objects about which the enterprise needs information. An object refers to any person, place, or thing, that represents the rules in the model. The entity-relationship model is validated against the process definitions to ensure that any entities or relationships required are present and that all entities or relationships are used.

The entity-relationship diagram must represent the data needed to satisfy the data scope (Fig. 7A+B) and support the objectives for the project as established in the analysis phase project charter. It must correctly represent the facts or business rules about entity...
types as understood by the Claims, Claims Operations Regions and Banking Operations departments. This information is contained in associated descriptions about entities and attributes and is kept in the project encyclopedia or knowledge base. Entities related to the logical unit of action are shown in Fig. 13.

Fig. 13: Business Area Entity-Relationship Diagram
(Business Area Data Model B4000)
Analysis Phase Deliverable Name: Conceptual System Design

The objective of the conceptual system design report is to demonstrate that the idea, concept or change, as described by the deliverables that preceded it, can be accomplished by automation. The conceptual system design report summarizes the options for automating the business area and recommends an approach to take that is within the constraints of the application architecture of the enterprise. The report outlines what external components (such as screens, reports and forms), and what processing, storage, and other technical structures will be required to build the system. The conceptual system design report represents a bridge from the logical to physical domains as the focus of the project shifts from understanding what to do (data and process) to how it can be done (technical design). This results in a transformation that changes the perspective of the project, the audience, and the tools and techniques required to achieve it.

Analysis Phase Work Product: Business Transaction to Procedure Diagram

The business transaction to procedure diagram represents the transformation of elementary processes into business transactions. This is not a decomposition but rather a lateral shift away from the logical understanding of the business problem towards the design of its physical implementation. A business transaction is the method of implementing one or more elementary processes in the design environment. In other words, a business transaction is a representation of ‘how’ an elementary process gets
implemented as a series of procedures whereas an elementary process describes 'what' should be implemented in an information system.

A vast range of considerations go into identifying business transactions. The process of selecting business transactions is a highly creative one, which benefits from the various insights and skills of the project team and draws on their business knowledge, appreciation of information technology possibilities, and organizational development skills. The team decides which parts of the business transactions will be automated and which should remain manual, then explores different ways in which this can be achieved.

Fig. 14 describes how the CPAY system will process data in response to certain business events. The process Manage Cheque Inventory has been renamed Cheque Inventory Subsystem as development of the system begins to consider aspects of physical implementation. The elementary processes have become three business transactions under the Cheque Inventory Subsystem. The elementary process Record Blank Cheques Received is now a business transaction by the same name. Each business transaction typically implements one or more elementary processes.

Analysis Phase Work Product: Application Structuring Diagram

The application structuring diagram is a graphic depiction of the major functions to be performed by the new system. This diagram shows the logical boundaries of the second phase of the CPAY2B application. Fig. 15 shows that the business transaction Record Blank Cheques Received has two candidate procedures: Assign Cheques and Receive Cheques. A procedure represents one individual work step (act) within a business
transaction. The information in this diagram is augmented by the context diagram (see: Fig. 8) which reveals internal inputs and outputs to the business area.

**Analysis Phase Work Product: Procedure Descriptions**

Procedures become the individual steps required to accomplish the business transaction. The procedure description is intended to verify and communicate to users, management, and the rest of the project team how the elementary process will function as a procedure.

The properties which comprise the descriptions in Fig. 16 are intended to convey the following meanings:

- **Name** - communicates what the procedure does and how it is executed (operational mode) from the perspective of the application

- **Definition** - relevant business rules, logic and operations, including any significant business exceptions are outlined

- **Activating Events** - the events or physical means by which the procedure is activated

- **Operational Standards** - refers to how the step is executed using some kind of technology. The most common operational modes are on-line, batch, and manual.

- **Procedural Outline** - step by step explanation of how users will perform the procedure. Note that this is the most important property in the description because it explains how the business transaction will be accomplished.
Fig. 14: Business Transaction to Procedure Diagram
(Conceptual System Design Report M0000)
Fig. 15: Application Structuring Diagram
(Business Requirements CPAY 2B)
Process: BT:RECORD BLANK CHEQUES RECEIVED

Definition
PURPOSE
To assign and receive CHEQUE stock and update INVENTORY.

ELEMENTARY PROCESS
RECORD BLANK CHEQUES RECEIVED

TRIGGER
Name

Banking Operations receives a CHEQUE ORDER from a Claims Office or a Claims Office receives blank CHEQUE stock from BANKING OPERATIONS.

PROCEDURAL OUTLINE
Upon receipt of a CHEQUE ORDER, Banking Operations will assign CHEQUE SERIES if stock is available. Upon receipt of the CHEQUE SERIES, a Claims Office will record the receipt of the CHEQUE INVENTORY and inventory stocking details will be updated.

IMPLEMENTATION IMPORTANCE
Low

OPERATIONAL STANDARD
A Claims Office should update inventory as soon as it is received.

Process: EP:RECORD BLANK CHEQUES RECEIVED

Definition
RECORD BLANK CHEQUES RECEIVED accepts cheque stock detail from BANKING OPERATIONS which consists of new CHEQUE SERIES. The Claim Office Cheque Order Receipt Date is recorded, the CHEQUE INVENTORY is updated to increment Quantity on Hand, and CHEQUE SERIES and CHEQUES are created for each series received. The EMPLOYEE who updated the CHEQUE INVENTORY is recorded and must be currently assigned to the same Claim Office that manages the inventory.

Purpose: To update the blank cheque inventory at a Claim Office.

Activating Events:
Name: BANKING OPERATIONS sends cheque stock detail
Type: External
External-Data Flow: cheque stock detail
External-Frequency:

Location: Claim Offices

Responsibility: Claim Offices

Operational Standards: All blank cheques received by a Claim Office must be recorded the same day.

Fig. 16: Procedure Descriptions [extracts from]
(Business Requirements CPAY2B)
Design Phase Documentation

Design Phase Deliverable: Design Phase Project Charter

The Design Phase Charter is the planning document that addresses the technical approach to the design problem that will be taken. The technological solution recommended in the conceptual system design report is the input to this document. It is an iterative refinement of the scope, objectives, and overall approach of the project, and includes roles and responsibilities, deliverables, project plan, and costs.

Design Phase Work Product: Design Phase Goals and Objectives

Project goals identify the problems that the system will address and state the approach that the development will take. Goals in the design phase are more specific than in the analysis phase because they now concern how solutions can be accomplished with available technology.

The design phase begins with the recommendation that the claims payment cheque inventory function be automated. Goals and objectives relevant to the second phase of CPAY are illustrated in Fig. 17. The triggering problem can be traced back through the planning and analysis phases to link the goals of the design phase to the business function and activities that initiated the project.

9 Planning is always the first stage of every phase. It should not be confused with the planning phase of the methodology which pertains to the strategic planning for the enterprise.
Currently, cheque ordering is a manual process which each claim centre must perform. There is an opportunity to allow automatic ordering of cheques for claim centres which would prevent claim centres from running out of cheques and would also allow better administrative efficiency of managing claim cheque inventories.

