1. This document has been approved for public release and sale; its distribution is unlimited.

New York University
School of Engineering and Science
Department of Industrial Engineering
and Operations Research

160

AFOSR-68-0095-Vol.1. May, 1968

A MATHEMATICAL MODEL OF THE FLOW OF DATA IN A MANAGEMENT INFORMATION SYSTEM

Volume 1

Eugene D. Homer

Research sponsored by Air Force Office
of Scientific Research, Office of
Aerorpace Research, United States Air Force

Grant No. AF-AFOSR-621-66
Project No. 9749-01

Department of Industrial Engineering and Operations Research
School of Engineering and Science
NEW YORK UNIVERSITY
University Heights
Bronx, N.Y. 10453

A MATHEMATICAL MODEL OF
THE FLOW OF DATA IN A MANAGEMENT INFORMATION SYSTEM

A THESIS
by
Eugene D. Homer
B.I.E., M.I.E.

Submitted to the Graduate Division of the School of Engineering and Science in partial fulfillment of the requirements for the degree of Doctor of Philosophy at New York University.

New York
May 1968

TABLE OF CONTENTS

TABLE OF CONTENTS (cont'd)

TABLE OF CONTENTS (cont'd)

ACKNOWLEDGMENTS

The author wishes to express his appreciation to those persons and agencies which made possible the completion of this research:

The Applied Mathematics Division of the Air Force Office of Scientific Research which supported most of the research under Grants AF-AFOSR 621-64, AF-AFOSR 621-65, and AF-AFOSR 621-66.

Professor Norman N. Barish, Chairman of the Department of Industrial Engineering and Operations Research, whose encouragement initiated this project, and whose kindness, thoughtfulness, and sound advice were always in evidence.

To Professors Leon H. Herbach and Saul Blumenthal, whose comments were invaluable.

To the various students in the author's graduate class in Systems Analysis who suffered through the earlier versions of the models, and whose comments and criticisms helped to clarify the exposition.

To Mrs. Helen Nadler for her painstaking work in typing drafts and final copy.

And to the author's wife and family who bore the brunt of his involvement in this work, who accompanied him on his trips through gloom and through joy with sympathy and understanding.

ABSTRACT

Previous work in the formulation of a matrix model for the flow of data in a management information system is extended to include three additional types of systems phenomena. These are:

(a) data filtering, both conditional and unconditional,

(b) internally generated items of data, and

(c) the semantic relation of denotation among items of data.

The concept of macro-model analysis vs. micro-model analysis is introduced. Methods are given for solving the models for either the number of paths connecting inputs to outputs or for complete specification of all such paths.

CHAPTER I

INTRODUCTION

1. Models of Management Information Systems

A management information system is a complex structure of people, equipment, facilities, supplies, data, documents, flow patterns, organizational relatiorships and decision activities, operating within a set of demands and constraints established by the nature of the situation, management policy, laws and regulations, contractual agreements, cost factors, and the pressures of the ecology. In the analysis of such a system, existing or proposed, the intricacy of the interrelationships among the many components makes the use of some modelling technique almost mandatory.

Traditionally, systems analysts have favored the use of two types of models to summarize and depict their knowledge of a management information system. The first is a simple tabular arrangement of information; the second a graphic model, or "flow chart". (Both methods are frequently referred to by the comon label of "charts".) The use of charting techniques is widespread, and is described in many books on systems analysis (for example, [1; 4; 17; 18; 25; 32; 33; 35; 40]), in many pamphlets (for example, [7; 10; 11; 28; 30; 34]), and in many papers and reports (for example, [12; 13; 21; 22; 29; 31; 41; 43]). The technique of charting has as its purpose the consolidation of information about a system, and the display of that information in a form such that correlations and sequential flow can be readily observed. However, tabulations and flow charts have one characteristic in common with the verbal descriptions that they replace; they are qualitative models. They cannot be subjected to mathematical manipulation to yield quantitative descriptors or measures.

A quantitative management information systems model was proposed by Lieberman [26] and expanded upon somewhat by Kozmetsky and Kircher [24, Appendix 4]. A weakness of that model was pointed out and a generalized form was proposed by Homer [23]. In these models, systems are described in the form of matrices, and are subject to various matrix manipulations. The matrices are concise descriptors of the flow of information in a system. Used as qualitative models, they offer an advantage over the often voluminous flow charts that depict the flow of documents in a system. Used as quantitative models, they offer an objective way of evaluating systems and the effects of changes upon systems.

The purpose of this thesis is to evaluate prior work on matrix models of management information systems, and to present the results of research to incorporate further systems phenomena into the models.

2. Background

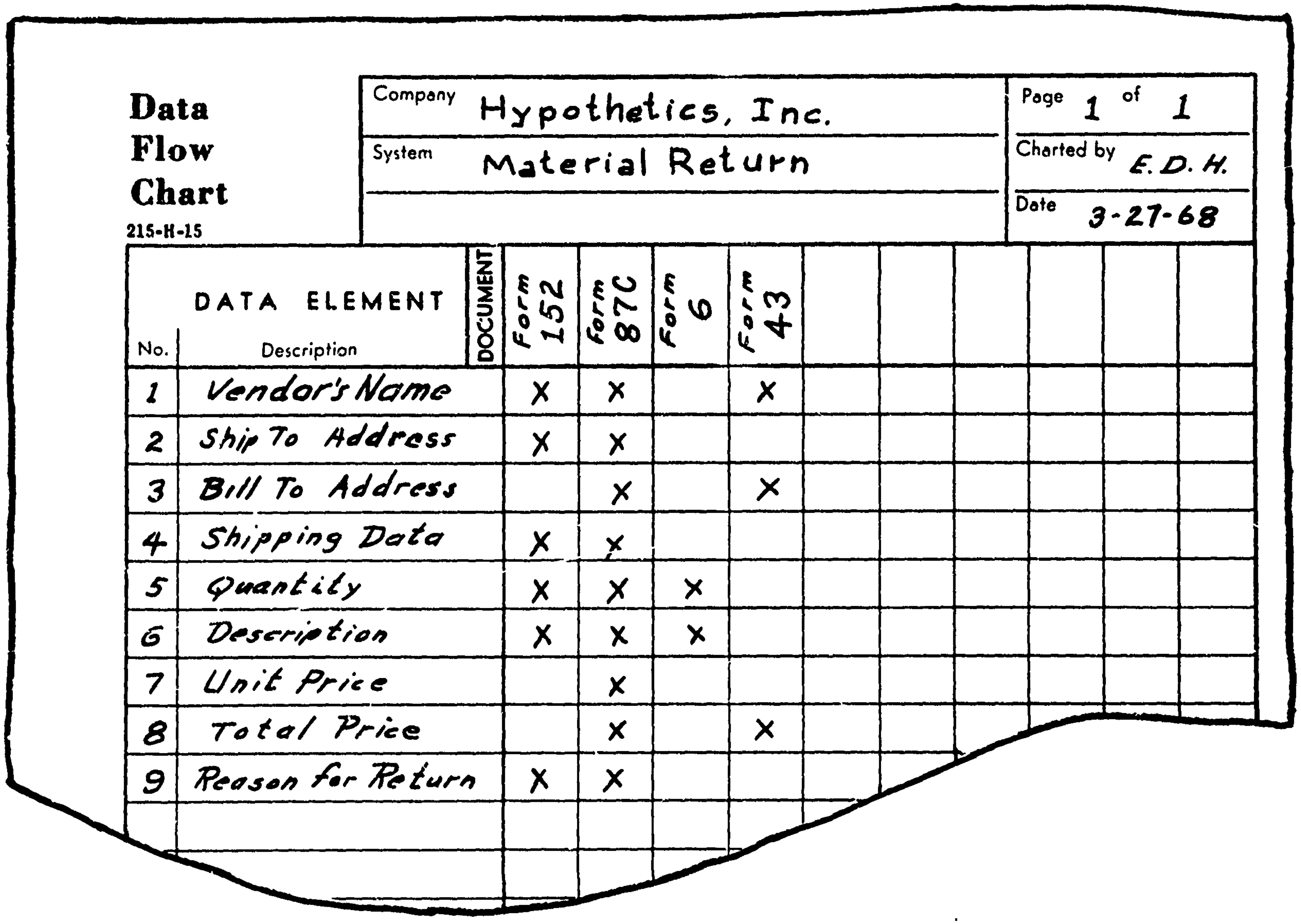
That quantitative management information systems models should have taken the form of matrices is logical. In fact, the concept could have developed from either tabular charts or flow charts, for both of these devices are mathematically related to matrices. (This is not to say that the development of the systems matrix model did actually proceed in such a logical fashion: there is no such indication in the literature.)

A table of numeric data bears a physical resemblence to a matrix, in that both are rectangular arrays of numbers. Even tables of qualitative data may have their entries encoded to yield numeric arrays. Hadley [14, p.60] states that a matrix "is simply a convenient way of representing arrays (tables) of numbers". Thus, we may view a tabular arrangement of facts pertaining to an information system as a precursor of the systems matrix model.

Several such tables have appeared in the literature. Neuschel mentions them [32, pp.196-197; 33, pp. 196-198], as does Barish [1, pp. 160-161]. A similar chart was independently developed by Homer [21; 22, pp. 61-66].

A typical tabular chart is shown in Figure 1. Here, each row of the chart represents an item of data (or an aggregation of items of data treated as a single item). Each column represents a form or document. The cell entries are either "blank" or "X". An X in the cell in the i-th row and the j-th column means that the item of data named in the label of row i appears on the document named by the label of column j; a blank in the cell indicates the contrary. It remains only to encode the cell entries in some numeric scheme to have a rectangular array of numbers. For example, the number 0 might be substituted for a blank and the number 1 for an X. The result is a matrix if the numerical elements of the array are members of a ring [37, p. 12]. Figure 2 is the matrix derived from Figure 1 by the above substitution.

Figure 1



Data Flow Chart

215-H-15

Company	Hypothetics, Inc.	Page 1 of 1	
System	Material Return	Charted by	E.D.H.
		Date	3-27-68

DATA ELEMENT No.	Description	DOCUMENT Form 152	Form 87C	Form 6	Form 43						
1	Vendor's Name	X	X		X						
2	Ship To Address	X	X								
3	Bill To Address		X		X						
4	Shipping Data	X	X								
5	Quantity	X	X	X							
6	Description	X	X	X							
7	Unit Price		X								
8	Total Price		X		X						
9	Reason for Return	X	X								

Figure 2

	152	87C	6	43
1	1	1	0	1
2	1	1	0	0
3	0	1	0	1
4	1	1	0	0
5	1	1	1	0
6	1	1	1	0
7	0	1	0	0
8	0	1	0	0
9	1	1	0	0

Graphic flow charts of information systems exist in a multitude of formats. A very simple one is shown in Figure 3. These flow charts can be visualized as directed graphs [15; 16] by considering each symbol representing a processing step or a document to be a node, and each flow line to be an arc oriented in the direction of the flow. Figure 4 is such a representation of the flow chart of Figure 3.

It is known that directed graphs and matrices are isomorphic [15]. Particularly useful in systems analysis is the adjacency matrix [16, p.15], A, which is defined as a square matrix with one row and one column corresponding to each node of the directed graph, and with $a_{ij} = 1$ if a directed arc exists from node i to node j, and $a_{ij} = 0$ otherwise. Figure 5 shows the adjacency matrix constructed from the network of Figure 4.

An early application of this concept to systems analysis was made by Hare [17, pp. 21-22]. However, Hare used the matrices he derived from networks only to describe the system; that is, as tables of systems characteristics, without any attempt at quantitative manipulations. Even earlier, Warshall [41] had depicted computer flow charts as networks from which he derived matrices. Warshall's interest, however, was confined to using the matrices for decomposition and simplification of computer programs.

Whether or not a matrix derived from a chart is useful for any purpose other than as an alternate way of storing and displaying information is a question that requires an analysis of the meanings of the

Figure 3

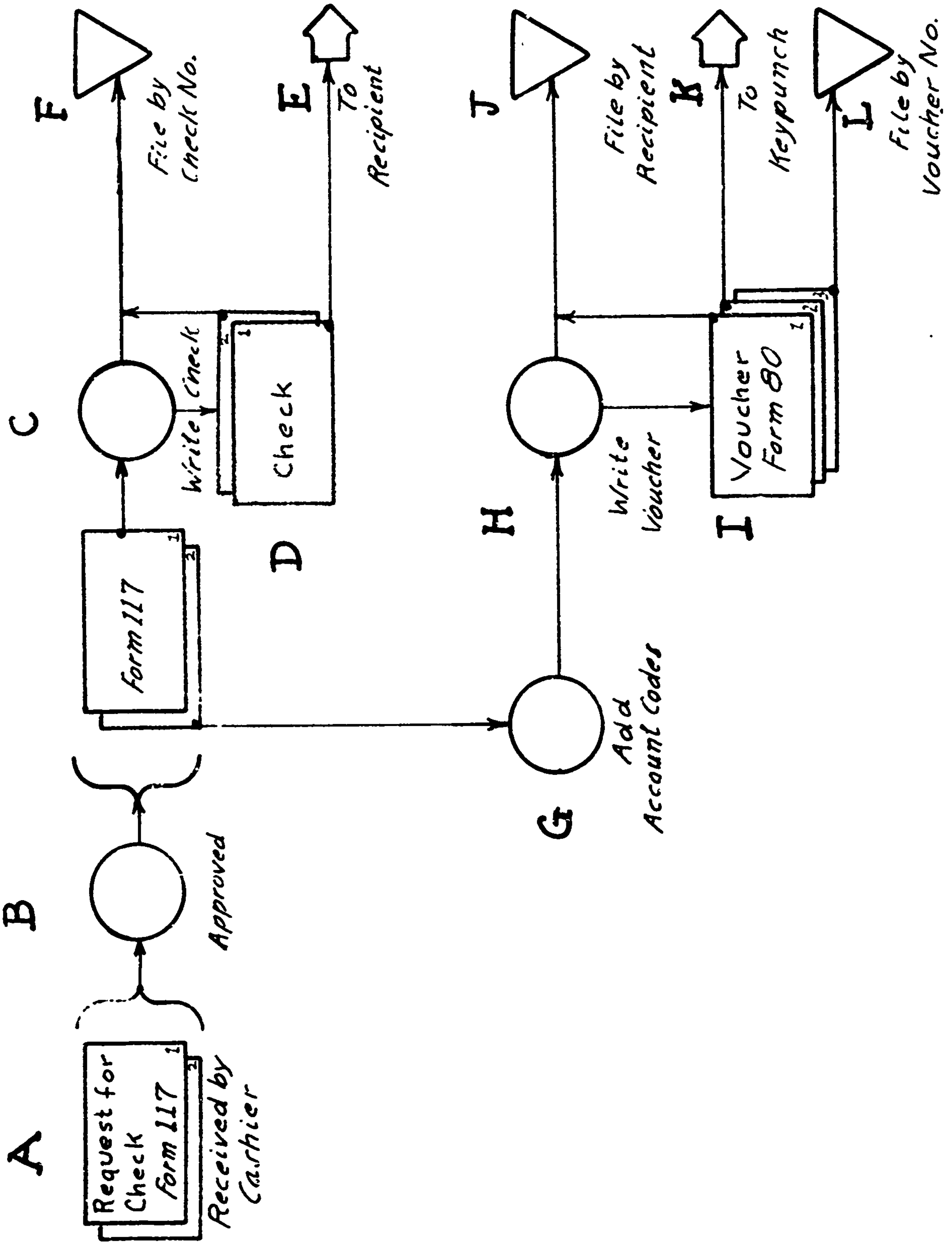

A
Request for Check Form 117
1
2
Received by Cashier
B
Approved
Form 117
1
2
C
Write Check
D
Check
1
2
E
To Recipient
F
File by Check No.
G
Add Account Codes
H
Write Voucher
I
Voucher Form 80
1
2
3
J
File by Recipient
K
To Keypunch
L
File by Voucher No.

Figure 4

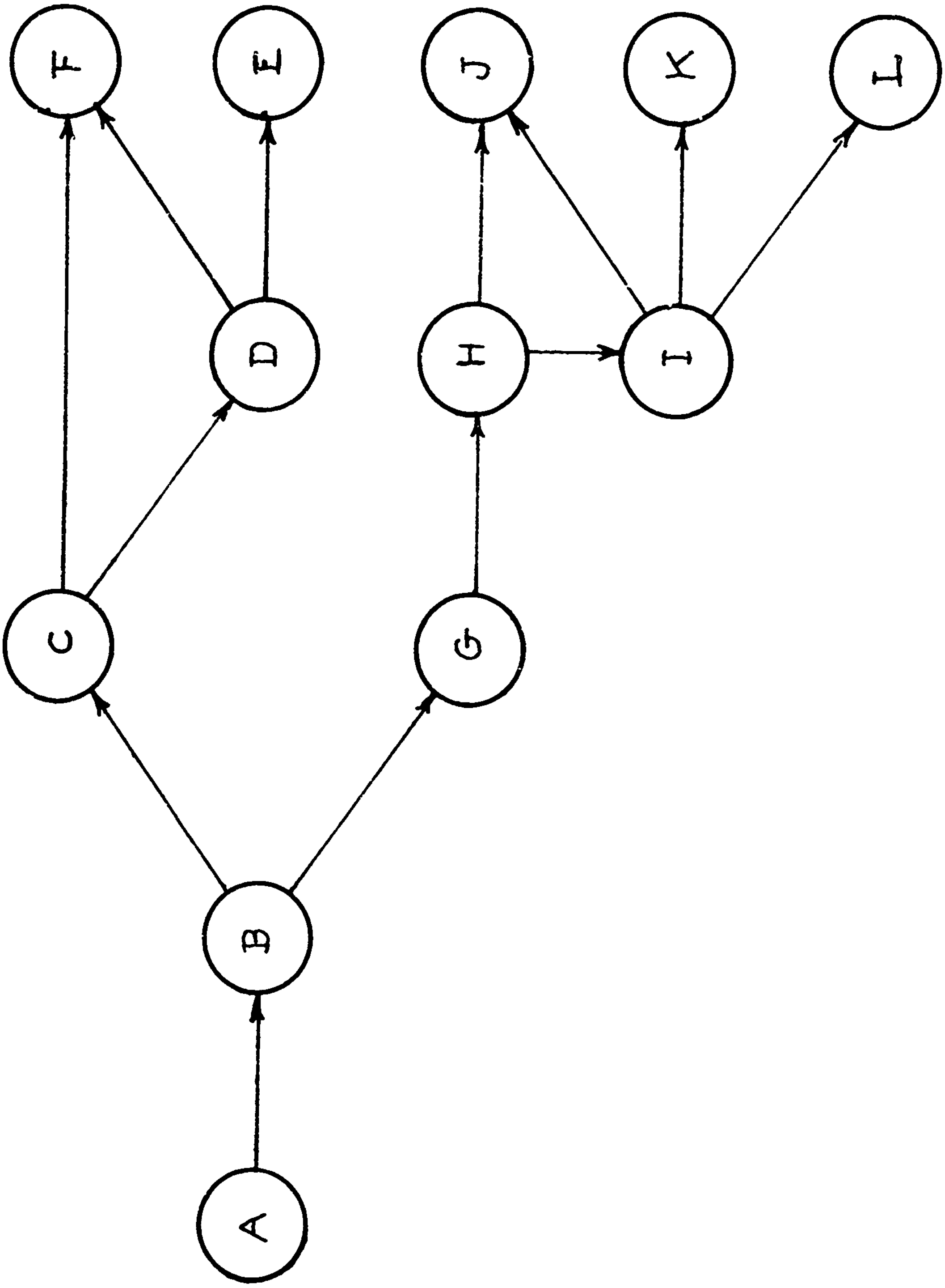


Figure 5

	A	B	C	D	E	F	G	H	I	J	K	L
A	0	1	0	0	0	0	0	0	0	0	0	0
B	0	0	1	0	0	0	1	0	0	0	0	0
C	0	0	0	1	0	1	0	0	0	0	0	0
D	0	0	0	0	1	1	0	0	0	0	0	0
E	0	0	0	0	0	0	0	0	0	0	0	0
F	0	0	0	0	0	0	0	0	0	0	0	0
G	0	0	0	0	0	0	0	1	0	0	0	0
H	0	0	0	0	0	0	0	0	1	1	0	0
I	0	0	0	0	0	0	0	0	0	1	1	1
J	0	0	0	0	0	0	0	0	0	0	0	0
K	0	0	0	0	0	0	0	0	0	0	0	0
L	0	0	0	0	0	0	0	0	0	0	0	0

numerical values in the cells and the nature of the connectivity between the entities represented by the rows and columns of the matrix. Davis states:

> "But matrices are more than arrays of numbers. The individual numbers stand in very special relationship to one another, and their totality constitutes a mathematical object that will be manipulated according to certain rules and regulations, interpreted in a variety of ways, and applied in still others." [5, p.2]

The first published model found which meets Davis' criterion for a matrix is that of Lieberman [26]. This is the model that forms the foundation for the work reported in this thesis.

It would appear that the systems matrix model has its roots in the two most widely used techniques of traditional systems analysis; tabular charts and flow charts. However, since there is nothing in the literature indicating just how the matrix model was developed, it can only be concluded that these roots are logical, but not necessarily historical.

3. Organization of the Thesis

Chapters 2 and 3 are devoted to a review of prior work on matrix models of information systems: Chapter 2 treats of Lieberman's work and Chapter 3 of Homer's. Chapter 4 is a new treatment of the model discussed in Chapter 3; a treatment designed to form the basis of further work. It introduces a more formal description of the model in terms of a relation between components called the "direct connectivity" relation; it presents a network description associated with the matrix model; it introduces a new proof of the validity of the algorithm for solving the model; and it presents a method of specifying each path between components, rather than merely counting them.

Additional systems phenomena, previously untreated, are considered in Chapters 5, 6 and 7. Chapter 5 takes up the case of incomplete transfer of data between components, called "filtering", in both the unconditional form and the conditional form. This is done by means of a model which depicts connectivity between components with respect to each item of data. Data which is generated within the system under investigation is studied in Chapter 6, especially the case of data generated from and replacing data received from outside the system. In Chapter 7, the concept of "knowledge redundancy" due to semantic relations among items of data is introduced into the model.

The thesis concludes with a summary, conclusions, and suggestions for further research in Chapter 8.

Throughout this thesis, statements which are basic to further development are preceded by a letter followed by a sequential number. Letters have been assigned as follows:

Letter	Statements refer to:
A	algorithm for solving the systems-matrix
C	connectivity relation
D	denotation relation
M	systems-matrix
P	paths in the system
S	system characteristics
T	transmittance of connectivity

CHAPTER 2

THE LIEBERMAN MODEL

1. The Basic Model

We present first a brief description of the concept of using matrix methods to model management information systems. This section is based primarily upon the work of Lieberman [26]. The notation employed is that of this author. In this first section we shall use as undefined terms the words "item of data", "report", and "business function".

The Lieberman model takes the form of a series of matrices. The first matrix, M_0, shows which items of data appear on each source report, where a source report is the first form on which data is entered within the systems under investigation, and is considered a first-level report. The second matrix, M_1, shows which source reports are used to prepare second-level reports. Each subsequent matrix, M_j, except for the last one, shows which j-th level reports are used to prepare (j+1)-th level reports. The final matrix, M_n, shows which n-th level reports are used in support of each of the various business functions within the scope of the system.

More formally, let:

d_i = the i-th element of data in the system; $i=1,2,\ldots,p_0$.

$R_{k(j)}$ = the k-th report at the j-th level; $j=2,3,\ldots,n$; $k=1,2,\ldots,p_j$.

$R_{k(1)}$ = the k-th source report; $k=1,2,\ldots,p_1$.

B_q = the q-th business function in the system; $q=1,2,\ldots,p_{(n+1)}$.

M_0 = a matrix containing a row associated with each d_i and a column associated with each $R_{k(1)}$. Then M_0

will be a $p_0 \times p_1$ matrix. The element $(m_0)_{r,s}$ will be 1 if d_r appears on $R_{s(1)}$; 0 otherwise.

M_j = a matrix containing a row associated with each $R_{k(j)}$ and a column associated with each $R_{k(j+1)}$; $j=1,2,\ldots,(n-1)$. Then M_j will be a $p_j \times p_{(j+1)}$ matrix. The element $(m_j)_{r,s}$ will be 1 if $R_{r(j)}$ is used to prepare $R_{s(j+1)}$; 0 otherwise.

M_n = a matrix containing a row associated with each $R_{k(n)}$ and a column associated with each B_q. Then M_n will be a $p_n \times p_{(n+1)}$ matrix. The element $(m_n)_{r,s}$ will be 1 if $R_{r(n)}$ is used in the performance of B_s; 0 otherwise.

The matrices M_j describe the flow of information through a system from the first appearance of an item of data to the performance of the various business functions. The set of matrices is a descriptive model of information flow which is considerably more concise than a flow chart depicting the same information. Figure 6 illustrates the Lieberman model for a simple case, $n = 3$.

Corresponding to the example of Figure 6 is the directed graph shown in Figure 7. Obviously, it is significantly more difficult to obtain the flow information from the network than it is from the matrices. This alone would lead one to conclude that the Lieberman model is a more useful descriptive model than the corresponding flow chart. But, the power of the Lieberman model is not limited to its usefulness as a descriptive tool. This is a quantitative model, and can be subjected

Figure 6

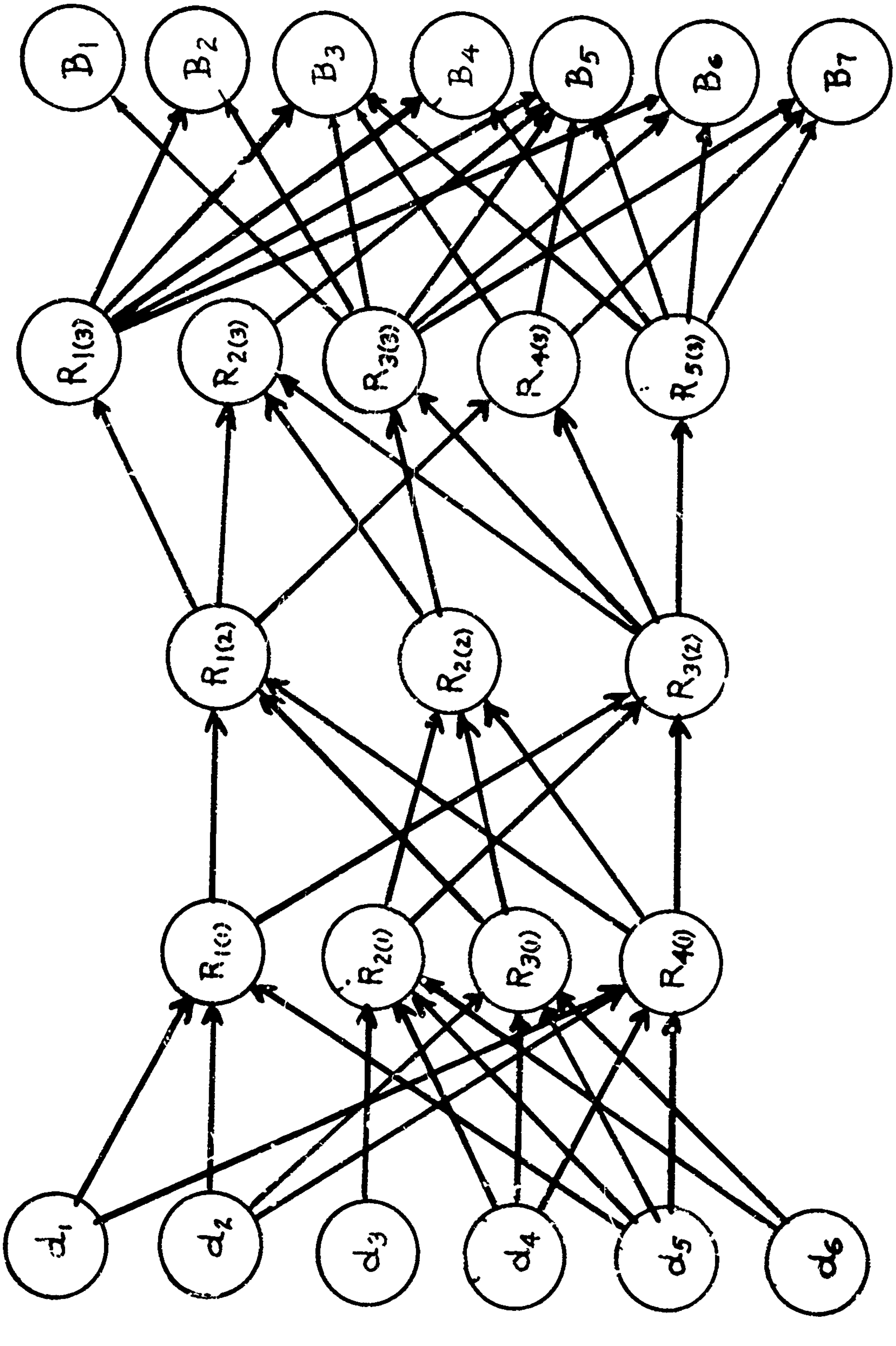
$$M_0 = \begin{array}{c|cccc} & R_{1(1)} & R_{2(1)} & R_{3(1)} & R_{4(1)} \\ \hline d_1 & 1 & 0 & 0 & 1 \\ d_2 & 1 & 0 & 1 & 1 \\ d_3 & 0 & 1 & 0 & 0 \\ d_4 & 0 & 1 & 1 & 1 \\ d_5 & 1 & 1 & 1 & 1 \\ d_6 & 0 & 1 & 1 & 0 \end{array}$$

$$M_1 = \begin{array}{c|ccc} & R_{1(2)} & R_{2(2)} & R_{3(2)} \\ \hline R_{1(1)} & 1 & 0 & 1 \\ R_{2(1)} & 0 & 1 & 1 \\ R_{3(1)} & 1 & 1 & 0 \\ R_{4(1)} & 1 & 1 & 1 \end{array}$$

$$M_2 = \begin{array}{c|ccccc} & R_{1(3)} & R_{2(3)} & R_{3(3)} & R_{4(3)} & R_{5(3)} \\ \hline R_{1(2)} & 1 & 1 & 0 & 1 & 0 \\ R_{2(2)} & 0 & 1 & 1 & 0 & 0 \\ R_{3(2)} & 0 & 1 & 1 & 1 & 1 \end{array}$$

$$M_3 = \begin{array}{c|ccccccc} & B_1 & B_2 & B_3 & B_4 & B_5 & B_6 & B_7 \\ \hline R_{1(3)} & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ R_{2(3)} & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ R_{3(3)} & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ R_{4(3)} & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ R_{5(3)} & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{array}$$

Figure 7



B_1
B_2
B_3
B_4
B_5
B_6
B_7
$R_{1(3)}$
$R_{2(3)}$
$R_{3(3)}$
$R_{4(3)}$
$R_{5(3)}$
$R_{1(2)}$
$R_{2(2)}$
$R_{3(2)}$
$R_{1(1)}$
$R_{2(1)}$
$R_{3(1)}$
$R_{4(1)}$
d_1
d_2
d_3
d_4
d_5
d_6

to matrix manipulation. For example, let:

$$\mathfrak{M}_{a,b} = \prod_{j=a}^{b} M_j, \qquad 0 \leq a < b \leq n .$$

Such a product can always be found, since, by definition, the number of columns of M_j is equal to the number of rows of $M_{(j+1)}$, and the matrices are therefore compatible for multiplication.

