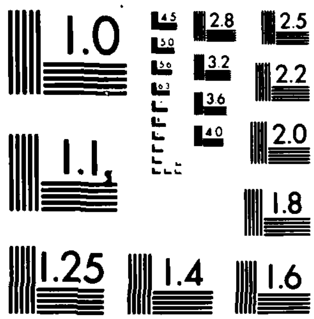
AD-A129 620 A QUANTITATIVE ANALYSIS OF THE EFFECT OF ORGANIZATIONAL STRUCTURE ON SOFT..(U) POLYTECHNIC INST OF NEW YORK BROOKLYN DEPT OF ELECTRICAL ENGI.. A G COMIER MAY 82 1/1

UNCLASSIFIED POLY-EE/CS-82-006 N00014-75-C-0858 F/G 9/2 NL

END
DATE
FILMED
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

ADA129620

Polytechnic Institute of New York

(12)

A QUANTITATIVE ANALYSIS OF THE

EFFECT OF ORGANIZATIONAL STRUCTURE

ON SOFTWARE ENGINEERING MANAGEMENT

BY

ANTOINE GERARD COMIER

Prepared for
Office of Naval Research
Contract N00014-75-C-0858

Report No. POLY-EE/CS 82-006

Polytechnic Institute of New York
Department of Electrical Engineering
and Computer Science
333 Jay Street
Brooklyn, New York 11201

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER

2. GOVT ACCESSION NO. AD-A129620

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
A Quantitative Analysis of the Effect of Organizational Structure on Software Engineering Management

5. TYPE OF REPORT & PERIOD COVERED
Technical

6. PERFORMING ORG. REPORT NUMBER
POLY EE-82-006

7. AUTHOR(s)
Antoine Gerard Comier

8. CONTRACT OR GRANT NUMBER(s)
N00014-75-C-0858

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Dept. of Elec. Engineering & Computer Science
Polytechnic Institute of New York
333 Jay Street, Brooklyn, New York 11201

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS
Office of Naval Research
Arlington, Virginia 22217

12. REPORT DATE
May, 1982

13. NUMBER OF PAGES
76

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)

15. SECURITY CLASS. (of this report)
Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)
Reproduction in whole or in part is permitted for any purpose of the United States Government.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)
- - - -

18. SUPPLEMENTARY NOTES
- - - -

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
software engineering, management, graph model, information content

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)
The effect of organizational structure on software engineering management has been quantified using graph models of the organizational flow of control and information. Values are assigned to each unit of information transmitted (α) and to each link formed (β). The model is constrained by the maximum number of interfaces that an individual can handle effectively. Optimization of the organizational structure is achieved by minimizing the result of trade-offs between the α's and the β's.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

A QUANTITATIVE ANALYSIS OF THE EFFECT OF ORGANIZATIONAL STRUCTURE ON SOFTWARE ENGINEERING MANAGEMENT

BY

ANTOINE GERARD COMIER

Prepared for
Office of Naval Research
Contract N00014-75-C-0858

Report No. POLY EE/CS 82-006

Polytechnic Institute of New York
Department of Electrical Engineering
and Computer Science
333 Jay Street
Brooklyn, New York 11201

AN ABSTRACT

The effect of organizational structure on software engineering management has been quantified using graph models of the organizational flow of control and information. Values are assigned to each unit of information transmitted (α) and to each link formed (β). The model is constrained by the maximum number of interfaces that an individual can handle effectively. Optimization of the organizational structure is achieved by minimizing the result of trade-offs between the α's and the β's.

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 General

Software engineering encompasses all activities required for the planning, design, development, generation, maintenance, enhancement and modification of software. Since the most critical resource in software engineering is creative human thought, people management is of paramount importance. The importance of the management function on software engineering projects has been recognized repeatedly in the literature. One survey done in 1975 on the Safeguard Data Processing System concluded that "the shortage of experienced software managers on the project posed a more serious challenge than the shortage of experienced programmers."[1]

Since an organization's management policy is implemented in its organizational structure it follows that the particular organizational structure selected for the management of a software project will contribute significantly to the degree of success achieved. This effect was recognized in a report on software modelling studies produced

---

[1] J.D. Musa and F.N. Woomer, Jr., "Safeguard Data Processing System: Software Project Management," Bell System Technical Journal, 1975, pp. S245-S259.

in 1977, which came to the conclusion that "... the organizational structure of the project team has a large influence on the productivity, reliability and quality of the software produced."[1] An article in the December, 79 issue of Datamation asserted the fact more forcefully: "Software managers who succeed in establishing effective organizations will enjoy development rates 1,200% better than managers who fail."[2]

If the effect that the organizational structure has on software engineering can be quantified, it will be possible to select the optimum structure for any given project. A quantitative model has been developed which utilizes graphs of the organizational flow of control and information. Values are assigned to each unit of information transmitted (α) and each link formed (β). The model is constrained by the maximum number of interfaces that an individual can handle effectively. Optimization of the organizational structure is achieved by minimizing the result of trade-offs between the α's and the β's.

---

1 M.L. Shooman and H. Ruston, "Final Report: Software Modelling Studies," Program in Software Engineering Poly-EE-77-042, SRS112, Sept. 30, 1977 pp. 23 and 24.

2 Edmund B. Daly, "Organizing for successful software development," Datamation, December 1979, pp. 107 to 120.

1.2 Software engineering defined

Software engineering is not the same as computer science. Computer science is the result of an endeavor to transform computer programming from an art into a science.[1] At present it includes the following branches or subfields: programming languages and systems, theory of computation, numerical analysis, artificial intelligence, and computer architecture.[2]

Software engineering on the other hand is a direct product of programming experience. Ever escalating software costs and continual failures of software projects forced software engineers to search for ways of improving software quality and reduce software costs. Investigation of methodologies used in other fields of engineering uncovered the fact (practically axiomatic) that software engineering methodology should consist of a disciplined application of the process of iteration to the specification, design, coding and testing of software.

In summary, the computer scientist's responsibility is to develop computer knowledge and techniques, while the software engineer's responsibility is to apply the techniques to produce quality software in

---

1 D. Knuth, "Computer Programming as an Art" (1974 ACM Turing Award Lecture), CACM, Vol. 17, No. 12, Dec. 1974, pp. 667-673.

2 P. Hsia, "Software Engineering and Computer Science," Computer magazine Vol. 12, No. 10, Oct. 1979, pp. 87-88.

a cost-effective manner. Stated differently, the function of the (computer) scientist is to know, while that of the (software) engineer is to do.[1] The computer scientist adds to the store of verified, systemized knowledge of the computer-centered world; the software engineer brings this knowledge to bear on practical problems.[2]

---

1 "Engineering" Encyclopedia Britannica, 15th ed., Vol. 6 (Macropaedia), p. 860.

2 P. Hsia, op. cit. p. 88.

1.3 Software engineering management

Software engineering encompasses all activities required to plan, design, code, test and modify software in order to meet a set of requirements.

In most software projects development costs are the total system costs, since the prototype software system is the only one produced. This is in sharp contrast with hardware projects where large-scale production constitutes a major portion of system cost. Since the most critical (and expensive) resource in software development is creative human thought, people-management takes on increased significance. It becomes vital that the optimum number and mix of designers, coders, and testers be assigned to the project; that they be organized into the optimum structure; and that they be provided with the required management tools for controlling development efforts and documenting results.

Thus, Management of software engineering consists of managing human creativity so as to maximize productivity. This necessitates identifying what productivity is. Brooks has shown that it is not necessarily proportional to charged man-hours.[1] I submit that productivity is primarily a product of management policy as implemented in organizational structure. I further submit that what is implemented in these key areas is a manifestation of management philosophy. The relationship between philosophy, policy and organizational structure is depicted in Figure 1.

---

1 F.P. Brooks, Jr., "The Mythical Man-Month", Addison-Wesley Pub. Co., Reading, Mass., 1975, pp. 83-90.

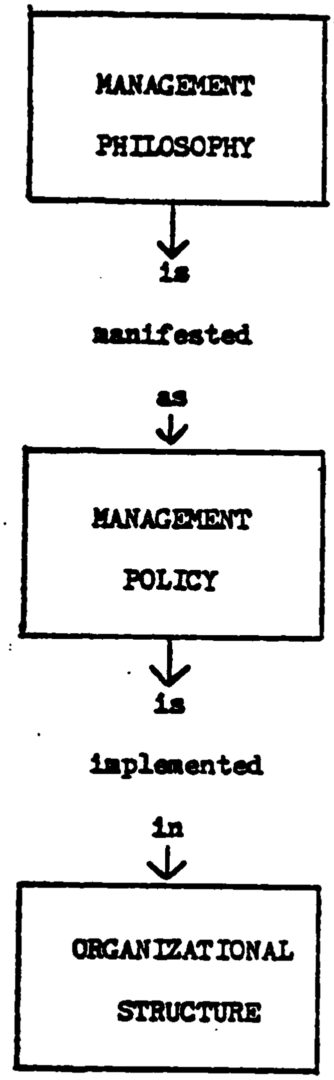


FIGURE 1

RELATIONSHIP BETWEEN MANAGEMENT PHILOSOPHY

POLICY AND ORGANIZATIONAL STRUCTURE

Since management philosophy is at the core of the solution to the software development management problem, it is vital that the optimum philosophy be selected, asserted, and understood. What is the perceived purpose of the people-management (or more properly stated, LEADERSHIP) function?

I subscribe to the philosophy advanced by Greenleaf a distinguished leader in the management area, who states that the true leader has to be a servant first.[1]

A philosophy of servant-leadership translates into a management policy that facilitates and fosters creativity thereby maximizing productivity. Such a policy is not altruistic but rather based on sound principles of effectiveness and efficiency.

---

1 Robert K. Greenleaf, "Servant Leadership; A Journey into the Nature of Legitimate Power and Greatness", Paulist Press, New York, N.Y., 1977. Mr. Greenleaf has served as director of management research at A.T.&T. for seven years; teaching positions at Dartmouth College and Harvard University; consultant to M.I.T., Ohio University, Ford Foundation, R.K. Mellon Foundation, Lilly Endowment, Brookings Institution, and the American Foundation for Management Research.

1.4 Management policy and Organizational structure

A supportive management policy is implemented chiefly through an organization which provides managers with authority commensurate with their responsibilities, and which promotes and rewards free exchange of information between individuals, sections and departments.

Organizational structure determines the flow of control - who controls which resources (human and others), and who has to account for which results. It also determines the flow of information - who has access to what, and what information is collected.

There are practically as many different organizational structures as there are managers. However, three distinct types (project, functional, and matrix)[1] emerge, and other structures can be considered to be a composite of the basic types. These are best illustrated by an example. Let us assume a new development organization is required to develop two projects: Project A and Project B. Each project has three major functions to perform: real-time software development (operating systems), support software development (compilers), and hardware development (computers).

---

1 Daly, op. cit., p. 10

Figure 2 shows six separate organizational entities, one entity for each technology for each project. Now the manner in which we combine these separate organizations will give us a project organization structure, a functional organization structure, or a matrix organisation structure.