On occasion, cheque orders have been misplaced or misdirected. This system should allow better tracking or ordering, filling and receiving of cheques at claims centres.

This project provides an opportunity to standardize how manual and local cheques are acquired and depleted from a claims centre inventory.

Fig. 17: Design Phase Goals and Objectives CPAY 2B [excerpt from]
(Design Phase Charter PM1100)

**Design Phase Work Product: Design Phase Project Scope**

The project scope during this phase identifies the logical boundaries of data and process required by the design based on descriptions from the conceptual system design report. It defines which business transactions and procedures will be automated and what data they will use. It also describes which entities need to be created and which are pre-existing, and therefore able to be reused or modified by the activities of the design phase project (Fig. 18).
PROJECT SCOPE

Based on the scope outlined in the CPA2Y Conceptual System Design (CSD) document, the following has been identified as the logical scope of the project:

Process Scope

The process scope of CPA2Y PHASE 2B is defined as the following Business Transactions and Procedures as shown in Figure 1:

- Re-order Blank Cheques Business Transaction:
  - Generate Cheque Order Procedure
  - Maintain Cheque Order Procedure
  - Review Cheque Order Procedure

- Record Blank Cheques Received Business Transaction:
  - Assign Cheques Procedure
  - Receive Cheque Procedure

- Maintain Re-Order Information Business Transaction:
  - Maintain Re-Order Information Procedure

- Reset Next Cheque Number Business Transaction
  - Reset Next Cheque Number Process

- Maintain Exchange Rate Business Transaction
  - Maintain Exchange Rate

For further detail on the procedures identified above, refer to Appendix A.

To implement CPA2Y Phase 2B, enhancements will need to performed on CPA2Y Phase 1, COPS, and CDUP. It is anticipated that an enhancement project will deliver these changes at the same time this project is completed in March 1994.

Enhancements to CDUP, CPA2Y Phase 1, and COPS are also required to implement US Cheque Printing and Multiple Cheque Printing. US Cheque Printing and Multiple Cheque Printing will not be delivered in this project.

Data Scope

The data scope of CPA2Y PHASE 2B includes the following entities as shown in Figure 2:

- Cheque Series (new)
- Cheque Order (new)
- Cheque Inventory (new)
- Exchange Rate (new)
- Cheque (new)
- Cheque Type (existing but not in production)
- Currency (existing)
- Organization Unit (existing)
- Position (existing)
- Employee Position Assignment (existing)
- Employment (existing)
- Employee (existing)

Fig. 18: Design Phase Project Scope CPA2Y2B
(Design Phase Charter PM1100)
Design Phase Deliverable: Business System Design

The business system design deliverable specifies the appearance and operation of the application from the perspective of the end-user. The purpose of this report is to fully define and reach agreement on specific components of the application before proceeding with technical specifications for construction. Detailed design involves translating the conceptual system design into a set of specifications for programming. This report describes the on-line, batch, and manual procedures that must be executed in order to complete the business transactions. It includes the logical database design for the application as well as specifications for all screens, reports, and forms.

Design Phase Work Product: Screen Presentation Description

The screen presentation description explains how the user will interact with the screen and how the application will be used. It is used by systems designers as the starting point for programming, and by users, as a functional specification. The screen presentation description explains how the procedures will effect the data (update, inquire, add, delete). It is a narrative description of the same relationships presented in the intra-model association diagram (see: Fig. 12). Screen design is still a conceptual or logical activity; technical specifications are developed later that will utilize a specific technology for the construction of the final screen layout.

Fig. 19 describes how the business transaction will act on the data (update, inquire, add, delete). The section 'Key fields' describes how a row in the database will be retrieved. 'Navigation' describes how the user can move around the screen and what
processes can be executed during the transaction. 'Edits' refers to the set of constraints (business rules) applied to the procedure (attributes or data elements).

---

**Presentation: RECEIVE CHEQUE SERIES**

**Definition**

**DESCRIPTION:**

The purpose of the RECEIVE CHEQUE SERIES conversation is to allow Claim Centres to acknowledge receipt of the cheque series sent by Banking Operations. Inquiry access will also be available.

Upon initial entry to this screen, the Claim Centre will default to the operator's claim centre. The Cheque Type must be entered. The Order Date will default to the most recent order for the Claim Centre and Cheque Type specified. The Order Date may be entered to reference specific orders. The Order Sent Date, Quantity Ordered, Quantity Sent and Cheque Series are for display only.

If an order is correct, 'Y' should be entered in the Order Verified field and F6 entered to save the verified order. This will prevent Banking Operations from making any changes to the assigned cheque series.

Once an order is accepted as correct, the Order Received date is set to the current date. The Order Received By is automatically set to the Id of the employee that received the order. New entries should be created in the CHEQUE table for every CHEQUE in the CHEQUE SERIES. The status of each of these CHEQUES should be set to 'AV' for Available For Use and the Status Date should be set to the current date.

If an error is discovered in the order after it has been verified as correct, the order may still be set back to not verified. All CHEQUES should be deleted from the CHEQUE SERIES and added again when the order is verified as correct.

The Order Verified Flag may not be changed from 'Y' to 'N' if any of the CHEQUE SERIES are active. It may also not be changed if another CHEQUE ORDER for the same claim centre and cheque type was added at a later date.

If an order is incorrect, 'N' should be entered in the Order Verified field and F6 should be entered to save the information. This will give Banking Operations update access to modify the assigned series. Banking Operations must be manually notified to correct the problem. The Claims Office can then re-execute the Receive Cheque Series transaction and verify the order.

---

Fig. 19: Screen Presentation Description (Receive Cheque Series)
(Business System Design CPAY2B)

**Design Phase Work Product: Logical Database Design Diagram**

The logical database design diagram is the first step toward the physical organization of data in a relational database. It depicts the structure of how data will be stored and accessed by the computer system. The term logical means that the design is independent of technology constraints.
The logical database design diagram represents a specific set of entities, relationships, and attributes which can be implemented as a complete unit (or table). This diagram is based on the analysis and business rules represented in the entity-relationship diagram. The physical version of the entity is the database table. Each cell represents a data structure or logical record which is the unit of data storage on which a computer program operates. Each cell describes the physical characteristics of the columns that comprise the tables in a database. In a relational database such as this, the entities (e.g., cheque order, cheque series, cheque, cheque inventory, cheque type) become the tables, the data, that gets captured after it is built, become the rows, and the relationships among the entities can become columns within other tables. Tables are related to one another via repeating fields or keys. Access to the database can be made according to this database structure, which is controlled by a database management system. The database management system converts the program into physical addresses that retrieve the data required by the program, which this structure represents.