The row labels of $\mathfrak{M}_{a,b}$ are those of M_a and the column labels of $\mathfrak{M}_{a,b}$ are those of M_b. Each element of $\mathfrak{M}_{a,b}$, $(\mathfrak{m}_{a,b})_{i,j}$, can be interpreted as the number of "paths", or routes, through which the entity represented by row i reaches the entity represented by column j. In particular, the elements of $\mathfrak{M}_{0,n}$ represent the number of times each d_i is made available to each B_q.

In the illustrative example of Figure 6, the product

$$\mathfrak{M}_{0,3} = \prod_{j=0}^{3} M_j$$

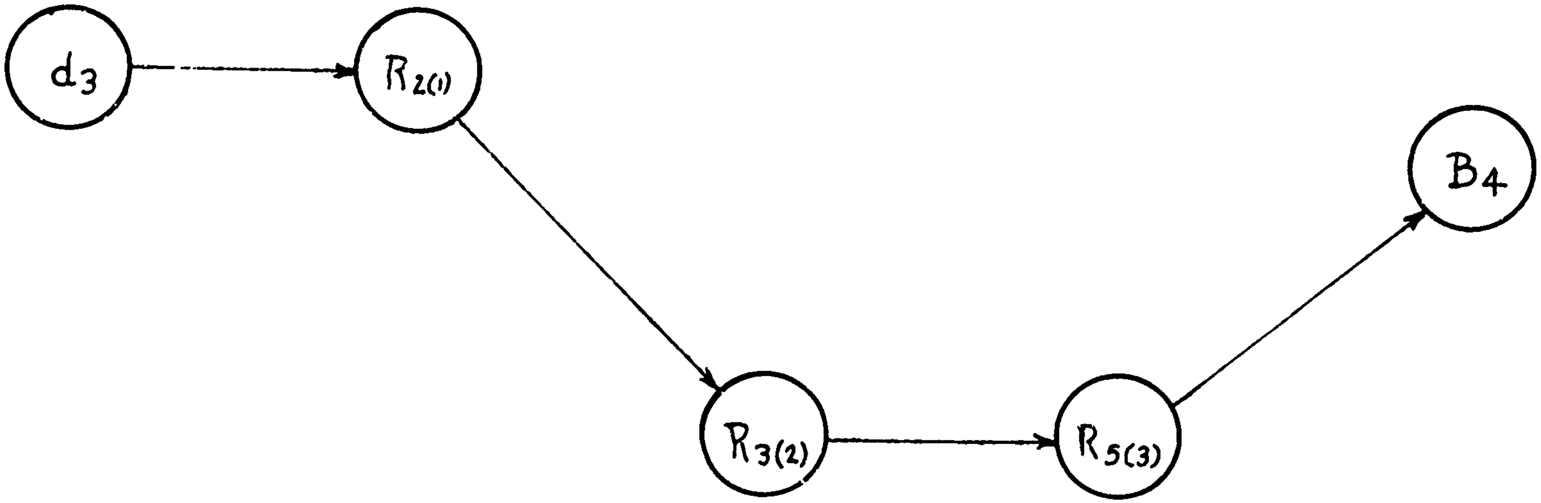
is shown in Figure 8. From this product we can see, for example, that there is only one way in which d_3 reaches B_4, but there are 27 ways in which d_5 reaches B_5. These paths can be determined from an analysis of Figure 7, but only with considerable difficulty. For comparison, the subgraph and path listing for the paths from d_3 to B_4 and from d_5 to B_5 are shown in Figures 9 and 10, respectively. It is obvious that the amount of effort that would be required to obtain the total information contained in $\mathfrak{M}_{0,3}$ by an analysis of a flow chart or network would be considerable, since this almost trivially small illustrative example contains a total of 348 paths.

Figure 8

$$m_{0,3} = \begin{array}{c||c|c|c|c|c|c|c||}
 & B_1 & B_2 & B_3 & B_4 & B_5 & B_6 & B_7 \\
\hline\hline
d_1 & 3 & 5 & 11 & 4 & 16 & 7 & 9 \\
\hline
d_2 & 4 & 7 & 14 & 5 & 21 & 9 & 11 \\
\hline
d_3 & 2 & 2 & 4 & 1 & 6 & 3 & 4 \\
\hline
d_4 & 5 & 7 & 13 & 4 & 20 & 9 & 11 \\
\hline
d_5 & 6 & 9 & 18 & 6 & 27 & 12 & 15 \\
\hline
d_6 & 3 & 4 & 7 & 2 & 11 & 5 & 6 \\
\hline\hline
\end{array}$$

Figure 9

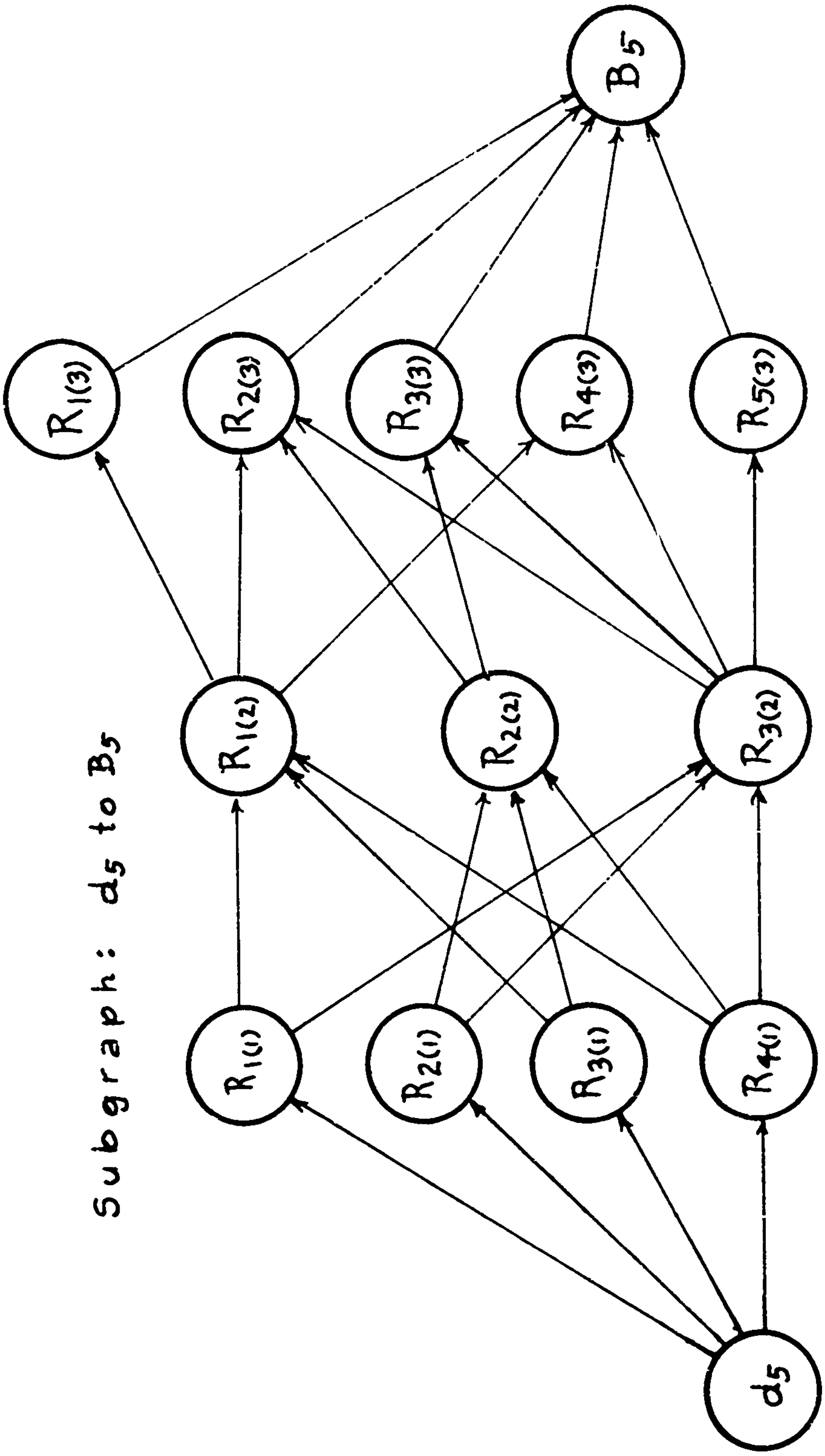
Subgraph: d_3 to B_4



Path Listing :

1. $d_3 \rightarrow R_{2(1)} \rightarrow R_{3(2)} \rightarrow R_{5(3)} \rightarrow B_4$

Figure 10



Subgraph: d_5 to B_5

cont'd

Figure 10 *(cont'd)*

Path Listing:

1. $d_5 \rightarrow R_{1(1)} \rightarrow R_{1(2)} \rightarrow R_{1(3)} \rightarrow B_5$
2. $d_5 \rightarrow R_{1(1)} \rightarrow R_{1(2)} \rightarrow R_{2(3)} \rightarrow B_5$
3. $d_5 \rightarrow R_{1(1)} \rightarrow R_{1(2)} \rightarrow R_{4(3)} \rightarrow B_5$
4. $d_5 \rightarrow R_{1(1)} \rightarrow R_{3(2)} \rightarrow R_{2(3)} \rightarrow B_5$
5. $d_5 \rightarrow R_{1(1)} \rightarrow R_{3(2)} \rightarrow R_{3(3)} \rightarrow B_5$
6. $d_5 \rightarrow R_{1(1)} \rightarrow R_{3(2)} \rightarrow R_{4(3)} \rightarrow B_5$
7. $d_5 \rightarrow R_{1(1)} \rightarrow R_{3(2)} \rightarrow R_{5(3)} \rightarrow B_5$
8. $d_5 \rightarrow R_{2(1)} \rightarrow R_{2(2)} \rightarrow R_{2(3)} \rightarrow B_5$
9. $d_5 \rightarrow R_{2(1)} \rightarrow R_{2(2)} \rightarrow R_{3(3)} \rightarrow B_5$
10. $d_5 \rightarrow R_{2(1)} \rightarrow R_{3(2)} \rightarrow R_{2(3)} \rightarrow B_5$
11. $d_5 \rightarrow R_{2(1)} \rightarrow R_{3(2)} \rightarrow R_{3(3)} \rightarrow B_5$
12. $d_5 \rightarrow R_{2(1)} \rightarrow R_{3(2)} \rightarrow R_{4(3)} \rightarrow B_5$
13. $d_5 \rightarrow R_{2(1)} \rightarrow R_{3(2)} \rightarrow R_{5(3)} \rightarrow B_5$
14. $d_5 \rightarrow R_{3(1)} \rightarrow R_{1(2)} \rightarrow R_{1(3)} \rightarrow B_5$
15. $d_5 \rightarrow R_{3(1)} \rightarrow R_{1(2)} \rightarrow R_{2(3)} \rightarrow B_5$
16. $d_5 \rightarrow R_{3(1)} \rightarrow R_{1(2)} \rightarrow R_{4(3)} \rightarrow B_5$
17. $d_5 \rightarrow R_{3(1)} \rightarrow R_{2(2)} \rightarrow R_{2(3)} \rightarrow B_5$
18. $d_5 \rightarrow R_{3(1)} \rightarrow R_{2(2)} \rightarrow R_{3(3)} \rightarrow B_5$
19. $d_5 \rightarrow R_{4(1)} \rightarrow R_{1(2)} \rightarrow R_{1(3)} \rightarrow B_5$
20. $d_5 \rightarrow R_{4(1)} \rightarrow R_{1(2)} \rightarrow R_{2(3)} \rightarrow B_5$
21. $d_5 \rightarrow R_{4(1)} \rightarrow R_{1(2)} \rightarrow R_{4(3)} \rightarrow B_5$
22. $d_5 \rightarrow R_{4(1)} \rightarrow R_{2(2)} \rightarrow R_{2(3)} \rightarrow B_5$
23. $d_5 \rightarrow R_{4(1)} \rightarrow R_{2(2)} \rightarrow R_{3(3)} \rightarrow B_5$
24. $d_5 \rightarrow R_{4(1)} \rightarrow R_{3(2)} \rightarrow R_{2(3)} \rightarrow B_5$
25. $d_5 \rightarrow R_{4(1)} \rightarrow R_{3(2)} \rightarrow R_{3(3)} \rightarrow B_5$
26. $d_5 \rightarrow R_{4(1)} \rightarrow R_{3(2)} \rightarrow R_{4(3)} \rightarrow B_5$
27. $d_5 \rightarrow R_{4(1)} \rightarrow R_{3(2)} \rightarrow R_{5(3)} \rightarrow B_5$

2. Definitions

We return now to the question of the undefined terms "item of data", "report", and "business function", as used in the above and in [23]. Lieberman [26] attempted some explanation of these terms, and some further classification of entities. While he did not define "item of data", he did distinguish between "identification type" and "quantitative type" of information, which he symbolized by "i" and "q", respectively. Although Lieberman maintained this distinction throughout [26], no apparent purpose was served by the dichotomy. Lieberman also distinguished "source data forms" from "report forms". Again, this distinction appears to be unnecessary to the Lieberman model. Finally, Lieberman defined a "business function" to be "a set of managerial activities which are assigned to a group according to types of duties" [26].

3. Additional Concepts

In Kozmetsky and Kircher [24, Appendix 4] there is an additional concept, the time ordering of information: "This refers to the relative time that information arises, or is made into a report, not the specific date this occurs" [24, p. 277]. Following the definition, no further mention is made of the concept except for the paragraph:

> "The order of time is greater for the report forms on each higher level, in each higher-level matrix. This is necessary, since the system predicates that the reports on the higher levels are using data taken from those on lower levels, and thus must succeed them in time." [24, p. 281]

Kozmetsky and Kircher also introduce the concept of "the informal communication channel", which "designates the transfer of items by means other than documents" [24, p. 278]. Later in this work [24, p.285], the informal communication links are depicted by means of a connectivity matrix on the set of business functions; i.e., a squar^e matrix such that each row and each column represent a business function (ordered identically) and each element, $b_{i,j}$, is defined as:

$$b_{i,j} = \begin{cases} 1, \text{ if } B_i \text{ interacts with } B_j, \; i \neq j, \\ 0, \text{ otherwise.} \end{cases}$$

One other concept introduced in [24, pp. 285-286] is that of establishing a matrix depicting the optimum data requirements of each business function, and comparing this with the above defined $\mathfrak{M}_{0,n}$, to ascertain excessive routing of data or lack of desirable data.

CHAPTER 3

A GENERALIZATION OF THE LIEBERMAN MODEL

1. The Systems Structure

Perhaps the most obvious shortcoming of the Lieberman model [26], pointed out by Homer [23], is the fact that a well-ordered data processing structure must exist in order for the model to be applied. It is necessary for the analyst to be able to assign "levels" to the various reports (here, data items are assumed to be at level zero and business functions at level (n+1)) such that only entities of level (j-1) serve as inputs to entities of level j, and, in return, entities of level j become inputs to only those entities of level (j+1). When such a structure does exist, the resulting matrices are such that the column labels of M_j are identical to the row labels of $M_{(j+1)}$, and M_j and $M_{(j+1)}$ are compatible for multiplication.

That such well-ordered structures seldom, if ever, appear in actual practice is a conjecture that finds ready acceptance among experienced analysts. Rather, the following conditions are frequently found in actual cases:

1. Reports of level j are prepared not only from reports of level (j-1), but also from lower level reports.
2. Business functions are performed not only on the basis of n-th level reports, but also on the basis of lower level reports.
3. Not all business functions are of the same level.
4. Data items enter the system at a level higher than the first.

5. Reports prepared outside the scope of the study enter the system at some level higher than the first.
6. Some reports are terminal at levels less than $(n+1)$; that is, they are not used by any business function within the scope of the study.

All of these conditions lead to Lieberman matrices which are not conformable for multiplication, for there is no longer a guarantee that the row labels of M_j will be identical to the column labels of $M_{(j-1)}$. An example of such a system is shown in the form of a network in Figure 11, and in the form of a series of Lieberman matrices in Figure 12. (This example is from [23]).

2. Handling Non-Conformable Matrices - Method 1

Two methods of handling such situations were presented by Homer [23]. The first method calls for the formation of new matrices, N_j, from the given matrices, M_j, by the addition of "dummy" rows and columns as required to insure compatibility for matrix multiplication. The rules for the formation of the enlarged matrices are repeated below:

"1. Start with M_n. Add a row to M_n for each column in M_{n-1} not already represented by a row in M_n. All elements of added rows will be zero.

"2. For any row which contains only zeros, add a column with the identical heading. The new columns will have all elements equal to zero, except that elements formed by the intersection of identical rows and columns will have the value 1.

Figure 11

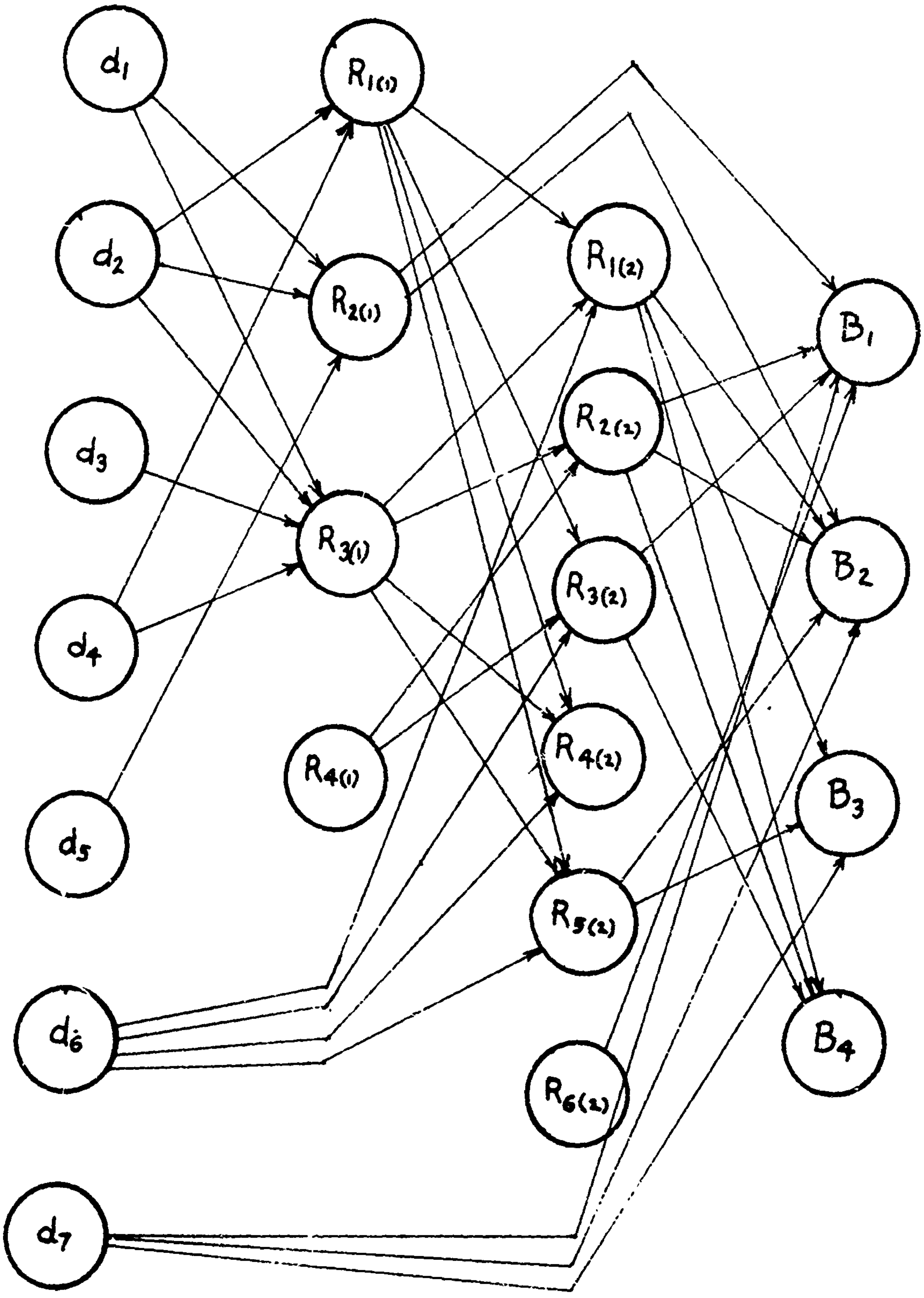

d_1
d_2
d_3
d_4
d_5
d_6
d_7
$R_{1(1)}$
$R_{2(1)}$
$R_{3(1)}$
$R_{4(1)}$
$R_{1(2)}$
$R_{2(2)}$
$R_{3(2)}$
$R_{4(2)}$
$R_{5(2)}$
$R_{6(2)}$
B_1
B_2
B_3
B_4

Figure 12

$$M_0 = \begin{array}{c||c|c|c||}
 & R_{1(1)} & R_{2(1)} & R_{3(1)} \\
\hline
d_1 & 0 & 1 & 1 \\
d_2 & 1 & 1 & 1 \\
d_3 & 0 & 0 & 1 \\
d_4 & 1 & 0 & 1 \\
d_5 & 0 & 1 & 0 \\
\hline
\end{array}$$

$$M_1 = \begin{array}{c||c|c|c|c|c||}
 & R_{1(2)} & R_{2(2)} & R_{3(2)} & R_{4(2)} & R_{5(2)} \\
\hline
R_{1(1)} & 1 & 0 & 1 & 1 & 1 \\
R_{3(1)} & 1 & 1 & 0 & 1 & 1 \\
R_{4(1)} & 0 & 1 & 1 & 0 & 0 \\
\hline
d_6 & 1 & 0 & 1 & 1 & 1 \\
\hline
\end{array}$$

$$M_2 = \begin{array}{c||c|c|c|c||}
 & B_1 & B_2 & B_3 & B_4 \\
\hline
R_{2(1)} & 1 & 1 & 0 & 0 \\
R_{1(2)} & 0 & 1 & 1 & 1 \\
R_{2(2)} & 1 & 1 & 0 & 1 \\
R_{3(2)} & 1 & 0 & 0 & 1 \\
R_{5(2)} & 0 & 1 & 1 & 0 \\
R_{6(2)} & 1 & 0 & 0 & 0 \\
d_7 & 1 & 1 & 1 & 0 \\
\hline
\end{array}$$

"3. Add a column to M_{n-1} for each row in M_n not already represented by a column in M_{n-1}. All elements of added columns will be zero.

"4. For any column which contains only zeros, add a row with the identical heading. The new rows will have all elements equal to zero, except that the elements formed by the intersection of identical rows and columns will have the value 1.

"5. Repeat steps 1 and 2 for matrix M_{n-1}. It may be necessary to make further adjustments in M_n as a result of changes in the columns of M_{n-1}.

"6. Repeat steps 3 and 4 for matrix M_{n-2}. Continue the process back through M_0." [23, p. 505].

Applying these rules to the matrices of Figure 12, we arrive at the set of matrices shown in Figure 13. These, of course, can be successively multiplied, since they have been constructed so as to be compatible. The product,

$$\mathfrak{N}_{0,2} = N_0 \cdot N_1 \cdot N_2$$

is shown in Figure 14.

What this method actually does is to imbed the problem in a higher dimension space. For example, let us suppose that we wish to multiply two matrices, A and B, where A is 3 X 3 and B is 4 X 4. If we consider the matrix A as the representation of a transformation in 3-space and the matrix B as the representation of a transformation in 4-space, we see that the "product"; A•B, even though multiplication of the matrices is not defined, is a transformation from 3-space to 4-space.

Figure 13

$$N_0 = \begin{array}{c||c|c|c|c|c|c|c|}
 & R_{1(1)} & R_{3(1)} & R_{4(1)} & d_6 & R_{2(1)} & R_{6(2)} & d_7 \\ \hline
d_1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
d_2 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\
d_3 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
d_4 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
d_5 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
R_{4(1)} & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
d_6 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
R_{5(2)} & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
d_7 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ \hline
\end{array}$$

$$N_1 = \begin{array}{c||c|c|c|c|c|c|c|c|}
 & R_{2(1)} & R_{1(2)} & R_{2(2)} & R_{3(2)} & R_{4(2)} & R_{5(2)} & R_{6(2)} & d_7 \\ \hline
R_{1(1)} & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\
R_{3(1)} & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\
R_{4(1)} & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
d_6 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\
R_{2(1)} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
R_{6(2)} & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
d_7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ \hline
\end{array}$$

$$N_2 = \begin{array}{c||c|c|c|c|c|}
 & B_1 & B_2 & B_3 & B_4 & R_{4(2)} \\ \hline
R_{2(1)} & 1 & 1 & 0 & 0 & 0 \\
R_{1(2)} & 0 & 1 & 1 & 1 & 0 \\
R_{2(2)} & 1 & 1 & 0 & 1 & 0 \\
R_{3(2)} & 1 & 0 & 0 & 1 & 0 \\
R_{4(2)} & 0 & 0 & 0 & 0 & 1 \\
R_{5(2)} & 0 & 1 & 1 & 0 & 0 \\
R_{6(2)} & 1 & 0 & 0 & 0 & 0 \\
d_7 & 1 & 1 & 1 & 0 & 0 \\ \hline
\end{array}$$

NOTE: *Row and column labels circled are those added to the matrizes of Figure 12.*

Figure 14

$$\mathcal{N}_{0,2} = \begin{array}{c|ccccc} & B_1 & B_2 & B_3 & B_4 & R_{4(2)} \\ \hline d_1 & 2 & 4 & 2 & 2 & 1 \\ d_2 & 3 & 6 & 4 & 4 & 2 \\ d_3 & 1 & 3 & 2 & 2 & 1 \\ d_4 & 2 & 5 & 4 & 4 & 2 \\ d_5 & 1 & 1 & 0 & 0 & 0 \\ R_{4(1)} & 2 & 1 & 0 & 2 & 0 \\ d_6 & 1 & 2 & 2 & 2 & 1 \\ R_{6(2)} & 1 & 0 & 0 & 0 & 0 \\ d_7 & 1 & 1 & 1 & 0 & 0 \end{array}$$

Reordering the rows and columns to agree with the sequence in which the entities enter the system:

$$\mathcal{N}_{0,2} = \begin{array}{c|ccccc} & R_{4(2)} & B_1 & B_2 & B_3 & B_4 \\ \hline d_1 & 1 & 2 & 4 & 2 & 2 \\ d_2 & 2 & 3 & 6 & 4 & 4 \\ d_3 & 1 & 1 & 3 & 2 & 2 \\ d_4 & 2 & 2 & 5 & 4 & 4 \\ d_5 & 0 & 1 & 1 & 0 & 0 \\ d_6 & 1 & 1 & 2 & 2 & 2 \\ d_7 & 0 & 1 & 1 & 1 & 0 \\ R_{4(1)} & 0 & 2 & 1 & 0 & 2 \\ R_{6(2)} & 0 & 1 & 0 & 0 & 0 \end{array}$$

If we create a new matrix, C, by adding a fourth row and a fourth column to A, such that:

$$c_{4,j} = 0; \; j = 1, 2, 3,$$

$$c_{i,4} = 0; \; i = 1, 2, 3,$$

$$c_{4,4} = 1;$$

and

$$c_{i,j} = a_{i,j}; \quad \text{elsewhere,}$$

the transformation represented by A is now represented by C, but the 3-space of A is now a 3-dimensional hyperplane in 4-space. The transformation C•B is identical to the transformation A•B, and multiplication of the matrices is defined.