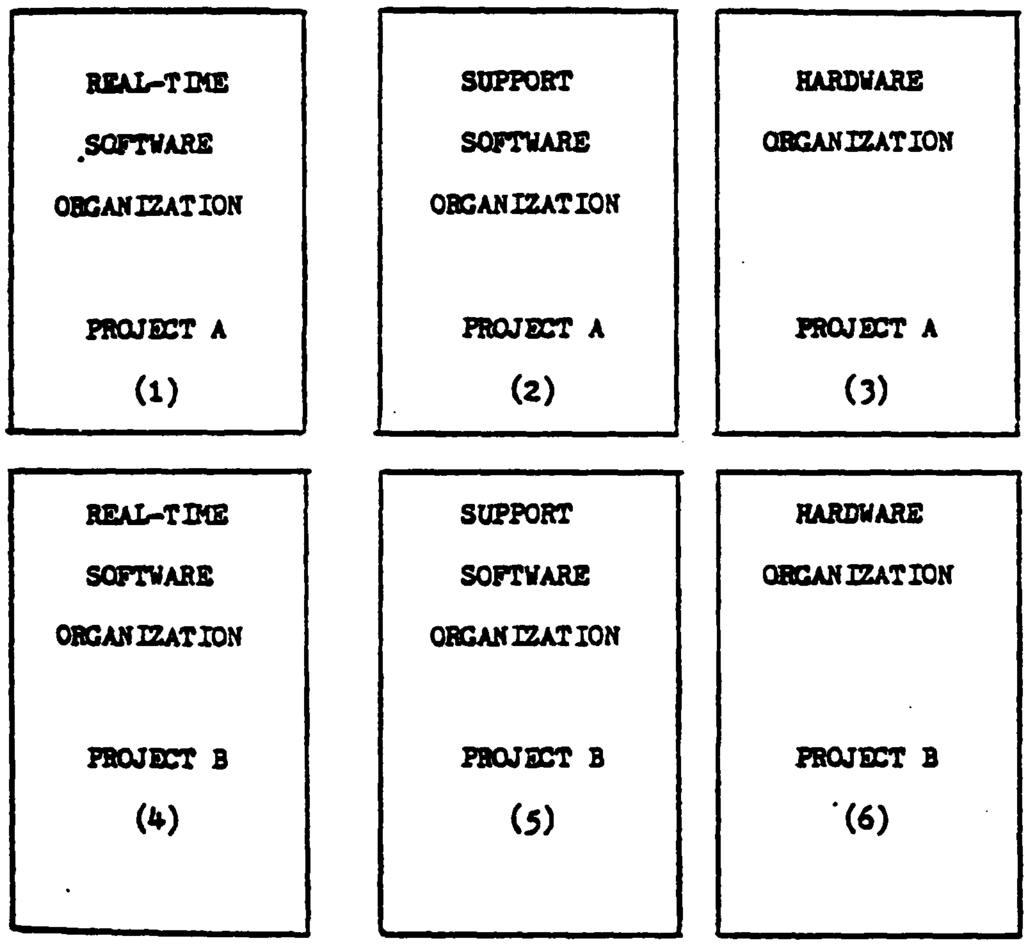


FIGURE 2

SIX ORGANIZATIONAL ENTITIES

In a project organization structure (Figure 3) all resources required to complete a project are organized under a single line manager who performs both the technical and administrative functions. A functional organization (Figure 4) groups all the people associated with one specialty under a functional manager (e.g., all real-time software development for all projects). A matrix organization (Figure 5) attempts to incorporate the advantages of the other two basic structures, project and functional. Personnel are grouped functionally for technical and administrative purposes, but are responsive to a project manager. The project manager decides what will be done, while the functional manager decides how to do the job, and supplies all resources.

The organizational structures that will be examined using the quantitative graph model will be made up of elements of the above three basic types in varying degrees, and therefore will be advantageous or disadvantageous depending on the value/cost/effort associated with the model parameters. The parameter values, in turn, will be determined by the management exigencies of a given project.

FIGURE 3

PROJECT ORGANIZATION

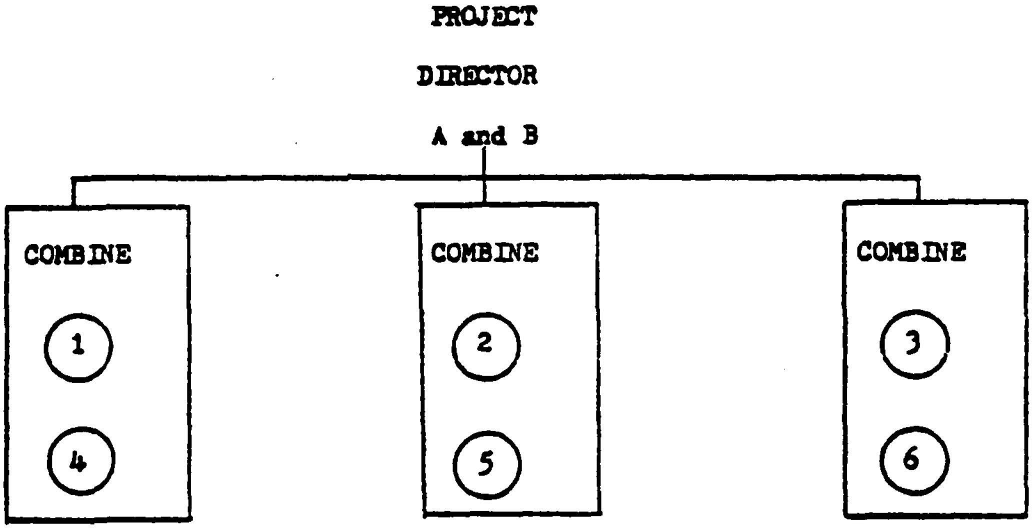


FIGURE 4

FUNCTIONAL ORGANIZATION

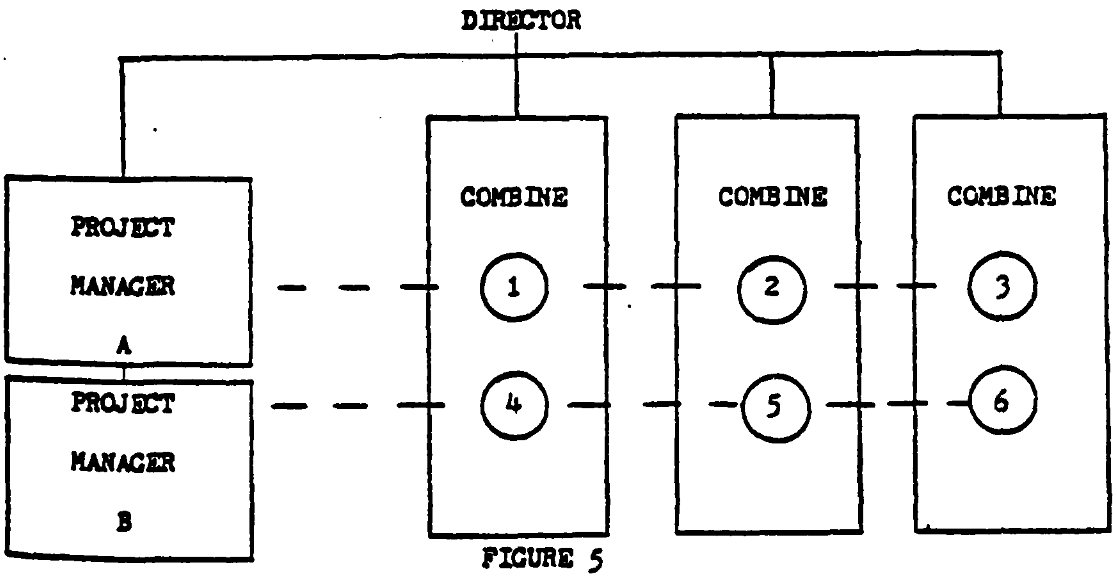


FIGURE 5

MATRIX ORGANIZATION

CHAPTER 2

QUANTITATIVE MODEL THEORY

2.1 General - Productivity models

Most quantitative models which examine the effect of organizational structure on productivity assume a "worst case" situation. Shooman[1] observes that software development productivity is not a direct function of charged time. Charged time represents raw man hours composed of personal time (coffee breaks, conversations, etc.), communication time, and lastly, productive time. He assumes that the proportion of personal time is fixed at 10% regardless of the organizational structure. However, the remaining time divisions are highly dependent on the organizational structure. He proposes a model which breaks the total time (T) into development time (Td), and communication time (Tc).

$$T = Td + Tc \qquad (2.1)$$

He then postulates that for a project team consisting of Nd workers, every team member communicates with every other team member. Thus the number of interfaces is the number of combinations of Nd taken two at a time.

$$\binom{Nd}{2} = \frac{Nd!}{2!\ Nd!} = \frac{Nd\ (Nd - 1)}{2} \qquad (2.2)$$

If there are L total lines of code to be developed and Td is measured in months, then the productivity (P) in lines/month is given by

$$P = \frac{L}{Nd\ Td} \qquad (2.3)$$

---

1 M.L. Shooman, "Software Engineering: Reliability, Design, Management", McGraw-Hill, New York, 1980."

If we assume that a certain fraction (K) of the total work time (T) is spent communication with <u>each</u> interface, then Tc the total communication time for all the interfaces is given by

$$Tc = KT\ Nd\ (Nd - 1)/2 \tag{2.4}$$

Substituting equation 2.4 in equation 2.1 and solving for T yields

$$T = \frac{Td}{1 - KNd\ (Nd - 1)/2} \tag{2.5}$$

Multiplying both sides of equation 2.5 by Nd and substituting from equation 2.3 yields

$$NdT = \frac{L/P}{1 - KNd\ (Nd - 1)/2} \tag{2.6}$$

The numerator in equation 2.6 predicts a linear variation in man months with program length; however, the denomination factor produces a plot which curves upward indicating a decrease in productivity due to the increase in Nd for larger programs.

Similarly, Tausworthe[1] measures software team productivity by defining "index of productivity" (P) in terms of the total number of lines of code (L), number of workers on the project (W), and the average time each worker spent developing the software (T) by the formula

$$P = \frac{L}{WT} \tag{2.7}$$

T is then split into productive time (Tp) and non-productive time (Tnp) spent interfacing with each of the other team members.

$$T = Tp + (W - 1)\ Tnp \tag{2.8}$$

---

1 R.C. Tausworthe, "Standardized Development of Computer Software", Prentice Hall, New Jersey, 1977, Chapter 10.

He then postulates that the individual productivity level (Pi) that each team member must sustain during his "productive" time periods so that the team have overall productivity P is given by

$$Pi = \frac{L}{Tp} = \frac{WP}{1 - (W - 1)(Tnp/T)} \qquad (2.9)$$

After extensive formula manipulation he arrives at the conclusion that the amount of code that a project can produce per day has a maximum value, found to be

$$\frac{L}{T}_{max} = Pi \; \frac{1 + (Tnp/T)}{2\,(Tnp/T)} \left[\frac{1 + (Tnp/T)}{2} \right] \qquad (2.10)$$

where the figure in braces represents the loss in personnel efficiency. This maximum production rate is achieved when the team size is

$$W = \frac{1 + (Tnp/T)}{2\,(Tnp/T)} \qquad (2.11)$$

He then concludes that a project hoping to deliver L lines within time T using W workers having individual integrated-task productivities Pi must keep their non-productive index (Tnp/T) within the bound

$$\frac{Tnp}{T} < \frac{1 - (L/WTPi)}{W - 1} \qquad (2.12)$$

if there is to be success.