In Fig. 20, the logical data structures for all business transactions being automated by the CPAY2B project are depicted. The entities used by *Record Blank Cheques Received* are: Cheque Series; Cheque Order; Cheque Inventory; Cheque; Cheque Type; Organizational Unit; Position; Employee Position Assignment; Employment; Employee.
Fig. 20: Logical Database Diagram - Claims Cheque Inventory
(Business System Design CPAY2B)
Design Phase Work Product: Screen Flow Diagram

This diagram provides an overview of screen navigation and access from the user's viewpoint. The screen is the mechanism for users to exchange information with the computer. The diagram verifies that the workflow matches how the system will interact with users; and confirms that both the system designers and users understand the approach, and that it is ready for construction.

Fig. 21 depicts which screens or procedures are accessible to which users, as well as, the entry and exit points for users to escape the application. The two procedures that comprise the Record Blank Cheques Received action are separate screens (Assign Blank Cheques and Receive Blank Cheques). Each of these screens is linked via an access path to the authorized user or participant in the business transaction. e.g. Banking Operations Department assigns blanks cheques to a claims office and the claims office verifies receipt through the screen Receive Blank Cheques. It is also evident from the screen flow diagram that the claims office cannot assign cheques but can only receive blank stock.
Fig. 21: Screen Flow Diagram
(Business System Design CPAY 2B)
Design Phase Work Product: Screen Specification Design

Each screen typically equals one procedure in a transaction. A screen is a type of display environment with specific physical and functional characteristics. Eventually, each data structure is developed into a screen. The specification stage of screen design verifies that the design is functionally efficient and aesthetically pleasing, before developing complete and accurate documentation for the final layout. This work product is used in conjunction with the screen description by programmers as they start to build the programs that automate the procedure.

Fig. 22 shows what the screen looks like and how it will be used. It indicates which fields borrow data from existing tables in the database and which are entered by the users. Edit and control functions are described for user-modified fields. This specification is used as a prototype in the development process to evaluate such things as screen navigation, spatial layout, and field titling with users and system designers.

<table>
<thead>
<tr>
<th>CLAIMS PAYMENT SYSTEM</th>
<th>XXXXXXXX</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPAYRCV</td>
<td>RECEIVE CHEQUE SERIES · XXXXXXXX</td>
</tr>
<tr>
<td>COMMAND:</td>
<td>XXXXXXXX (DDMMYYYY)</td>
</tr>
<tr>
<td>HH:MM:SS</td>
<td>XXXXXXXX</td>
</tr>
<tr>
<td>ORDER VERIFIED:</td>
<td>A (Y/N)</td>
</tr>
<tr>
<td>CLAIM CENTRE:</td>
<td>ZZ9 Xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx</td>
</tr>
<tr>
<td>CHEQUE TYPE:</td>
<td>XXXXXXXX</td>
</tr>
<tr>
<td>ORDER DATE:</td>
<td>XXXXXXXX</td>
</tr>
<tr>
<td>QUANTITY ORDERED:</td>
<td>XXXXXXXX</td>
</tr>
<tr>
<td>QUANTITY SENT:</td>
<td>ZZZZZ</td>
</tr>
<tr>
<td>QUANTITY SENT DATE:</td>
<td>XXXXXXXX</td>
</tr>
</tbody>
</table>

Fig. 22: Specification Screen Design (Receive Cheque Series)
(Business System Design CPAY 2B)
Design Phase Deliverable Name: Technical System Design

The technical system design report represents the transformation of the design requirements into the physical environment. The report includes the physical database design and complete specifications for the construction of the application software, including data access requirements, performance requirements, and database procedure and utility design.

Design Phase Work Product: Program Structure Chart

A program structure chart presents a graphic image of a computer program and is used as a blueprint by a programmer when constructing a computer program. It is a graphic display of the hierarchy of modules or instructions in a program and the relationships between the modules. Modules comprise the structure of programs and applications. To deal with the complexity of logic, programmers break programs into modules. Although the structure chart becomes the input to module action diagrams and eventually to construction (code generation), it does not reference any earlier diagrams or knowledge objects from previous phases because the development has shifted from the logical domain of problem identification to the physical realm of technology platforms and the existing technology architecture. A structure chart and its associated module action diagrams form a complete specification which is used as an implementation model for a program. Issues considered during physical design include: file organization (random or index); access method (hash, index, pointers); establishment of file size to allocate disk space; security procedures; audit requirements; and deletion decisions.
Fig. 23: Program Structure Chart: Receive Cheque Series (Module PY150X)  
(Business System Design CPAY 2B)

Fig. 24: Program Structure Chart: Receive Cheque Series (PY150XX) [Detail of first level]  
(Business System Design CPAY 2B)
The program structure chart (Fig. 23 and 24) depicts the transition from logical to physical design for the purposes of coding the program. The execution of these procedures accomplishes the business transaction *Record Blank Cheques Received* which has become program PY150X.
Construction Phase Documentation

Construction Phase Deliverable Name: Application System

This deliverable includes all of the software constructed and tested during the construction phase. The software is constructed according to the specifications and development approach described in the design phase.

Construction Phase Work Product: Module Action Diagram

A module action diagram details the logic and structure for a module of programmed code using actions and brackets. It combines process logic notations with graphics and text to support the definition of technical rules. Brackets are the basic structuring device and are used to show the hierarchy or structure of a specification of procedural logic. Brackets show the flow of control between actions and are used to organize related actions. An action is a statement to manipulate, access, or communicate data. It may be expressed in formal, structured or natural language. The action diagram is used with the structure chart to represent the structure of a program and communication between the data.

The purpose of the module action diagram is to fully specify the procedural steps taken by a program and its modules in the automation of a business transaction. In the case of CPAY2B, the action diagram became the input for code generation.\textsuperscript{10} In this abbreviated form, which combines a work flow structure with natural language comments

\textsuperscript{10} The computer-aided software engineering design tool “Application Development Workbench” (ADW) was used to generate COBOL (Common Business Oriented Language) code from the module action diagram.
** PY150X21 - EDIT CHEQUE TYPE

Left justify

MOVE ZEROS TO WMAME-TALLY
EXAMINE WMAME-CS-CT-TYPE-CDE TALLYING LEADING SPACES
ADD 1 TO WMAME-TALLY
MOVE WMAME-CS-CT-TYPE-CDE(WMAME-TALLY:) TO CHEQUE-CS-CT-TYPE-CDE

** Edit Cheque Type Exists

** Call to Cheque Type Table

MOVE WMAME-CS-CT-TYPE-CDE TO CPGCT-CS-CT-TYPE-CDE
MOVE 100 TO DMSXX-DB2-ACCEPT-STATUS(1)
Read Rel. Table

<table>
<thead>
<tr>
<th>CHEQUE TYPE</th>
<th>Find Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPGCT</td>
<td>Using: CT_SLCT</td>
</tr>
</tbody>
</table>

MOVE 0 TO DMSXX-DB2-ACCEPT-STATUS(1)