From a practical standpoint, this method is not entirely satisfactory. The task of setting up the new matrices, N_j, is a tedious one, and the number of calculations required to obtain the desired result, $\mathfrak{N}_{0,n}$, is increased due to the increased number of rows and columns at almost every stage. Also, the analyst still has the burden of carefully classifying all reports into levels, for the way in which this is done may affect the complexity of the enlarged matrices.

3. Handling Non-Conformable Matrices - Method 2

A more direct approach to the problem is the second method of [23]. In order to explore this method further, we adopt a slightly different vocabulary.

The "items of data" of the Lieberman model will be referred to as system inputs. More generally, we will accept reports as inputs to the system, as well as items of data, for those reports which are prepared outside the system under consideration. We remark also that an input may be an aggregation of data items if it is more convenient for the analyst to treat a group of data items as one entity.

Outputs from the system are any terminal activities of the system. Thus, the business functions of the Lieberman model are system outputs, as are terminal reports (historical records) generated by the system.

All components of the system which are neither inputs nor outputs are intermediate entities. An intermediate entity is one which is produced by the system from either inputs or other intermediate entities or both, and which is used to produce other intermediate entities or to facilitate an output. (Note that an intermediate document may, in addition, be a terminal report, but that it is not considered an output.) Lieberman's reports and source documents are both intermediate entities. In this scheme it is not necessary to classify intermediate entities into levels, and they may be symbolized by the single-subscripted R_k.

We establish one matrix, S, referred to in the following as a systems-matrix, which will represent the entire information system under consideration. The following rules are used for setting up S:

"1. A row is established for each d_i and for each R_k in the system; i.e., for each component except the B_q's.

"2. A column is established for each R_k and for each B_q in the system; i.e., for each component of the system except the d_i's.

"3. As before, the number 1 will be inserted in each cell to represent [one entity used in the production of another] ...

"4. Each R_k will be represented by both a column and a row. Into each cell formed by the intersection of an identical row and column (r_{kk}), the value -1 will be inserted.

"5. All other cells will be labelled zero." [23, p. 508].

In Figure 15, we show an S matrix constructed for the example depicted in Figures 6 and 7. Similarly, in Figure 16, we show an S matrix for the case illustrated in Figures 11, 12, and 13. Note that the double-subscript notation for intermediate entities is used in Figures 15 and 16, but only to facilitate comparison with previous figures.

The first advantage offered by the systems-matrix model over the Lieberman model is that S may be analyzed for consistency very rapidly. A comparable examination of the Lieberman M_j's would be, at the least, cumbersome. The analysis consists of a rapid scan of S according to the following rules:

"a. Should any column contain only zeros, the [entity] represented is outside the scope of the problem being investigated, and the column should be removed.

Figure 15

$$S = \begin{array}{c|ccccccccccccccccccc}
 & R_{1(1)} & R_{2(1)} & R_{3(1)} & R_{4(1)} & R_{1(2)} & R_{2(2)} & R_{3(2)} & R_{1(3)} & R_{2(3)} & R_{3(3)} & R_{4(3)} & R_{5(3)} & B_1 & B_2 & B_3 & B_4 & B_5 & B_6 & B_7 \\
\hline
d_1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
d_2 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
d_3 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
d_4 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
d_5 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
d_6 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
R_{1(1)} & -1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
R_{2(1)} & 0 & -1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
R_{3(1)} & 0 & 0 & -1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
R_{4(1)} & 0 & 0 & 0 & -1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
R_{1(2)} & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
R_{2(2)} & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
R_{3(2)} & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
R_{1(3)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\
R_{2(3)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
R_{3(3)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\
R_{4(3)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\
R_{5(3)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 1 & 1 & 1 & 1
\end{array}$$

Figure 16

$$S = \begin{array}{c|cccccccccccccc}
 & R_{1(1)} & R_{2(1)} & R_{3(1)} & R_{4(1)} & R_{1(2)} & R_{2(2)} & R_{3(2)} & R_{4(2)} & R_{5(2)} & R_{6(2)} & B_1 & B_2 & B_3 & B_4 \\
\hline
d_1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
d_2 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
d_3 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
d_4 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
d_5 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
d_6 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
d_7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\
R_{1(1)} & -1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
R_{2(1)} & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
R_{3(1)} & 0 & 0 & -1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
R_{4(1)} & 0 & 0 & 0 & -1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
R_{1(2)} & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\
R_{2(2)} & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\
R_{3(2)} & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\
R_{4(2)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
R_{5(2)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 1 & 0 \\
R_{6(2)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0
\end{array}$$

"b. Should any row contain only zeros, the [entity] represented is not a component of the system, and the row should be removed.

"c. Should any column contain -1 as the only non-zero entry, the component represented by that column is really an input to the system. The column should be removed.

"d. Should any row contain -1 as the only non-zero entry, the component represented by that row is really an output of the system. The row should be removed." [23, p. 508].

These rules should be applied iteratively to S until no more changes can be made. If in the process both a row and a column representing the same entity are removed, this is an indication that that entity was not a component of the system under investigation.

It should be noted at this point that failure to apply the above simplification rules or failure to apply them to completion will not affect the final results, except to the extent that errors are made in determining the correct solution area (see below).

When these rules are applied to the matrix of Figure 16, the columns labelled $R_{4(1)}$ and $R_{6(2)}$ are removed, since these entities are system inputs, and the row labelled $R_{4(2)}$ is removed, since this report is a syst output. The reduced matrix is shown in Figure 17.

The solution area of S is defined to be the set of all cells of S, $\{s_{ij}\}$, such that row i represents an input and column j represents an output. A row represents an input if it does not contain an entry

Figure 17

$$S = \begin{array}{c||c|c|c|c|c|c|c|c|c|c|c|c}
 & R_{1(1)} & R_{2(1)} & R_{3(1)} & R_{1(2)} & R_{2(2)} & R_{3(2)} & R_{4(2)} & R_{5(2)} & B_1 & B_2 & B_3 & B_4 \\
\hline
d_1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
d_2 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
d_3 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
d_4 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
d_5 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
d_6 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
d_7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\
R_{1(1)} & -1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
R_{2(1)} & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
R_{3(1)} & 0 & 0 & -1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
R_{4(1)} & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
R_{1(2)} & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\
R_{2(2)} & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\
R_{3(2)} & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 1 \\
R_{5(2)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 1 & 0 \\
R_{6(2)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0
\end{array}$$

of -1. A column represents an output if it does not contain en entry of -1. In Figure 17, the solution area is bordered by heavy lines. Note that the cells constituting the solution area need not be adjacent to each other.

We define the solution to the model to be a matrix, S*, with row and column labels identical to S, containing the desired product in the solution area and arbitrary entries elsewhere. In [23] it is shown that S* may be obtained from S by performing only the following elementary column operations on S:

1. Multiplication of a column by a scalar, and
2. Addition of columns.

These two operations are to be performed iteratively so as to reduce to zero all those cells which lie in output columns but not within the solution area. Upon completion of these operations, S will have been transformed to S*, and the desired results will appear in the solution area.

Stated differently, we define a critical area of S to be a set of cells of S, $\{s_{k,\ell}\}$, such that row k represents an intermediate entity and column ℓ represents an output. Then, when the two elementary column operations prescribed above are performed so as to transform the critical area to all zeros, it will be found that the solution area has been transformed so as to contain the desired results. The "desired results" are, of course, the cells of $\mathfrak{N}_{0,n}$.

Although we have not specified what that portion of S* not included in either the solution area or the critical area should be, the algorithm leaves them identical to S.

A demonstration of the mathematical validity of the systems-matrix approach appears in [23, Appendix].

Figure 18 illustrates S* derived from S as shown ir Figure 17. Note that the values within the solution area correspond exactly to those of $\mathfrak{N}_{0,2}$ in Figure 14.

Figure 18

$S^* =$

	$R_{1(1)}$	$R_{2(1)}$	$R_{3(1)}$	$R_{1(2)}$	$R_{2(2)}$	$R_{3(2)}$	$R_{4(2)}$	$R_{5(2)}$	B_1	B_2	B_3	B_4
d_1	0	1	1	0	0	0	1	0	2	4	2	2
d_2	1	1	1	0	0	0	2	0	3	6	4	4
d_3	0	0	1	0	0	0	1	0	1	3	2	2
d_4	1	0	1	0	0	0	2	0	2	5	4	4
d_5	0	1	0	0	0	0	0	0	1	1	0	0
d_6	0	0	0	1	0	1	1	1	1	2	2	2
d_7	0	0	0	0	0	0	0	0	1	1	1	0
$R_{1(1)}$	-1	0	0	1	0	1	0	1	0	0	0	0
$R_{2(1)}$	0	-1	0	0	0	0	0	0	0	0	0	0
$R_{3(1)}$	0	0	-1	1	1	0	0	1	0	0	0	0
$R_{4(1)}$	0	0	0	0	1	1	0	0	2	1	0	2
$R_{1(2)}$	0	0	0	-1	0	0	0	0	0	0	0	0
$R_{2(2)}$	0	0	0	0	-1	0	0	0	0	0	0	0
$R_{3(2)}$	0	0	0	0	0	-1	0	0	0	0	0	0
$R_{5(2)}$	0	0	0	0	0	0	0	-1	0	0	0	0
$R_{6(2)}$	0	0	0	0	0	0	0	0	1	0	0	0

CHAPTER 4

A BASIS FOR EXTENSIONS TO THE MODEL

1. Introduction

In this chapter, we will develop a systems-matrix model of a management information system which will be essentially the same as the model developed in section 3 of Chapter 3 (Method 2 of [23]), and which will be used as the basis for further extensions. We will, however, adopt a slightly different scheme of notation than previously employed. This scheme will be somewhat less cumbersome than that previously used, since many of the distinctions of the Lieberman model need not be retained. Some restrictions will be placed on the notation, but only because this will simplify later explanations and discussion.

In this chapter, we will define only those terms and concepts necessary to discuss the model as developed so far. Additional definitions will appear as additional ideas are presented.

Throughout the rest of this thesis, the symbol I will refer to the identity matrix, and the symbol Z will refer to the zero matrix. The order of these matrices will be apparent from the usage.

2. System Components

We view a management information system as being composed of three classes of components: inputs, intermediate entities, and outputs, as discussed in Chapter 3, section 3. These components are symbolized by c_s; $s = 1,2,\ldots,n$; where:

$$n = n_3 + n_4 + n_5,$$

and,

n_3 = the number of inputs to the system,

n_4 = the number of intermediate entities in the system, and

n_5 = the number of outputs from the system.

(It is convenient to index the n's from 3 to 5, for in Chapter 6 we will partition the class of inputs into two classes of sizes n_1 and n_2, $n_1 + n_2 = n_3$.) We refer to the set of all components of the system by:

$$C_s = \left\{ c_s \mid 1 \leq s \leq n \right\} .$$

Indexing of the components will be restricted, as a matter of convenience, so that the index of any output will be greater than the index of any intermediate entity, which will be greater than the index of any input. This allows us to use the following notation for subsets of components:

S1. inputs: $C_{s3} = \left\{ c_s \mid \leq s \leq n_3 \right\}$

S2. intermediate entities: $C_{s4} = \left\{ c_s \mid (n_3+1) \leq s \leq (n_3+n_4) \right\}$

S3. outputs: $C_{s5} = \left\{ c_s \mid (n_3+n_4+1) \leq s \leq n \right\} .$

Obviously:

S4. $C_s = C_{s3} \cup C_{s4} \cup C_{s5}$

and

S5. $C_{s3} \cap C_{s4} = C_{s3} \cap C_{s5} = C_{s4} \cap C_{s5} = C_{s3} \cap C_{s4} \cap C_{s5} = \Phi,$

where Φ indicates the null (empty) set. Stated in words, S4 and S5 mean that the sets of inputs, intermediate entities and outputs are mutually exclusive and exhaustive.

3. Direct Connectivity and Paths

If there is a flow of data directly from component c_i to c_j, we shall say that c_i is directly connected to c_j. Direct connectivity may exist only between members of classes of components as shown below:

Direct Connectivity

	From	To
C1.	C_{s3}	C_{s4}
C2.	C_{s3}	C_{s5}
C3.	C_{s4}	C_{s4}
C4.	C_{s4}	C_{s5}

We impose the rule:

C5. If c_i is directly connected to c_j, then $i < j$, as a further convenient restriction on the indexing of components. This rule follows for C1, C2 and C4 from S1, S2 and S3. That it is feasible for C3 follows from S1, S2 and S3, and from C6, C7 and P2, below. Obviously, the direction of direct connectivity is significant.

Direct connectivity is a relation on the set C_s, symbolized by:

$$c_i \oplus\!\!\rightarrow_0 c_j.$$

(The denial of the relation is denoted by the symbol $\overline{\oplus\!\!\rightarrow}_0$.) The relation is:

C6. anti-symmetric: if $c_i \oplus\!\!\rightarrow_0 c_j$, then it must be true that $c_j \overline{\oplus\!\!\rightarrow}_0 c_i$,

C7. anti-reflexive: $c_i \overline{\oplus\!\!\rightarrow}_0 c_i$, C8. non-transitive: if $c_i \oplus\!\!\rightarrow_0 c_j$ and $c_j \oplus\!\!\rightarrow_0 c_k$, not necessarily true that $c_i \oplus\!\!\rightarrow_0 c_k$.

A path exists from c_i to c_j if:

P1. $c_i \oplus\!\!\rightarrow_0 c_j$, or

P2. there exists a set of components, $\{c_k\}$; $q = 1,2,\ldots,p$; $i < k_1 < k_2 < \ldots < k_p < j$; such that

$$c_i \oplus\!\!\rightarrow_0 c_{k_1} \oplus\!\!\rightarrow_0 c_{k_2} \oplus\!\!\rightarrow_0 \cdots \oplus\!\!\rightarrow_0 c_{k_p} \oplus\!\!\rightarrow_0 c_j.$$

We say that the path is of length p and write:

$$c_i \oplus\!\!\rightarrow_p c_j.$$

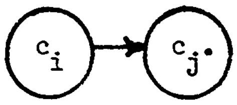
Direct connectivity is a path of length zero, by P1.

4. Network Representation

The system, C_s, may be represented by a directed graph or network wherein there is a node from each $c_s \in C_s$, and a directed arc between nodes whenever direct connectivity holds between the components associated with the nodes. The direction of the arc agrees with the implied direction of the relation. Thus, if:

$$c_i \oplus\!\!\rightarrow_0 c_j$$

then the network will contain:



We will frequently refer to this network, but only as a conceptual aid. The network should not be regarded as a part of the model.

We note that it is not possible for the network to contain a loop (i.e., an arc directed from one component back to itself) because of C7.

Neither can it contain a circuit (i.e., a path of non-zero length which includes the same node more than once) because of the strict inequality imposed on the indices by P2.

5. Matrix Representation

The system, C_s, may be represented by a matrix, S, of order $(n_3 + n_4) \times (n_4 + n_5)$, whose elements are defined as:

$$\text{M1.} \quad s_{i,j} = \begin{cases} 1: \text{ if } c_i \quad {}_0\, c_j;\ i \neq j, \\ -1;\ \text{if } i = j \\ 0;\ \text{otherwise}\ ; \end{cases}$$

and:

$$1 \leq i \leq (n_3 + n_4);$$

$$(n_3 + 1) \leq j \leq n\ .$$

This matrix is akin to, but not identical to, the adjacency, or incidence matrix commonly associated with a directed graph [15;16, p.15]. It differs by the fact that S is not square and by the sub-diagonal of -1's. Note also that the indexing of the columns assigns to the first column the value $(n_3 + 1)$ rather than 1. This is a matter of convenience, not necessity. Each row of S is labelled c_i, and each column of S is labelled c_j. The matrix S may be decomposed into submatrices, as shown below, with the notation $S_{\alpha,\beta}$ indicating a submatrix depicting the set ι paths from the elements of the set C_α to the elements of the set C_β. (If $\alpha = \beta$ the submatrix represents the set of paths among elements of the set C_α.)

M2. $$S = \begin{array}{c||c|c||} & j=(n_3+1)\ \dots\ (n_3+n_4) & j=(n_3+n_4+1)\ \dots\ n \\ \hline\hline \begin{matrix} i = 1 \\ \vdots \\ n_3 \end{matrix} & S_{s3,s4} & S_{s3,s5} \\ \hline \begin{matrix} i = (n_3+1) \\ \vdots \\ (n_3+n_4) \end{matrix} & S_{s4,s4} - I & S_{s4,s5} \\ \hline\hline \end{array}$$

Obviously, the matrix S as described above is identical to the matrix S as developed in [23] and described in Chapter 3, except for notation. All the previous remarks about S clearly still apply. For example, the scanning rules [23] quoted in Chapter 3 may be summarized by the following:

M3. Every row and every column of S shall contain at least one entry of +1.

This is equivalent to saying that every member of C_s must be directly connected to at least one other component.

6. Determination of S^*

We return now to a discussion of the submatrices, or blocks, of S as depicted by M2. These are listed below, with their orders:

Block	Order
$S_{s3,s4}$	$n_3 \times n_4$
$S_{s3,s5}$	$n_3 \times n_5$
$(S_{s4,s4}-I)$	$n_4 \times n_4$
$S_{s4,s5}$	$n_4 \times n_5$.

The orders of the blocks are important in that they indicate which combinations of blocks are conformable for multiplication. Thus, the product $(S_{s3,s4})\cdot(S_{s4,s5})$ is defined, but the product $(S_{s3,s5})\cdot(S_{s4,s5})$ is not.

The solution of the model, S*, is described in Chapter 3. In terms of the notation employed in this chapter:

$$\text{M4.}\quad S^* = \left[\begin{array}{c|c} S^*_{s3,s4} & S^*_{s3,s5} \\ \hline S^*_{s4,s4} & S^*_{s4,s5} \end{array}\right]$$

By analogy to the previous definition of S* (Chapter 3), we can specify the blocks of S* to be:

$$\text{M5.}\quad S^* = \left[\begin{array}{c|c} S_{s3,s4} & S^*_{s3,s5} \\ \hline S_{s4,s4}-I & Z \end{array}\right]$$

where $S^*_{s3,s5}$ is the previously defined "solution area". Our problem now is to determine the algebraic form of $S^*_{s3,s5}$.

The block $S_{s4,s4}$ is an adjacency matrix within the subset C_{s4}. We define an adjacency matrix, A, for the entire system, C_s, to be an n X n matrix, with elements:

$$A = \begin{array}{c|c|c|c|} & j=1\ldots n_3 & j=(n_3+1)\ldots(n_3+n_4) & j=(n_3+n_4+1)\ldots n \\ \hline \begin{array}{c} i = \quad 1 \\ \vdots \\ n_3 \end{array} & Z & S_{s3,s4} & S_{s3,s4} \\ \hline \begin{array}{c} i = (n_3+1) \\ \vdots \\ (n_3+n_4) \end{array} & Z & S_{s4,s4} & S_{s4,25} \\ \hline \begin{array}{c} i = (n_3+n_4+1) \\ \vdots \\ n \end{array} & Z & Z & Z \\ \hline \end{array}$$

The matrix A represents the number of paths of length zero in the system. Each non-zero block, $S_{\alpha,\beta}$, represents the number of paths of length zero from components in C_α to components in C_β .

It is well known (for example, [16, p.112]) that if $B = A^k$; $k = 1, 2, 3, \ldots,$ then $b_{i,j}$ represents the number of paths of length $k-1$ from c_i to c_j. Squaring the adjacency matrix, A, we obtain:

$$A^2 = \left[\begin{array}{c|c|c} Z & (S_{s3,s4})\cdot(S_{s4,s4}) & (S_{s3,s4})\cdot(S_{s4,s5}) \\ \hline Z & (S_{s4,s4})^2 & (S_{s4,s4})\cdot(S_{s4,s5}) \\ \hline Z & Z & Z \end{array}\right]$$

Note that all products are defined. The blocks of A^2 may be interpreted according to this example: the product $(S_{s3,s4})\cdot(S_{s4,s5})$ represents the number of paths of length 1 from a component in C_{s3} to a component in C_{s5}. Continuing:

$$A^3 = \left[\begin{array}{c|c|c} Z & (S_{s3,s4})\cdot(S_{s4,s4})^2 & (S_{s3,s4})\cdot(S_{s4,s4})\cdot(S_{s4,s5}) \\ \hline Z & (S_{s4,s4})^3 & (S_{s4,s4})^2\cdot(S_{s4,s5}) \\ \hline Z & Z & Z \end{array}\right]$$

$$A^4 = \left[\begin{array}{c|c|c} Z & (S_{s3,s4})\cdot(S_{s4,s4})^3 & (S_{s3,s4})\cdot(S_{s4,s4})^2\cdot(S_{s4,s5}) \\ \hline Z & (S_{s4,s4})^4 & (S_{s4,s4})^3\cdot(S_{s4,s5}) \\ \hline Z & Z & Z \end{array}\right]$$

And, by induction:

$$A^k = \left[\begin{array}{c|c|c} Z & (S_{s3,s4})\cdot(S_{s4,s4})^{k-1} & (S_{s3,s4})\cdot(S_{s4,s4})^{k-2}\cdot(S_{s4,s5}) \\ \hline Z & (S_{s4,s4})^k & (S_{s4,s4})^{k-1}\cdot(S_{s4,s5}) \\ \hline Z & Z & Z \end{array}\right]$$

In A^k, the interpretation of the upper right block is, clearly, the number of paths of length $(k-1)$ from C_{s3} to C_{s5}.

The solution area of S^*, denoted $S^*_{s3,s5}$, must contain the total number of paths, of any length, from inputs (C_{s3}) to outputs (C_{s5}), which is the sum of the upper right blocks of all the powers of A; i.e.:

$$S^*_{s3,s5} = (S_{s3,s5}) + (S_{s3,s4})\cdot(S_{s4,s5}) + (S_{s3,s4})\cdot(\sum_{k=3}^{\infty}(S_{s4,s4})^{k-2})\cdot(S_{s4,s5}) \tag{4-1}$$

Earlier, we imposed rule C5, and showed that it was feasible for case C3. Case C3 is represented in S by $S_{s4,s4}$. The application of C5 to that submatrix means that $S_{s4,s4}$ is strictly upper triangular. It is known ([9, p.81], for example) that a strictly triangular matrix is nilpotent of index ρ, that is, there exists an integer, ρ, such that:

$$(S_{s4,s4})^k = Z; \quad \text{for} \quad k \geq \rho,$$

and

$$(S_{s4,s4})^k \neq Z; \quad \text{for} \quad k < \rho.$$

This means that the summation in equation (4-1) is finite, and we may write:

$$\begin{aligned} S^*_{s3,s4} &= (S_{s3,s4}) + (S_{s3,s4})\cdot(S_{s4,s5}) + (S_{s3,s4})\cdot(\sum_{k=3}^{\rho+1}(S_{s4,s4})^{k-2})\cdot(S_{s4,s5}) \\ &= (S_{s3,s5}) + (S_{s3,s4})\cdot(\sum_{k=2}^{\rho+1}(S_{s4,s4})^{k-2})\ (S_{s4,s5}) \\ &= (S_{s3,s5}) + (S_{s3,s4})\cdot(\sum_{k=0}^{\rho-1}(S_{s4,s4})^{k})\ (S_{s4,s5})\ , \end{aligned} \tag{4-2}$$

since $\quad (S_{s4,s4})^0 = I\ ,$

and $\quad (S_{s3,s4})\cdot(S_{s4,s5}) = (S_{s3,s4})\cdot I\cdot(S_{s4,s5}) = (S_{s3,s4})\cdot(S_{s4,s4})^0\cdot(S_{s4,s5}).$

Since $S_{s4,s4}$ is nilpotent, we know that

$$\sum_{k=0}^{\rho-1} (S_{s4,s4})^k = (I - S_{s4,s4})^{-1} \tag{4-3}$$

(This is one of those facts of matrix algebra that is almost inevitably left as "an exercise for the reader". See, for example. [5, p.111; 20, p.68].) Using equation (4-3), we may re-write equation (4-2) as:

$$S^*_{s3,s5} = (S_{s3,s5}) + (S_{s3,s4})\cdot(I - S_{s4,s4})^{-1}\cdot(S_{s4,s5}).$$

This permits us to specify the solution matrix, following M5, as:

$$S^* = \left[\begin{array}{c|c} S_{s3,s4} & (S_{s3,s5}) + (S_{s3,s4})\cdot(I - S_{s4,s4})^{-1}\cdot(S_{s4,s5}) \\ \hline S_{s4,s4} - I & Z \end{array}\right]. \tag{4-4}$$

7. The Systems-Matrix Algorithm

We now postulate that there exists at least one matrix, T, of order $(n_3 + n_4) \times (n_3 + n_4)$, such that post-multiplication of S by T will yield S^*:

$$S\cdot T = S^* \tag{4-5}$$

In block notation, equation (4-5) is:

$$\left[\begin{array}{c|c} S_{s3,s4} & S_{s3,s5} \\ \hline S_{s4,s4}-I & S_{s4,s5} \end{array}\right] \cdot \left[\begin{array}{c|c} T_{11} & T_{12} \\ (n_3 \times n_3) & (n_3 \times n_4) \\ \hline T_{21} & T_{22} \\ (n_4 \times n_3) & (n_4 \times n_4) \end{array}\right]$$

$$= \left[\begin{array}{c|c} S_{s3,s4} & (S_{s3,s5}) + (S_{s3,s4}) \ (I - S_{s4,s4})^{-1} \ (S_{s4,s5}) \\ \hline S_{s4,s4} - I & Z \end{array}\right] \tag{4-6}$$

From equation (4-6), we can obtain the following equations by block multiplication:

$$(S_{s3,s4})\cdot(T_{11}) + (S_{s3,s5})\cdot(T_{21}) = S_{s3,s4} \tag{4-7}$$

$$(S_{s4,s4} - I)\cdot(T_{11}) + (S_{s4,s5})\cdot(T_{21}) = S_{s4,s4} - I \tag{4-8}$$

$$(S_{s3,s4})\cdot(T_{12}) + (S_{s3,s5})\ (T_{22}) = (S_{s3,s5}) + (S_{s3,s4})\cdot(I - S_{s4,s4})^{-1}\cdot(S_{s4,s5}) \tag{4-9}$$

$$(S_{s4,s4} - I)\cdot(T_{12}) + (S_{s4,s5})\cdot(T_{22}) = Z \tag{4-10}$$

Equations (4-7) and (4-8) are satisfied by:

$$T_{11} = I \tag{4-11}$$

and

$$T_{21} = Z \tag{4-12}$$

(We do not claim that these values are unique.) Equation (4-10) may be written:

$$-(I - S_{s4,s4})\cdot(T_{12}) = (S_{s4,s5})\cdot(T_{22}) \ ,$$

whence:

$$T_{12} = (I - S_{s4,s4})^{-1}\cdot(S_{s4,s5})\cdot(T_{22}) \ , \tag{4-13}$$

since the existence of $(I - S_{s4,s4})^{-1}$ has already been demonstrated.

Substituting equation (4-13) in equation (4-9):

$$(S_{s3,s4})\cdot(I - S_{s4,s4})^{-1}\cdot(S_{s4,s5})\cdot(T_{22}) + (S_{s3,s5})\cdot(T_{22})$$

$$= (S_{s3,s5}) + (S_{s3,s4})\cdot(I - S_{s4,s4})^{-1}\cdot(S_{s4,s5}) \ , \tag{4-14}$$

which is satisfied by:

$$T_{22} = I \ . \qquad (4\text{-}15)$$

Then, from equation (4-13):

$$T_{12} = (I - S_{s4,s4})^{-1} \cdot (S_{s4,s5}) \ . \qquad (4\text{-}16)$$

By combining equations (4-11), (4-12), (4-15) and (4-16):

$$T = \left[\begin{array}{c|c} I & (I - S_{s4,s4})^{-1} (S_{s4,s5}) \\ \hline Z & I \end{array}\right] \qquad (4\text{-}17)$$

Obviously, from equation (4-17), T is an upper triangular matrix, with a main diagonal of all 1's. Since det(T) = 1, T is non-singular. It is well known (for example, [8, pp.217-228; 9, pp.125-129; 19, pp.23-26, 20, pp.95-98; 37, pp.102-106] that:

a. a non-singular matrix is equivalent to the product of a finite sequence of elementary matrices, and,

b. post-multiplication by elementary matrices is equivalent to elementary column operations.

Therefore, S may be transformed to S* by performing a finite sequence of elementary column operations on S. The algorithm is:

A1. Find an entry in $S_{s4,s5}$ which is non-zero. If there are none, the algorithm is finished, and the desired results are in block $S_{s3,s5}$. If one is found, call the row in which it occurs r and the column t.

A2. Search row r for a negative entry. Call the column in which one occurs d.

A3. Add to column t the scalar product of column d and $s_{r,t}$.

Go to Step A1.

8. The Validity of the Algorithm

For this algorithm to be correct, it must be true that:

a. it is always possible to perform the stated steps,

b. the algorithm will terminate, and

c. the correct solution will have been generated when the algorithm terminates.

Steps A1 and A3 are obviously always executable. Step A2 can be performed if there exists an entry of -1 in the row being scanned. Since $S_{s4,s4}$ is strictly upper triangular and therefore contains only zeros on its main diagonal, the submatrix $(S_{s4,s4} - I)$ will contain entries of -1 on its main diagonal. Only the rows of $S_{s3,s5}$ are scanned. Therefore, for each row subject to scanning there is an entry of -1 in the initial block $(S_{s4,s4} - I)$. This block is never changed, since Step A3 operates only on columns in blocks $S_{s3,s5}$ and $S_{s4,s5}$. Therefore, Step A2 can always be executed.