2.2 Quantitative graph model

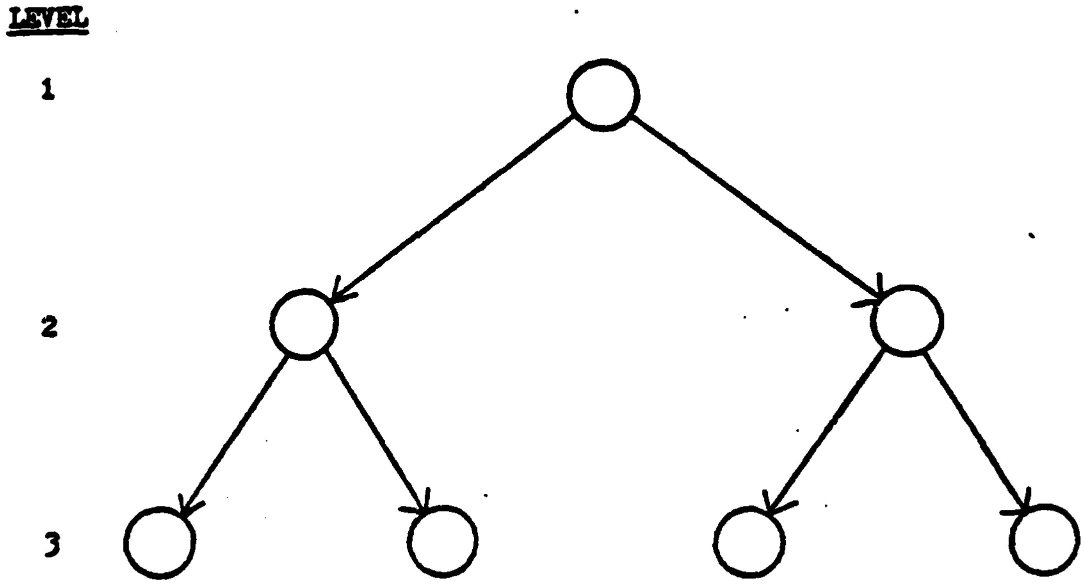
2.2.1 Concept

The preceding two models assume that each team member interacts with every other team member. What if the team is organized into a different structure? A quantitative model has been developed[1] which utilizes graphs of the organizational flow of control and flow of information structures to evaluate the effect of alternate organizational structures on the number of communication paths, and on the volume of information flowing through each path. The concept is illustrated in Figure 6. The lateral paths in the information flow model are analogous to the pre-review discussions implemented in the "Generic Engineer" concept,[2] and are considered to be essential to effective monitoring of progress. The models can easily be extended or transported to desired level of detail.

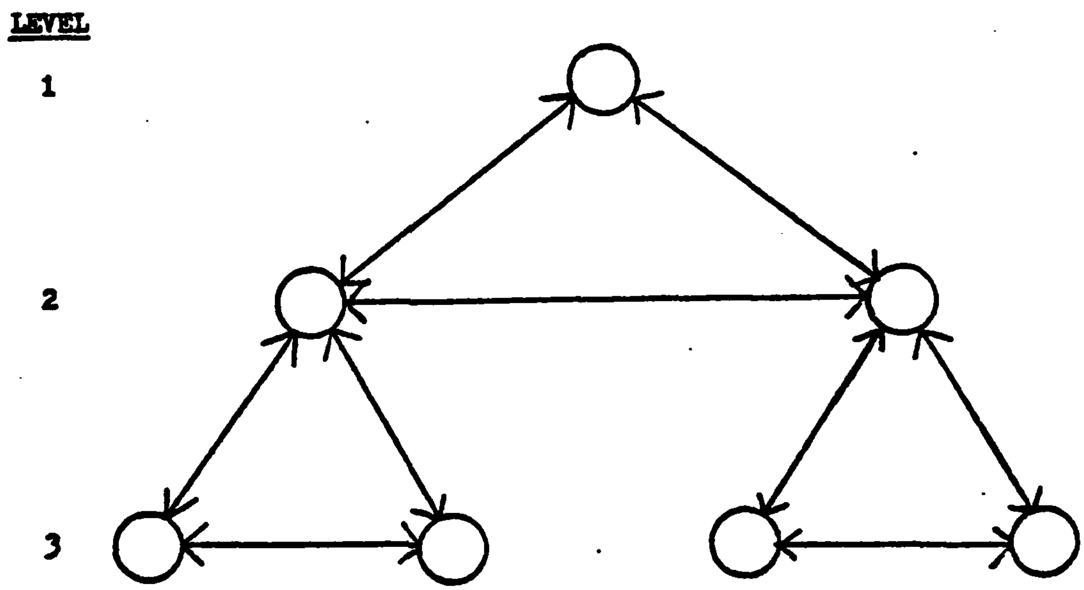
---

1 A. Kershenbaum, "Software Management Models: A Graph Theoretic Approach to System Morphology", Unpublished summary, presented at the SOFTY research meeting, Feb. 5, 1979, Polytechnic Institute of New York, Farmingdale, N.Y.

2 J.B. Synnott, III, (Bell Laboratories), "Managing Software Development - Requirements to Delivery", Proceedings, Computer Software and Application Conference, 78, Palmer House, Chicago, p. 19.



(a) FLOW OF CONTROL



(b) FLOW OF INFORMATION

FIGURE 6

QUANTITATIVE GRAPH MODEL

2.2.2 Graph model parameters

2.2.2.1 Basic parameters

To illustrate the graph parameters we will decompose a problem, P, into three subproblems - P1, P2, P3 - of identical size. Thus instead of solving P, we can solve P1, P2 and P3. The three subproblems are in general related to one another, i.e., some effort must be expended in having them communicate with one another or coordinate them. (This may take the form of engineers spending time coordinating their proposed solutions to software subproblems). We wish to study the effect of the shape of the system on its overall cost or complexity. Two alternative organizations will be compared. In case 1 (Figure 7 (a)) each person communicates directly with every other person. In case 2 (Figure 7 (b)) all communication is routed through a central point (P2).

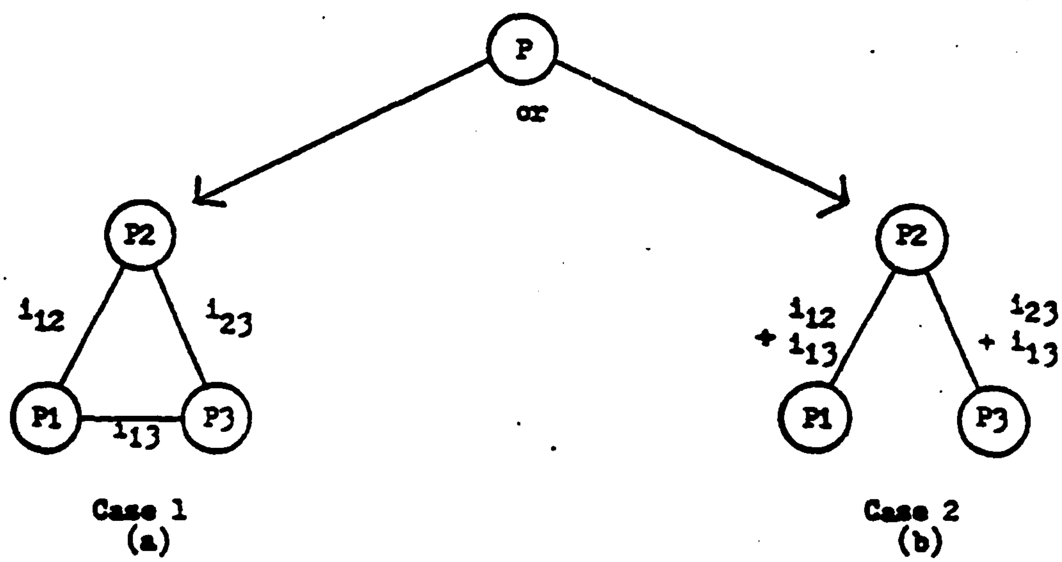


FIGURE 7

Let i_{ab} be the amount of information exchange required between persons a and b in order to complete their tasks. We define a constant, α as the amount of effort per unit of information exchanged, i.e., the total effort required to exchange i_{ab} units of information is αi_{ab}.

We define another constant, β , as the amount of effort associated with the existance of a direct communication path between any pair of persons. This corresponds to the overhead cost of establishing direct communication which is independent of the amount of information exchanged (e.g. the cost of having a meeting, not counting the time spent actually exchanging information). For problem P, letting $i_{ab} = I$ for all a b, (i.e., constant information exchange) it can be shown (see Appendix A) that direct communication (Figure 7 (a)) is more efficient if $\beta < \alpha I$, and the use of an intermediary (Figure 7 (b)) is better for $\alpha I < \beta$. Generalizing, in the case where the i_{ab}'s are different, the use of an intermediary is efficient if

$$\beta > \alpha i_{ab}$$

and there is some c such that direct interfaces between (a and c) and (b and c) exist.

It is apparent that the value assigned to the model parameters, α and β will determine the optimum structure. This is realistic since the importance given to accuracy of information and interpersonal interaction will vary from one project to another.

A gross analysis of the model as developed to this point reveals that:

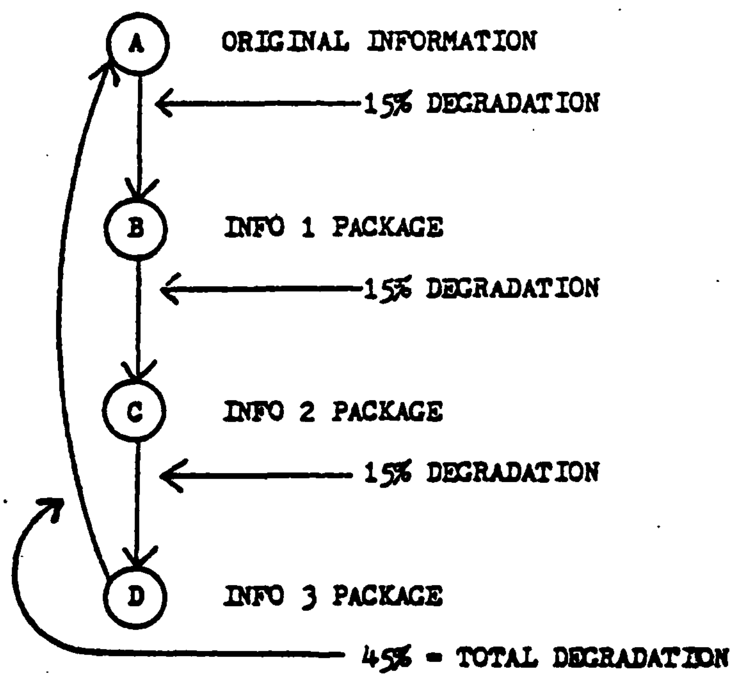
(i) if αi_{ab} is small relative to β - implying either that very little effort is required to transmit one unit of information over one link (α is small), or that very few units of information are required to be exchanged between a and b (i_{ab} is small) - then the use of intermediaries is favored.