** Check return code

IF CPGCT-STATUS NOT = SPACES.
** Set Cheque Type does not exist message

Invoke

<table>
<thead>
<tr>
<th>CM01 - STACK HELD MESSAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>CM01</td>
</tr>
</tbody>
</table>

HIGHLIGHT WMAME-CS-CT-TYPE-CDE
SET ZZGSP-KEY-ERRORS TO TRUE
END-IF

** Edit Cheque Type valid for Claim Center

IF NOT ZZGSP-KEY-ERRORS

** Call to Cheque Inventory Table

MOVE ZZGSP-USER-LOCATION TO WMAME-CS-ORGUNIT-ID
MOVE WMAME-CS-ORGUNIT-ID TO CPGCI-CS-ORGUNIT-ID
MOVE WMAME-CS-CT-TYPE-CDE TO CPGCI-CS-CT-TYPE-CDE
MOVE 100 TO DMSXX-DB2-ACCEPT-STATUS(1)
Read Rel. Table

<table>
<thead>
<tr>
<th>CHEQUE INVENTORY</th>
<th>Find Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPGCI</td>
<td>Using: CI_SLCT</td>
</tr>
</tbody>
</table>

MOVE 0 TO DMSXX-DB2-ACCEPT-STATUS(1)

** Check return code

IF CPGCI-STATUS NOT = SPACES.
** Set Cheque Type not valid for Claim Center

Invoke

<table>
<thead>
<tr>
<th>CM01 - STACK HELD MESSAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>CM01</td>
</tr>
</tbody>
</table>

HIGHLIGHT WMAME-CS-CT-TYPE-CDE
SET ZZGSP-KEY-ERRORS TO TRUE
END-IF
END-IF

Fig. 25: Module Action Diagram (PY150X)
(Business System Design CPAY 2B)
and some high level coding statements, the program is easier for programmers and
designers to read and conceptualize than is the actual written code. A module becomes a
section with a specific function within the generated COBOL source code. Fig. 25 shows
the procedural instructions for one module of code from program PY150X.

Project Documentation Summary

The type and nature of every system development project is different from the next
because every problem and solution is different. The work products and deliverables
illustrated were selected to be representative of the entire system development life cycle.
The complexity of system development is underscored by the presentation of CPAY
documentation and the realization that the selection presented is only a small percentage
of the total documentation.

The work product profiles introduced the descriptive framework used by the
information engineering methodology and provided an explanation of some of the
terminology used in the practice of information management. Tracing the movement of
the logical unit of action through the development phases highlighted how its form was
affected by the requirements of database technology. In the planning phase, the unit had
not yet developed as a separate action, though its seed existed in the high-level plans of
the enterprise. In the analysis phase the unit became a process, then an elementary process
required to execute the Manage Cheque Inventory process. During design, the logical
unit of action became part of a business transaction that decomposed into procedures, and
later into structured modules used to generate code.
The transformation of the unit of action through the phases of system development revealed how the constraints of database technology, imposed during the design and construction phases caused the action to fragment and change. In effect, this simple unit of action is a microcosm of the larger CPAY system, the business function it belongs to, and the goals of the enterprise.

The significance of information engineering and its documentary products will be examined in respect to recordkeeping in the next chapter.
CHAPTER THREE
THE SIGNIFICANCE OF INFORMATION ENGINEERING
FOR RECORDKEEPING

In the last chapter, the documentary products of a system development project were presented to illustrate information engineering techniques and methods, and the system development life cycle. The emphasis of the presentation was on the order of the phases, the nature of information engineering activities, and on the modes of representation used to depict these activities. At the same time, a logical unit of action was followed through the phases of system development to show the impact of the process on the unit and the fragmentation of its form.

It is outside the boundaries of this study to analyze the archival nature of the physical data record after the system is implemented in an office environment and is used to capture data. To attempt a definitive evaluation that determines the archival nature and the reliability of the record would be a substantive enough investigation in its own right to support the work of another thesis. Such a study would require a detailed analysis of all variables that could effect the integrity of the record, and they are numerous. Some discussion of these questions will come up in this chapter, but its goal is ultimately to demonstrate the usefulness of information engineering as a framework for analyzing information in electronic form, and the importance for archivists of understanding the
system development process and its representation. The impact of these findings on traditional archival concepts will also be discussed.

Data and information in electronic form challenge some basic archival concepts and definitions, and for good reason. The difficulty in determining what we have in electronic form makes it difficult to do anything with the information, whether it is in a form that constitutes records or not. Records are documents created in the course of conducting organizational business. They contain information about the activities and transactions of which the record forms a part and the context in which the record is created. Traditional archival methods accomplish the task of capturing contextual meaning by capturing evidence of the origin and genesis of the records, by maintaining original order among the records, by creating histories of their administrations, and by identifying the relationships among the records in larger aggregations, like *fonds* and series.\(^1\)

The archival profession has been struggling to define records in electronic form. The problem that these records present to archivists is complicated by their distributed form. If these records do not have a stable and complete physical form, and take what shape they do have from the rules and relationships embedded in their coded programs, how can they be identified and preserved as evidence in our bureaucratic and juridical systems? David Bearman, in what has been called the “new provenance theory”, equates

---

1 According to Heather MacNeil, adherence to the principle of *respect des fonds* requires that the internal and external structure of the various contexts of document creation be analyzed in order to better reveal their relation to one another. Internal structure allows us to infer the genesis of the form and transmission of the documents, the procedural relationships among them and the purposes they served in an administration or personal environment. External structure is established in the process of delegating authority and function which is reflected in hierarchies, function and associated activities. See: H. MacNeil, “Archival Theory and Practice: Between Two Paradigms,” *Archivaria 37* (Spring 1994): 6-20.
evidence with information about the content, structure and context of a record.² Ron
Weissman says that information cannot constitute evidence if it is deprived of structure,
that records have both a structure and are created within structures, and that their value is
due to these structures.³ Others have written that the preservation of impartiality depends
on the preservation of context which is the truth of the activities that the record represents
by virtue of its archival nature.⁴ Context is the basis of a new view of provenance, and has
emerged in archival literature from the US, Australia and Canada as the new paradigm that
replaces the physical record.⁵ At the heart of these discussions is the importance of the
record as evidence of action and transaction.

Information engineering does not set out to create or manage records. The
foundation of its information structure is data, not records or transactions. This is a
problem for archivists. Both information and records are aggregates of data that have
purpose and meaning in the world of record creators, but when the development of an
information system moves from the analysis phase to the design phase, the logical unit of
action (which is our record) is lost. The effect of database technology is to fragment the
logical record into a physical shape that aims to share and reuse the data in the most

² David Bearman, "Record Keeping Systems," Archivaria 36 (Autumn 1993), 17. Bearman uses
the term context to mean the circumstance of creation enveloping the transaction, such as time, date,
author etc. This thesis will use the term context to refer to external structure.