The algorithm will terminate when the block $S_{s4,s5}$ has been reduced to Z. To see that this reduction to all zeros will always occur, we select, without loss of generality, the bottom-most non-zero entry in some column of $S_{s4,s5}$, say, $s_{r,t}$, in Step A1. Steps A2 and A3 will reduce that entry to zero. In so doing, multiples of positive values in column d will be added to column t, but any such additions to column t will be above row r, because of the strict upper triangularity of $S_{s4,s4}$. As we repeat the steps of the algorithm on the same column, we never add entries to column t in a row already reduced to zero. Therefore, we will, after a finite number of iterations, reduce the entire column t to all zero entries in rows (n_3+1) to (n_3+n_4). This is true for all columns from (n_3+n_4+1) to n. Therefore, $S_{s4,s5}$ can be reduced to all zeros and the algorithm will terminate.

Consider repeating the steps of the algorithm only for the original entries of $S_{s4,s5}$. The effect of doing this is to change $S_{s4,s5}$ to $(S_{s4,s5} + S_{s4,s5} \cdot (S_{s4,s4} - I)) = (S_{s4,s5} \cdot S_{s4,s4})$; and to change $S_{s3,s5}$ to $(S_{s3,s5} + S_{s3,s4} \cdot (S_{s4,s5}))$. Call this cycle 1. Repeat the algorithm, but only for the entries added to $S_{s4,s5}$ by cycle 1. When this has been done (cycle 2), the lower right block will have been changed to $(S_{s4,s5} \cdot S_{s4,s4} + S_{s4,s5} \cdot S_{s4,s4} \cdot (S_{s4,s4} - I)) = (S_{s4,s5} \cdot (S_{s4,s4})^2)$; and the upper right block (the solution area) will have been changed to

$$(S_{s3,s5} + S_{s3,s4} \cdot (S_{s4,s5}) + S_{s3,s4} \cdot S_{s4,s4} \cdot S_{s4,s5}))$$

$$= (S_{s3,s5} + S_{s3,s4} \cdot (\sum_{k=0}^{1} (S_{s4,s4})^k)\ S_{s4,s5}).$$

It is obvious that at the completion of cycle ρ (where ρ is the index of nilpotency of $S_{s4,s4}$), the lower right block will be $S_{s4,s5} \cdot (S_{s4,s4}) = Z$; and the solution area will be

$$(S_{s3,s5} + S_{s3,s4} \cdot (\sum_{k=0}^{\rho-1} (S_{s4,s4})^k) \cdot S_{s4,s5}),$$ which is the desired result.

9. The Determination of Paths

The solution area of the model developed in the previous sections indicates the number of paths by which an input can reach an output. An associated problem is to identify each of these paths, rather than to enumerate them. This can be accomplished by a relatively minor change in the model. We call the systems-matrix for this problem SP, and we establish it exactly as we established S, except that wherever there is an entry of -1 in S, we set

$$sp_{i,i} = -(1/c_i),$$

where c_i is a symbol for a component, not a numerical value. We also impose the restriction that multiplication is not commutative. This is because we shall interpret the indicated products of symbols to mean a path. If a cell in the solution area, say, $s_{i,j}$, contains the entry $c_a \cdot c_b \cdot c_d$ we shall say that there exists the path of length 3: $c_i \underset{0}{\oplus\!\!\rightarrow} c_a \underset{0}{\oplus\!\!\rightarrow} c_b \underset{0}{\oplus\!\!\rightarrow} c_d \underset{0}{\oplus\!\!\rightarrow} c_j$. Because of C6, the product cannot be commutative.

In the algorithm, the only change necessary is to specify in Step A3 that to column t be added the scalar product of column d post-multiplied by $(c_d \cdot s_{r,t})$. We still take $1/c_i$ to be the multiplicative inverse of c_i; that is,

$$-(1/c_i) \cdot c_i = -1 .$$

In Figure 19, we show the matrix SP for the case illustrated in Figure 11. The solution area of SP*, arrived at by application of the systems-matrix algorithm, is shown in Figure 20. (Note that in Figures 19

Figure 19

$$SP = \begin{array}{c|cccccccccccc}
 & R_{1(1)} & R_{2(1)} & R_{3(1)} & R_{1(2)} & R_{2(2)} & R_{3(2)} & R_{4(2)} & R_{5(2)} & B_1 & B_2 & B_3 & B_4 \\
\hline
d_1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
d_2 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
d_3 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
d_4 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
d_5 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
d_6 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
d_7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\
R_{1(1)} & -\frac{1}{R_{1(1)}} & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
R_{2(1)} & 0 & -\frac{1}{R_{2(1)}} & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
R_{3(1)} & 0 & 0 & -\frac{1}{R_{3(1)}} & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
R_{4(1)} & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
R_{1(2)} & 0 & 0 & 0 & -\frac{1}{R_{1(2)}} & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\
R_{2(2)} & 0 & 0 & 0 & 0 & -\frac{1}{R_{2(2)}} & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\
R_{3(2)} & 0 & 0 & 0 & 0 & 0 & -\frac{1}{R_{3(2)}} & 0 & 0 & 1 & 0 & 0 & 1 \\
R_{5(2)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{R_{5(2)}} & 0 & 1 & 1 & 0 \\
R_{6(2)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0
\end{array}$$

Figure 20

	$R_{4(2)}$	B_1	B_2
d_1	$R_{3(1)}$	$R_{3(1)} \cdot R_{2(2)} + R_{2(1)}$	$R_{3(1)}(R_{5(2)}+R_{2(2)}+R_{1(2)}) + R_{2(1)}$
d_2	$R_{3(1)}+R_{1(1)}$	$R_{3(1)} \cdot R_{2(2)} + R_{2(1)} + R_{1(1)} \cdot R_{3(2)}$	$R_{3(1)}(R_{5(2)}+R_{2(2)}+R_{1(2)})+R_{2(1)} + R_{1(1)}(R_{5(2)}+R_{1(2)})$
d_3	$R_{3(1)}$	$R_{3(1)} \cdot R_{2(2)}$	$R_{3(1)}(R_{5(2)}+R_{2(2)}+R_{1(2)})$
d_4	$R_{3(1)}+R_{1(1)}$	$R_{3(1)} \cdot R_{2(2)} + R_{1(1)} \cdot R_{3(2)}$	$R_{3(1)}(R_{5(2)}+R_{2(2)}+R_{1(2)})+R_{1(1)}(R_{5(2)}+R_{1(2)})$
d_5	0	$R_{2(1)}$	$R_{2(1)}$
d_6	1	$R_{3(2)}$	$R_{5(2)} + R_{1(2)}$
d_7	0	1	1
$R_{4(1)}$	0	$R_{3(2)} + R_{2(2)}$	$R_{2(2)}$
$R_{6(2)}$	0	1	0

	B_3	B_4
d_1	$R_{3(1)}(R_{5(2)}+R_{1(2)})$	$R_{3(1)}(R_{2(2)}+R_{1(2)})$
d_2	$R_{3(1)}(R_{5(2)}+R_{1(2)}) + R_{1(1)}(R_{5(2)}+R_{1(2)})$	$R_{3(1)}(R_{2(2)}+R_{1(2)}) + R_{1(1)}(R_{3(2)}+R_{1(2)})$
d_3	$R_{3(1)}(R_{5(2)}+R_{1(2)})$	$R_{3(1)}(R_{2(2)}+R_{1(2)})$
d_4	$R_{3(1)}(R_{5(2)}+R_{1(2)}) + R_{1(1)}(R_{5(2)}+R_{1(2)})$	$R_{3(1)}(R_{2(2)}+R_{1(2)}) + R_{1(1)}(R_{3(2)}+R_{1(2)})$
d_5	0	0
d_6	$R_{5(2)}+R_{1(2)}$	$R_{3(2)} + R_{1(2)}$
d_7	1	0
$R_{4(1)}$	0	$R_{3(2)} + R_{2(2)}$
$R_{6(2)}$	0	0

and 20, the notation of Chapter 3 has been retained to facilitate comparison.)

Looking at the cell representing the connectivity from d_1 to $R_{4(2)}$ in Figure 20, we find the single entry $R_{3(1)}$. This depicts the existence of only one path, of length 1, connecting these two components, namely:

$$d_1 \oplus\!\!\rightarrow_0 R_{3(1)} \oplus\!\!\rightarrow_0 R_{4(2)} .$$

From d_3 to B_1 there is a path of length 2:

$$d_3 \oplus\!\!\rightarrow_0 R_{3(1)} \oplus\!\!\rightarrow_0 R_{2(2)} \oplus\!\!\rightarrow_0 B_1 .$$

From d_6 to B_2 there are two paths of length 1:

$$d_6 \oplus\!\!\rightarrow_0 R_{5(2)} \oplus\!\!\rightarrow_0 B_2 ,$$

and

$$d_6 \oplus\!\!\rightarrow_0 R_{1(2)} \oplus\!\!\rightarrow_0 B_2 .$$

Similarly, there are two paths of length 2 from d_4 to B_1. From d_3 to B_2 there are three paths of length 2, since

$$R_{3(1)} - (R_{5(2)} + R_{2(2)} + R_{1(2)})$$

$$= R_{3(1)} \cdot R_{5(2)} + R_{3(1)} \cdot R_{2(2)} + R_{3(1)} \cdot R_{1(2)} .$$

From d_2 to B_2, there are six paths, one of length 1 and five of length 2. Finally, the entry of 1 indicates a path of length zero (direct connectivity), and the entry of 0 indicates no path.

It should be noted that the matrix SP and its associated algorithm can be formalized to a "path algebra" with the operations "is directly connected to" and "and" over the elements of the power set of C_s. However, at this point, there is nothing to be gained by such a strategy. Rather, that approach would hinder the mixed models (numeric and symbolic) that will be developed in a later chapter, and would inhibit rather than help the type of intuitive understanding of a model that the analyst should have.

Chapter 5

INFORMATION FILTERING

1. Introduction

The systems-matrix model developed in Chapter 4 has incorporated in it an assumption which is now challenged. In speaking of the direct connectivity relation between two components in C_{s4} or from a component in C_{s4} to a component in C_{s5}, denoted:

$$c_i \oplus\!\!\rightarrow_{0} c_j \text{ ,}$$

we have implicity assumed that if a path exists from some input, c_k, to c_i, then a path will exist from c_k to c_j. In other words, the model assumes that whatever input items have been received by an intermediate entity will be transmitted to every other component to which that intermediate entity is directly connected. This assumption has lead to the fact that, in our previous models, any input to a system could be traced to one or more system outputs. In fact, the violation of that rule was used as an indicator that what the analyst thought was an intermediate entity was truly an output.

In practice, management information systems need not necessarily possess this characteristic. Data may be "dropped", or "lost", within a system. In the preparation of c_j from c_i, not all of the input items contained in c_i may appear in c_j. We shall refer to this phenomenon as <u>information filtering</u>, or, simply, <u>filtering</u>.

Filtering may be classified as:

1. systematic filtering, if the filtering is a deliberate, planned part of the management information system (or results in a deterministic fashion from a planned part of the system), or,
2. random filtering, if the loss of information results from noise, error, equipment failure, or other non-planned, randomly occurring events.

In this chapter, only systematic filtering will be considered, and the term "filtering" will be interpreted to mean "systematic information filtering".

Two types of filtering are recognized:

1. unconditional filtering occurs when certain specific items of data (input items) in c_i are never transmitted to c_j, and,
2. conditional filtering occurs when the transmittal or non-transmittal of some or all of the items in c_i to c_j is conditioned upon the values of one or more input items (called criteria input items). The criteria input items may or may not be included in the set of input items subject to filtering.

In the case of unconditional filtering, the filtering action is independent of the data being processed, exactly which items of data are transmitted and which are not transmitted is determined by systems design. The two limiting cases of unconditional filtering are:

1. zero filtering: all of the data from c_i are transmitted to each c_j to which c_i is directly connected. This covers the assumption of the previous models.

2. total filtering: none of the data from c_i is transmitted to any c_j. This is equivalent to saying that c_i is not directly connected to any c_j. Then, this is the previously discussed case wherein c_i is a system output.

From the above we can see that only those instances of unconditional filtering lying between the limiting cases (i.e., partial filtering) present a new problem.

In conditional filtering, not only the specific filtering action, but also the filtering process itself may not be completely specified by the systems design, but may be a function of specific values of data processed. It is possible that zero filtering may occur at some times, total filtering may occur at other times, and various degrees of partial filtering at still other times. Conditional filtering may involve a change of state: should zero filtering occur, the process changes to a non-filtering one; should total filtering occur, the process may change to a non-filtering one with one or more intermediate entities becoming outputs. Thus, in conditional filtering the limiting cases as well as partial filtering must be considered. Furthermore, both non-filtering and unconditional filtering are special cases of conditional filtering, as will be developed in this chapter. Therefore, the discussion following will concentrate on the conditional filtering problem.

2. The Relation of Micro-Direct Connectivity

The concept of direct connectivity as present in Chapter 4 is meaningful only when there is zero filtering unconditionally throughout the system.

A model which employs such an assumption will be referred to as a *macro-model*, and the analysis of such a system as *macro-analysis*. In this chapter we will be concerned with *micro-models* and *micro-analysis*, applied to systems wherein we must investigate the transfer of information between components with reference to each item of data in the system. The relation of direct connectivity tells us only that there is, or is not, a transfer of some information from one component to another. We need a more detailed relation between components to indicate just which specific items of data are transferred.

Conceptually, the macro-model pictured a direct connectivity relation from c_i to c_j as a single directed arc from the node labelled c_i to the node labelled c_j. We now need a relation which can be depicted as a group, or aggregation, of identically directed arcs between two nodes. Each arc in the group will represent the transfer of one specific item of data, and there will be one arc in the group for each item of data transferred between the two nodes. Then we may say that c_i is *directly connected with respect to input c_k* to c_j, if an arc labelled c_k is directed from c_i to c_j on our conceptual network. This will be symbolized:

$$c_i \oplus\!\xrightarrow[0]{k} c_j, \; 1 \leq k \leq n_3 \, .$$

In order to distinguish between the aggregate and the detailed direct connectivity relations, we will call the previously defined one *macro-direct connectivity*, and the one defined in this section *micro-direct connectivity*. We may state:

C9. The component c_i is macro-directly connected to component c_j if and only if c_i is micro-directly connected to c_j for at least one item of data.

It can be readily verified that all the statements made in Chapter 4, section 3, about macro-direct connectivity, statements C1 through C8, apply as well to micro-direct connectivity.

We also may extend our definition of a path to the micro-model. The path previously defined in Chapter 4, section 3, P1 and P2, we will now refer to as a macro-path. We say that a micro-path with respect to input c_k exists from c_i to c_j if:

P3. $c_i \oplus\!\!\xrightarrow[0]{k} c_j$, or

P4. there exists a set of components, $\{c_{r_q}\}$;$q = 1,2,\ldots,p$;

$i < r_1 < r_2 < \ldots < r_p < j$; such that

$$c_i \oplus\!\!\xrightarrow[0]{k} c_{r_1} \oplus\!\!\xrightarrow[0]{k} c_{r_2} \oplus\!\!\xrightarrow[0]{k} \ldots\ldots \oplus\!\!\xrightarrow[0]{k} c_{r_p} \oplus\!\!\xrightarrow[0]{k} c_j \, .$$

We say that the micro-path with respect to c_k is of length p and write:

$$c_i \oplus\!\!\xrightarrow[p]{k} c_j \, .$$

Micro-direct connectivity is a path of length zero, by P3.

Since the relation of macro-direct connectivity holds between two components which are micro-directly connected with respect to at least one item of data, we can readily observe that:

P5. A macro-path exists between two components if and only if a micro-path with respect to at least one item of data exists between the same two components.

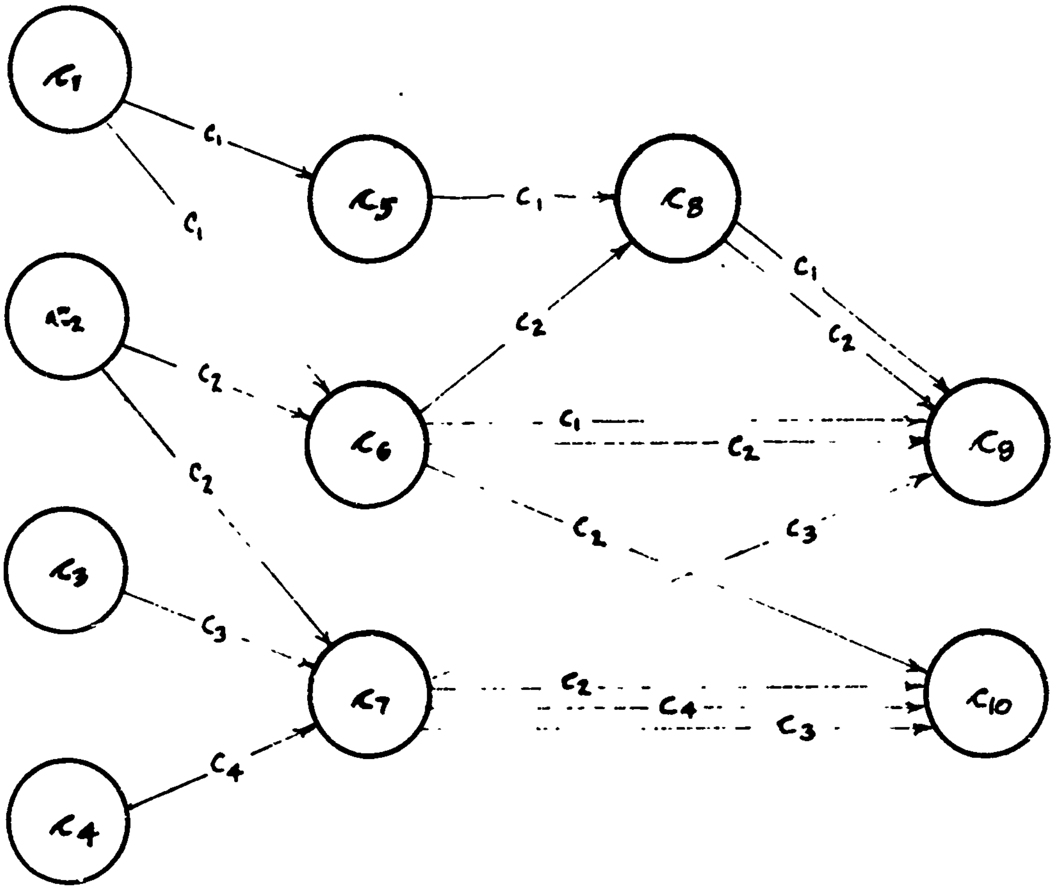
We may now define filtering in a somewhat more precise fashion. We will say that filtering is a system characteristic:

S6. An item of data c_k, is filtered out of the flow of information from component c_i to c_j if there does not exist a micro-direct connectivity relation with respect to c_k from c_i to c_j.

Note that this definition of filtering is worded so that if c_i is not macro-directly connected to c_j (that is, if there is no flow of any information from c_i to c_j) then we can say that total filtering exists between c_i and c_j. Worded this way, we can use the concept of filtering as a characteristic of the macro-model as well as of the micro-model. This enables a systems analyst to change from macro-analysis to micro-analysis, or vice-versa, with a minimum of difficulty, and thereby increases the flexibility of the systems-matrix approach.

As an example, we depict, in Figure 21, the conceptual network associated with a system which is to be subjected to micro-analysis.

Figure 21



	κ_9	κ_{10}
κ_1	2	0
κ_2	2	2
κ_3	1	1
κ_4	0	1

Note, for instance, that while c_7 is macro-directly connected to c_9, it is micro-directly connected only with respect to c_3. At the bottom of Figure 21, we display in a small matrix the total number of micro-paths from each input item to each output. In this simple example, that information may be readily obtained by direct examination of the network. In more complex cases, of course, such direct examination and enumeration would become unwieldy. It is our purpose to apply the systems-matrix approach to generate this information as the solution to the systems-matrix.

3. Transmittance Values and Vectors

The network scheme developed in section 2, above, is a good one pictorially, but is difficult to work with mathematically because of the varying number of arcs that may exist between each pair of components. In the interest of uniformity we now put forth an alternative way of indicating micro-direct connectivity. We will visualize each node representing an intermediate entity to be connected to every other node representing an intermediate entity and to every node representing an output, and that the connection will consist of an aggregation of n_3 arcs, one for each input in the system. On each arc will be a number, or a variable, which denotes whether or not that arc transmits the input item it represents between the nodes it connects. In the case of unconditional filtering, that number will always be either a 1 (transmits) or a 0 (does not transmit). Conditional filtering will be indicated by a variable, which, when evaluated, will be either a 1 or a 0.

More formally:

S7. From every system component c_i, $n_3 < i \le (n_3+n_4)$, to every other system component c_j, $(n_3+1) \le j \le n$, $i \ne j$, there will be a set of k transmittance values, $t_{i,j,k}$, $1 \le k \le n_3$, defined as follows:

$$t_{i,j,k} = \begin{cases} 0; & \text{if } c_k \text{ is never transmitted} \\ & \text{from } c_i \text{ to } c_j, \\ 1\text{-}v_{i,j}; & \text{if } c_k \text{ is not transmitted from } c_i \\ & \text{to } c_j \text{ under certain conditions,} \\ 1; & \text{otherwise} \end{cases}$$

The symbolic value, $1\text{-}v_{i,j}$, will be explained below (section 5 of this chapter).

We may make the following statements about transmittance values:

T1. If $t_{i,j,k} = 1$, c_i is unconditionally micro-directly connected with respect to input c_k to c_j.

T2. If $t_{i,j,k} = 1\text{-}v_{i,j}$, c_i is conditionally micro-directly connected with respect to input c_k to c_j.

T3. If $t_{i,j,k} = 0$, c_i is not micro-directly connected with respect to input c_k to c_j.

T4. c_i is not macro-directly connected to c_j if and only if $t_{i,j,k} = 0$ for all k.

T5. The micro-path of length P with respect to input c_k as specified in P4 exists if and only if:

$$t_{i,r_1,k} \cdot \left[\prod_{i=1}^{\rho-1} t_{r_i,r_{i+1},k} \right] \cdot t_{r_p,j,k} \neq 0 \ .$$

The definition of $t_{i,j,k}$ in S7 is worded so as to remove one cause of ambiguity. It is possible for an input item of data, c_k , to fail to reach c_j either because the flow from c_i to c_j is filtered ($t_{i,j,k} = 0$), or because c_k never reached c_i . In the latter case, as will be shown below (section 5 of this chapter), the value of $t_{i,j,k}$ is arbitrary. We have defined $t_{i,j,k}$ to be independent of whether or not c_k exists in the set of items that reach c_i; we will show that $t_{i,j,k}$ is a function only of the logic of the system structure.

In S7 we have left undefined the micro-direct connectivity of input items to other components. It becomes a little difficult to visualize an aggregation of n_3 arcs eminating from a component which is the component represented by one of the arcs. One might puzzle over the significance of the arc labelled, say, c_5, directed from the node, say, c_4 . Instead of trying to give meaning to this, we will consider the entire set of input items, C_{s3}, as constituting a single component with an aggregation of n_3 arcs directed from this "super-component" to every intermediate entity and to every output. In order not to have to change the specification of the range of the indices in S7, we must label this new component with an index number greater than n_3, but less than (n_3+1). Arbitrarily, we will refer to the component representing the subset C_{s3} as $c_{(n_3+\frac{1}{2})}$.

Obviously, the drawing of a network of even a simple system according to the above scheme would be so complex in appearance as to be a detriment to visualization. We will continue to draw networks as before, but with the understanding that, in micro-models, each arc drawn represents an aggregation of n_3 arcs. We will replace the transmittance value which is connected with each individual arc with a transmittance vector:

S8. The transmittance vector from c_i to c_j ; $n_3 < i \leq (n_3+n_4)$; $(n_3+1) \leq j \leq n$; $i \neq j$; is defined as:

$$T_{i,j} = (t_{i,j,1}, t_{i,j,2}, \dots , t_{i,j,k}, \dots , t_{i,j,n_3}) .$$

We note that we may further simplify the drawing of networks by omitting an arc when its associated transmittance vector is:

$$T_{i,j} = (0,0, \dots, 0).$$

Such will be the case, for example, when c_i is not macro-directly connected to c_j .

We digress momentarily to note that we will make frequent use of vectors that have all elements equal to the same value. We will denote such vectors by:

$$V_z(a) = (a, a, \dots , a) ,$$

where the subscript z states the size of the vector. Thus, in the preceding paragraph:

$$T_{i,j} = (0,0, \dots, 0) = V_{n_3}(0).$$

4. Input Content and Cargo

A remark in section 3, above, indicated that the question of whether or not some intermediate entity has received some input item may be significant. In order to be able to discuss such a question, we introduce additional terms:

S9. The input content of a component, c_i, is stated by a vector, D_i, of size n_3, defined as follows:

$$D_i = (d_{i,1}, d_{i,2}, \dots, d_{i,k}, \dots, d_{i,n_3}); \ (n_3+1) \leq i \leq n$$

where $d_{i,k}$ represents the number of paths by which c_k can reach c_i, $1 \leq k \leq n_3$. We define the input content of $c_{(n_3+\frac{1}{2})}$ as:

$$D_{(n_3+\frac{1}{2})} = V_{n_3}(1) \; .$$

In an unconditional filtered system, each $d_{i,k}$ will be a non-negative integer. In a conditional filtered system, as will be shown below, $d_{i,k}$ may also assume a functional form (which becomes a non-negative integer when the function is evaluated).

S10. The cargo from c_i to c_j is the input content received by c_j from c_i, and is described by a vector of size n_3:

$$R_{i,j} = (r_{i,j,1}, r_{i,j,2}, \dots , r_{i,j,k}, \dots , r_{i,j,n_3}).$$

Obviously, $R_{i,j} = V_{n_3}(0)$ when c_i is not macro-directly connected to c_j.

The cargo from c_i to c_j will be identical to the input content of c_i unless the transfer of one or more input items is prevented by filtering. If this is the case for, say, c_k from c_i to c_j, then it will be indicated by $t_{i,j,k} = 0$, and $r_{i,j,k}$ should be zero. Otherwise, $t_{i,j,k} = 1$, and $r_{i,j,k}$ should be equal to $d_{i,k}$. Therefore:

$$r_{i,j,k} = \left(d_{i,k}\right)\cdot\left(t_{i,j,k}\right)$$

and,

$$R_{i,j} = \left[(d_{i,1})\cdot(t_{i,j,1}),\ \ldots,\ (d_{i,n_3})\cdot(t_{i,j,n_3})\right].$$

(This is in accord with customary usage in signal-flow graph theory; for example, [3, p.187 ff; 6, p.87 ff; 27, Chapter 1; 36, Chapter 1].)

Again, we digress to introduce special notation for a case that will appear frequently. If A and B are two vectors of the same size, z, and if $C = (a_i\cdot b_i,\ a_2\cdot b_2,\ \ldots,\ a_z\cdot b_z)$, we will denote this by:

$$C = A \otimes B\ ,$$

where the operator $\otimes$ indicates element-by-element vector multiplication. Then we may write:

$$R_{i,j} = D_i \otimes T_{i,j} \qquad (5\text{-}1)$$

S11. The input content of c_j, $(n_3+1) \leq j \leq n$, is equal to the sum of all the cargos terminating in c_j:

$$D_j = \sum_i R_{i,j}\ . \qquad (5\text{-}2)$$

Substituting equation (5-1) in equation (5-2):

$$D_j = \sum_i (D_i \otimes T_{i,j}), \qquad (5\text{-}3)$$

and, from S9:

$$D_{(n_3+1)} = V_{n_3}(1). \qquad (5\text{-}4)$$

Without loss of generality, we may stipulate that each of the vectors we have discussed so far be a column vector. Then any D_j for $(n_3+n_4+1) \leq j \leq n$ is a column from the solution area of S^*; i.e., $S^*_{s3,s5}$ (M4, Chapter 4, section 6). Equations (5-3) and (5-4) define the solution area recursively when applied to j such that $(n_3+n_4+1) \leq j \leq n$.

5. Filtering Functions

The determination of a transmittance value (S7) is clear when the system under study contains only unconditional filtering. For a system including conditional filtering, the term $1\text{-}v_{i,j}$ must be determined. We will do this so as to be able to state a single definition for the transmittance value, $t_{i,j,k}$, regardless of the type of filtering present, treating $t_{i,j,k}$ as a function of independent variables to be introduced.

We define a filtering function on each micro-direct connectivity relation in the system. This filtering function is based on two elements, one of which indicates the conditions under which filtering takes place; the other, the input items affected by the filtering action.