(ii) if $\alpha\, i_{ab}$ is large relative to β - implying that the overhead cost of a direct link is small compared to the cost of exchanging information - then the use of direct communication is favored.

2.2.2.2 Refinement of the parameters

2.2.2.2.1 Information degradation

Software engineering management information is transmitted verbally, in writing, and through transcription. The accepted retention rate for information transmitted verbally is between 40% and 60%. The integrity of information transmitted in writing and via transcription is higher. A realistic average degradation of information content for any one transfer of information will therefore be taken to be 15%.



DEGRADATION OF SOFTWARE ENGINEERING

MANAGEMENT INFORMATION UPON TRANSFER

FIGURE 8

We illustrate the concept of information degradation in Figure 8. An original information package is to be exchanged successively among four individuals (A,B,C and D). In being transmitted from A to B the original information content suffers a 15% degradation yielding the new information package, info 1. Info 1 in turn is degraded by 15% in the transmission from B to C yielding info 2. Info 2, when transmitted from C to D, is similarly degraded yielding info 3. If D were to communicate the info 3 package back to A, we would find that little more than half the original information content remains.

Our graph model takes into account information degradation by incorporating it into α, the cost of transmitting one unit of information over one link. The degradation cost is essentially the cost of checking for and correcting errors. Thus, information degradation is directly proportional to the number of information units transmitted, and to the number of links over which the information is propagated.

2.2.2.2.2 Capacity constraints

There is a limit to the number of interfaces that one individual can handle. (Were this not so, then the lower boundary on the number of links required would be represented by the case in which there exists only one intermediary through which everyone communicates.) This capacity constraint has been studied extensively for organizational flow of control. Theories and techniques for optimizing an individual's span of control abound. In an organization where the work is simple, routine, and repetitive - like the basic kind of assembly work - a supervisor might be able to handle 25 to 30 people and do all the necessary supervisory work. If, however, the work managed is variable the supervisor must spend more time to set objectives, to train, and to put in new methods, and consequently cannot handle as many people.

In the realm of software engineering management is quite a complex function; therefore, an individual supervisor can supervise only a limited number of people. Edward Schleh[1] attacks some traditional methods of "spanning the gap". He refers specifically to the tendency among companies to feel that they have so many supervisors at the first level that they could easily cut out one or two, have each remaining one handle a little more, and still get by. This is an illusion. One large paper plant did this and found, within three years, that its costs increased 15 percent, and quality slipped. Costs and quality improved only

---

1 Edward C. Schleh, "Managing for Success: Capitalizing on each individual" IEEE Engineering Management Review, Volume 7, Number 4, December 1979, pp.33-41.

after the span of control for each foreman was decreased and each could handle his work.

On the other hand, executives often fail to grasp that many of their communication problems come from too long a management chain. If the first supervisory level is not beyond its span of control, the second and third can handle many more managers. In one plant a superintendent supervised four foremen. When the foremen were set up with more manageable span of control and trained to supervise, a superintendent could supervise eight to nine foremen. Communication problems up and down the line were greatly decreased because problems were solved in most cases by the foremen.

When applied to organizational flow of control and organizational flow of information graphs, capacity constraints may cause an otherwise optimal solution to become unfeasible. For example, the lower bound solution of making one person an intermediary for all others, generally optimal for $\beta >> \alpha$ may be unfeasible as it places a tremendous burden on the intermediary and may violate his capacity constraints. In such a situation, (not uncommon in the functional organization structure), it may be necessary to introduce additional personnel strictly as intermediaries in order to satisfy the capacity constraints. For a given person (a), if we let his total capacity equal Ca and the capacity required to solve the subproblem assigned to him equal to Ra, then his spare capacity available for communication (Sa) is equal to:

$$Sa = Ca - Ra$$

In order for a feasible solution to exist

$$Sa \geq \left(\sum_b \alpha \, i_{ab} \right) + \beta$$

i.e., each person must have at least enough spare capacity to handle his own communication requirements plus one interface to someone else.

CHAPTER 3

APPLICATION OF THE QUANTITATIVE GRAPH MODEL TO SELECTED ORGANIZATIONAL STRUCTURES

3.1 General

The quantitative graph model will be applied to the three basic organization structures depicted in Figures 3, 4 and 5 (project, functional and matrix), and to selected composite organizational structures. The flow of control graph will be taken as forming a lower constraint on the number of links required for transmitting information without violating the capacity constraints of the persons involved. Adoption of an upper constraint was more difficult. The theoretical upper limit occurs when there exists a direct link between every pair of persons in the organization. This is obviously not desirable. An intermediate information model which consists of the control graph links plus links for direct lateral exchange of information was selected.

In the remainder of this chapter, the graph models will utilize a solid line (———) to represent a direct link between two individuals, and a broken line (- - - - -) to represent requirements for information exchange which are satisfied via the use of intermediaries (i.e., for which direct links do not exist).

The control and information graphs will be compared for each organizational structure and a crossover point between direct communication and the use of an intermediary established based on the relative values of β and α i.

In a real-life situation the process could be applied either as described above or in reverse. By initially determing the value of α and β for a particular organization structure, and analyzing the flow of information within the organization one would then be able to select the optimum organizational structure which minimizes the total cost/effort of the sum of all the α i's plus the β 's. Figures 9 to 18 graphically depict the organizational information flow within the three basic organization structures and within two real-life organization structures - the Canadian Forces WSM and the Texas Instruments OST structures - which are composites of the basic organization structures.

As an illustration of how the graph models work, let us consider Figure 9 and Figure 10 which depict the information flow within the project organization structure (one of the three basic organization structures). Figure 9 depicts information flow through the control graph model which represents a lower bound on the number of direct links required for transmitting information without violating the capacity constraints of the persons involved.

The solid lines represent actual direct links available for transmitting information. Both the solid lines and the broken lines represent a requirement to exchange information between the individuals joined by the line. Hence the information exchange requirements $i_{a,b}$ for the case in which a and b are joined by a broken line must be transmitted along an existing solid line (e.g., (1) has a requirement to exchange information with (4) as indicated by the broken line joining them. Since no direct link exists between (1) and (4) the unit of information $i_{1,4}$ must be transmitted successively along graph paths 1,2 and 2,4.) (note: graph path a,b represents a direct link between a and b along which information can be transmitted, and is depicted by a solid line). Figure 9 includes the information flow along graph path 1,2 ($i_{1,2} + i_{1,4} + i_{1,5} + i_{1,6} + i_{2,3}$) and along graph path 3,9 ($i_{1,9} + i_{3,9} + i_{7,9} + i_{8,9}$) for illustrative purposes only. Table 1 lists all the graph paths along with the information transmitted along each.

Figure 10 depicts information flow through the information graph model which contains the direct links found in the control graph model (Figure 9) plus direct links for lateral exchange of information. The information exchange requirements are identical to those found in the control model. Therefore, utilization of the information graph model results in an <u>increase</u> in the number of direct links between persons (the lateral links), and a corresponding <u>decrease</u> in the number of graph paths along which the units of information have to be transmitted. Table 2 lists all the graph paths found in Figure 10 along with the information transmitted along each. Similarly, Figures 11 to 18 and Tables 3 to 6 and 8 to 11 depict the information flow within the remaining basic and composite organization structures.

3.2 Basic organization structures

3.2.1 Project organization structure graph models

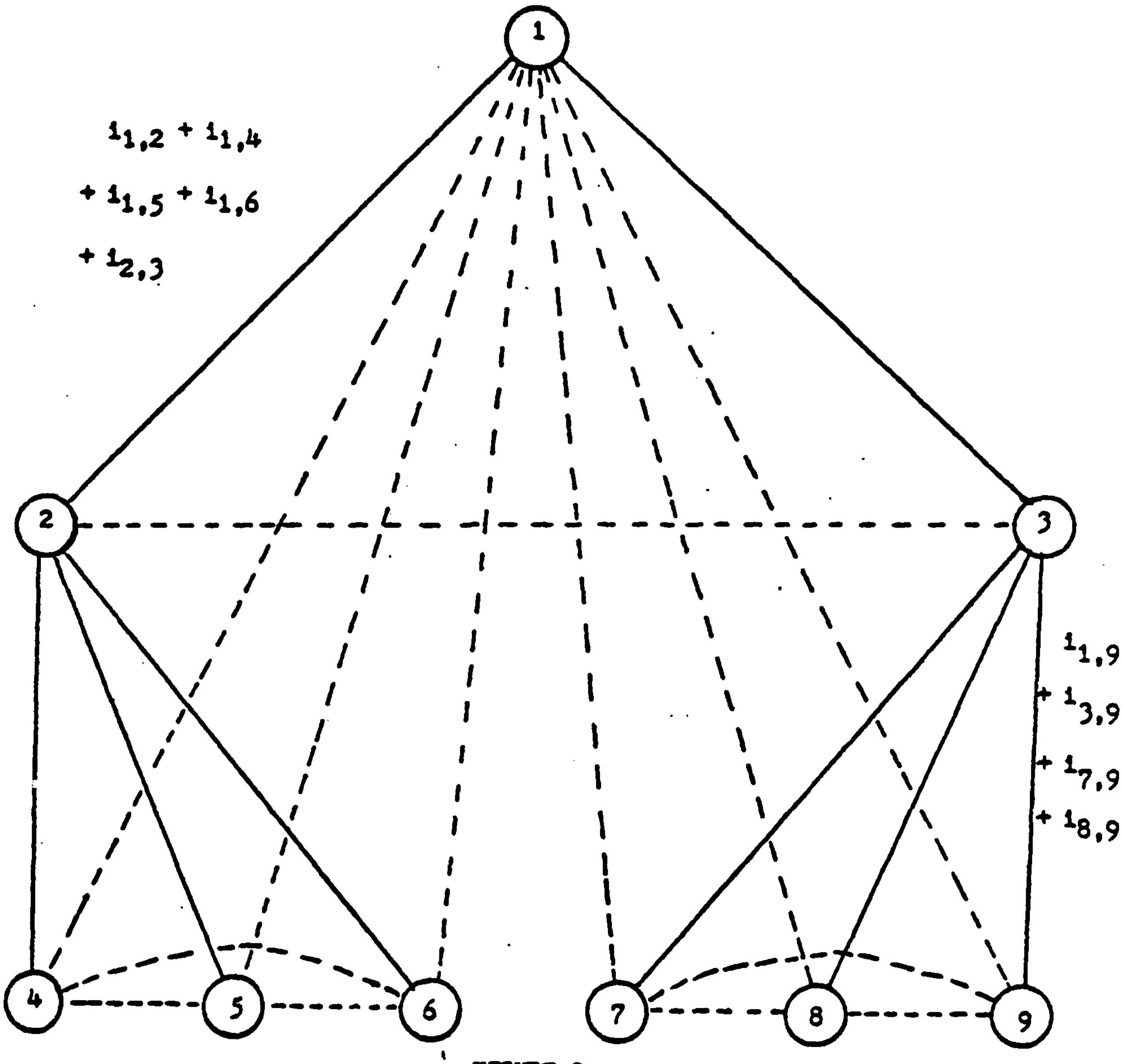


FIGURE 9

PROJECT ORGANIZATION STRUCTURE

CONTROL GRAPH MODEL

TABLE 1

PROJECT ORGANIZATION STRUCTURE CONTROL MODEL

GRAPH PATH	INFORMATION TRANSMITTED
1,2	$i_{1,2} + i_{1,4} + i_{1,5} + i_{1,6} + i_{2,3}$
1,3	$i_{1,3} + i_{1,7} + i_{1,8} + i_{1,9} + i_{2,3}$
2,4	$i_{1,4} + i_{2,4} + i_{4,5} + i_{4,6}$
2,5	$i_{1,5} + i_{2,5} + i_{4,5} + i_{5,6}$
2,6	$i_{1,6} + i_{2,6} + i_{4,6} + i_{5,6}$
3,7	$i_{1,7} + i_{3,7} + i_{7,8} + i_{7,9}$
3,8	$i_{1,8} + i_{3,8} + i_{7,8} + i_{8,9}$
3,9	$i_{1,9} + i_{3,9} + i_{7,9} + i_{8,9}$