⁵ See for example, Frank Upward and Sue McKemmish, "Somewhere Beyond Custody" in Archives and Manuscripts 22 (May 1994), 136-149.
efficient and economical way possible, and information engineering is designed to operate within this particular set of constraints. The information engineering methodology is appropriate for use as a framework for analysis because it represents the most comprehensive of current approaches in terms of the system development process.\(^6\) The fact that information engineering links organizational goals to business functions and processes by integrating strategic planning into its approach makes it very sympathetic to archival goals. These goals are to derive administrative or organizational structure from the facts or circumstances of record creation, so that we can rely on records as representations of truth about transactions and activities. An enterprise also needs to rely on the truth of its transactions for operational continuity and competitive advantage, and therefore recognizes the integrity of its records as critical to achieving these objectives. Archival and information engineering analyses seem to demonstrate similar intent to reveal structure, form, and activity during the analytical process. While the goals of the analysis may be different, the difference is one of focus not of substance.

As indicated in the literature of the profession, archivists have finally reached the consensus that to save data by itself is not enough, that the structure that gives data its meaning and controls access to it must also be retained. The fundamental difference between the way information is stored in electronic form and the way it is represented in traditional paper records is the separation of the structure from content in the electronic form.\(^6\)

---

\(^6\) In a recent research note published by the Gartner Group, the information engineering methodology “Navigator” was ranked in the top five of methodologies used by organizations to control and manage the system development process. Gartner Group Inc., “A Perfect Match: Methodology and Process Management” *Application Development & Management Strategies Research Note*. ADM: T-480-1081 (October 24, 1994), 2.
environment. In an electronic system the structure (type) is independent of the data (instance) it processes. A manual record system, on the other hand, represents both content and structure simultaneously in a single document.

The separation of content from structure in the electronic environment is a response to the requirements of the database for storage and retrieval. The arrangement of the database and how data is captured should be reviewed in order to clarify this idea.

- **Screens** are an interface device used to present, capture or manipulate data (the screen configuration closely resembles a record in a paper form but they are virtual records and are lost after the procedure has been executed by the system). The screen is not a layer of the database but rather the interface through which the user interacts with the database.

- **Programs** are the coded instructions that will locate and perform operations on the data in the database.

- **Subschema** is the view of the database appropriate to a particular system that defines the arrangement of the data to be viewed by the programs.

- **Schema** is the actual or existing business data that is referenced by the subschema and stored in the database.

At the subschema level, the system is structured with “types” of things or categories of data representation which exist independently of time and value. This structure was revealed by the work products of the last chapter, and by the logical unit of action in particular. The work products represent the types of entities and processes required by the system to execute a specific business procedure. The incremental layering of work products builds up the behaviour of the data and process and reveals all its dimensions, except that of “instance”.
When a procedure is executed by the system, as for example when a user receives blank cheque stock and verifies acceptance through interaction with the database (via the screen), an “entity instance” is created. An instance is the specific occurrence of a data value in a database and is the smallest unit of named data that has meaning. An act consists of facts, and once an act has been executed, values are assigned to the facts and each becomes an entity instance. This instance can become a record if, and only if, the value assigned to the act is preserved by the system. The purpose of this explanation is to show that what we consider to be contextual or meaningful to our understanding of the record resides almost entirely in the type (structure) and not in the instance (content) of the type in an electronic environment. In other words, a record exists partially in the rules, relationships and competencies embedded in the design of the programs (structure) of the computer system, and partially in the actual data that represents the execution of the act that has been entered into the database (instance). These structures are embedded with the knowledge of the business in the form of procedure and transaction. In this way, the structure forms part of the record even before the action occurs to create it.

The concepts of type and instance are at the heart of the confusion archivists have about records in electronic form. The distinction between structure (type) and data (instance) must clearly be understood before archivists begin to reinvent appraisal or description. This thesis has explored the process of developing the structure of systems,

---

7 The meaning that resides in the structures contained within a system is what the term “metadata” refers to, and what the concept of “type” represents.

8 Several archivists and educators have acknowledged the importance of the separation of structure from content before, however it is not understood and accepted by the whole of the profession and is not always taken into consideration when discussing appraisal or description of electronic records.
and has shown how the methodology connects administrative relationships, business functions, and transactions, to the logical unit of action (type). This thesis has not discussed or presented an occurrence of the action by itself (instance), and therefore cannot analyze any physical examples of executed actions or data records. In the presentation of type or structure it has examined only the potential for action to occur.

As technology continues to advance, and business systems continue to take advantage of technology, it is important that archivists understand the system development process and become proficient in certain technical areas as part of their continued professional education. The information engineering framework and the system development process represent prerequisites of knowledge for the study of the electronic record form. Such an understanding must precede any discussion of program strategies, appraisal, or description of records in electronic form, since it is necessary to know intimately the beast in question before building the zoo. David Bearman has remarked, “the failure of archivists to understand record systems in their practice with paper records has left us without the analytical tools with which to approach electronic records.”

Whether or not archivists have failed to understand manual systems is debatable, but Bearman is correct in implying that the profession does not have, as yet, a conceptual framework for understanding the electronic counterpart of the record. It is this understanding that is essential to the evolution of archival technical knowledge and of their responsibility as archivists for ensuring the reliability, authenticity, availability, and integrity of all records regardless of form.

---

9 Bearman, “Record Keeping Systems”, 27.
It is, and will continue to be, virtually impossible to acquire the *fonds* of an organization without encountering some degree of automation in the information system environment of the 1990s and beyond. Technical skills, therefore, are necessary in order to review information systems, to explore record issues and to formulate meaningful recommendations about their resolution. Without understanding the system in relation to the processes and activities of the enterprise, it will not be possible to identify what data in the system constitutes reliable evidence of activities, and which activities and competencies created or used the record.¹⁰ Because the foundation of the information structure is data (entities) and not records or transactions, we cannot plan for the preservation of and access to the contents of the database with the current arsenal of practices.

The phenomenon of the database has produced the situation where the information can be displayed in ways different from the way the person who originally created or had access to it did. The problem from an archival perspective is that most systems do not retain specific information about how users "see" data, or information about how it was used in a business context. So, the data cannot be seen as constituting records, although what is done with the data, if preserved in a stable form might. It is a matter of stability. Each record must be stable in order to exist as evidence of action. To know that the information existed somewhere in the organization at the time that some transaction took place is very different from being able to say definitely that this person had access to that view of that information, had permission to see that view, could bring together the data,

¹⁰ ibid.
had the software tools to do it, could have analyzed it in this specific way, and effected these specific actions with it. To say that the data was present somewhere in the electronic environment means nothing from an archival point of view. From that point of view, a record needs to be in a stable form, have identifiable persons create it, and be connected with definite business actions of those persons. Without these characteristics, there can be no guarantee of its meaning and value as evidence.