S12. The <u>value of the filtering function</u> is defined as:

$$v_{i,j} = \begin{cases} 1; \text{ if filtering takes place between } c_i \text{ and } c_j, \\ 0; \text{ otherwise .} \end{cases}$$

S13. The <u>blocking vector of the filtering function,</u> $B_{i,j}$, describes which input items are filtered out if filtering takes place. This is a vector of size n_3, with the k-th component defined as:

$$b_{i,j,k} = \begin{cases} 1; \text{ if } c_k \text{ is filtered out of the flow} \\ \quad c_i \text{ to } c_j \text{ when } v_{i,j} = 1, \\ 0; \text{ otherwise .} \end{cases}$$

S14. The <u>filtering function,</u> $F_{i,j}$, a vector of size n_3, is defined as the scalar product:

$$F_{i,j} = v_{i,j} \cdot B_{i,j}; \; n_3 < i \leq (n_3+n_4);$$
$$(n_3+1) \leq j \leq n; \; i \neq j .$$

Thus, the k-th element of $F_{i,j}$ is:

$$f_{i,j,k} = \begin{cases} v_{i,j}; \text{ if } b_{i,j,k} = 1, \\ 0; \text{ otherwise,} \end{cases}$$
$$1 \leq k \leq n_3 .$$

The value of $v_{i,j}$ is the truth value of the condition upon which filtering takes place. Suppose, for example, that filtering occurs if and only if the value of c_k is less than some quantity Q. Then we may say:

$$v_{i,j} = (c_k < Q),$$

which is, of course, a logical proposition. For multiple criteria, we may have expressions such as:

$$v_{i,j} = (\ (c_3 < Q) \wedge (c_6 = c_7)\ ),$$

$$v_{i,j} = (\ (c_3 \cdot c_4) \geq (c_7 + c_8)\ ),$$

$$v_{i,j} = (\ (c_1 < Q) \vee (c_2 = 0)\ ),$$

$$v_{i,j} = (((c_2 \neq c_7) \wedge (c_5 \neq 0)\ ) \vee (c_2 = \text{"inbound"})),$$

and so on.

For an example of a blocking vector, consider a system with $n_3 = 7$, and the existence of $F_{12,17}$ such that if $v_{12,17} = 1$, all the information contained in c_{12} is transmitted to c_{17} except for c_4 and c_6. Then:

$$B_{12,17} = (0,0,0,1,0,1,0),$$

and

$$F_{12,17} = (0,0,0,v_{12,17}, 0, v_{12,17}, 0).$$

For unconditional filtering, $v_{i,j} = 1$. For zero filtering, $v_{i,j} = 0$; but in this case we need not recognize the existence of $F_{i,j}$. For total filtering, $B_{i,j} = V_{n_3}(1)$.

More complicated filtering functions may occur. It is possible that the filtering action may take several different forms, depending on the values of several criteria. As an example, consider the following set of blocking vectors, each of which may become the controlling blocking vector between c_i and c_j upon the satisfaction of its associated criterion:

$$B_{i,j} = \begin{cases} (0, 1, 1, 1, 1, 0, 0); & \text{if } c_2 < 0, \\ (0, 0, 1, 1, 0, 1, 0); & \text{if } c_2 = 0, \\ (0, 0, 0, 0, 1, 0, 1); & \text{if } c_2 > 0. \end{cases}$$

We can express each condition as a sub-function, $G_{i,j,h}$; $h = 1, 2, \ldots ,$:

$$G_{i,j,1} = (v_{i,j,1})\cdot(0,1,1,1,1,0,0);\ v_{i,j,1} = (c_2 < 0),$$

$$G_{i,j,2} = (v_{i,j,2})\cdot(0,0,1,1,0,1,0);\ v_{i,j,2} = (c_2 = 0),$$

$$G_{i,j,3} = (v_{i,j,3})\cdot(0,0,0,0,1,0,1);\ v_{i,j,3} = (c_2 > 0).$$

Then:

$$F_{i,j} = \sum_h G_{i,j,h} \qquad (5\text{-}5)$$

In this example,

$$F_{i,j} = (0,\ v_{i,j,1},\ (v_{i,j,1} + v_{i,j,2}),\ (v_{i,j,1} + v_{i,j,2}),$$
$$(v_{i,j,1} + v_{i,j,3}),\ v_{i,j,2},\ v_{i,j,3}).$$

In this example, we were careful to define the criteria such that they were mutually exclusive and exhaustive; i.e.,

$$\sum_h v_{i,j,h} = 1,$$

for all possible values of the criteria. This is an important point in order to avoid ambiguity. In this way no element of $F_{i,j}$ will be anything other than 0 or 1, after the $v_{i,j}$'s have been evaluated.

It is this which permits us to couple the $v_{i,j}$'s by arithmetic operations rather than logical operators.

It is important here to note that the constraints of information flow in the system under study, which in the macro-model resulted in some components being connected, others not, are depicted in the micro-model by filtering functions. Thus, in the model of Chapter 4, we might have a situation with the following characteristics:

a. $n_3 = 4$

b. c_1 is macro-directly connected to c_5

c. c_2 is macro-directly connected to c_5 and c_6

d. c_3 is macro-directly connected to c_6 and c_7

e. c_4 is macro-directly connected to c_6 and c_7 .

In the micro-model of this chapter, the description of these characteristics is:

a. $c_{4\frac{1}{2}}$ is micro-directly connected to c_5, c_6 and c_7, through the filtering functions described below:

b. $F_{4\frac{1}{2},5} = (1)\cdot(0,0,1,1) = (0,0,1,1)$

c. $F_{4\frac{1}{2},6} = (1)\cdot(1,0,0,0) = (1,0,0,0)$

d. $F_{4\frac{1}{2},7} = (1)\cdot(1,1,0,0) = (1,1,0,0)$.

In cases involving the macro-direct connectivity from an intermediate entity to another component, systems design will impose either zero filtering or total filtering; <u>i.e.</u>, either $F_{i,j} = V_{n_3}(0)$ or $F_{i,j} = V_{n_3}(1)$, respectively.

In addition to these system design imposed unconditional filtering functions, there may be one or more conditional filtering functions on an arc. In this case, equation (5-5) governs.

The transmittance of an arc is either 0 or 1; that is, an input item, c_k, is either transmitted from c_i to c_j ($t_{i,j,k} = 1$) or it is not ($t_{i,j,k} = 0$). In order for $t_{i,j,k}$ to be equal to 1, the filtering function, $F_{i,j}$, must be inoperative on c_k. This, in turn, will occur when $v_{i,j} = 0$ and $b_{i,j,k}$ is arbitrary, or when $b_{i,j,k} = 0$ and $v_{i,j}$ is arbitrary. In either case, $f_{i,j,k} = 0$. In order for $t_{i,j,k}$ to be equal to zero, $F_{i,j}$ must be operative on c_k, which means that both $v_{i,j}$ and $b_{i,j,k}$ must be equal to 1, and, therefore, $f_{i,j,k} = 1$. Thus, it can be seen that:

$$t_{i,j,k} = 1 - f_{i,j,k} \,,$$

or,

$$T_{i,j} = V_{n_3}(1) - F_{i,j} \,. \qquad (5\text{-}6)$$

In the case of zero filtering, by definition:

$$v_{i,j} = 0 \,,$$

$$F_{i,j} = v_{i,j} \cdot B_{i,j} = V_{n_3}(0) \,,$$

and, therefore,

$$T_{i,j} = V_{n_3}(1) - F_{i,j} = V_{n_3}(1) \,.$$

In the case of total conditional filtering, by definition:

$$B_{i,j} = V_{n_3}(1) \,,$$

so that, when $v_{i,j} = 1$,

$$F_{i,j} = v_{i,j}{}'B_{i,j} = V_{n_3}(1)$$

and

$$T_{i,j} = V_{n_3}(1) - F_{i,j} = V_{n_3}(0) \ .$$

6. Matrix Representation of a System with Filtering

The matrix S, developed in Chapter 4, has as elements, $s_{i,j}$, the values 1 or 0, depending upon whether macro-direct connectivity does or does not, respectively, exist from c_i to c_j. However, the development of the model in this chapter will not permit such a simple matrix representation, for now we must depict the micro-direct connectivity relation from c_i to c_j, which is an n_3-tuple rather than a single figure.

We could use the previously developed matrix S as a guide for constructing a similar, but larger, matrix to represent the micro-model. Since there are n_3 possible micro-direct connectivity relations between two intermediate entities, the intersection of c_i and c_j in the matrix will be an $(n_3 \times n_3)$ submatrix instead of a scalar. Each row of the submatrix, $c_{i,k}$ will correspond to a single arc leaving c_i, each column, $c_{j,k}$, to a single arc reaching c_j. The intersection of intermediate entities and outputs will also be represented by $(n_3 \times n_3)$ submatrices. Input components will be represented by a set of n_3 rows, as in S, the entire set representing $c_{(n_3+1)}$. The resulting matrix, which we will call SF', will be of order $(n_3+n_3 \cdot n_4) \times (n_3 \cdot n_4+n_3 \cdot n_5)$. The entries in this matrix will be similar to those in S. Let $sf'_{i,j,k}$ be the cell formed by the intersection of the row representing $c_{i,k}$ and the column representing $c_{j,k}$. This cell will be set equal to $t_{i,j,k}$, which represents the k-direct connectivity from c_i to c_j;

unless $i = j$, in which case the cell will be set equal to -1. All other cells will be set equal to 0.

Obviously, each submatrix of SF' will be either Z or an $(n_3 \times n_3)$ diagonal matrix. A cell of a submatrix not on the major diagonal represents the micro-direct connectivity from one input item contained in c_i to a different input item in c_j. No such connectivity exists. Therefore, all cells not on the major diagcnal must be 0. It is known (for example, [9, p.81]) that the product of two square diagonal matrices of the same size is a square diagonal matrix of the same size. Furthermore, the diagonal elements of the product are the products of the corresponding elements of the two diagonal matrices being multiplied.

If X and Y are two diagonal matrices of size $(z \times z)$, then:

$$X \cdot Y = Y \cdot X = W$$

and

$$w_{i,j} = \begin{cases} x_{i,j} \cdot y_{i,j}; & \text{for } i = j, \\ 0; \text{ otherwise.} \end{cases}$$

Let X', Y', and W' be column vectors of size z, formed of the diagonal elements of X, Y and W, respectively:

$$X'_i = X_{i,i}$$

$$Y'_i = Y_{i,i}$$

and

$$W'_i = W_{i,i} \quad ,$$

and let

$$W = X \cdot Y.$$

Then it can be seen that

$$W' = X' \otimes Y' ,$$

where $\otimes$, as before, indicates the element-by-element multiplication of two equal size vectors.

The foregoing permits us to represent each $(n_3 \times n_3)$ submatrix of SF' by a column vector of size n_3, and to represent the operation of ordinary matrix multiplication by the operation $\otimes$. Thus, each submatrix of SF' collapses to $V_{n_3}(0)$, $V_{n_3}(-1)$, or $T_{i,j}$. The collapsed systems-matrix is then of order $(n_3 + n_3 \cdot n_4) \times (n_4 + n_5)$. However, we can consider this as an $(1 + n_4) \times (n_4 + n_5)$ matrix by entering in place of the elements of each column vector the symbol denoting that vector. This new matrix, with vector cell entries, we will call SF, whose elements are defined as:

M6.
$$sf_{i,j} = \begin{cases} V_{n_3}(-1); \text{ if } i = j, \\ T_{i,j}; \text{ otherwise,} \end{cases}$$

and:

$$n_3 < i \leq (n_3 + n_4);\ (n_3 + 1) \leq j \leq n .$$

M7.

$$SF = \begin{array}{r||c|c||}
 & j = (n_3 + 1) \ldots (n_3 + n_4) & j = (n_3 + n_4 + 1) \ldots n \\
\hline
i = (n_3 + \frac{1}{2}) & SF_{s3,s4} & SF_{s3,s5} \\
\hline
i = (n_3 + 1) & & \\
\vdots & SF_{s4,s4} + I(V(-1)) & SF_{s4,s5} \\
(n_3 + n_4) & & \\
\hline
\end{array}$$

where $I(V(-1))$ is a symbol for a diagonal of vectors $V_{n_3}(-1)$.

M8. Every row and every column of SF will contain at least one entry not equal to either $V_{n_3}(0)$ or $V_{n_3}(-1)$.

M9.

$$SF^* = \left[\begin{array}{c|c} SF^*_{s3,s4} & SF^*_{s3,s5} \\ \hline SF^*_{s4,s4} & SF^*_{s4,s5} \end{array}\right]$$

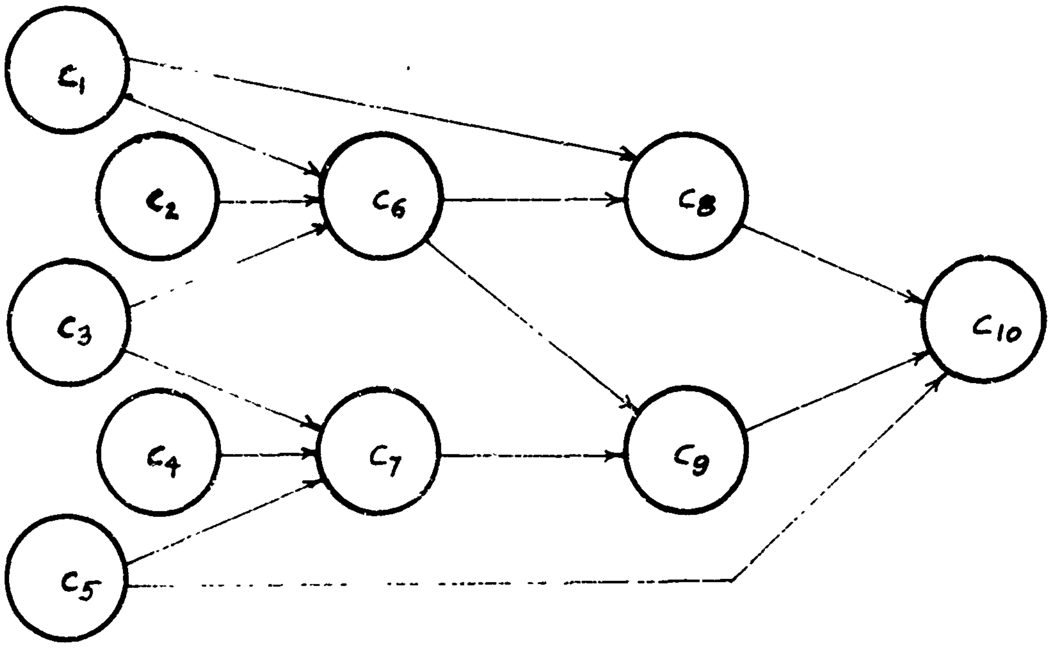
M10.

$$SF^* = \left[\begin{array}{c|c} SF_{s3,s4} & SF^*_{s3,s5} \\ \hline SF_{s4,s4} + I(V(-1)) & Z \end{array}\right]$$

The proof that SF* can be developed by elementary column operations on SF follows the same arguments as those used in Chapter 4, section 7, for the transformation of S* from S.

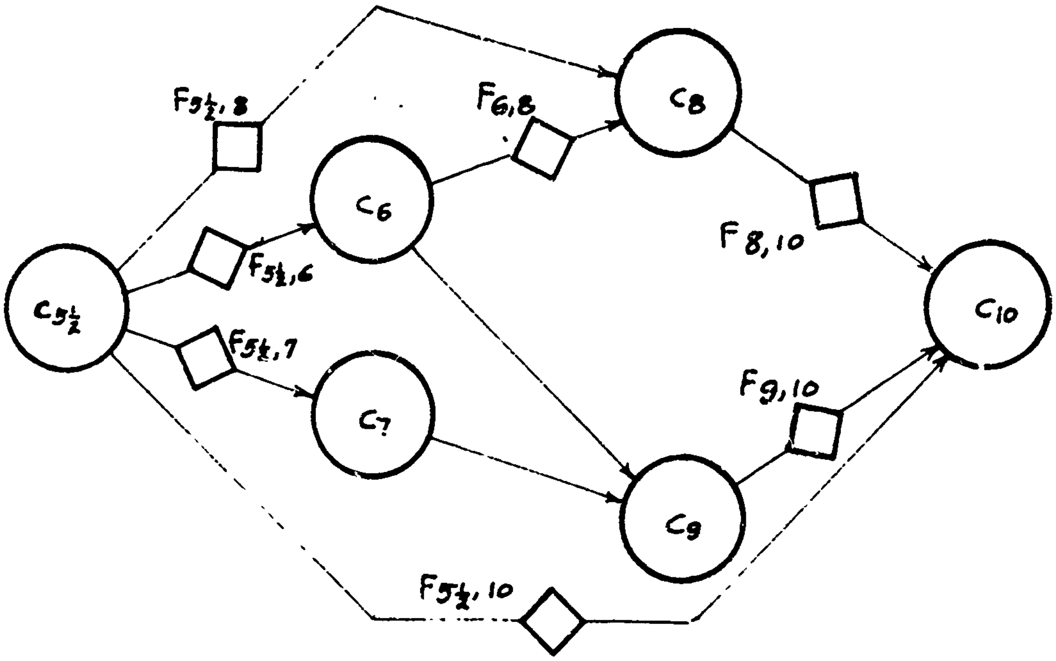
In Figure 22, we depict a simple system with zero filtering and the corresponding matrices S and S* . In Figure 23, we introduce filtering, both conditional and unconditional. In the interest of visual simplicity we have omitted arcs to which we know are associated $T_{i,j}$'s of $V_5(0)$.

Figure 22



$$S = \begin{array}{c||c|c|c|c|c||}
 & c_6 & c_7 & c_8 & c_9 & c_{10} \\ \hline
c_1 & 1 & 0 & 1 & 0 & 0 \\
c_2 & 1 & 0 & 0 & 0 & 0 \\
c_3 & 1 & 1 & 0 & 0 & 0 \\
c_4 & 0 & 1 & 0 & 0 & 0 \\
c_5 & 0 & 1 & 0 & 0 & 1 \\
c_6 & -1 & 0 & 1 & 1 & 0 \\
c_7 & 0 & -1 & 0 & 1 & 0 \\
c_8 & 0 & 0 & -1 & 0 & 1 \\
c_9 & 0 & 0 & 0 & -1 & 1 \\ \hline
\end{array}
\qquad
S^* = \begin{array}{c||c|c|c|c|c||}
 & c_6 & c_7 & c_8 & c_9 & c_{10} \\ \hline
c_1 & 1 & 0 & 1 & 0 & 3 \\
c_2 & 1 & 0 & 0 & 0 & 2 \\
c_3 & 1 & 1 & 0 & 0 & 3 \\
c_4 & 0 & 1 & 0 & 0 & 1 \\
c_5 & 0 & 1 & 0 & 0 & 2 \\
c_6 & -1 & 0 & 1 & 1 & 0 \\
c_7 & 0 & -1 & 0 & 1 & 0 \\
c_8 & 0 & 0 & -1 & 0 & 0 \\
c_9 & 0 & 0 & 0 & -1 & 0 \\ \hline
\end{array}$$

Figure 23



i,j	$v_{i,j}$	$B_{i,j}$	$F_{i,j}$	Explanation
$5\frac{1}{2},6$	(1) $c_3 < 0$	0, 0, 0, 1, 1 0, 0, 1, 0, 0	0, 0, $v_{5\frac{1}{2},6}$, 1, 1	System design If $c_3 < 0$, filter c_3
$5\frac{1}{2},7$	(1) $c_3 = 0$	1, 1, 0, 0, 0 0, 0, 1, 0, 0	1, 1, $v_{5\frac{1}{2},7}$, 0, 0	System design If $c_3 = 0$, filter c_3
$5\frac{1}{2},8$	(1)	0, 1, 1, 1, 1	0, 1, 1, 1, 1	System design
$5\frac{1}{2},10$	(1) $c_3 > 0$	1, 1, 1, 1, 0 1, 1, 1, 1, 1	1, 1, 1, 1, $v_{5\frac{1}{2},10}$	System design If $c_3 > 0$, filter all data
6, 8	$c_1 = c_2$	0, 1, 1, 0, 0	0, $v_{6,8}$, $v_{6,8}$, 0, 0	If $c_1 = c_2$, filter c_2 and c_3
8, 10	$c_1 > c_2$	1, 1, 0, 0, 0	$v_{8,10}$, $v_{8,10}$, 0, 0, 0	If $c_1 > c_2$, filter c_1 and c_2
9, 10	$(c_4 + c_5) < 50$	0, 0, 0, 0, 1	0, 0, 0, 0, $v_{9,10}$	If the sum of c_4 and c_5 is lessthan 50, filter c_5

cont'd

Figure 23 *(cont'd)*

Computed $T_{i,j}$'s

i, j	$T_{i,j}$
$5\frac{1}{2}, 6$	$1, 1, 1-v_{5\frac{1}{2},6}, 0, 0$
$5\frac{1}{2}, 7$	$0, 0, 1-v_{5\frac{1}{2},7}, 1, 1$
$5\frac{1}{2}, 8$	$1, 0, 0, 0, 0$
$5\frac{1}{2}, 10$	$0, 0, 0, 0, 1-v_{5\frac{1}{2},10}$
$6, 8$	$1, 1-v_{6,8}, 1-v_{6,8}, 1, 1$
$8, 10$	$1-v_{8,10}, 1-v_{8,10}, 1, 1, 1$
$9, 10$	$1, 1, 1, 1, 1-v_{9,10}$

NOTES

1. Where no arc appears between nodes, the filtering function may be assumed to be $V_5(1)$; $T_{i,j} = V_5(0)$.

2. Where no filtering function is listed for an arc which does appear, it may be assumed to be $V_5(0)$; $T_{i,j} = V_5(1)$.

3. For $i = j$, $T_{i,j} = V_5(-1)$.

Beneath the drawing we indicate the nature of each filtering function. Cases of arcs without filtering functions listed represent the trivial cases of $F_{i,j} = V_5(0)$. From these $F_{i,j}$'s we compute the $T_{i,j}$'s by equation (5-6). The desired solution is D_{10}, which we compute using equations (5-3) and (5-4):

$$D_6 = D_{5\frac{1}{2}} \otimes T_{5\frac{1}{2},6} = V_5(1) \otimes T_{5\frac{1}{2},6} = T_{5\frac{1}{2},6}$$

$$D_7 = V_5(1) \otimes T_{5\frac{1}{2},7} + D_6 \otimes V_5(0) = T_{5\frac{1}{2},7}$$

$$D_8 = V_5(1) \otimes T_{5\frac{1}{2},8} + D_6 \otimes T_{6,8} + D_7 \otimes V_5(0)$$

$$= T_{5\frac{1}{2},8} + T_{5\frac{1}{2},6} \otimes T_{6,8}$$

$$D_9 = V_5(1) \otimes V_5(0) + D_6 \otimes V_5(1) + D_7 \otimes V_5(1) + D_8 \otimes V_5(0)$$

$$= D_6 + D_7 = T_{5\frac{1}{2},6} + T_{5\frac{1}{2},7}$$

$$D_{10} = V_5(1) \otimes T_{5\frac{1}{2},10} + D_6 \otimes V_5(0) + D_7 \otimes V_5(0)$$

$$+ D_8 \otimes T_{8,10} + D_9 \otimes T_{9,10}$$

$$= T_{5\frac{1}{2},10} + (T_{5\frac{1}{2},8} + T_{5\frac{1}{2},6} \otimes T_{6,8}) \otimes T_{8,10}$$

$$+ (T_{5\frac{1}{2},6} + T_{5\frac{1}{2},7}) \otimes T_{9,10} \,. \qquad (5\text{-}7)$$

Evaluating equation (5-7), which is the desired solution:

$$D_{10} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1-v_{5\frac{1}{2},6} \end{bmatrix} + \left[\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 1-v_{5\frac{1}{2},6} \\ 0 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 1-v_{6,8} \\ 1-v_{6,8} \\ 1 \\ 1 \end{bmatrix} \right] \otimes \begin{bmatrix} 1-v_{8,10} \\ 1-v_{8,10} \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

$$+ \left[\begin{bmatrix} 1 \\ 1 \\ 1-v_{5\frac{1}{2},6} \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1-v_{5\frac{1}{2},7} \\ 1 \\ 1 \end{bmatrix} \right] \otimes \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1-v_{9,10} \end{bmatrix}$$

$$= \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1-v_{5\frac{1}{2},10} \end{bmatrix} + \begin{bmatrix} 2-2v_{8,10} \\ (1-v_{6,8})\cdot(1-v_{8,10}) \\ (1-v_{5\frac{1}{2},6})\cdot(1-v_{6,8}) \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ (1-v_{5\frac{1}{2},6}) + (1-v_{5\frac{1}{2},7}) \\ 1 \\ 1-v_{9,10} \end{bmatrix}$$

$$= \begin{bmatrix} 3-2v_{8,10} \\ 2-v_{6,8}-v_{8,10}+v_{6,8}\cdot v_{8,10} \\ 3-2v_{5\frac{1}{2},6}-v_{5\frac{1}{2},7}-v_{6,8}+v_{5\frac{1}{2},6}\cdot v_{6,8} \\ 1 \\ 2-v_{5\frac{1}{2},10}-v_{9,10} \end{bmatrix} \qquad (5\text{-}8)$$

In Figure 24, we show the matrix SF, corresponding to the network of Figure 23. We show this in two forms, first, symbolically, secondly, with each vector symbol expanded. The second representation is only to aid the reader perform subsequent computations. The solution area is outlined in heavy lines in Figure 24.

We modify the algorithm of Chapter 4, section 7, so that it is applicable to SF:

A4. Find an entry in $SF_{s4,s5}$ which is not all zeros. If there are none, the algorithm is finished, and the desired results are in block $SF_{s3,s5}$. If one is found, call the row in which it occurs r and the column t.

A5. Search row r for an entry of the form $V_{n_3}(-a)$. Call the column in which one occurs d.

A6. Add to column t the element-by-element product of the vector entries in column d and the vector entry in $s_{r,t}$. Go to Step A4.

Applying this algorithm to the matrix SF of Figure 24, we obtain the matrix SF* shown in Figure 25. We note that the block $SF^*_{s3,s5}$ is equivalent to equation (5-7); therefore its evaluation will be identical to equation (5-8). This solution is, of course, largely in functional form, in that it models the number of paths from an input to an output in terms of the logical $v_{i,j}$'s. To depict this in a form useful to the systems analyst, we employ the format of a truth table, evaluating each functional expression in $SF^*_{s3,s5}$ for all possible combinations of values of the $v_{i,j}$'s. Of course, when a numeric value only appears in

Figure 24

$$
SF = \begin{array}{c|c|c|c|c|c}
 & 6 & 7 & 8 & 9 & 10 \\
\hline
5\frac{1}{2} & T_{5\frac{1}{2},6} & T_{5\frac{1}{2},7} & T_{5\frac{1}{2},8} & V_5(0) & \boxed{T_{5\frac{1}{2},10}} \\
\hline
6 & V_5(-1) & V_5(0) & T_{6,8} & V_5(1) & V_5(0) \\
\hline
7 & V_5(0) & V_5(-1) & V_5(0) & V_5(1) & V_5(0) \\
\hline
8 & V_5(0) & V_5(0) & V_5(-1) & V_5(0) & T_{8,10} \\
\hline
9 & V_5(0) & V_5(0) & V_5(0) & V_5(-1) & T_{9,10}
\end{array}
=
\begin{array}{c|c|c|c|c|c}
 & 6 & 7 & 8 & 9 & 10 \\
\hline
5\frac{1}{2} & \begin{matrix}1\\1\\1-v_{5\frac{1}{2},6}\\0\\0\end{matrix} & \begin{matrix}0\\0\\1-v_{5\frac{1}{2},7}\\1\\1\end{matrix} & \begin{matrix}1\\0\\0\\0\\0\end{matrix} & \begin{matrix}0\\0\\0\\0\\0\end{matrix} & \boxed{\begin{matrix}0\\0\\0\\0\\1-v_{5\frac{1}{2},10}\end{matrix}} \\
\hline
6 & \begin{matrix}-1\\-1\\-1\\-1\\-1\end{matrix} & \begin{matrix}0\\0\\0\\0\\0\end{matrix} & \begin{matrix}1\\1-v_{6,8}\\1-v_{6,8}\\1\\1\end{matrix} & \begin{matrix}1\\1\\1\\1\\1\end{matrix} & \begin{matrix}0\\0\\0\\0\\0\end{matrix} \\
\hline
7 & \begin{matrix}0\\0\\0\\0\\0\end{matrix} & \begin{matrix}-1\\-1\\-1\\-1\\-1\end{matrix} & \begin{matrix}0\\0\\0\\0\\0\end{matrix} & \begin{matrix}1\\1\\1\\1\\1\end{matrix} & \begin{matrix}0\\0\\0\\0\\0\end{matrix} \\
\hline
8 & \begin{matrix}0\\0\\0\\0\\0\end{matrix} & \begin{matrix}0\\0\\0\\0\\0\end{matrix} & \begin{matrix}-1\\-1\\-1\\-1\\-1\end{matrix} & \begin{matrix}0\\0\\0\\0\\0\end{matrix} & \begin{matrix}1-v_{8,10}\\1-v_{8,10}\\1\\1\\1\end{matrix} \\
\hline
9 & \begin{matrix}0\\0\\0\\0\\0\end{matrix} & \begin{matrix}0\\0\\0\\0\\0\end{matrix} & \begin{matrix}0\\0\\0\\0\\0\end{matrix} & \begin{matrix}-1\\-1\\-1\\-1\\-1\end{matrix} & \begin{matrix}1\\1\\1\\1\\1-v_{9,10}\end{matrix}
\end{array}
$$

Figure 25

$$SF^{*} = \begin{array}{c|c|c|c|c|c} & 6 & 7 & 8 & 9 & 10 \\ \hline 5\frac{1}{2} & T_{5\frac{1}{2},6} & T_{5\frac{1}{2},7} & T_{5\frac{1}{2},8} & V_5(0) & \boxed{\begin{array}{r} T_{5\frac{1}{2},10}+T_{5\frac{1}{2},8}\otimes T_{8,10}+T_{5\frac{1}{2},7}\otimes T_{9,10} \\ +T_{5\frac{1}{2},6}\otimes(T_{9,10}+T_{6,8}\otimes T_{8,10}) \end{array}} \\ \hline 6 & V_5(-1) & V_5(0) & T_{6,8} & V_5(1) & V_5\ (0) \\ \hline 7 & V_5(0) & V_5(-1) & V_5(0) & V_5(1) & V_5\,(0) \\ \hline 8 & V_5(0) & V_5(0) & V_5(-1) & V_5(0) & V_5\ (0) \\ \hline 9 & V_5(0) & V_5(0) & V_5(0) & V_5(-1) & V_5\ (0) \\ \hline \end{array}$$

$$SF^{*}_{53,55} = \begin{bmatrix} 3-2\,v_{8,10} \\ 2-v_{6,8}-v_{8,10}+v_{6,8}\cdot v_{8,10} \\ 3-2v_{5\frac{1}{2},6}-v_{5\frac{1}{2},7}-v_{6,8}+v_{5\frac{1}{2},6}\cdot v_{6,8} \\ 1 \\ 2-v_{5\frac{1}{2},10}-v_{9,10} \end{bmatrix} = D_{10}$$

a cell of the solution area, that indicates a deterministic number of paths, independent of any conditional filtering action. In Figure 26 we display the truth table expansion of the solution shown in Figure 25, showing the values of each element of D_{10} for every possible combination of values of the logical $v_{i,j}$'s.