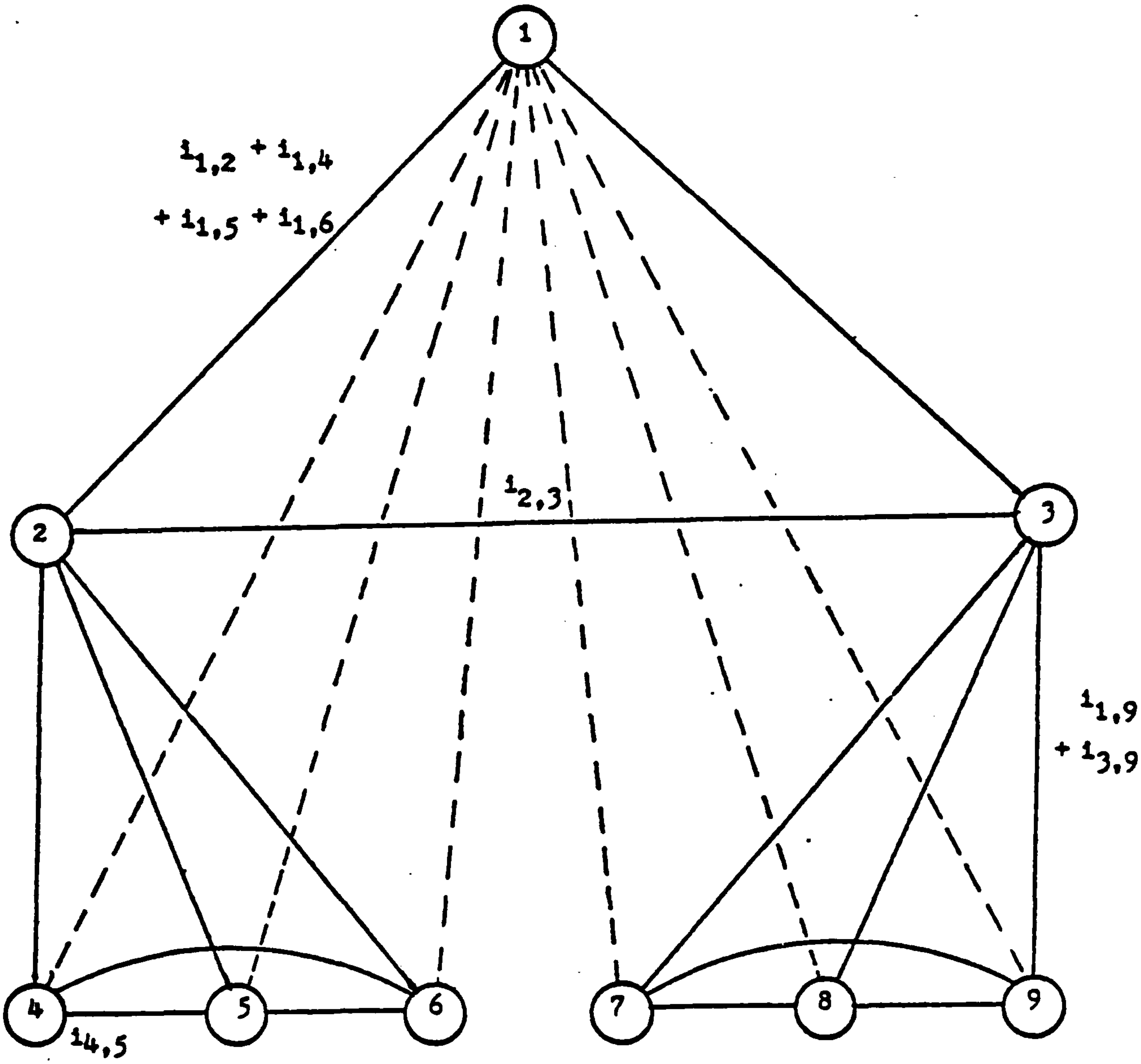


FIGURE 10

PROJECT ORGANIZATION STRUCTURE

INFORMATION GRAPH MODEL

TABLE 2

PROJECT ORGANIZATION STRUCTURE INFORMATION MODEL

GRAPH PATH	INFORMATION TRANSMITTED
1,2	$i_{1,2} + i_{1,4} + i_{1,5} + i_{1,6}$
1,3	$i_{1,3} + i_{1,7} + i_{1,8} + i_{1,9}$
2,3	$i_{2,3}$
2,4	$i_{1,4} + i_{2,4}$
2,5	$i_{1,5} + i_{2,5}$
2,6	$i_{1,6} + i_{2,6}$
3,7	$i_{1,7} + i_{3,7}$
3,8	$i_{1,8} + i_{3,8}$
3,9	$i_{1,9} + i_{3,9}$
4,5	$i_{4,5}$
4,6	$i_{4,6}$
5,6	$i_{5,6}$
7,8	$i_{7,8}$
7,9	$i_{7,9}$
8,9	$i_{8,9}$

3.2.2 **Functional organization structure graph models**

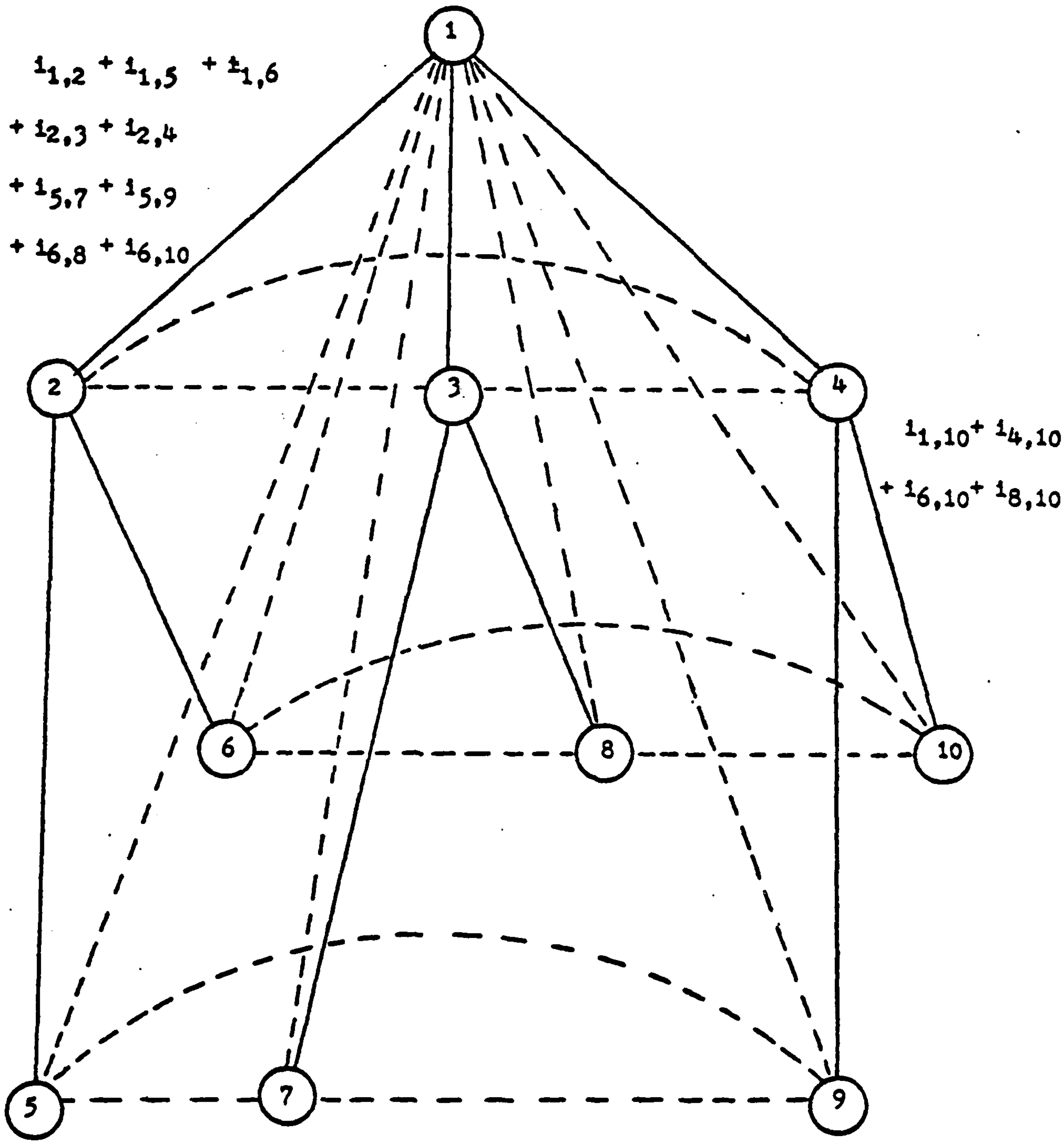


FIGURE 11

FUNCTIONAL ORGANIZATION STRUCTURE

CONTROL GRAPH MODEL

TABLE 3

FUNCTIONAL ORGANIZATION STRUCTURE CONTROL MODEL

GRAPH PATH	INFORMATION TRANSMITTED
1,2	$i_{1,2} + i_{1,5} + i_{1,6}$
	$+ i_{2,3} + i_{2,4}$
	$+ i_{5,7} + i_{5,9}$
	$+ i_{6,8} + i_{6,10}$
1,3	$i_{1,3} + i_{1,7} + i_{1,8}$
	$+ i_{2,3} + i_{3,4}$
	$+ i_{5,7} + i_{7,9}$
	$+ i_{6,8} + i_{8,10}$
1,4	$i_{1,4} + i_{1,9} + i_{1,10}$
	$+ i_{2,4} + i_{3,4}$
	$+ i_{5,9} + i_{7,9}$
	$+ i_{6,10} + i_{8,10}$
2,5	$i_{1,5} + i_{2,5} + i_{5,7} + i_{5,9}$
2,6	$i_{1,6} + i_{2,6} + i_{6,8} + i_{6,10}$
3,7	$i_{1,7} + i_{3,7} + i_{5,7} + i_{7,9}$
3,8	$i_{1,8} + i_{3,8} + i_{6,8} + i_{8,10}$
4,9	$i_{1,9} + i_{4,9} + i_{5,9} + i_{7,9}$
4,10	$i_{1,10} + i_{4,10} + i_{6,10} + i_{8,10}$