The emphasis on the enterprise view of system development is particular to information engineering among system development methodologies, though not new to archival science. General archival theory sets out the concepts which guide analysis and explains how organizations and persons generate and organize their records in the conduct of business. Information engineering regards a system as a subset of the whole enterprise. Because strategic planning is included in the development process, systems are required to serve overall enterprise objectives in some way, rather than merely parochial or local ones. Systems developed in this manner will not pose a complete mystery to archivists if they attempt to understand the nature of the bureaucratic action connected to the particular system. In other words, the connection of the system to the facts of its administrative context, and to the acts or actions which brought them into being, can be found in the smallest unit of meaning and can be revealed through the documentation of the information engineering process. Hilary Jenkinson emphasizes the importance of gaining a precise understanding of the process governing the way an office conducts its business and creates its records as a way to evaluate the reliability or truth of an action as evidence.¹¹

Heather MacNeil makes the point that to map a trajectory from administrative structure to bureaucratic procedure to documentation illuminates the symbiotic relationship that exists between the contextual realms of provenance and documentation, and demonstrates the constant mediation between process and product inherent in that relationship.\textsuperscript{12}

Information engineering formalizes what MacNeil refers to as "mediation" into a formal technique called iteration.\textsuperscript{13}

Contextual information identifies and explains the administrative relationships that govern how the business operates, which in turn affect how information is created and used within the enterprise. To prove that things are as they should be, and as they appear to be in "reality", or to say that records are reliable in electronic form, depends in part on the integrity of the processes and data which form the basis of the creation of records in electronic form. To determine that a record exists and that it is complete and genuine and has an archival nature, the area of analysis must expand to include all the activities of the system, including those in the technology environment, the application, and the end users. To investigate the integrity of computer records all procedures must be reviewed and monitored that pertain to: the implementation of the system; the security of the program; the computer operations (assurance that programmed procedures are consistently applied and correct data files are used by the programs); and the system software.\textsuperscript{14} This scope of investigation is expansive but necessary if one is to try and make effective and efficient


\textsuperscript{13} The iterative development technique is described on page 29.

decisions about the existence of the completeness and reliability of records. The perspective of this task is nearly unmanageable, but not an impossibility. It is a very complicated process, but it is possible to trace the origin of most actions and determine what larger transactions are associated with the action. It is possible to check what processes accessed what tables in the database and what terminals entered or processed data. By reviewing the system documentation, it is also possible to know how specific programs acted on specific data. The point here being that it is possible to determine specific facts about an act by examining certain physical pieces of evidence, and by using the documentation from the development process. However, the complexity of ensuring completeness and stability in this ever-changing environment makes this type of investigation wholly impractical for the purposes of archival description and appraisal.

Data is represented by entities and an entity is the foundation of the information structure in a database environment. An information system transforms data into information by means of process which occurs during an event. An elementary process represents a complete set of logical activities made in response to a single activating event. Elementary processes are equivalent to what we perceive as acts, and like acts, can change or alter the relations or state of something. An elementary process either creates, reads, updates or deletes data during the execution of an act. These are the core processes that

---

15 Secondary use of system development documentation for analyzing the physical record is limited because it does not necessarily reflect what was actually built and implemented. Designs are generally modified during construction and implementation and if the encyclopedias are not updated by the project team, final changes are not necessarily documented. Also, it is not always known before the system becomes operational how the database management system will manage certain physical operations on the data. However, the documentary products of the planning and analysis phases remain stable in the view they provide of the external structure of the administration.

16 These actions (excluding delete) are remarkably similar to the elements which comprise the requirements for functional records as set out by the University of Pittsburgh study. The Pittsburgh study
are performed on data and which are at the heart of how the system processes information. This is also evident in the dependency of the models on each other, that is, how data and process are bound together.

The description of each elementary process identifies its relationship to a part of the enterprise strategy (goals and objectives), process, data, rules, and the persons required to execute the transaction. The elementary process is bound to the functions of the business area of which the transaction that it is connected to forms a part. As explained earlier, every instance (data record) maps back to a meta description (type) of the instance. The description of the elementary process also has relationships to other higher level processes and functions. An elementary process is an action that has a discrete beginning and end. A parent process on the other hand, represents an on-going function of the enterprise and resembles what archivists call a “transaction”. Each elementary process under a parent process (or transaction) must occur once in order to execute the larger process. For example in the case under study the elementary processes Re-order Blank Cheques, Record Blank Cheques Received, and Reset Next Cheque Number each must occur to execute the parent process Manage Cheque Inventory.

The role of the elementary process in the information structure of the system can be likened to the activity of building a wall. If a wall is a structure that has meaning and form then the integrity of the structure depends on the quality of its individual uses Capture for Create; Maintain for Update; and Access for Read. Bearman, “Record Keeping Systems”, 29.

17 Bearman describes the same idea when he writes, “the complex relations between data structures are links to the elements of the business environment in which the system operates.” Bearman, 25.
components, such as the quality and techniques used in construction, and how the parts relate to and balance each other. The bricks in this case are the individual objects of process, data, strategy, etc., that comprise the scope of the project (business area). This makes the mortar the set of techniques which bind all of the objects together, and make them a recognizable structure as opposed to a pile of construction materials. The set of techniques is the methodology itself, which in this case is information engineering. When you know that the tools, techniques and products have certain inherent standards, and the process can be repeated similarly every time, then you can make some assumptions about the reliability and integrity of the structure.

At the data and process level, the use of an information engineering methodology for the development of information systems promotes structure and completeness by providing a logical and smooth progression of work from one activity to another. The procedures, tools, and techniques used within the framework of the methodology support the creation, refinement and quality of the products delivered in each phase of system development. The standard of deliverables that comprise the project documentation as dictated by the methodology, allow one to make the assumption that processes are complete and have been properly authorized.

The iterative nature of system development using an information engineering approach addresses the criteria for quality at every stage of the process. Quality is the intent of Martin’s "interlocking formal techniques" and how he and his co-founders envisioned that information engineering could improve the system development process. Data integrity is also an objective of the knowledge base, which functions as a control of
how information is managed. Quality assurance is inherent within the techniques and practices that produce the final work products. That is to say, the opportunity to accomplish the quality assurance aspects of the methodology are embedded in the criteria for the creation of the individual work products and deliverables. The standards and controls which support data integrity ensures that an accurate and complete translation exists between data and process activities. Iteration is the essence of information engineering and is achieved by moving back and forth between the models to ensure that each is synchronized with the other, and that the next step adequately represents and builds on what came before. The work is checked to verify that goals and objectives established for the phase or the project as a whole have been met, and that they still support the larger enterprise mission.