7. Path Determination

We can determine the paths by which an input reaches an output for a system with filtering in the same way as we did for the macro-model of Chapter 4. We form the matrix SFP in the same way as SF, except that:

$$spf_{i,i} = V_{n_3}(-1/c_i)$$

instead of $V_{n_3}(-1)$. We also impose the non-commutative restriction on the operation $\otimes$. Applying this method to the system in Figure 23, we arrive at the SFP shown in Figure 27, and the SFP* and evaluation of D_{10} shown in Figure 28. In Figure 29 we display the truth table expansion of D_{10}, but in a different form than was used in Figure 26. In Figure 29 we expand each element of D_{10} in a separate table, using only those $v_{i,j}$'s which affect that element. The result is five tables instead of one, but each is easier to interpret. Note the further simplification through the use of "else" columns and "don't care" entries (-).

Figure 26

$V_{5\frac{1}{2},6}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$V_{5\frac{1}{2},7}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
$V_{5\frac{1}{2},10}$	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
$V_{6,8}$	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
$V_{8,10}$	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
$V_{9,10}$	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
$d_{10,1}$	3	3	1	1	3	3	1	1	3	3	1	1	3	3	1	1	3	3	1	1	3	3	1	1	3	3	1	1	3	3	1	1
$d_{10,2}$	2	2	1	1	1	1	1	1	2	2	1	1	1	1	1	1	2	2	1	1	1	1	1	1	2	2	1	1	1	1	1	1
$d_{10,3}$	3	3	3	3	2	2	2	2	3	3	3	3	2	2	2	2	2	2	2	2	1	1	1	1	2	2	2	2	1	1	1	1
$d_{10,5}$	2	1	2	1	2	1	2	1	1	0	1	0	1	0	1	0	2	1	2	1	2	1	2	1	1	0	1	0	1	0	1	0

$V_{5\frac{1}{2},6}$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
$V_{5\frac{1}{2},7}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
$V_{5\frac{1}{2},10}$	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
$V_{6,8}$	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
$V_{8,10}$	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
$V_{9,10}$	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
$d_{10,1}$	3	3	1	1	3	3	1	1	3	3	1	1	3	3	1	1	3	3	1	1	3	3	1	1	3	3	1	1	3	3	1	1
$d_{10,2}$	2	2	1	1	1	1	1	1	2	2	1	1	1	1	1	1	2	2	1	1	1	1	1	1	2	2	1	1	1	1	1	1
$d_{10,3}$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$d_{10,5}$	2	1	2	1	2	1	2	1	1	0	1	0	1	0	1	0	2	1	2	1	2	1	2	1	1	0	1	0	1	0	1	0

Figure 27

$$
SFP = \begin{array}{c||c|c|c|c|c||}
 & 6 & 7 & 8 & 9 & 10 \\
\hline
5\frac{1}{2} & T_{5\frac{1}{2},6} & T_{5\frac{1}{2},7} & T_{5\frac{1}{2},8} & V_5(0) & \boxed{T_{5\frac{1}{2},10}} \\
\hline
6 & V_5(-\frac{1}{C_6}) & V_5(0) & T_{6,8} & V_5(1) & V_5(0) \\
\hline
7 & V_5(0) & V_5(-\frac{1}{C_7}) & V_5(0) & V_5(1) & V_5(0) \\
\hline
8 & V_5(0) & V_5(0) & V_5(-\frac{1}{C_8}) & V_5(0) & T_{8,10} \\
\hline
9 & V_5(0) & V_5(0) & V_5(0) & V_5(-\frac{1}{C_9}) & T_{9,10} \\
\hline\hline
\end{array}
$$

Figure 28

$$SFP^* = \begin{array}{c|c|c|c|c|c} & 6 & 7 & 8 & 9 & 10 \\ \hline 5\frac{1}{2} & T_{5\frac{1}{2},6} & T_{5\frac{1}{2},7} & T_{5\frac{1}{2},8} & V_5(0) & \boxed{D_{10}\ \textit{(See below)}} \\ \hline 6 & V_5(-\frac{1}{c_6}) & V_5(0) & T_{6,8} & V_5(1) & V_5(0) \\ \hline 7 & V_5(0) & V_5(-\frac{1}{c_7}) & V_5(0) & V_5(1) & V_5(0) \\ \hline 8 & V_5(0) & V_5(0) & V_5(-\frac{1}{c_8}) & V_5(0) & V_5(0) \\ \hline 9 & V_5(0) & V_5(0) & V_5(0) & V_5(-\frac{1}{c_9}) & V_5(0) \end{array}$$

$$\begin{aligned} D_{10} = {} & T_{5\frac{1}{2}.10} + T_{5\frac{1}{2},8} \otimes V_5(c_8) \otimes T_{8,10} + T_{5\frac{1}{2},7} \otimes V_5(c_7) \otimes V_5(c_9) \otimes T_{9,10} \\ & + T_{5\frac{1}{2},6} \otimes V_5(c_6) \otimes [V_5(c_9) \otimes T_{9,10} + T_{6,8} \otimes V_5(c_8) \otimes T_{8,10}] \end{aligned}$$

$$= \begin{bmatrix} c_8(1-v_{8,10}) + c_6[c_9 + c_8(1-v_{8,10})] \\ c_6[c_9 + (1-v_{6,8})\, c_8(1-v_{8,10})] \\ (1-v_{5\frac{1}{2},7})c_7 + (1-v_{5\frac{1}{2},6})c_6[c_9 + (1-v_{6,8})c_8] \\ c_7 \cdot c_9 \\ (1-v_{5\frac{1}{2},10}) + c_7(1-v_{9,10}) \end{bmatrix}$$

Figure 29

a)

$v_{8,10}$	0	1
$d_{10,1}$	$c_8 + c_6(c_9+c_8)$	$c_6 \cdot c_9$

b)

$v_{6,8}$	0	else
$v_{8,10}$	0	
$d_{10,2}$	$c_6(c_9+c_8)$	$c_6 \cdot c_9$

c)

$v_{5\frac{1}{2},6}$	0	0	0	0	1	1
$v_{5\frac{1}{2},7}$	0	0	1	1	0	1
$v_{6,8}$	0	1	0	1	-	-
$d_{10,3}$	$c_7+c_6(c_9+c_8)$	$c_7 + c_6 \cdot c_9$	$c_6(c_9+c_8)$	$c_6 \cdot c_9$	c_7	0

d)

$d_{10,4}$: $c_7 \cdot c_9$ (independent of any $v_{i,j}$'s)

e)

$v_{5\frac{1}{2},10}$	0	0	1	1
$v_{9,10}$	0	1	0	1
$d_{10,5}$	$1 + c_7$	1	c_7	0

CHAPTER 6

INTERNALLY GENERATED DATA

1. Sources of Data

In the models proposed so far, we have considered data as being information coming from outside the system under consideration, and we have referred to such data as input items. However, data may also be generated within a system. It is the purpose of this chapter to explore this phenomenon and to incorporate it into the systems-matrix model.

Data which is generated outside the scope of the system being modelled will be called external data; that which is generated by the system, internal data. Internal data may be of two types:

a. meta-data, data about data, or data about the system;

b. resultant data, data which is formed by operations on other data.

Some examples may help clarify the use of these terms.

Consider a system set up to process invoices. Inputs to the system are data regarding shipments to customers, selling process of commodities, and discounts to various classes of trade. Outputs are invoices and various management reports. All input items are, obviously, external data. If one of the management reports contains the number of typing errors per typist, that information is internal meta-data. If the external data includes the quantity shipped and the unit price of a commodity, the extended price on the invoice (arrived at by multiplying the two specified input items) is internal resultant data.

For our purposes in the model proposed, all meta-data may be treated as input items, since we do not require that input be limited to any one "level". Internal resultant data must be further discussed.

2. Processes for Generating Internal Resultant Data

Two types of internal resultant data processes must be recognized:

a. an additional resultant data process is a process which generates one or more new items of data from data received by the process, and transmits both the new items and the items from which the resultant data were generated.

b. a replacement resultant data process is a process which generates one or more new items of data from data received by the process, and transmits the new items but not the items from which the resultant data were generated.

Thus, an additional process furnishes to some direct connected component both the resultant data and the data which generated the resultant data. It suffices, as a matter of fact, that only enough of the original data be transmitted to permit the reconstitution of all of the original data by means of an operation which is the inverse of the operation which formed the resultant data. An example might be a process which receives an "old" inventory balance (OB) and the quantity received into stock (R); computes the new inventory balance (NB); and transmits any one of the following sets of data items:

$\{OB,R,NB\}$,

$\{OB,NB\}$,

$\{R,NB\}$.

A replacement process does not transmit enough information for the original data to be reconstituted. In the above example, a replacement process would transmit NB only. Perhaps a better example is a process which receives all invoices and transmits only the number of invoices and the average dollar amount per invoice.

3. Resultant Data Components in the Model

The problem introduced by the recognition of resultant data is twofold. First, we must assure ourselves that if the generating process is a replacement process, our model automatically "blocks" the flow of the data which was used to compute the resultant data. Second, regardless of the type of generating process, we should, in some way, indicate that when an item of resultant data is received by another component in the system, it carries with it some information about the data from which it was generated, even though it may not carry that information in a way which will permit reconstitution of the original data. We propose to incorporate this phenomenon into our systems-matrix model by a single mechanism regardless of whether a process is additional or replacement.

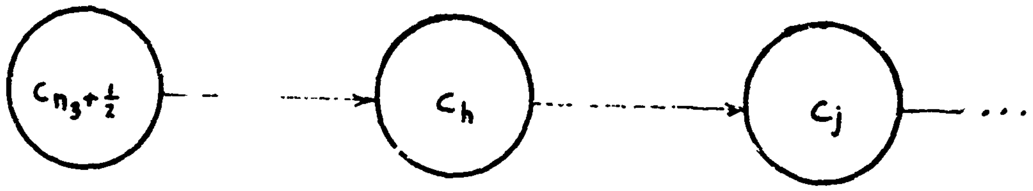
Let c_h be a component of C_s which is an item of data formed within the system by a combination of other data elements; i.e., an item of internal resultant data. We will treat this component as if it had been produced by a replacement process, for an item of resultant data produced by an additional process may be depicted as the result of two processes, one a replacement process, the other a direct connectivity process that transmits the original information to the same component that receives the new data item. This decomposition, incidentally, gives us the additional flexibility of easily depicting partial additional processes, by inserting a filtering function on the flow of the original information. In Figure 30, we depict, as networks, both replacement and additional processes viewed in this way.

This resultant data component, c_h, is an intermediate entity with respect to the input components from which it is generated. However, from the viewpoint of the components to which c_h is directly connected, it is an input. Since such ambiguity cannot be allowed to persist, we will classify resultant data as secondary inputs, calling externally generated components primary inputs. Thus, the components called simply "inputs" in previous chapters will now be called "primary inputs". We will assume that a system contains n_1 items of primary inputs and n_2 items of secondary inputs, and that

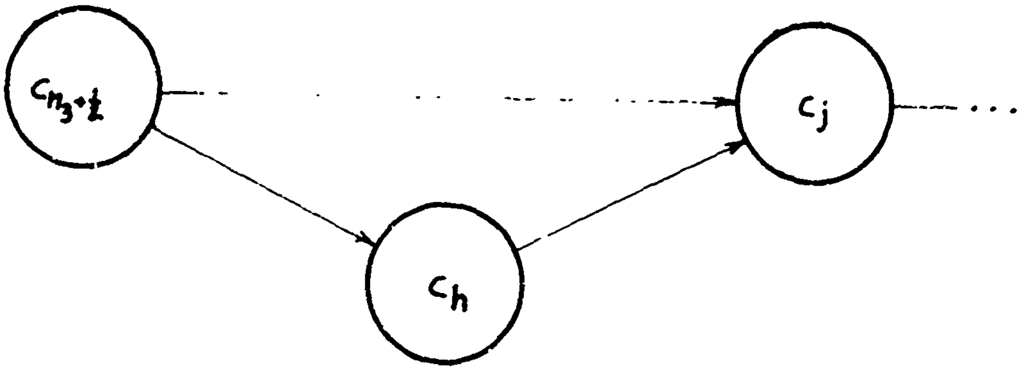
$$n_1 + n_2 = n_3 ,$$

where, as before, n_3 represents the total number of inputs to a system.

Figure 30



a) Replacement Resultant Data Process



b) Additional Resultant Data Process

4. Macro-Model with Resultant Data

Our first model incorporating internally generated data will be a macro-model; i.e., a model without filtering, and, therefore, without consideration of individual data item flow between components. We will modify the model of Chapter 4 to accommodate secondary input items. By extension of previous usage, we use the following additional subset notation:

S15. primary inputs: $C_{s1} = \left\{ c_s \middle| 1 \le s \le n_1 \right\}$

S16. secondary inputs: $C_{s2} = \left\{ c_s \middle| (n_1 + 1) \le s \le (n_1 + n_2) = n_3 \right\}$.

S17. $C_{s3} = C_{s1} \cup C_{s2}$.

S18. $C_{s1} \cap C_{s2} = \phi$.

Note that the above, taken with S1, S2, and S3, imposes on the indexing of components the following restriction: if $c_i \in C_{s1}$, $c_h \in C_{s2}$, $c_j \in C_{s4}$ and $c_k \in C_{s5}$, then:

$$1 \le i < h < j < k \le n .$$

In our new model, direct connectivity is permitted, in addition to C1, C2, C3, and C4, between the following sets:

Direct Connectivity

	From	To
C9.	C_{s1}	C_{s2}
C10.	C_{s2}	C_{s2}
C11.	C_{s4}	C_{s2}

Statement C11 implies that we can no longer stipulate C5. Instead, we impose:

C5a. Direct connectivity may not exist in such a fashion as to form a loop or a circuit; that is, no path may include the same component more than once.

Note also that S17 may be applied to C1 and C2 to yield:

	Direct Connectivity	
	From	To
C1a.	C_{s1}	C_{s4}
C1b.	C_{s2}	C_{s4}
C2a.	C_{s1}	C_{s5}
C2b.	C_{s2}	C_{s5}

The problem involved in depicting the flow of data in a management information system with internal data generated by a replacement process is one of blocking paths from primary inputs to secondary inputs to outputs, yet in some way indicating that intelligence based upon certain primary inputs is included in the flow along the path from secondary inputs to outputs. We elect to use a scheme similar to the SP matrix of Chapter 4, section 9. We will describe a matrix SR, which will employ a mixed notational scheme; that is, a combination of non-negative integers and symbols. An element of the solution area, $SR^*_{s1,s5}$, might contain the entry:

$$sr_{i,j} = \alpha + \beta_{c_h} \gamma ,$$

where:

$$1 \leq i \leq n_1,$$
$$(n_1+1) \leq h \leq n_3,$$
$$(n_3+n_4+1) \leq j \leq n \ ,$$

and where α, β, and γ are non-negative integers. This expression may be interpreted as:

a. c_i reaches c_j in its original form through α paths.

b. c_i reaches c_h through β paths.

c. c_h reaches c_j (and, thereby, intelligence about c_i reaches c_j) through γ paths.

As usual, if α is 0, it may be omitted; if either β or γ are 1, they may be omitted; and if either β or γ are 0, the entire term containing c_h may be omitted. (If all terms are 0, an entry of a single 0 is indicated.) The γ arises through application of the non-commutative multiplication process; the β through the application of the rule: $c_h + c_h = 2c_h$.

Since secondary inputs may form paths, an expression such as:

$$sr_{i,j} = c_{h_1}(\alpha_1 + \beta_1 c_{h_2}\ \gamma + c_{h_3}(\alpha_2 + \beta_2 c_{h_2}\gamma))$$

may occur. This would be interpreted:

a. c_i is connected to c_{h_1} through one path,

b. c_{h_1} is connected to c_j through α_1 paths, to c_{h_2} through β_1 paths, and to c_{h_3} through one path,

c. c_{h_2} is connected to c_j through γ paths,

d. c_{h_3} is connected to c_j through α_2 paths, and to c_{h_2} through β_2 paths.

The system with secondary inputs may be represented by a matrix, SR, of order $(n_3+n_4) \times (n_2+n_4+n_5)$, whose elements are defined as:

M11.
$$sr_{i,j} = \begin{cases} 1; \text{ if } c_i \overset{\oplus}{\underset{0}{\rightarrow}} c_j;\ i \neq j, \\ -1; \text{ if } (n_3+1) \leq i = j \leq (n_3+n_4), \\ -(1/c_i); \text{ if } (n_1+1) \leq i = j \leq n_2\ , \\ 0; \text{ otherwise} \end{cases}$$

and:

$$1 \leq i \leq (n_3+n_4),$$
$$(n_1+1) \leq j \leq n\ .$$

Each row of SR is labelled c_i; and each column, c_j. Thus, all primary inputs are represented by rows; all outputs by columns; and all secondary inputs and intermediate entities by both rows and columns. We may decompose the matrix SR into blocks:

M12.
$$SR = \begin{array}{c|c|c|c} & j=(n_1+1)\ldots n_3 & j=(n_3+1)\ldots(n_3+n_4) & j=(n_3+n_4+1)\ldots n \\ \hline i = 1 \\ \vdots \\ n_1 & SR_{s1,s2} & SR_{s1,s4} & SR_{s1,s5} \\ \hline i = (n_1+1) \\ \vdots \\ n_3 & SR_{s2,s2}-I(C) & SR_{s2,s4} & SR_{s2,s5} \\ \hline i = (n_3+1) \\ \vdots \\ (n_3+n_4) & SR_{s4,s2} & SR_{s4,s4}-I & SR_{s4,s5} \\ \hline \end{array}$$

where $I(C)$ is a symbol for the diagonal matrix with entries of $1/c_i$.

Obviously:

M13. Every row and every column of SR will contain at least one entry of +1.

Both a partial solution matrix, SR_*, and a solution matrix, SR^*, are recognized. Following the development of Chapter 4:

M14.

$$SR_* = \left[\begin{array}{cc|c} SR_{s1,s2} & SR_{s1,s4} & SR_{*s1,s5} \\ \hline SR_{s2,s2}-I(C) & SR_{s2,s4} & SR_{*s2,s5} \\ \hline SR_{s4,s2} & SR_{s4,s4}-I & Z \end{array}\right]$$

M15.

$$SR^* = \left[\begin{array}{cc|c} SR_{s1,s2} & SR_{s1,s4} & SR^*_{s1,s5} \\ \hline SR_{s2,s2}-I(C) & SR_{s2,s4} & Z \\ \hline SR_{s4,s2} & SR_{s4,s4}-I & Z \end{array}\right]$$

The interpretation of the two solutions are as follows:

a. SR_* displays in $SR_{*s2,s5}$ the number of paths through which each secondary input reaches each output; and, in $SR_{*s1,s5}$, the number of paths through which each primary input reaches each output in its original form.

b. SR^* displays in $SR^*_{s1,s5}$, in the previously described mixed notational scheme, the number of paths through which primary inputs reach each output.

It can be readily seen that SR is, in a sense, a hybrid of S and SP. The algorithm for manipulating S can be applied to SR to transform it into SR_*; then the algorithm for manipulating SP can be applied to SR_* to transform it into SR^*. Thus, we are applying the path-finding technique of Chapter 4, section 9, to the problem of identification of primary inputs replaced by secondary inputs.

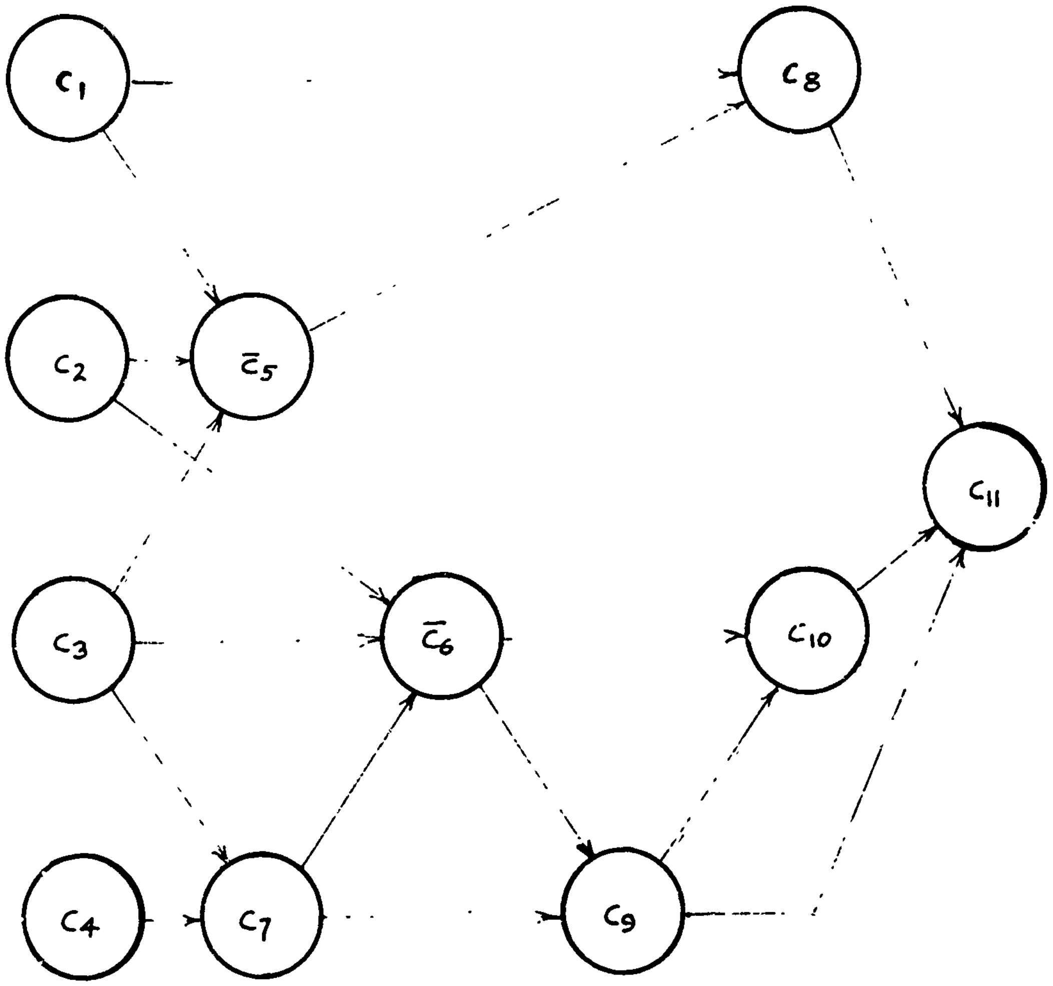
Figure 31 is a network drawing of a macro-model of a system which has four primary inputs and two secondary inputs. The matrix corresponding to this network, SR, is shown in Figure 32. The partial solution matrix, SR_*, is shown in Figure 33, and the solution matrix, SR*, in Figure 34. Note that we have employed the device of indicating secondary inputs by a bar over the component symbol: $\bar{c}_h$; $(n_1+1) \leq h \leq (n_1+n_2)$. This is a convention of convenience, not necessity, since the indexing of the components is unique; it simply permits rapid identification of secondary inputs.

It is quite obvious that we can employ the method of Chapter 4, section 9, to generate the identity of each path. To do this requires that we set up a systems-matrix, SRP, identical to SR, except that wherever a -1 appears in row i of SR, we set:

$$srp_{i,i} = -(1/c_i) \ .$$

For the system illustrated by the network of Figure 30, Figure 35 depicts SRP; Figure 36, SRP_*; and Figure 37, SRP* together with simplified equivalent expressions for the solution.

Figure 31



$n_1 = 4$	$c_{s1} = \{ c_1, c_2, c_3, c_4 \}$
$n_2 = 2$	$c_{s2} = \{ \bar{c}_5, \bar{c}_6 \}$
$n_3 = 6$	$c_{s3} = \{ c_1, c_2, c_3, c_4, \bar{c}_5, \bar{c}_6 \}$
$n_4 = 4$	$c_{s4} = \{ c_7, c_8, c_9, c_{10} \}$
$n_5 = 1$	$c_{s5} = \{ c_{11} \}$

Figure 32

$$SR = \begin{array}{c|ccccccc}
 & \bar{c}_5 & \bar{c}_6 & c_7 & c_8 & c_9 & c_{10} & c_{11} \\
\hline
c_1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
c_2 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
c_3 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
c_4 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
\bar{c}_5 & -\frac{1}{\bar{c}_5} & 0 & 0 & 1 & 0 & 0 & 0 \\
\bar{c}_6 & 0 & -\frac{1}{\bar{c}_6} & 0 & 0 & 1 & 1 & 0 \\
c_7 & 0 & 1 & -1 & 0 & 1 & 0 & 0 \\
c_8 & 0 & 0 & 0 & -1 & 0 & 0 & 1 \\
c_9 & 0 & 0 & 0 & 0 & -1 & 1 & 1 \\
c_{10} & 0 & 0 & 0 & 0 & 0 & -1 & 1
\end{array}$$

Figure 33

$$SR_* = \begin{array}{c||c|c|c|c|c|c|c||}
 & \bar{c}_5 & \bar{c}_6 & c_7 & c_8 & c_9 & c_{10} & c_{11} \\
\hline
c_1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\
\hline
c_2 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
\hline
c_3 & 1 & 1 & 1 & 0 & 0 & 0 & 2 \\
\hline
c_4 & 0 & 0 & 1 & 0 & 0 & 0 & 2 \\
\hline
\bar{c}_5 & -\frac{1}{\bar{c}_5} & 0 & 0 & 1 & 0 & 0 & 1 \\
\hline
\bar{c}_6 & 0 & -\frac{1}{\bar{c}_6} & 0 & 0 & 1 & 1 & 3 \\
\hline
c_7 & 0 & 1 & -1 & 0 & 1 & 0 & 0 \\
\hline
c_8 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\
\hline
c_9 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\
\hline
c_{10} & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\
\hline
\end{array}$$

Figure 34

$$5R^{*} = \begin{array}{c||cccccc|c||}
 & \bar{c}_5 & \bar{c}_6 & c_7 & c_8 & c_9 & c_{10} & c_{11} \\
\hline
c_1 & 1 & 0 & 0 & 1 & 0 & 0 & 1+\bar{c}_5 \\
c_2 & 1 & 1 & 0 & 0 & 0 & 0 & \bar{c}_6 \cdot 3 + \bar{c}_5 \\
c_3 & 1 & 1 & 1 & 0 & 0 & 0 & 2 + 2 \cdot \bar{c}_6 \cdot 3 + \bar{c}_5 \\
c_4 & 0 & 0 & 1 & 0 & 0 & 0 & 2 + \bar{c}_6 \cdot 3 \\
\bar{c}_5 & -\frac{1}{\bar{c}_5} & 0 & 0 & 1 & 0 & 0 & 0 \\
\bar{c}_6 & 0 & -\frac{1}{\bar{c}_6} & 0 & 0 & 1 & 1 & 0 \\
c_7 & 0 & 1 & -1 & 0 & 1 & 0 & 0 \\
c_8 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\
c_9 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\
c_{10} & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\
\hline
\end{array}$$

Figure 35

$$SRP = \begin{array}{c||c|c|c|c|c|c|c||}
 & \bar{c}_5 & \bar{c}_6 & c_7 & c_8 & c_9 & c_{10} & c_{11} \\
\hline\hline
c_1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
\hline
c_2 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
\hline
c_3 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
\hline
c_4 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
\hline
\bar{c}_5 & -\frac{1}{\bar{c}_5} & 0 & 0 & 1 & 0 & 0 & 0 \\
\hline
\bar{c}_6 & 0 & -\frac{1}{\bar{c}_6} & 0 & 0 & 1 & 1 & 0 \\
\hline
c_7 & 0 & 1 & -\frac{1}{c_7} & 0 & 1 & 0 & 0 \\
\hline
c_8 & 0 & 0 & 0 & -\frac{1}{c_8} & 0 & 0 & 1 \\
\hline
c_9 & 0 & 0 & 0 & 0 & -\frac{1}{c_9} & 1 & 1 \\
\hline
c_{10} & 0 & 0 & 0 & 0 & 0 & -\frac{1}{c_{10}} & 1 \\
\hline\hline
\end{array}$$