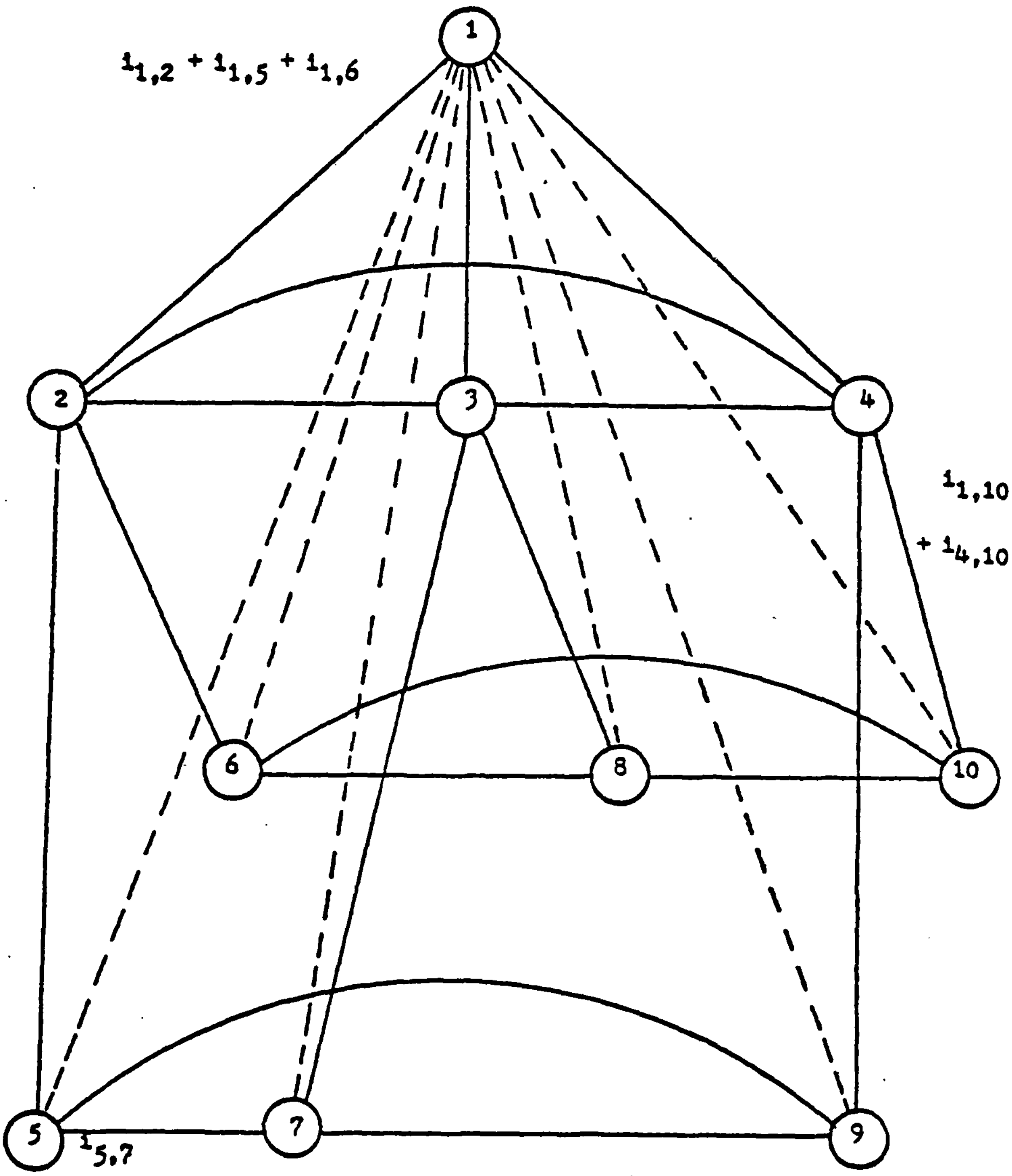


FIGURE 12

FUNCTIONAL ORGANIZATION STRUCTURE

INFORMATION GRAPH MODEL

TABLE 4

FUNCTIONAL ORGANIZATION STRUCTURE INFORMATION MODEL

GRAPH PATH	INFORMATION TRANSMITTED
1,2	$i_{1,2} + i_{1,5} + i_{1,6}$
1,3	$i_{1,3} + i_{1,7} + i_{1,8}$
1,4	$i_{1,4} + i_{1,9} + i_{1,10}$
2,3	$i_{2,3}$
2,4	$i_{2,4}$
2,5	$i_{1,5} + i_{2,5}$
2,6	$i_{1,6} + i_{2,6}$
3,4	$i_{3,4}$
3,7	$i_{1,7} + i_{3,7}$
3,8	$i_{1,8} + i_{3,8}$
4,9	$i_{1,9} + i_{4,9}$
4,10	$i_{1,10} + i_{4,10}$
5,7	$i_{5,7}$
5,9	$i_{5,9}$
6,8	$i_{6,8}$
6,10	$i_{6,10}$

3.2.3 Matrix organization structure

"Figure 13 and Figure 14" notes:

1) Information exchange requirements over two or more organizational levels (e.g. 1,7; 1,13; 4,13) have not been included (as dashes) in the graph, in order to make the graph more readable. However, they are included under "information transmitted" and taken into consideration when calculating the tradeoffs between β and α I.

2) $i_{1,8}$; $i_{1,9}$; $i_{1,10}$; $i_{1,11}$; $i_{1,12}$; and $i_{1,13}$ appear twice. Project information is transmitted via (2) and (3), while technical information is transmitted via (4).

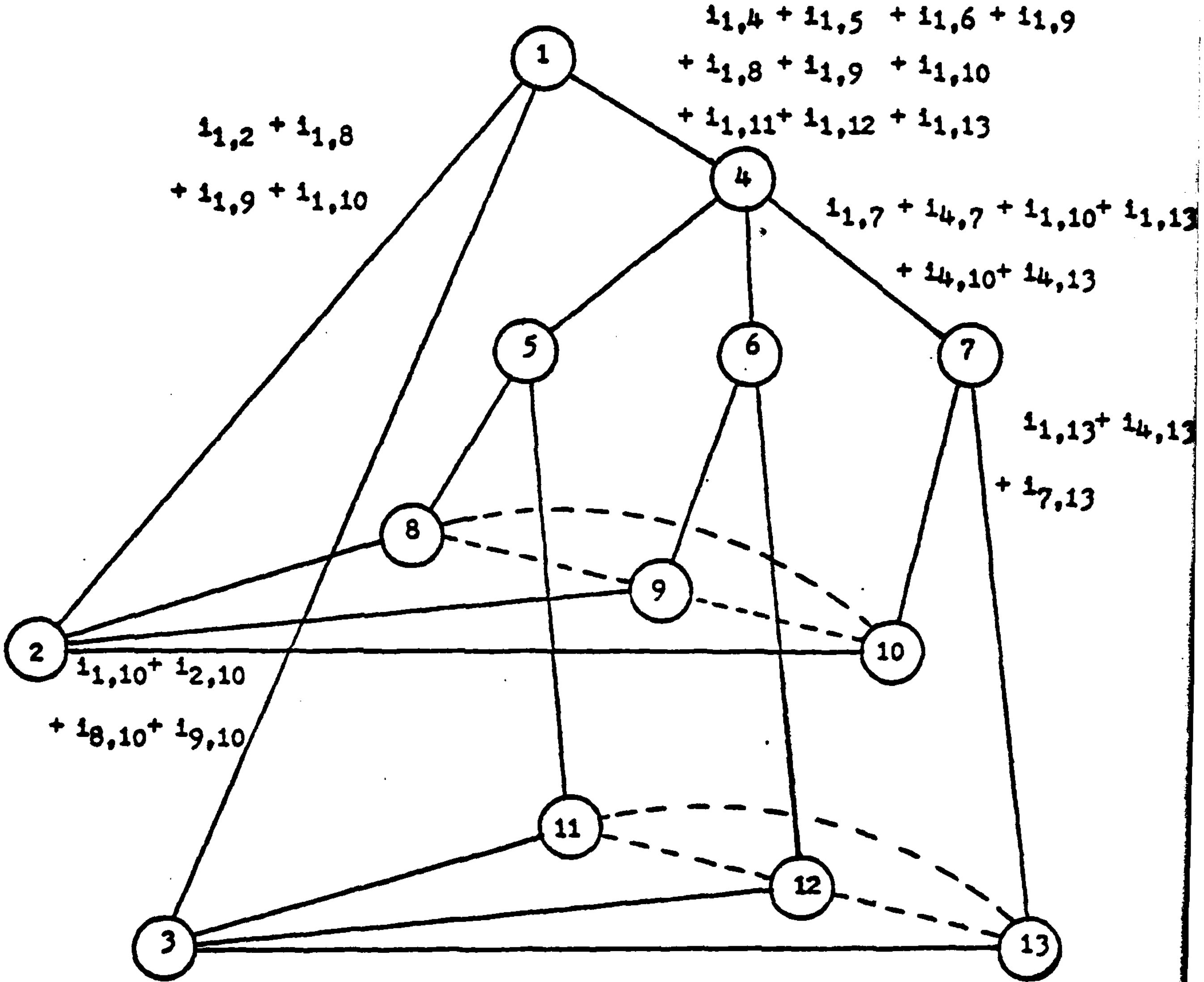


FIGURE 13

MATRIX ORGANIZATION STRUCTURE

CONTROL GRAPH MODEL

TABLE 5

MATRIX ORGANIZATION STRUCTURE CONTROL MODEL

GRAPH PATH	INFORMATION TRANSMITTED
1,2	$i_{1,2} + i_{1,8} + i_{1,9} + i_{1,10}$
1,3	$i_{1,3} + i_{1,11} + i_{1,12} + i_{1,13}$
1,4	$i_{1,4} + i_{1,5} + i_{1,6} + i_{1,7} + i_{1,8} + i_{1,9} + i_{1,10} + i_{1,11} + i_{1,12} + i_{1,13}$
2,8	$i_{1,8} + i_{2,8} + i_{8,9} + i_{8,10}$
2,9	$i_{1,9} + i_{2,9} + i_{8,9} + i_{9,10}$
2,10	$i_{1,10} + i_{2,10} + i_{8,10} + i_{9,10}$
3,11	$i_{1,11} + i_{3,11} + i_{11,12} + i_{11,13}$
3,12	$i_{1,12} + i_{3,12} + i_{11,12} + i_{12,13}$
3,13	$i_{1,13} + i_{3,13} + i_{11,13} + i_{12,13}$
4,5	$i_{1,5} + i_{4,5} + i_{1,8} + i_{1,11} + i_{4,8} + i_{4,11}$
4,6	$i_{1,6} + i_{4,6} + i_{1,9} + i_{1,12} + i_{4,9} + i_{4,12}$
4,7	$i_{1,7} + i_{4,7} + i_{1,10} + i_{1,13} + i_{4,10} + i_{4,13}$
5,8	$i_{1,8} + i_{4,8} + i_{5,8}$
5,11	$i_{1,11} + i_{4,11} + i_{5,11}$
6,9	$i_{1,9} + i_{4,9} + i_{6,9}$
6,12	$i_{1,12} + i_{4,12} + i_{6,12}$
7,10	$i_{1,10} + i_{4,10} + i_{7,10}$
7,13	$i_{1,13} + i_{4,13} + i_{7,13}$