The entire systems development life cycle reinforces the understanding and accuracy of the components of the design. A transformation junction refers to the point in the process when development moves from one domain to the other, that is, passes from the logical, to the technical, or to the physical domain. At these junctions an individual interpretation by the developer or designer can affect the outcome of the project. This is why the completeness of the information models must be assured before development proceeds further. Models are mapped and compared to guarantee accuracy so that new

\[\text{18} \text{ These transformations generally correspond to the different cells depicted in Zachman’s Information Systems Architecture as seen in fig. 3. The first transformation is from strategic goals to project objectives (which translates vision to action plan); the second, from objectives (action plan) to deliverables. Much of this work appears in the charter before work even begins. The next transformation is from the entity relationship diagram (logical) to logical database design, and from logical database design to physical database design (which is shown in Data column of framework). In the Process column, the elementary process to business transaction diagram represents a major transformation. Strategy representation and models affect both the data and process columns equally.}\]
work is not based on mistakes or oversights. An example of an iteration is verification that all processes have data to act on, and that all data will be used by a process. If there is doubt, additional views (representations or models) can be created as a mechanism to further test, up to a given point, the quality of the work. Iteration is a critical practice at these transformation points in the system development process, and is documented by obtaining consensus between the parties that the transformation represents the most appropriate decision or interpretation of the business. Acceptance is formalized by a contractual agreement between all parties involved in the project. The signatures on this record confirm that the plans or products of the phase accurately depict the business function, and that all parties approve the work and are in agreement with the approach. The involvement of the creator throughout the development of the system is central to the methodology, as information engineering relies on the creators' input to ensure that processes accurately depict policies and practices which govern business transactions.

The connection between iteration and integrity is drawn in order to demonstrate that the elementary process is a fundamental piece of the larger system and, as such, represents a microcosm of the larger whole. (The reader may also recall that the elementary process was the "logical unit of action" traced through the presentation of project documentation in the last chapter.) The methodology, therefore, ensures that integrity is built into each work product of the project by the controls inherent to it. This aspect of information engineering effectively binds the elementary process to the data it acts upon, to the organizational unit and business function of which it forms a part, to the strategy or goals that define the reason the action occurs, and to the policies and
procedures represented by the business rules that govern its behaviour. It follows then that compliance to the procedures of the methodology makes a certain level of completeness and accuracy inevitable and that the documentary products are reliable representations of these relations.

The trustworthiness and reliability of these relationships depends on the integrity of the data and process. Its linkage from the microcosmic world of entities and elementary processes outward to functions and corporate goals is tied to organizational information management practices. Information engineering builds upon the degree and extent of the information management practices that are resident within an enterprise. Information management is not an aspect of information engineering per se. It is mentioned simply to explain its relationship to data integrity.

Data integrity is achieved through the sound information management practices of the enterprise. Data administration is all of the activities required to make accurate data available to authorized personnel. Its focus is entirely conceptual. Data analysts identify, catalogue, control, and coordinate the data needs of the organization. The objectives of data administration are to maximize the availability of data and exercise control over it. This function provides planning, control, and direction for the inventory, privacy, recovery, retention and standardization of data, and is often represented in the internal standards and policies of the enterprise.

Database administration is a separate activity that is usually performed by another group of specialists within the information technology department. It relates to the work of data administration but in a more physical sense. The role of the database administrator
is to protect data that is essential to the success of an organization, coordinate all departments that use the database and provide service to users of the database. The database administrator is responsible for the physical database design (subschema), database implementation, database monitoring specification, database availability and database security.

It should be understood that in order to analyze the integrity, accuracy, availability and reliability of information in electronic form, archivists will inevitably become increasingly dependent on other information specialists. The reason is that the activity of securing the information itself is increasingly an electronic process that requires a specialist's expertise. For example, to ensure the security of information by limiting access to only authorized users requires special access software. There are certain built-in control facilities of the database management system that also protect data within the database from unauthorized alteration, disclosure or destruction. These control features reduce the set of users and processes to a limited, identifiable and certified group which is allowed to perform functions against the data secured in the database.

Data administrators have nearly as many words for types of information as Eskimos have for snow, to use a cliché. The language of the methodology is exact because it has a different audience with whom it must communicate for each phase of development. This has the effect of refining the terminology and modes of representation and improving communication between parties. As a profession, archivists have experienced a similar experience of language affecting practice. The variation in terminology and definition is evident in archival methods and in our publications across
countries and continents. The concept of “function” for example, is still fluid in archival science. The problem of vocabulary also exacerbates the international debate about electronic records in archival literature; it also impedes making progress in this area and reaching consensus. Different interpretations, different levels of knowledge and experience, and a general lack of training in the computing disciplines, combined with a lack of consistency of usage have kept solutions, consensus and even a genuine understanding of the problem of electronic records somewhat beyond our reach.

In system development, the partitioning of language and technique into elements of technical design, physical construction and logical analysis is largely a result of necessity. The computer itself is a dumb machine, which is why we must be extremely precise when communicating with it. The electronic environment requires that there is no ambiguity in meaning or instruction, probably because all analysis and design eventually and inevitably becomes code. The fact that the precision of the concepts and language aids communication during and after the process is more a result of the demands of the technology than of design. The boundaries between technical, logical and physical transformation points represent the boundaries between a specific lexicon and a specific set of representations that help developers, users and management define the nuances of the actions being automated.

John Zachman’s name has been synonymous with information architecture since the publication of his article, which set forth a schema to help articulate and understand the spectrum of activities involved in developing a complex information system.¹⁹ The

"Zachman framework" graphically depicts the boundaries between perspectives and audiences, and is one of several proprietary concepts incorporated into information engineering from other sources. Zachman's architecture can be used, among other things, to orient oneself in terms of the purpose of the work products, the sum of knowledge which each contains, and the perspective of the project that the work products embody. As seen in Fig. 3 on page 50, Zachman's architecture offers a mechanism for identifying the interdependencies of data and processes and the perspectives of individual representations. For example, in the planning phase the information captured consists of business subjects and high level relationships of interest to executives. During the analysis phase, the information that is represented in the work products is aimed at the record creators or persons actually doing the work being automated. The entity relationship model, for example, is developed to depict the perceptual requirements and needs of the owner/user of the future application system, and consists of objects such as business rules, transactions, relationships and other information necessary to execute specific business activities. During the design phase, the perspective of the relational model is used to place the subset of the data model in a position to become automated. The logical model of the design is developed which depicts, from a designer's perspective, the system as it is constrained by economic, functional or time limitations. Physical space is allocated and identified within the database during the construction phase, and the audience changes again to the database specialists who need to understand the quantifiable requirements of the new system in time and space.
In summary, the information engineering methodology represents a sound and logical framework for defining, managing and integrating all components of a system development project. It defines a top down, structured approach to strategic planning that confirms that the information needs of a business area are consistent with enterprise goals, and that these associations and linkages between goals and activities form the basis of the final system. Systems developed in the structured, organized manner of information engineering, with requirements for reliable records built into their design, are the systems that have the potential to support accountability and the requirements for corporate memory.