Figure 36

$$
SRP_{*} = \begin{array}{c||c|c|c|c|c|c|c||}
 & \bar{c}_5 & \bar{c}_6 & c_7 & c_8 & c_9 & c_{10} & c_{11} \\
\hline\hline
c_1 & 1 & 0 & 0 & 1 & 0 & 0 & c_8 \\
\hline
c_2 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
\hline
c_3 & 1 & 1 & 1 & 0 & 0 & 0 & c_7 c_9 (1+c_{10}) \\
\hline
c_4 & 0 & 0 & 1 & 0 & 0 & 0 & c_7 \cdot c_9 (1+c_{10}) \\
\hline
\bar{c}_5 & -\frac{1}{\bar{c}_5} & 0 & 0 & 1 & 0 & 0 & c_8 \\
\hline
\bar{c}_6 & 0 & -\frac{1}{\bar{c}_6} & 0 & 0 & 1 & 1 & c_{10} + c_9 (1+c_{10}) \\
\hline
c_7 & 0 & 1 & -\frac{1}{c_7} & 0 & 1 & 0 & 0 \\
\hline
c_8 & 0 & 0 & 0 & -\frac{1}{c_8} & 0 & 0 & 0 \\
\hline
c_9 & 0 & 0 & 0 & 0 & -\frac{1}{c_9} & 1 & 0 \\
\hline
c_{10} & 0 & 0 & 0 & 0 & 0 & -\frac{1}{c_{10}} & 0 \\
\hline\hline
\end{array}
$$

Figure 37

$$
SRP^{*} = \begin{array}{c||c|c|c|c|c|c|c}
 & \bar{c}_5 & \bar{c}_6 & c_7 & c_8 & c_9 & c_{10} & c_{11} \\
\hline
c_1 & 1 & 0 & 0 & 1 & 0 & 0 & c_8 + \bar{c}_5 \cdot c_8 \\
\hline
c_2 & 1 & 1 & 0 & 0 & 0 & 0 & \bar{c}_6 (c_{10} + c_9(1 + c_{10})) + \bar{c}_5 \cdot c_8 \\
\hline
c_3 & 1 & 1 & 1 & 0 & 0 & 0 & c_7 \cdot c_9(1 + c_{10}) + \bar{c}_6 (c_{10} + c_9(1 + c_{10})) + c_7 \bar{c}_6 (c_{10} + c_9(1 + c_{10})) + \bar{c}_5 c_8 \\
\hline
c_4 & 0 & 0 & 1 & 0 & 0 & 0 & c_7 c_9 (1 + c_{10}) + c_7 \bar{c}_6 (c_{10} + c_9(1 + c_{10})) \\
\hline
\bar{c}_5 & -\frac{1}{\bar{c}_5} & 0 & 0 & 1 & 0 & 0 & 0 \\
\hline
\bar{c}_6 & 0 & -\frac{1}{\bar{c}_6} & 0 & 0 & 1 & 1 & 0 \\
\hline
c_7 & 0 & 1 & -\frac{1}{c_7} & 0 & 1 & 0 & 0 \\
\hline
c_8 & 0 & 0 & 0 & -\frac{1}{c_8} & 0 & 0 & 0 \\
\hline
c_9 & 0 & 0 & 0 & 0 & -\frac{1}{c_9} & 1 & 0 \\
\hline
c_{10} & 0 & 0 & 0 & 0 & 0 & -\frac{1}{c_{10}} & 0
\end{array}
$$

Simplified Expressions for the Solution

c_1 : $(1 + \bar{c}_5) \cdot c_8$

c_2 : $\bar{c}_5 \cdot c_8 + \bar{c}_6 (c_9 (1 + c_{10}) + c_{10})$

c_3 : $c_7 c_9 (1 + c_{10}) + \bar{c}_5 c_8 + (1 + c_7) \bar{c}_6 (c_9 (1 + c_{10}) + c_{10})$

c_4 : $c_7 ((1 + \bar{c}_6) c_9 (1 + c_{10}) + \bar{c}_6 c_{10})$

5. Micro-Model with Resultant Data and Filtering

To depict a system which contains both internal resultant data generation processes and conditional or unconditional systematic information filtering requires a combination of the foregoing techniques and those of Chapter 5. As in Chapter 5, the vehicle for the transition from macro-to-micro-model will be the relation of micro-direct connectivity. Unlike the work in Chapter 5, the range of k will differ between different classes of components. We list below, the subsets between which the micro-direct connectivity relation is permitted, and the range of k in each case:

Micro-Direct Connectivity

	From	To	k
C12.	C_{s1}	C_{s2}	$1 \leq k \leq n_1$
C13.	C_{s1}	C_{s4}	
C14.	C_{s1}	C_{s5}	
C15.	C_{s2}	C_{s2}	$(n_1+1) \leq k \leq (n_1+n_2) = n_3$
C16.	C_{s2}	C_{s4}	
C17.	C_{s2}	C_{s5}	
C18.	C_{s4}	C_{s2}	$1 \leq k \leq n_3$
C19.	C_{s4}	C_{s4}	
C20.	C_{s4}	C_{s5}	

Figure 38 is a pictorial representation of the conceptual network associated with this model. We have drawn the arcs between nodes as if they were cables, each containing the indicated number of wires. In the several cases where two cables are "spliced" to form a third, the third cable contains a number of wires equal to the sun of the wires in the two cables joined to form it. Furthermore, each wire in each cable is labelled with a value according to C12 through C20, so that each wire is associated with input c_k.

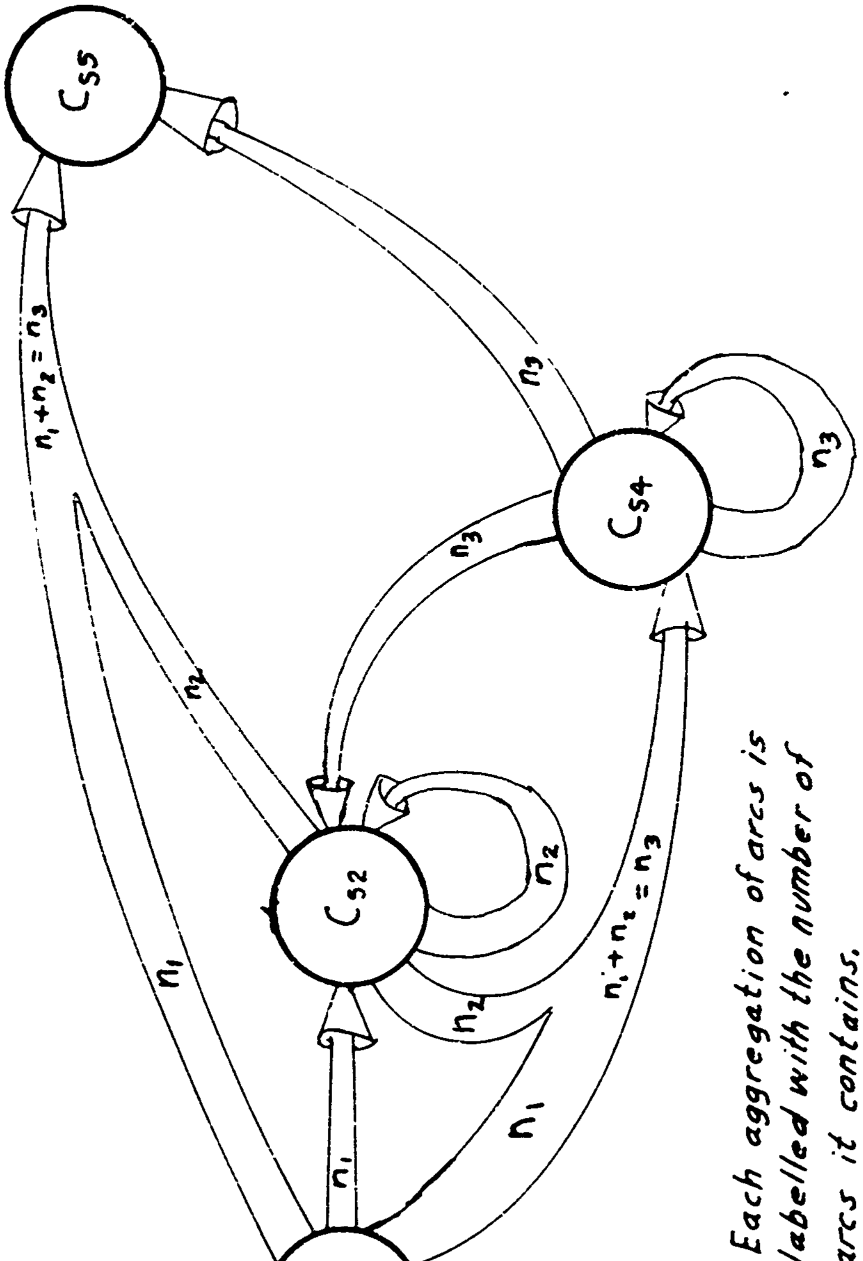
Basically, our model will follow closely the micro-model of Chapter 5, modified as required to account for the new class of secondary inputs and the difference in size of the micro-direct connectivity relations. We will follow the statements of Chapter 5, modifying them when necessary.

S19. The transmittance values, $t_{i,j,k}$, may assume the values described in S7. However, the range of k will vary according to C12 through C20.

As in Chapter 5, we consider a subset of input items as a component, except that now we confine the subset to primary inputs; that is, the set C_{s1}. We call this component $c_{(n_1+\frac{1}{2})}$. The component $c_{(n_3+\frac{1}{2})}$ refers to C_{s2}, the subsct of secondary inputs.

Statements T1 through T5, Chapter 5, section 3, hold here. However, in this case, $T_{i,j}$ is not defined for $i \le n_1$.

Figure 38



NOTE: Each aggregation of arcs is labelled with the number of arcs it contains.

S20. The transmittance vector, $T_{i,j}$, will be as described in S8, except for the size, which will vary according to C12 through C20.

S21. The input content of a component is as in S9, except that C12 through C20 will govern the size of D_i; and, we define:

$$D_{(n_1+\frac{1}{2})} = V_{n_1}(1) \ ,$$

and

$$D_{(n_3+\frac{1}{2})} = V_{n_2}(1) \ .$$

S22. The cargo of a micro-direct connectivity relation is as described in S10, except that the size of $R_{i,j}$ is governed by C12 through C20.

Statement S11 holds, as do equations (5-1) through (5-3).

With respect to filtering functions, S12 and S13 hold.

S23. S14 holds except that the size of the vector, $F_{i,j}$, is the same as the size of $T_{i,j}$.

The matrix model of the system with internal data generating and filtering processes may be described by a matrix, SRF, of size $(2 + n_4) \times (1 + n_4 + n_5)$.

M16.
$$srf_{i,j} = \begin{cases} V_{n_3}(-1); \text{ if } (n_3+1) \leq i = j \leq (n_3+n_4), \\ T_{i,j}-(1/c_j)\ e_j; \ i = (n_3+\frac{1}{2}) \text{ and } \\ \qquad (n_1+1) \leq j \leq (n_1+n_2) = n_3, \\ T_{i,j}; \text{ otherwise,} \end{cases}$$

and: $n_1 < i \leq (n_3+n_4)$; $(n_1+1) \leq j \leq n$,

where: e_j is the j-th unit vector; i.e., a vector of all zeros except for the j-th element, which is 1.

M17.

	$j = (n_3+\frac{1}{2})$	$j=(n_3+1)\ldots(n_3+n_4)$	$j=(n_3+n_4+1)\ldots n$
$i=(n_1+\frac{1}{2})$	$SRF_{s1,s2}$	$SRF_{s1,s4}$	$SRF_{s1,s5}$
$i=(n_3+\frac{1}{2})$	$SRF_{s2,s2}-I(C)$	$SRF_{s2,s4}$	$SRF_{s2,s5}$
$i=(n_3+1)$ $\vdots$ (n_3+n_4)	$SRF_{s4,s2}$	$SRF_{s4,s4}+I(V(-1))$	$SRF_{s4,s5}$

where: $I(V(-1))$ is a symbol for a diagonal of vectors, $V_{n_3}(-1)$, as in M7, and,

$I(C)$ is a symbol for the diagonal matrix with entries $(1/c_i)$, as in M12.

M18. Every row and every column of SRF will contain at least one entry not equal to $V\ (0)$, $V\ (-1)$, or $-(1/c_i)$.

M19.

$$SRF_* = \left[\begin{array}{c|c|c} SRF_{s1,s2} & SRF_{s1,s4} & SRF_{*s1,s5} \\ \hline SRF_{s2,s2}-I(C) & SRF_{s2,s4} & SRF_{*s2,s5} \\ \hline SRF_{s4,s2} & SRF_{s4,s4}+I(V(-1)) & Z \end{array}\right]$$

M20.

$$SRF^* = \begin{array}{ccc} SRF_{s1,s2} & SRF_{s1,s4} & SRF^*_{s1,s5} \\ \hline SRF_{s2,s2}-I(C) & SRF_{s2,s4} & Z \\ \hline SRF_{s4,s2} & SRF_{s4,s4}+I(V(-1)) & Z \end{array}$$

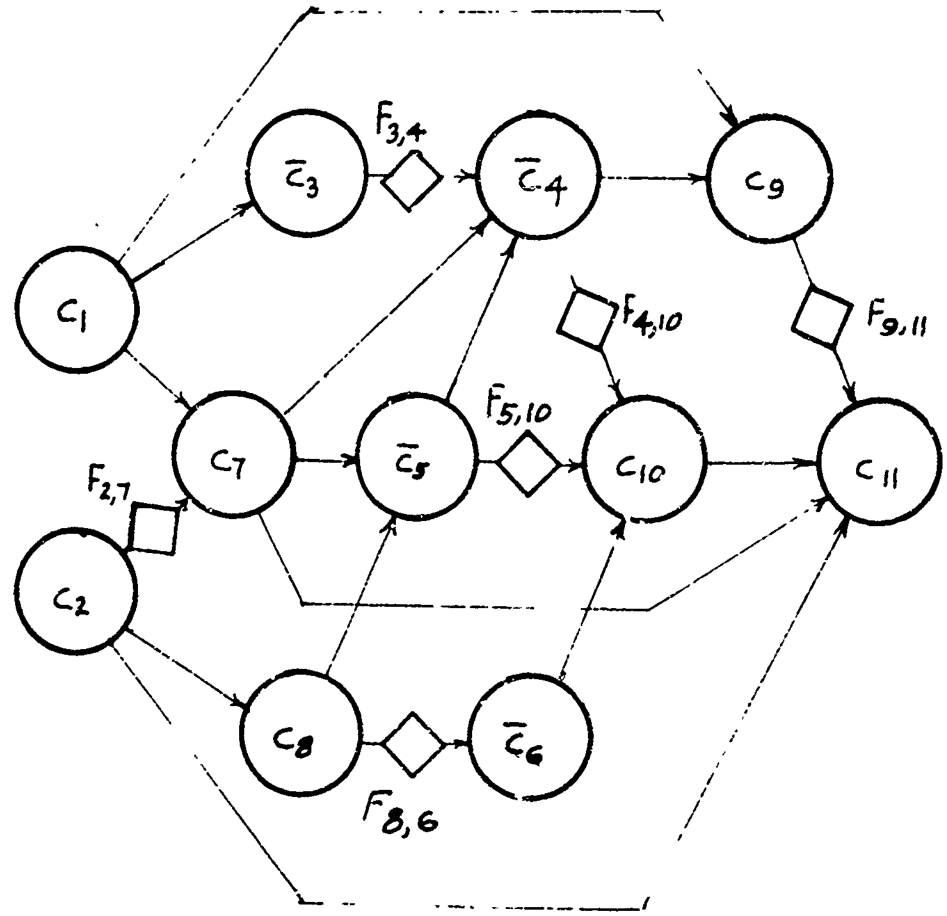
The interpretation of the partial solution and the solution is as previously described.

The application of the algorithm for micro-models represented by SF will reduce SRF to SRF_* if we consider the first and second rows of blocks of SRF to be one row of blocks, to form column vectors of size n_3. At this stage, the first row of SRF_* will contain vector entries of size n_1; and the second row, vector entries of size n_2. We now consider the first row as an n_1-row submatrix, and the second row as an n_2-row submatrix, both with scalar entries, and apply the algorithm developed for the macro-model, S, to reduce $SRF_{*s2,s5}$ to Z. The result will be SRF*.

The proof of this follows the same arguments used in Chapter 4, section 7.

In Figure 39, we show the network drawing of a system containing both internal data generating and filtering processes. The associated matrices, SRF, SRF_*, and SRF^* are shown in Figures 40, 41, and 42, respectively. A full evaluation of the results depicted in Figure 42 can be obtained in truth table format as was done in Chapter 5.

Figure 39



$n_1 = 2$ $c_{s1} = \{c_1, c_2\}$

$n_2 = 4$ $c_{s2} = \{c_3, c_4, c_5, c_6\}$

$n_3 = 6$ $c_{s3} = \{c_1, c_2, c_3, c_4, c_5, c_6\}$

$n_4 = 4$ $c_{s4} = \{c_7, c_8, c_9, c_{10}\}$

$n_5 = 1$ $c_{s5} = \{c_{11}\}$

$n = 11$

(cont'd)

Figure 39 *(cont'd)*

i,j	$T_{i,j}$	i,j	$T_{i,j}$
$2\frac{1}{2}, 3$	1, 0	7,4	$V_6(1)$
$2\frac{1}{2}, 7$	$1, 1-v_{2\frac{1}{2},7}$	7,5	$V_6(1)$
$2\frac{1}{2}, 8$	0, 1	7,11	$V_6(1)$
$2\frac{1}{2}, 9$	1, 0	8,5	$V_6(1)$
$2\frac{1}{2}, 11$	0, 1	8,6	$1, 1-v_{8,6}, 1, 1, 1, 1$
$6\frac{1}{2}, 9$	0, 1, 0, 0	9, 11	$1, 1, 1, 1-v_{9,11}, 1, 1$
$6\frac{1}{2}, 10$	$0, 1-v_{4,10}, 1-v_{5,10}, 1$	10, 11	$V_6(1)$

j	$T_{6\frac{1}{2},j} - (1/c_j)\cdot(e_j)$
3	$-\frac{1}{c_3}, 0, 0, 0$
4	$1-v_{3,4}, -\frac{1}{c_4}, 1, 0$
5	$0, 0, -\frac{1}{c_5}, 0$
6	$0, 0, 0, -\frac{1}{c_6}$

NOTE: *Only non-zero* $T_{i,j}$*'s listed.*

Figure 40

SRF =

	$\bar{c}_3$	$\bar{c}_4$	$\bar{c}_5$	$\bar{c}_6$	c_7	c_8	c_9	c_{10}	c_{11}
$c_{2\frac{1}{2}}$	$T_{2\frac{1}{2},3}$	$V_2(0)$	$V_2(0)$	$V_2(0)$	$T_{2\frac{1}{2},7}$	$T_{2\frac{1}{2},8}$	$T_{2\frac{1}{2},9}$	$V_2(0)$	$T_{2\frac{1}{2},11}$
$c_{6\frac{1}{2}}$	$T_{6\frac{1}{2},3}-\frac{1}{\bar{c}_3}\cdot e_4$	$T_{6\frac{1}{2},4}-\frac{1}{\bar{c}_4}\cdot e_4$	$T_{6\frac{1}{2},5}-\frac{1}{\bar{c}_5}\cdot e_5$	$T_{6\frac{1}{2},6}-\frac{1}{\bar{c}_6}\cdot e_6$	$V_4(0)$	$V_4(0)$	$T_{6\frac{1}{2},9}$	$T_{6\frac{1}{2},10}$	$V_4(0)$
c_7	$V_6(0)$	$V_6(1)$	$V_6(1)$	$V_6(0)$	$V_6(-1)$	$V_6(0)$	$V_6(0)$	$V_6(0)$	$V_6(1)$
c_8	$V_6(0)$	$V_6(0)$	$V_6(1)$	$T_{8,6}$	$V_6(0)$	$V_6(-1)$	$V_6(0)$	$V_6(0)$	$V_6(0)$
c_9	$V_6(0)$	$V_6(0)$	$V_6(0)$	$V_6(0)$	$V_6(0)$	$V_6(0)$	$V_6(-1)$	$V_6(0)$	$T_{9,11}$
c_{10}	$V_6(0)$	$V_6(0)$	$V_6(0)$	$V_6(0)$	$V_6(0)$	$V_6(0)$	$V_6(0)$	$V_6(-1)$	$V_6(1)$

Figure 41

$SRF_* =$

	$\bar{c}_3$	$\bar{c}_4$	$\bar{c}_5$	$\bar{c}_6$	c_7	c_8	c_9	c_{10}	c_{11}
$c_{2\frac{1}{2}}$	$T_{2\frac{1}{2},3}$	$V_2(0)$	$V_2(0)$	$V_2(0)$	$T_{2\frac{1}{2},7}$	$T_{2\frac{1}{2},8}$	$T_{2\frac{1}{2},9}$	$V_2(0)$	$\begin{bmatrix} 2 \\ 2 - v_{2\frac{1}{2},7} \end{bmatrix}$
$c_{6\frac{1}{2}}$	$T_{6\frac{1}{2},3} - \frac{1}{\bar{c}_3} \cdot e_3$	$T_{6\frac{1}{2},4} - \frac{1}{\bar{c}_4} \cdot e_4$	$T_{6\frac{1}{2},5} - \frac{1}{\bar{c}_5} \cdot e_5$	$T_{6\frac{1}{2},6} - \frac{1}{\bar{c}_6} \cdot e_6$	$V_4(0)$	$V_4(0)$	$T_{6\frac{1}{2},9}$	$T_{6\frac{1}{2},10}$	$\begin{bmatrix} 0 \\ 2 - v_{4,10} - v_{9,11} \\ 1 - v_{5,10} \\ 1 \end{bmatrix}$
c_7	$V_6(0)$	$V_6(1)$	$V_6(1)$	$V_6(0)$	$V_6(-1)$	$V_6(0)$	$V_6(0)$	$V_6(0)$	$V_6(0)$
c_8	$V_6(0)$	$V_6(0)$	$V_6(1)$	$T_{8,6}$	$V_6(0)$	$V_6(-1)$	$V_6(0)$	$V_6(0)$	$V_6(0)$
c_9	$V_6(0)$	$V_6(0)$	$V_6(0)$	$V_6(0)$	$V_6(0)$	$V_6(0)$	$V_6(-1)$	$V_6(0)$	$V_6(0)$
c_{10}	$V_6(0)$	$V_6(0)$	$V_6(0)$	$V_6(0)$	$V_6(0)$	$V_6(0)$	$V_6(0)$	$V_6(-1)$	$V_6(0)$

(cont'd)

Figure 41 *(cont'd)*

Computation of solution area of SRF_*

$$\begin{bmatrix} T_{2\frac{1}{2},11} \\ V_4(0) \end{bmatrix} + \begin{bmatrix} V_2(0) \\ T_{6\frac{1}{2},11} \end{bmatrix} + \begin{bmatrix} T_{2\frac{1}{2},9} \\ T_{6\frac{1}{2},9} \end{bmatrix} \otimes \begin{bmatrix} T_{9,11} \end{bmatrix} + \begin{bmatrix} T_{2\frac{1}{2},7} \\ V_4(0) \end{bmatrix}$$

$$= \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1-v_{4,10} \\ 1-v_{5,10} \\ 1 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1-v_{9,11} \\ 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 1 \\ 1-v_{2\frac{1}{2},7} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} 2 \\ 2-v_{2\frac{1}{2},7} \\ 0 \\ 2-v_{4,10}-v_{9,11} \\ 1-v_{5,10} \\ 1 \end{bmatrix}$$

Figure 42

$$
SRF^{*} = \begin{array}{c|cccccccc|c}
 & \bar{c}_3 & \bar{c}_4 & \bar{c}_5 & \bar{c}_6 & c_7 & c_8 & c_9 & c_{10} & c_{11} \\
\hline
c_1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 2+(1+\bar{c}_5+\bar{c}_3(1-v_{3,4}))\cdot(\bar{c}_4(2-v_{4,10}-v_{9,11}))+\bar{c}_5(1-v_{5,10}) \\
c_2 & 0 & 0 & 0 & 0 & 1-v_{2,7} & 1 & 0 & 0 & 1+(1-v_{2,7})[1+(1+\bar{c}_5)\bar{c}_4(2-v_{4,10}-v_{9,11})+\bar{c}_5(1-v_{5,10})]+\bar{c}_5(1-v_{5,10})+\bar{c}_4(2-v_{4,10}-v_{9,11})+(1-v_{8,6})\bar{c}_6 \\
\bar{c}_3 & -\frac{1}{\bar{c}_3} & 1-v_{3,4} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\bar{c}_4 & 0 & -\frac{1}{\bar{c}_4} & 0 & 0 & 0 & 0 & 1 & 1-v_{4,10} & 0 \\
\bar{c}_5 & 0 & 1 & -\frac{1}{\bar{c}_5} & 0 & 0 & 0 & 0 & 1-v_{5,10} & 0 \\
\bar{c}_6 & 0 & 0 & 0 & -\frac{1}{\bar{c}_6} & 0 & 0 & 0 & 1 & 0 \\
c_7 & V_6(0) & V_6(1) & V_6(1) & V_6(0) & V_6(-1) & V_6(0) & V_6(0) & V_6(0) & V_6(0) \\
c_8 & V_6(0) & V_6(0) & V_6(1) & T_{8,6} & V_6(0) & V_6(-1) & V_6(0) & V_6(0) & V_6(0) \\
c_9 & V_6(0) & V_6(0) & V_6(0) & V_6(0) & V_6(0) & V_6(0) & V_6(-1) & V_6(0) & V_6(0) \\
c_{10} & V_6(0) & V_6(0) & V_6(0) & V_6(0) & V_6(0) & V_6(0) & V_6(0) & V_6(-1) & V_6(0)
\end{array}
$$

A combination of methods previously developed leads to a path-determining model for a system with both internal data generating and filtering processes. A matrix, SPRF, is required, identical to SRF, except that:

$$sprf_{i,i} = V_{n_3}(-1/c_i) \quad \text{for} \quad (n_3+1) \leq i \leq (n_3+n_4) \ .$$

This model is not further explored herein, since its construction and use are obvious from the foregoing.

CHAPTER 7

SEMANTIC RELATIONS AMONG PRIMARY DATA

1. Knowledge Redundancy

It is the prupose of this chapter to examine the items of data that constitute the primary inputs to a system from the standpoint of their semantic content. Particularly, we are interested in the fact that:

a. an item of data refers to some object, and

b. in a given set of data items, more than one item may have the same referend.

When this occurs, there exists a form of redundancy in the system which may be referred to as "knowledge redundancy", rather than "data redundancy". If two items of data each reach a certain system output through one path, there is no data redundancy. However, should these two items of data have the same referend, then the system output has received the same knowledge (although not the same item of data) twice. It is our objective to incorporate this phenomenon within the framework of the systems-matrix and the algorithms developed so far.

2. The Denotation Relation

The particular semantic relationship treated herein will be called the denotation relationship. The following definitions, which follow [2, p.191], introduce the relationship:

A system is concerned with entities. An entity is an object, a person, a concept, a thought, an instance, an event, an occurrence, etc. A data processing system is concerned with one or more classes of entities (*i.e.*, employees, customers, products) and with certain selected *properties* of these classes of entities. For example, a system might be concerned with the class of entities "employees" and the properties "employee number", "age", "sex", "name", "job number", and "hourly wage rate". Each property has assigned to it a set of *values*. Thus the value set associated with the property "age" might be the set of integers from 18 to 65, inclusive, and the value set associated with the property "sex" might be {"male", "female" }. As is postulated in [2, p.191], every property value set must include at least two values: Ω (undefined, or not relevant) and θ (missing, relevant but not known). Then, [2, p.192], every member of an entity class has assigned to it one and only one value from each property value set.

In the models we have been developing, each primary input represents a property, and associated with each primary input is a property value set. If $U(c_i)$, $1 \leq i \leq n_1$, is the property value set of property c_i, $U(c_i) = (u_{1i}, u_{2i}, \ldots, u_{qi})$, then associated with any specific entity, ℓ, is a specific value of c_i, $u_{\ell i}$, $1 \leq \ell \leq q$, $1 \leq i \leq n_1$. We may now state:

S24. If, given a specific entity, ℓ, and given $u_{\ell i} \in U(c_i)$, we can determine $u_{\ell j} \in U(c_j)$ by means of some finite procedure within the system, then we may say that $u_{\ell j}$ *is denoted by* $u_{\ell i}$. If this is true for all entities, we may say that c_j *is denoted by* c_i, or, symbolically, $c_j \leftarrow c_i$.

Denotation is a binary relation in the set-theoretic sense [39, pp.23-25], and may be expressed as a set of ordered pairs $\langle c_j, c_i \rangle \in \leftarrow$.

Denotation has the following properties:

D1. reflexive. $c_i \leftarrow c_i$.

D2. transitive. If $c_i \leftarrow c_j$, and $c_j \leftarrow c_k$, then $c_i \leftarrow c_k$.

D3. non-symmetric. If $c_i \leftarrow c_j$, then it may or may not be true that $c_j \leftarrow c_i$.