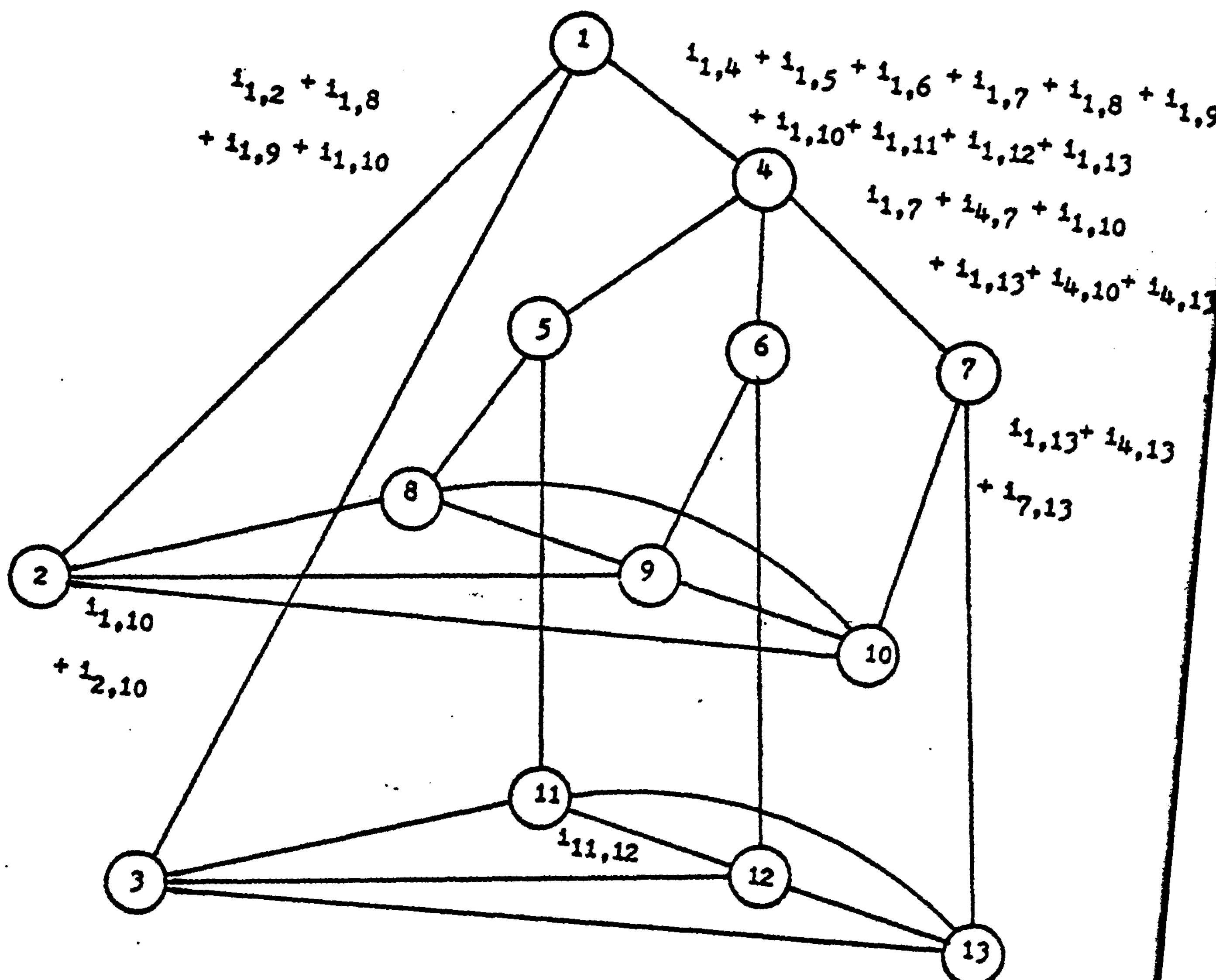


FIGURE 14

MATRIX ORGANIZATION STRUCTURE

INFORMATION GRAPH MODEL

TABLE 6

MATRIX ORGANIZATION STRUCTURE INFORMATION MODEL

GRAPH PATH	INFORMATION TRANSMITTED
1,2	$i_{1,2} + i_{1,8} + i_{1,9} + i_{1,10}$
1,3	$i_{1,3} + i_{1,11} + i_{1,12} + i_{1,13}$
1,4	$i_{1,4} + i_{1,5} + i_{1,6} + i_{1,7} + i_{1,8} + i_{1,9} + i_{1,10} + i_{1,11} + i_{1,12} + i_{1,13}$
2,8	$i_{1,8} + i_{2,8}$
2,9	$i_{1,9} + i_{2,9}$
2,10	$i_{1,10} + i_{2,10}$
3,11	$i_{1,11} + i_{3,11}$
3,12	$i_{1,12} + i_{3,12}$
3,13	$i_{1,13} + i_{3,13}$
4,5	$i_{1,5} + i_{4,5} + i_{1,8} + i_{1,11} + i_{4,8} + i_{4,11}$
4,6	$i_{1,6} + i_{4,6} + i_{1,9} + i_{1,12} + i_{4,9} + i_{4,12}$
4,7	$i_{1,7} + i_{4,7} + i_{1,10} + i_{1,13} + i_{4,10} + i_{4,13}$
5,8	$i_{1,8} + i_{4,8} + i_{5,8}$
5,11	$i_{1,11} + i_{4,11} + i_{5,11}$
6,9	$i_{1,9} + i_{4,9} + i_{6,9}$
6,12	$i_{1,12} + i_{4,12} + i_{6,12}$
7,10	$i_{1,10} + i_{4,10} + i_{7,10}$
7,13	$i_{1,13} + i_{4,13} + i_{7,13}$

TABLE 6 CONTINUED

MATRIX ORGANIZATION STRUCTURE INFORMATION MODEL

GRAPH PATH	INFORMATION TRANSMITTED
8,9	$i_{8,9}$
8,10	$i_{8,10}$
9,10	$i_{9,10}$
11,12	$i_{11,12}$
11,13	$i_{11,13}$
12,13	$i_{12,13}$

3.2.4 Analysis of the basic organization structure graph models

If we let $i_{a,b} = I$ for all a,b (i.e., constant information transfer), then summing the units of information transmitted over all paths in each of the control and information graphs for the project, functional and matrix organization structures (as listed in Tables 1,2, 3,4,5 and 6) yields the results shown in Table 7.

As an example, let us compare (refer to Tables 1, 2 and 7) the effort required to transfer information utilizing the control model or the information model for the project organization structure. The project control model (Table 1) contains 8 graph paths or direct links (8β), and transmits 34 units of information (34I) each of which require an amount of effort α. Thus the total information transfer effort for the model is $34(\alpha I) + \beta$.

The project information model (Table 2) contains 15 graph paths (15β) and transmits 27 units of information (27I) each of which require an amount of effort α. Thus the total information transfer effort for the model is $27(\alpha I) + 15\beta$.

TABLE 7

INFORMATION TRANSFER EFFORT FOR THE BASIC ORGANIZATION STRUCTURES

ORGANIZATION STRUCTURE	INFORMATION TRANSFER EFFORT	
	CONTROL MODEL	INFORMATION MODEL
PROJECT	$34\ (\alpha I) + 8\beta$	$27\ (\alpha I) + 15\beta$
FUNCTIONAL	$51\ (\alpha I) + 9\beta$	$28\ (\alpha I) + 16\beta$
MATRIX	$78\ (\alpha I) + 18\beta$	$72\ (\alpha I) + 24\beta$

If we now let $\alpha I = 1$ (i.e., normalize β with respect to αI) and equate the total effort expended in the control model with the total effort expended in the information model, we obtain the cost-effective cross-over point between direct communication and the use of an intermediary. From Table 7 we see that, for the project organization structure, the information model total effort is $27(\alpha I) + 15\beta$ while the control model total effort is $34(\alpha I) + 8\beta$. Normalizing with respect to αI and equating the two efforts:

$$27 + 15\beta = 34 + 8\beta$$
$$7\beta = 7$$
$$\beta = 1$$

Hence, for the project organization structure the overhead cost of maintaining a direct communication link (β) between a and b must <u>exceed</u> the total cost of exchanging information across the link (αi_{ab}) before the utilization of an intermediary becomes cost-effective. Table 13 shows the trade-offs between the control and information models over a range of values for $\frac{\beta}{\alpha I}$.

Similarly, for the functional organization model the cross-over point occurs at

$$28 + 16\beta = 51 + 9\beta$$
$$7\beta = 23$$
$$\beta = 3.3$$

and for the matrix organization model the cross-over point occurs at

$$72 + 24\beta = 78 + 18\beta$$
$$6\beta = 6$$
$$\beta = 1$$

The cross-over points for the project, functional and matrix organization structures are depicted graphically in Figure 19.

3.3 Composite organization structures

In this section two real-life organization structures will be examined. These do not correspond identically to any one of the three basic structures but rather display features of two or more of the generic structures. The first to be examined will be the Canadian Forces Weapon Systems Software Management (WSM) structure. This will be followed by the Texas Instrument Objectives, Strategies and Tactics (OST) structure.

3.3.1 The Canadian Forces WSM structure

The WSM structure is utilized by the Canadian Forces to manage the maintenance and modification of airborne embedded software.

The organization consists of a field support facility which implements urgent changes as required, and a headquarters facility which controls and implements periodic block changes. Complete and timely information exchange between the "patchers" in the field and the "updaters" at headquarters is vital. Figures 15 and 16, and Tables 8 and 9 document the structure and information flow for the control and information model.