The usefulness of information engineering to archivists as an analytical tool is limited by its focus on data and process. The onus is on the archival profession to adapt such an approach to service its own ends, and shift the perspective toward the transaction as the basic unit of analysis. The transaction is already represented in information engineering as a combination of data and process in the design and construction modules of the system development life cycle. Archivists need to learn to “talk the talk” in order to articulate what is necessary to preserve the evidence of affairs for which they are responsible. Evidence that was accessible and derivable from the custody, order of creation and receipt of paper records is missing from the electronic environment of the relational database. The utility of information engineering for archivists is in the information gathered in the process of replicating the activities of the enterprise for automation (especially during the planning and analysis phases). Even though the structure exists independently of the actions and events that constitute a record, it still
forms part of its meaning. And it is this structure that is linked to the larger context of the
fonds through the planning and organization models.

The analysis of project documentation demonstrates that there cannot be a
complete and reliable record in the archival sense unless such evidential requirements are
necessary to the business and as such, become part of the analysis and design phases. If
the entities involved in a transaction are required for memory, then the requirements that
particular data is retained and not destructively updated by another application will
become part of the physical database design. If the need for memory of actions and
transactions was not a stated business requirement, then evidence of activities in which the
entity participated over time may not be preserved. If a record can be designed as an
entity and becomes a table in the database with its own particular attributes, relationships
and physical specifications that are representative of a complete and reliable act, then
retention of the record may be achieved (that is, if controls are in place to protect the
integrity of the data used in the record from being changed by other programs using the
database). In the example of CPAY, the data record that results from Record Blank
Cheques Received exists as a record because the system was designed to retain all entity
instances created when the business transaction is executed. But the evidential nature of
archives requires that records are stable and are not altered in any way. Even though the
record is preserved by CPAY, it cannot be known without extensive investigation whether
it is stable and protected from manipulation by other programs. This does not mean that it
is impossible to design an architecture that has the ability to separate the evidence of
action from the myriad of other data associated with the application, only that in today’s
business culture there must be a business need for such an architecture to exist. It is up to archivists, auditors and lawyers to appraise the record for its retention value prior to creation, and to come forth with requirements that address evidential, legal and archival concerns, so that they are incorporated into analysis and translated into design.

Until such time that archivists are in a position to influence how systems are developed, information engineering and its documentation can be useful as a mechanism for understanding the purpose and the relationships of the process and data to the operations of the enterprise. Archivists can learn its terminology and vocabulary to communicate more precisely about the information they want to describe. It is possible to rely on the fact that the system developed in concert with an information engineering approach has been designed according to comprehensive methods that ensure the integrity of the relations, procedures and rules which represent the business. Archivists can learn the purpose of the representations and understand the sum of knowledge embodied by these forms as a vehicle for research and analysis. They might adapt some of these representations for reuse during appraisal to describe the relationship of business functions to activities and records. Archivists can use the framework of information engineering as an ideal with which to frame existing systems to determine what metadata is missing and what is required to understand the context of their creators. Most importantly, archivists can use the knowledge of the system development process to understand what is and what is not possible when they are invited to sit down at the development table and speak about requirements.
CONCLUSION

Information engineering is a discipline and a methodology used by information systems personnel to manage and control the complex process of developing information systems. Information engineering formalizes the process and adds rigour to the activities employed to plan, analyze, design and build systems. The integration of strategic planning into the methodology links the mission, structure and function of the enterprise to its activities and transactions, thereby making these relationships discernible in the smallest unit of meaning engineered during development. The union between strategy and activity makes information engineering suitable for use by archivists as an analytical framework to reveal the nature of systems, and the information or records which they capture and keep.

The application of the methodology to an actual project in this thesis, revealed the relationship of what is the structure of the system to the individual action (instance), and in this way illustrated the usefulness of the methodology in the examination of provenance (external and internal structure). The effect of the system development process on the action was demonstrated as the unit of action moved from the logical domain of planning and analysis to the physical realm of design and construction. Records can and do exist in this environment, however their nature is different from our traditional notion of records. Information engineering does not set out to create or manage records. If information is not managed as records, it cannot be appraised or described according to traditional
methods. Records can exist in electronic form; they are part structure (type) and part data (data instance of type). Each time the action occurs and data is entered into the system part of the structure is recreated, which effectively imprints elements of the structure on the final record (if it is preserved).

The methodology plays a role in the genesis of records in electronic form and has value as a research device to assist archival analysis. The quality criteria inherent in the method and its documentary products also has relevance to the archival characteristics of stability and completeness. The reliability of the system with respect to data integrity that results from inherent quality standards allows certain assumptions to be made.

The information engineering approach to systems development is stable, because its underpinnings are based on the strategies, objectives and functions of the business. Within the composition of an information engineering methodology there is a clear and distinct separation between the identification of actual business requirements, and the design decisions which deal with a specific system implementation which addresses these requirements. In other words, system development can be separated into domains of problem and solution. New developments in the computing industry tend to be associated with design and technology issues (solution domain) which are completely separate from matters of business strategy (problem domain). While the answers may differ, the questions remain the same. The process of identifying the business structure and functions which define the activities for automation will be constant queries. Technology solutions, however, will frequently change and reinvent themselves. There is no technology proxy for a formal, controlled and managed approach to planning and analysis, and as such, the
framework has significance to recordkeeping. Information engineering can be useful as a research framework for archivists even if it was not originally used in the development process. It represents an ideal of what can be known about a particular system because it links high level business goals to activities and actions. In this way, information engineering is sympathetic to archival requirements for context.

As previously stated, the study of information engineering is the study of the methods, techniques, concepts and principles used in the activity of developing information systems. A knowledge of the information engineering framework can assist the profession articulate archival considerations during system development, understand primary source metadata, or establish electronic records management programs. Further investigation of this form of record could be undertaken by any of the following studies:

- interdisciplinary study of analytical techniques for the purpose of developing procedures for gathering, classifying, analyzing, and interpreting data and information, to show relationships, dependencies and flows in the communication of data as transactions and records in electronic form

- study of information management and computer science lexicons for possible inclusion in archival glossary or in analytical techniques to describe, represent and express requirements and contextual information about records in electronic form

- use of information engineering framework to evaluate new developments in information science that represent business structure and function, such as object oriented analysis

- use of the information engineering framework during the analysis phase of a system development project to develop recommendations for preserving transactions with ongoing value

- development of an archival policy framework to address data security, computer operation and internal standards issues that affect the integrity of the physical record beyond the scope of methodology and internal audit considerations
• evaluate the use of system documentation in combination with other forms of metadata such as, knowledge base, encyclopedia, repository and dictionaries used within the enterprise in supporting electronic records management and archival requirements

The "paperless office" has so far turned out to be a bad joke. In spite of the fact that the electronic record is less secure and less permanent than paper at the present time, paper will eventually be supplanted in its role as the primary medium for recording knowledge and memory. For archivists to be able to participate in this societal shift, they need to accept responsibility for continuing their interdisciplinary education as an aspect of their profession as information specialists. The examination of information engineering in this thesis has been a window onto the range of skills and knowledge required by archivists to be versed and cognizant in the inevitable and emerging paperless state.


___________. "What is Archival Theory and Why is it Important?" *Archivaria* 37 (Spring 1994): 122-130.