When the denotation relation exists between c_i and c_j, we say that there is a semantic relationship between c_i and c_j. We do not mean necessarily that c_i and c_j are synonymous, but merely that, directly or indirectly, both items of data point to the same referend. If c_i and c_j are synonymous, then the denotation relation between them will be symmetric, which we indicate by writing:

$$c_i \leftrightarrow d_j \equiv (c_i \leftrightarrow c_j) \text{ and } (c_j \leftarrow c_i).$$

As an example, we consider the following denotation relations:

$\langle$customer account number$\rangle \leftrightarrow \langle$customer name$\rangle$ (7-1)

$\langle$salesman name$\rangle \leftarrow \langle$customer name$\rangle$ (7-2)

$\langle$salesman number$\rangle \leftrightarrow \langle$salesman name$\rangle$ (7-3)

By applying D2 (transitivity characteristic) to the above:

$\langle$salesman name$\rangle \leftarrow \langle$customer account number$\rangle$ (7-4)

$\langle$salesman number$\rangle \leftarrow \langle$customer name$\rangle$ (7-5)

Another application of D2 to (7-3) and (7-4) yields:

$$\langle \text{salesman number} \rangle \leftarrow \langle \text{customer account number} \rangle \qquad (7\text{-}6)$$

The significance of this relation can be seen in the following: Suppose it were necessary for all four of the items of data mentioned in the example above tc be included in some report; and that the information needed to prepare this reoort was communicated over a costly channel, with the cost being proportional t, the number of characters transmitted. Then, by establishing a proper reference file at the receiving end of the channel, the desired information for the report could be obtained if only customer account number were transmitted, since all other items are denoted by this one. (The customer name has the same power, but we assume that the customer account number has fewer characters.)

3. Incorporating Denotation in the Systems-Matrix

The denotation relation on a set of primary inputs, C_{s1}, may be depicted by an $(n_1 \times n_1)$ matrix, A, such that:

$$a_{i,j} = \begin{cases} 1; \text{ if } i = j \text{ (by D1)}, \\ 1; \text{ if } c_j \leftarrow c_i, \\ 0; \text{ otherwise} . \end{cases} \qquad (7\text{-}7)$$

To construct such a matrix, we must make sure that all denotation relations, both those stated and those derivable from the stated ones by the transitivity characteristic (D2), are represented by 1's in the appropriate cells. Such a matrix is called a reachability matrix, and its construction has been described by Harary, et.al. [16, pp.115-122].

We start by forming a matrix according to equation (7-7) from the given relations. This matrix we call A_o, and, since it contains entries of only 0 or 1, we treat it as a Boolean matrix. We form A_1 by multiplying A_o by itself in a Boolean sense. Boolean matrix multiplication is identical to ordinary matrix multiplication except for the rule: $1 + 1 = 1$. Thus, A_1 will also be a Boolean matrix. A_2 is formed by multiplying A_1 by A_o, in a Boolean sense. In general:

$$A_i = A_{(i-1)} \odot A_o; \quad i = 1, 2, \ldots,$$

where $\odot$ indicates Boolean matrix multiplication. Eventually, we reach a point such that [16, p.121]:

$$A_{(n+1)} = A_n \odot A_o = A_n \,.$$

Then, A_n is the reachability matrix of the denotation relation. The (i,j)-th cell of A_n will be 1 if $c_j \leftarrow c_i$ either directly or by repeated application of the transitivity characteristic (D2); 0, otherwise.

For the example stated above, let:

c_1 = customer account number

c_2 = customer name

c_3 = salesman name

c_4 = salesman number .

Then, from equation (7-7) and equations (7-1), (7-2) and (7-3):

$$A_c = \begin{array}{c|cccc|} & c_1 & c_2 & c_3 & c_4 \\ \hline c_1 & 1 & 1 & 0 & 0 \\ c_2 & 1 & 1 & 0 & 0 \\ c_3 & 0 & 1 & 1 & 1 \\ c_4 & 0 & 0 & 1 & 1 \\ \hline \end{array} ,$$

and,

$$A_1 = A_o \odot A_o = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

Two cells of A_1 contain 1's not contained in corresponding cells of A_0; namely, $(a_1)_{3,1}$ and $(a_1)_{4,2}$. The interpretation of these 1's are equations (7-4) and (7-5), above. Continuing:

$$A_2 = A_1 \odot A_o = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} .$$

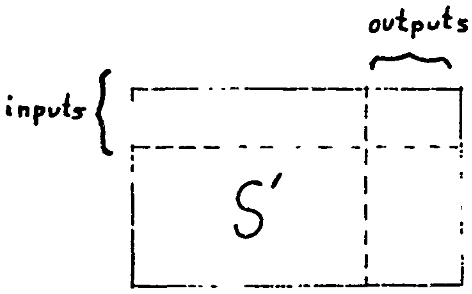
Here, we have gained one additional 1 in $(a_2)_{4,1}$, which corresponds to equation (7-6) above. Finally:

$$A_3 = A_2 \odot A_o = A_2 ,$$

and $A_n = A_2$.

The effect of the denotation relation on the flow of information in the system can be shown by pre-multiplying the previously defined solution area of any of the systems-matrices (S*, SP*, SF*, SPF*, SR*, SRF*, or SPRF*) by A_n. This can be readily done by incorporating A_n into the systems-matrix as follows:

Let S' be one of the matrices S, SP, SF, SPF, SR, SRF or SPRF. Denote by dotted lines the rows of S' corresponding to primary inputs and the columns of S' corresponding to outputs:



Then, define:

$$S'A = \left[\begin{array}{c:c:c} A_n & Z & \boxed{Z} \\ \hdashline -I & & \\ \hdashline Z & S' & \end{array}\right]$$

where the solution area is indicated by heavy lines. Application of any of the previously described algorithms will produce the desired results.

The effect of adding A_n to the systems-matrix is as follows: Suppose that, before A_n is applied, it is disclosed that an input, c_i, reaches an output c_j, through α paths, and that another input,

c_k, reaches c_j, through β paths. Suppose that $c_k \leftarrow c_i$. Then, A_n will contain a 1 in $(a_n)_{k,i}$. The effect of pre-multiplying the solution area by A_n will be to change the indication of the number of paths through which c_k reaches c_j from β to $(\alpha + \beta)$. This can be seen by observing that A_n is the sum of elementary permutation matrices, and the effect of pre-multiplying a matrix by an elementary permutation matrix is to add one row to another.

4. An Alternative Representation

The resulting figures, of the form $(\alpha + \beta)$, may be confusing to the analyst. He does not know, without considerable additional analysis, what portion of a particular number in a cell in the solution area is due to redundancy of information flow and what portion is due to knowledge redundancy. Therefore, it might be advantageous to indicate the solution in the form:

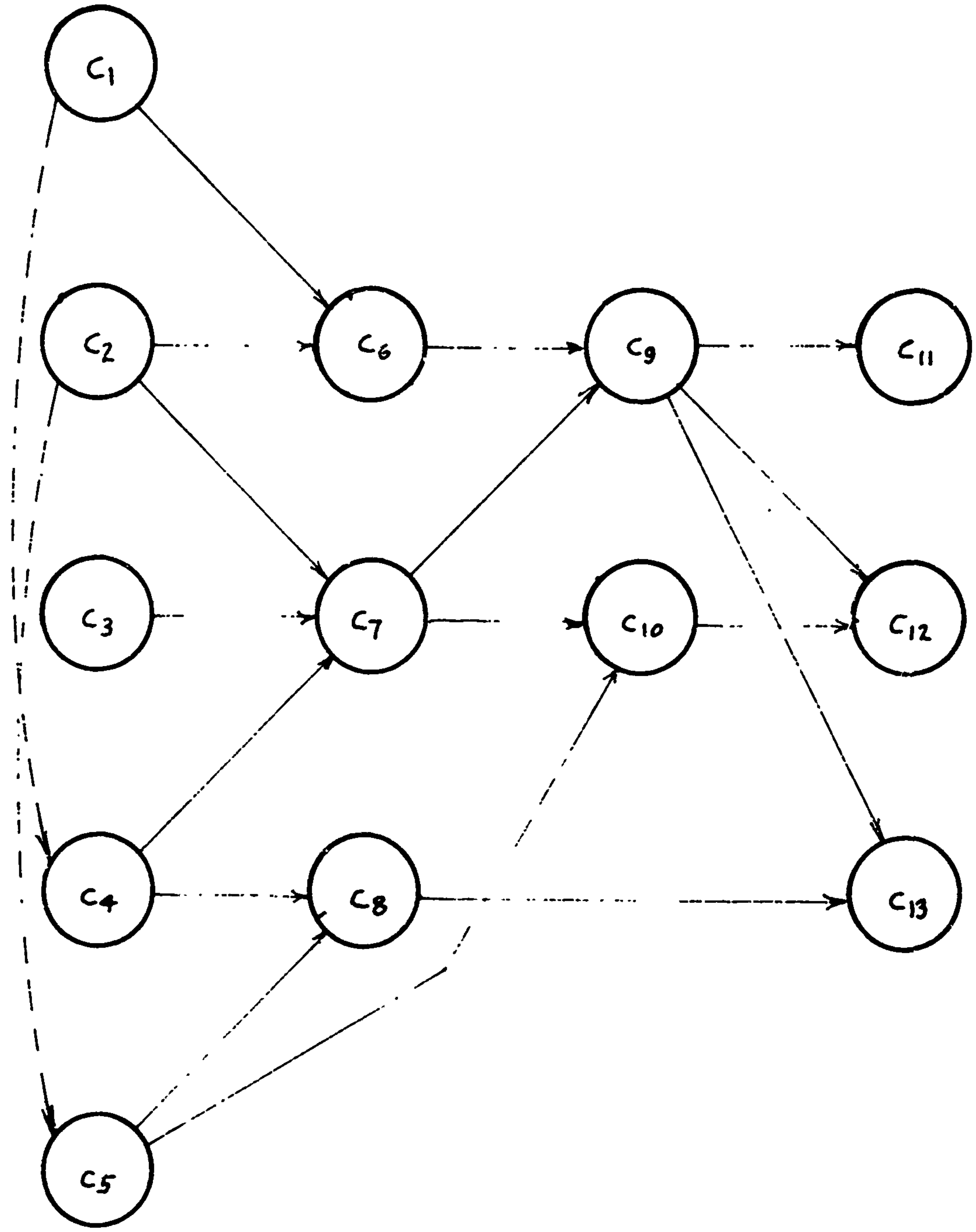
$$(s')^*_{k,j} = \beta + \sum_i \alpha_i \hat{c}_i ,$$

where β is the number of paths through which c_k reaches c_j due to information flow only, and α_i is the number of paths through which knowledge of the referend of c_k reaches c_j due to the denotation relation $c_k \leftarrow c_i$. The mechanism for doing this is simply to replace A_n by A'_n, where:

$$(a'_n)_{i,j} = \begin{cases} 1; & \text{if } i = j \ , \\ \hat{c}_j; & \text{if } i \neq j \text{ and } (a_n)_{i,j} = 1 \ , \\ 0; & \text{otherwise} \ . \end{cases}$$

By way of illustration, Figure 43 depicts a system with denotation relations. Considering a macro-model (S) of the system, Figure 44 shows the matrix SA', and Figure 45, SA'*.

Figure 43



NOTE: *Broken arcs represent* denotation *relations:*

$c_4 \leftarrow c_1$

$c_5 \leftarrow c_2$

Figure 44

$$SA' = \begin{array}{c||ccccccccccccc||}
 & c_1 & c_2 & c_3 & c_4 & c_5 & c_6 & c_7 & c_8 & c_9 & c_{10} & c_{11} & c_{12} & c_{13} \\
\hline
c_1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
c_2 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
c_3 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
c_4 & \hat{c}_1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
c_5 & 0 & \hat{c}_2 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
c_1 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
c_2 & 0 & -1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
c_3 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
c_4 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
c_5 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
c_6 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
c_7 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 1 & 0 & 0 & 0 \\
c_8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 \\
c_9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 1 & 1 \\
c_{10} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 \\
\hline
\end{array}$$

Figure 45

$$SA'^{*} = \begin{array}{c|ccccccccccccc}
 & c_1 & c_2 & c_3 & c_4 & c_5 & c_6 & c_7 & c_8 & c_9 & c_{10} & c_{11} & c_{12} & c_{13} \\
\hline
c_1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\
c_2 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 3 & 2 \\
c_3 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 1 \\
c_4 & \hat{\varepsilon}_1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1+\hat{\varepsilon}_1 & 2+\hat{\varepsilon}_1 & 2+\hat{\varepsilon}_1 \\
c_5 & 0 & \hat{\varepsilon}_2 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 2\hat{\varepsilon}_2 & 1+3\hat{\varepsilon}_2 & 1+2\hat{\varepsilon}_2 \\
c_1 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
c_2 & 0 & -1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
c_3 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
c_4 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
c_5 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
c_6 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
c_7 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 1 & 0 & 0 & 0 \\
c_8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\
c_9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\
c_{10} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0
\end{array}$$

CHAPTER 8

SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS

1. Summary

The flow of data in a management information system can be depicted in the form of a matrix, and matrix and scalar arithmetic operations can be used to transform the original matrix to one which contains certain defined characteristics or measures of the system. The matrix which depicts the system takes different forms depending upon the specific system characteristic or measure to be computed and the system factors to be included. The computing algorithm is essentially the same for all cases.

In the foregoing chapters we have used the model to compute two system characteristics:

a. the number of paths by which each system input reaches each system output, and,

b. the exact specification of each path by which each system input reaches each system output.

In addition, it has been shown that systems with more complex structures may be most adequately described by a combination of the above characteristics, and this has been done.

The models used may be categorized into two groups:

a. macro-models: depicting simple connectivity relations among inputs, intermediate reports, and outputs, and,

b. micro-models: depicting complex connectivity relations; that is, the connectivity among all system components with respect to each system input.

The use of micro-models permits the systems-matrix model to incorporate phenomena which had not been considered in earlier models. This thesis explores:

a. information filtering, conditional and unconditional,

b. internally generated information, and,

c. the semantic relation of denotation among items of data.

More specifically, this thesis has presented the following models:

Model	Chapter	Systems Factors Included	Computes[1]
S	4	simple connectivity	N
SP	4	simple connectivity	P
SF	5	complex connectivity, filtering	N
SPF	5	complex connectivity, filtering	P
SR	6	simple connectivity, internally generated data	NP
SRF	6	complex connectivity, filtering, internally generated data	NP
SPRF	6	complex connectivity, filtering, internally generated data	P
S'A	7	those of S', plus semantic denotation where S' is any of the above models.	see S'

[1] N means "number of paths".
P means "specification of paths"
NP means "combination of number of paths and specification of paths".

2. Conclusions

Our basic conclusion is that it is feasible and desirable to establish mathematical models of management information systems within the framework of matrix algebra.

In this thesis, we started with a previous model and algorithm, and progressively expanded the model to include several additional systems phenomena, formulating models of increasing scope and power. Yet this was accomplished with a logical expansion of the ideas upon which the original model was based. This, we feel, demonstrates that our approach is feasible. It leads us to believe that further extensions can be accomplished by the same method of building logically upon the foundations laid.

That our approach is desirable was demonstrated in Chapter 2, section 1 and Figures 6 through 10. We feel that the concept of matrix models of management information systems, a concept first proposed by Lieberman [26] and adhered to in our work, is the first breakthrough in the attempt to quantify the analysis and synthesis of management information systems and to replace intuition by measurement.

3. Suggestions for Further Research

The research reported in this thesis, while useful in itself, is not terminal. Several additional areas of investigation are proposed in the following paragraphs. No attempt is made to be exhaustive.

In our description of the models we have emphasized that the conceptual networks associated with the matrix contain no cycles or loops. This assured us that the matrices would be strictly triangular matrices, and, thusly, that the summation term in Equation (4-2) was finite and that the inverse in Equation (4-3) exists. Should the sub-matrix $S_{s4,s4}$ represent a network that does contain loops or cycles, the algorithm will fail. (Note the restriction of the prohibition to that part of the network represented by $S_{s4,s4}$. Thus, in Chapter 7, the matrix A_n does contain loops and possibly cycles, but does not cause the algorithm to fail since it is not part of $S_{s4,s4}$.)

It is quite easy to find countless examples of information systems which do contain cycles. One example might be the case of an inventory control system, wherein the primary input data included demand, the secondary input data included re-order point and standard order quantity, and the outputs included replenishment orders and recalculated values of re-order points. Replenishment orders are triggered by a comparison of available quantity and re-order point, and recalculation of the re-order point is triggered by too small a time interval between the current and the most recent previous replenishment orders. This seems to give rise to a cycle, since the decision to order at time t is a function of the re-order point, and the re-order point is a function of t.

In essence, however, this does not constitute a true cycle. There is a separation in time between the event "place an order" and the event "change the re-order point". Therefore, it is not immediately obvious that our algorithm will fail if we set up a model reflecting the time gap.

We have not, as yet, fully explored this problem. We think, however, that it might be appropriate to consider a system represented by a series of matrices, each single matrix representing the system at some particular time. The connectivity from an output to some intermediate entity would be a relation from an output at time t (represented by matrix S_t) to an intermediate entity at time t', $t < t'$, (represented by matrix $S_{t'}$). The conceptual network associated with such a model would be a three-dimensional one, with each node being a cylinder whose longitudinal axis was parallel to the time axis. Each matrix, S_t, would be associated with a plane of the network, perpendicular to the time axis and cutting it t units from the origin. Such a model would contain no loops or cycles in one plane.

The concept of a series of matrices in time introduces the possibility of investigating the dynamic behavior of systems. Suppose we introduce a change in the structure of an information system. For a while, the system will go through a transition phase until it reaches a state of equilibrium in the new structure. A series of systems-matrices, separated by time, might be a useful technique for studying transients.

Chapter 7 merely introduces the topic of one semantic relationsnip, denotation. This area is worthy of further research. One application of the concept which the author has used elsewhere is to answer the question: "Given a list of items of data which must appear on a report, and a list of denotation relations, what is the minimum amount of data which must be transmitted to the originator of the report?" This question can be answered, and is a form of the "covering problem" found in the minimization of Boolean functions, for the simple denotation relations of Chapter 7. The combinatorial problem gets rapidly out of hand, however, if more complex denotation relations are permitted; such as:

$$c_j \leftarrow c_i \boxdot c_k \,,$$

where the operator $\boxdot$ may represent an arithmetic operation or a set operation. In fact, under these conditions, it is no longer necessarily true that some power of A_0 will give all denotation relations arising through transitivity. Examples of such non-elementary denotation relations, which are quite common-place, are:

$$\langle\text{total price}\rangle \leftarrow \langle\text{unit price}\rangle \times \langle\text{quantity shipped}\rangle \,,$$

$$\langle\text{catalog number}\rangle \leftarrow \langle\text{product number}\rangle \wedge \langle\text{color code}\rangle \,.$$

Also in Chapter 7 mention was made of a paper by Bosack, *et.al.*[2]. It is our opinion that this is a work of major importance, which has received far less notice from other researchers than is warranted. One reason for this might be its abstract nature: systems analysts, by and large, are not attracted to such approaches, and those who can appreciate the paper are not usually interested in the still pragmatic area of systems analysis. We believe that further research can be done to connect the work of Bosack with the methodology of the approach taken in this thesis. This would give the systems-matrix model a more formal basis in abstract algebra, and would give the abstract formalism of Bosack a method of implementation.

Finally, we feel that much more can be done in the area of applications of the systems-matrix model. As examples, we cite three projects currently being pursued by the author:

a. The use of the systems-matrix model to investigate the effect upon information flow of changes in organizational work assignments.

b. The determination of the inherent delay time in a system, from receipt of input to issuance of output.

c. The propogation of errors throughout a system.

We feel that this list of further research topics will be increased in size with experience in the use of the model. Thus, we view the model technique proposed in this thesis as an open-ended research area.

BIBLIOGRAPHY

1. Barish, Norman N., Systems Analysis for Effective Administration, New York, Funk & Wagnalls Co., 1951.

2. Bosack, Robert, et.al., "An Information Algebra," Communications of the Association for Computing Machinery, Vol. 5, No. 4, (April, 1962), pp. 190-207.

3. Busacker, Robert G., and Thomas L. Saaty, Finite Graphs and Networks, New York, McGraw-Hill Book Co., 1965.

4. Business Systems, Cleveland, Ohio, Systems and Procedures Association, 1966.

5. Davis, Philip J., The Mathematics of Matrices, New York, Blaisdell Publishing Co., 1965.

6. Elmaghraby, Salah E., The Design of Production Systems, New York, Reinhold Publishing Corp., 1966.

7. Executive Office of the President, Bureau of the Budget, Process Charting, Washington, D.C., Superintendent of Documents, U.S. Printing Office, November, 1945.

8. Ficken, F.A., Linear Transformations and Matrices, Englewood Cliffs, N.J., Prentice-Hall, Inc., 1967.

9. Finkbeiner, Daniel T., II, Introduction to Matrices and Linear Transformations, Second Edition, San Francisco, W. H. Freeman and Company, 1966.

10. Flow Charting and Block Diagramming Techniques, Reference Manual C20-8008-0, White Plains, N.Y., International Business Machines Corp., undated.

11. Flowcharting Techniques, Form C20-8152, White Plains, N.Y., International Business Machines Corp., undated.

12. "Flowchart Symbols for Information Processing," Automation, Vol. 14, No. 1, (January, 1967), p.79.

13. Grad, Burton, and Richard G. Canning, "Information Process Analysis," The Journal of Industrial Engineering, Vol. 10, No. 6, (November-December, 1959), pp. 470-476.

14. Hadley, G., Linear Algebra, Reading, Mass., Addison-Wesley Publishing Co., Inc., 1961.

15. Harary, Frank, "Graph Theoretic Methods in the Management Sciences," Management Science, Vol. 5, No. 4, (July, 1959), pp.387-403. Reprinted in [38, pp.125-141].

16. ___________, Robert Z. Norman, and Dorwin Cartwright, Structural Models: An Introduction to the Theory of Directed Graphs, New York, John Wiley & Sons, Inc., 1965.

17. Hare, Van Court, Jr., Systems Analysis: A Diagnostic Approach, New York, Harcourt, Brace & World, Inc., 1967.

18. Herrman, Irvin A., Office Methods, Systems, and Procedures, New York, The Ronald Press Co., 1950.

19. Hoffman, Kenneth, and Ray Kunze, Linear Algebra, Englewood Cliffs, N.J., Prentice-Hall, Inc., 1961.

20. Hohn, Franz E., Elementary Matrix Algebra, New York, The Macmillan Co., 1958.

21. Homer, Eugene D., "A Method for the Simultaneous Design of Inputs, Outputs and Program for an Automatic Typewriter in an Integrated Data Processing System," unpublished masters thesis, New York University, 1958.

22. ___________, "Integrated Planning for Integrated Data Processing," Journal of Machine Accounting, Vol. 10, No. 12, (December, 1959), pp.42-49.

23. ___________, "A Generalized Model for Analyzing Management Information Systems," Management Science, Vol. 8, No. 4, (July, 1962), pp.500-515. Reprinted in [38, pp.197-212].

24. Kozmetsky, George, and Paul Kircher, Electronic Computers and Management Control, New York, McGraw-Hill Book Co., Inc., 1956.

25. Lazzaro, Vector (editor), Systems and Procedures, Englewood Cliffs, N.J., Prentice-Hall, Inc., 1959.

26. Lieberman, Irving J., "A Mathematical Model for Integrated Business Systems," Management Science, Vol. 2, No. 4, (July, 1956), pp.327-336. Reprinted in [42, pp.669-679].

27. Lorens, Charles S., Flowgraphs for the Modeling and Analysis of Linear Systems, New York, McGraw-Hill Book Co., Inc., 1964.

28. Lucas, Joseph W., More Profit - Less Paper through Work Simplification, Standard Oil of California, 1953.

29. Luft, A., L. Grinstein, and G. A. Pollak, "Development of a Qualitative Model of Information Flow at Management Levels in the Navy Supply System," Interim Report, Bayonne, N.J., U.S. Naval Supply Research and Development Facility, April, 1966.

30. Method of Charting Paperwork Procedures, ASA X2.3.4., New York, American Standards Association, Inc., 1959.

31. Nelson, Edward A., "A Charting Technique for the Analysis of Business Systems," Report P-3109, Santa Monica, Calif., The RAND Corp., 1965.

32. Neuschel, Richard F., Streamline Business Procedures, New York, McGraw-Hill Book Co., Inc., 1950.

33. _______________, Management by System (Second Edition of [32]), New York, McGraw-Hill Book Co., Inc., 1960.

34. Paperwork Flow Charting, Arvada, Colo., Capstone Bookpress, 1967.

35. Randall, Clarence B., Sally W. Weimer, and Maynard S. Greenfield, Systems and Procedures for Automated Accounting, Cincinnati, Ohio, South-Western Publishing Co., 1962.

36. Robichaud, Louis P. A., Maurice Boisvert, and Jean Robert, Signal Flow Graphs and Applications, Englewood Cliffs, N.J., Prentice-Hall, Inc., 1962.

37. Smiley, Malcolm F., Algebra of Matrices, Boston, Allyn and Bacon, Inc., 1965.

38. Starr, Martin K. (editor), Executive Readings in Management Science, New York, The Macmillan Co., 1965.

39. Stoll, Robert R., Set Theory and Logic, San Francisco, W. H. Freeman and Co., 1963.

40. Terry, George R., Office Management and Control, Fifth Edition, Homewood, Ill., Richard D. Irwin, Inc., 1966.

41. Warshall, Stephen, "An Algebraic Language for Flow Charts," Report No. TO-B-60-33, Burlington, Mass., Technical Operations, Inc., August, 1960.

42. Williams, Thomas H., and Charles H. Griffin (editors), Management Information - A Quantitative Accent, Homewood, Ill., Richard D. Irwin, Inc., 1967.

43. Young, John W., Jr., and Henry K. Kent, "Abstract Formulation of Data Processing Problems," The Journal of Industrial Engineering, Vol. 9, No. 6, (November-December, 1958), pp.471-479.

VITA

Eugene D. Homer was born in New York City in 1921. He received the degree of Bachelor of Industrial Engineering in 1954 and the degree of Master of Industrial Engineering in 1958, both from New York University.

His experience includes work in the field of production planning and control and factory administration for such companies as Garfield-Stahl Co. (textiles), Super Metal Manufacturing Co., American Lead Pencil Co. (now Venus Pencil Co.), Cutler-Hammer, Inc. (electrical switches), and Lightolier Corporation (naval lighting fixtures). His experience in the field of systems and procedures analysis and design includes five years with Emerson Radio and Phonograph Corporation, and several years as a management consultant.

He has been employed by New York University since 1957 as Instructor, Assistant Professor, Research Scientist, and, currently, Senior Research Scientist.

His publications and major reports include:

"The Evolution of a Form," The Office, May 1957.

"A Method for the Simultaneous Design of Inputs, Outputs and Program for an Automatic Typewriter in an Integrated Data Processing System," unpublished masters thesis, New York University, September 1958.

"Integrated Planning for Integrated Data Processing," Journal of Machine Accounting, December 1959.

"A Generalized Model for Analyzing Management Information Systems," Management Science, July 1962. Reprinted in: Starr, M.K., Executive Readings in Management Science, New York, The Macmillan Co., 1965.

"Some Remarks on the Computation of Production Requirements," 1962 International Meeting of the Institute of Management Sciences, University of Michigan, September 1962.

"Space Limited Aggregate Inventories with Phased Deliveries," *The Journal of Industrial Engineering*, June 1966.

"An Algorithm for Selecting and Sequencing Statements as a Basis for a Problem-Oriented Programming System," *Proceedings of the 21st National Conference*, Association for Computing Machinery, 1966.

"On the Computation of the Number of Elements in Combinations of Sets," *American Mathematical Monthly*, May 1967.

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1 ORIGINATING ACTIVITY (Corporate author)

New York University
Department of Industrial Engineering and Operations Research, Bronx, N. Y. 10453

2a. REPORT SECURITY CLASSIFICATION

UNCLASSIFIED

2b. GROUP

3 REPORT TITLE

A MATHEMATICAL MODEL OF THE FLOW OF DATA IN A MANAGEMENT INFORMATION SYSTEM

4 DESCRIPTIVE NOTES (Type of report and inclusive dates)

Scientific Interim

5 AUTHOR(S) (First name, middle initial, last name)

Eugene D. Homer

6 REPORT DATE

May 1968

7a. TOTAL NO. OF PAGES

150

7b. NO OF REFS

43

8a. CONTRACT OR GRANT NO

AF-AFOSR 621-66

b. PROJECT NO

9749-01

c. 6144501F

d. 681304

9a. ORIGINATOR'S REPORT NUMBER(S)

9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

AFOSR 68-0095

10 DISTRIBUTION STATEMENT

1. This document has been approved for public release and sale; its distribution is unlimited.

11 SUPPLEMENTARY NOTES

TECH, OTHER

12 SPONSORING MILITARY ACTIVITY

Air Force Office of Scientific Research
1400 Wilson Boulevard
Arlington, Virginia 22209

13 ABSTRACT

Previous work in the formulation of a matrix model for the flow of data in a management information system is extended to include three additional types of systems phenomena. These are: (a) data filtering, both conditional and unconditional, (b) internally generated items of data, and (c) the semantic relation of denotation among items of data. The concept of macro-model analysis vs. micro-model analysis is introduced. Methods are given for solving the models for either the number of paths connecting inputs to outputs or for complete specification of all such paths.

Security Classification

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Management Information Systems						
Information Science						
Matrix Algebra						
Networks						
Directed Graphs						

Security Classification