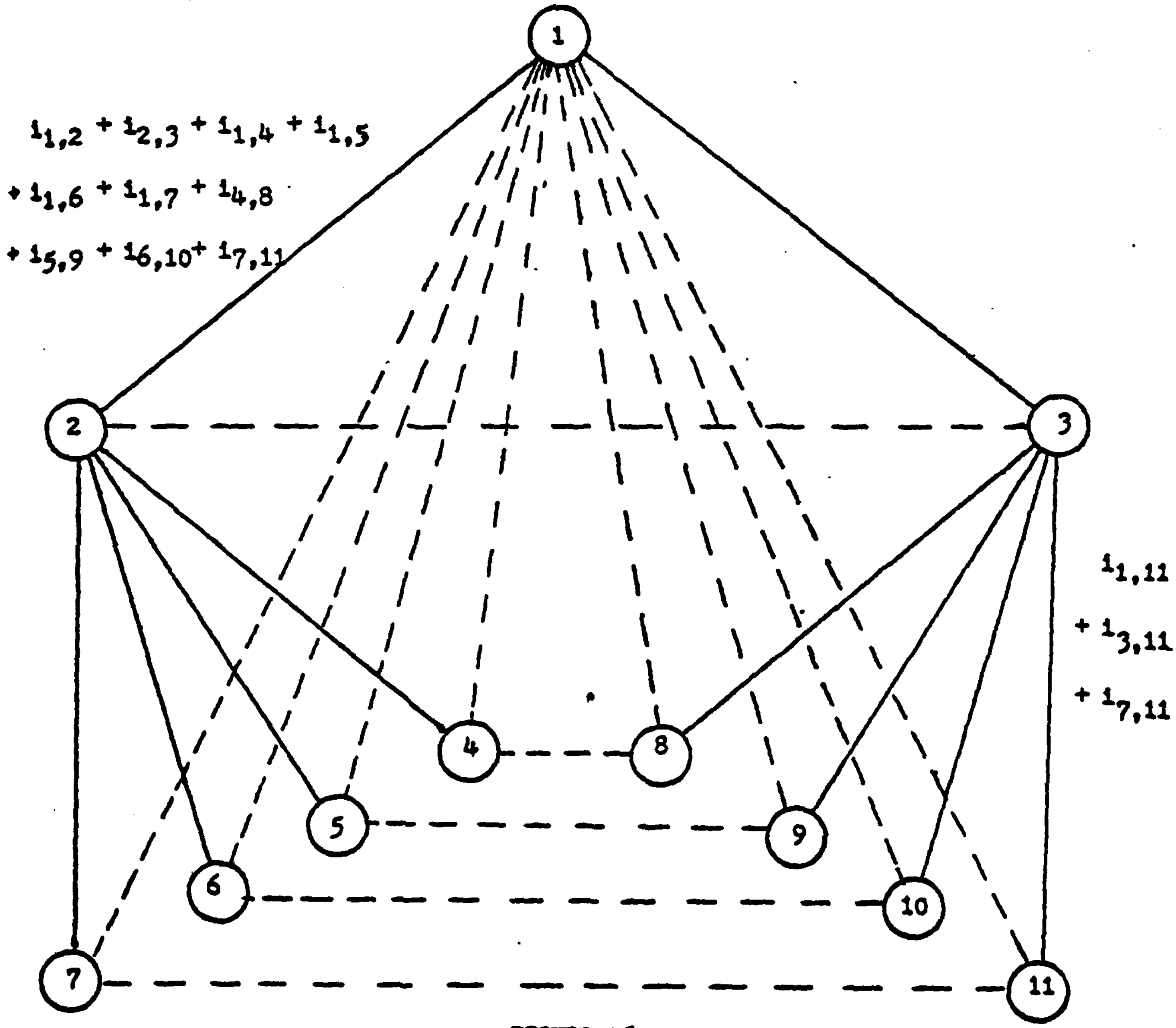


FIGURE 15

CANADIAN FORCES WSM

CONTROL GRAPH MODEL

TABLE 8

CANADIAN FORCES WSM CONTROL MODEL

GRAPH PATH	INFORMATION TRANSMITTED
1,2	$i_{1,2} + i_{2,3} + i_{1,4} + i_{1,5} + i_{1,6} + i_{1,7}$
	$+ i_{4,8} + i_{5,9} + i_{6,10} + i_{7,11}$
1,3	$i_{1,3} + i_{2,3} + i_{1,8} + i_{1,9} + i_{1,10} + i_{1,11}$
	$+ i_{4,8} + i_{5,9} + i_{6,10} + i_{7,11}$
2,4	$i_{1,4} + i_{2,4} + i_{4,8}$
2,5	$i_{1,5} + i_{2,5} + i_{5,9}$
2,6	$i_{1,6} + i_{2,6} + i_{6,10}$
2,7	$i_{1,7} + i_{2,7} + i_{7,11}$
3,8	$i_{1,8} + i_{3,8} + i_{4,8}$
3,9	$i_{1,9} + i_{3,9} + i_{5,9}$
3,10	$i_{1,10} + i_{3,10} + i_{6,10}$
3,11	$i_{1,11} + i_{3,11} + i_{7,11}$

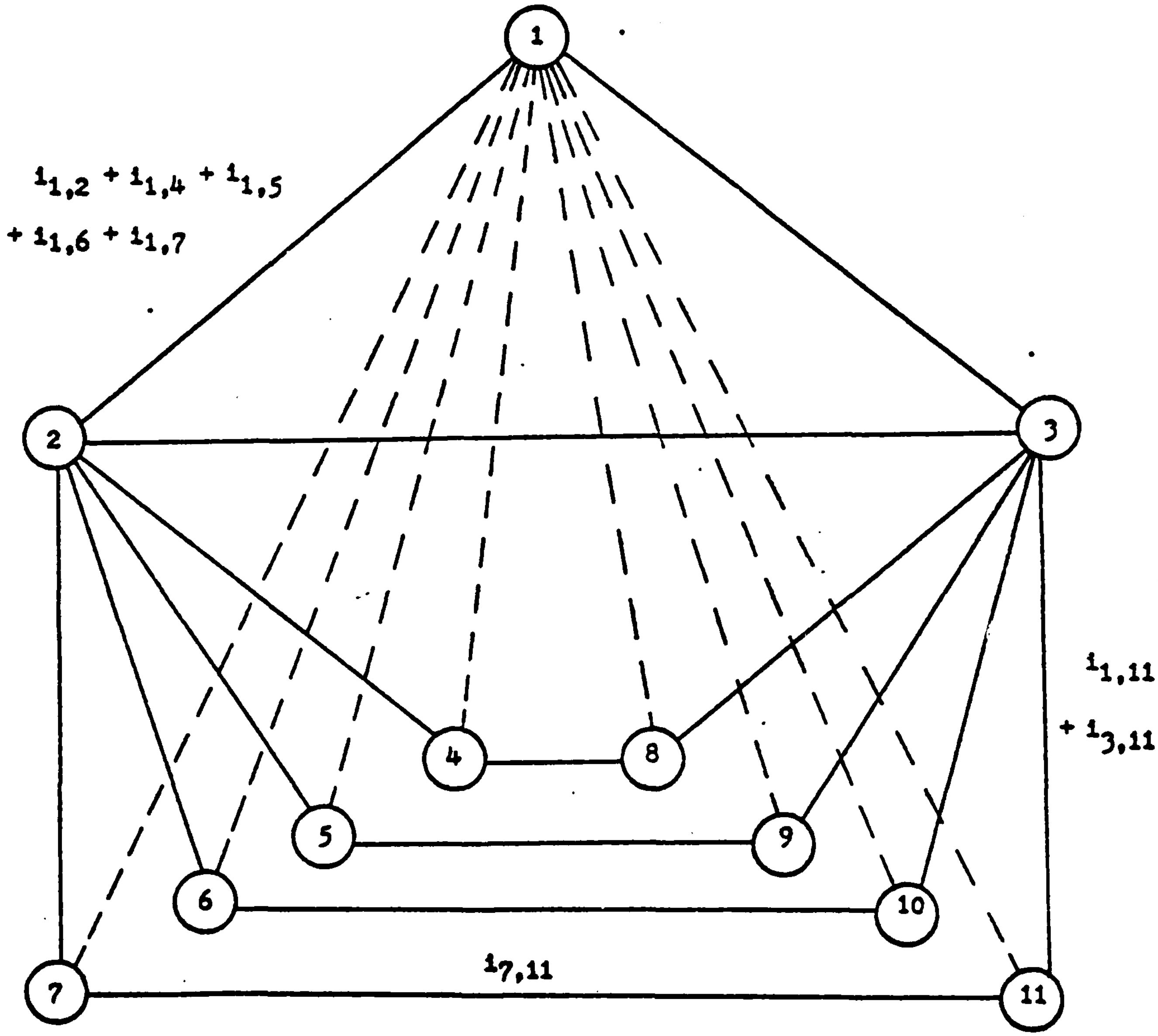


FIGURE 16

CANADIAN FORCES WSM

INFORMATION GRAPH MODEL

TABLE 9

CANADIAN FORCES WSM INFORMATION MODEL

GRAPH PATH	INFORMATION TRANSMITTED
1,2	$i_{1,2} + i_{1,4} + i_{1,5} + i_{1,6} + i_{1,7}$
1,3	$i_{1,3} + i_{1,8} + i_{1,9} + i_{1,10} + i_{1,11}$
2,3	$i_{2,3}$
2,4	$i_{1,4} + i_{2,4}$
2,5	$i_{1,5} + i_{2,5}$
2,6	$i_{1,6} + i_{2,6}$
2,7	$i_{1,7} + i_{2,7}$
3,8	$i_{1,8} + i_{3,8}$
3,9	$i_{1,9} + i_{3,9}$
3,10	$i_{1,10} + i_{3,10}$
3,11	$i_{1,11} + i_{3,11}$
4,8	$i_{4,8}$
5,9	$i_{5,9}$
6,10	$i_{6,10}$
7,11	$i_{7,11}$

(this page left blank intentionally)

3.3.2 The Texas Instruments OST structure

The Objectives, Strategies and Tactics (O.S.T.) system is utilized by Texas Instruments (T.I.) to manage the development and application of innovation.[1] It is T.I.'s way of clearly segregating "strategic" expense from "operating" expense. As a result of the O.S.T. system, two quite different organizational structures coexist at Texas Instruments.

The operating organization, which deals with day-to-day business activities, is a relatively permanent and conventional form of decentralized organization. This structure is overlaid by the OST organization, which is fluid, project-oriented and unbound except by funding limitations. The OST system provides the capability to create strategic programs that attack new opportunities without creating new permanent organizational structures; instead, resources are mobilized to achieve objectives, and then when the job is done, are remobilized in a different matrix for the next problem.

The OST manager wears two hats - one as the head of a strategic organization, the other as the head of a permanent organizational entity.

The OST system utilizes a "project" organizational structure, and overlays the "functional" organizational structure used by the operating organization. Thus it is similar to the "matrix" organization except that the temporary O.S.T. project managers are chosen from among the functional

---

1 Mark Shepherd, Jr. and Fred Bucy, Texas Instruments, "Innovation at Texas Instruments" Computer magazine Vol. 12, No. 9, Sept. 79, pp. 82-90

managers employed in the permanent operating organization. The resulting overlay organization is depicted in Figures 18 and 19. Broken circles represent the permanent operating organization. The solid circles overlaying them represent the temporary CST organization. Only the control structure of the operating organization is represented, and no analysis is performed since it is identical to the "functional" model presented in Figures 11 and 12.

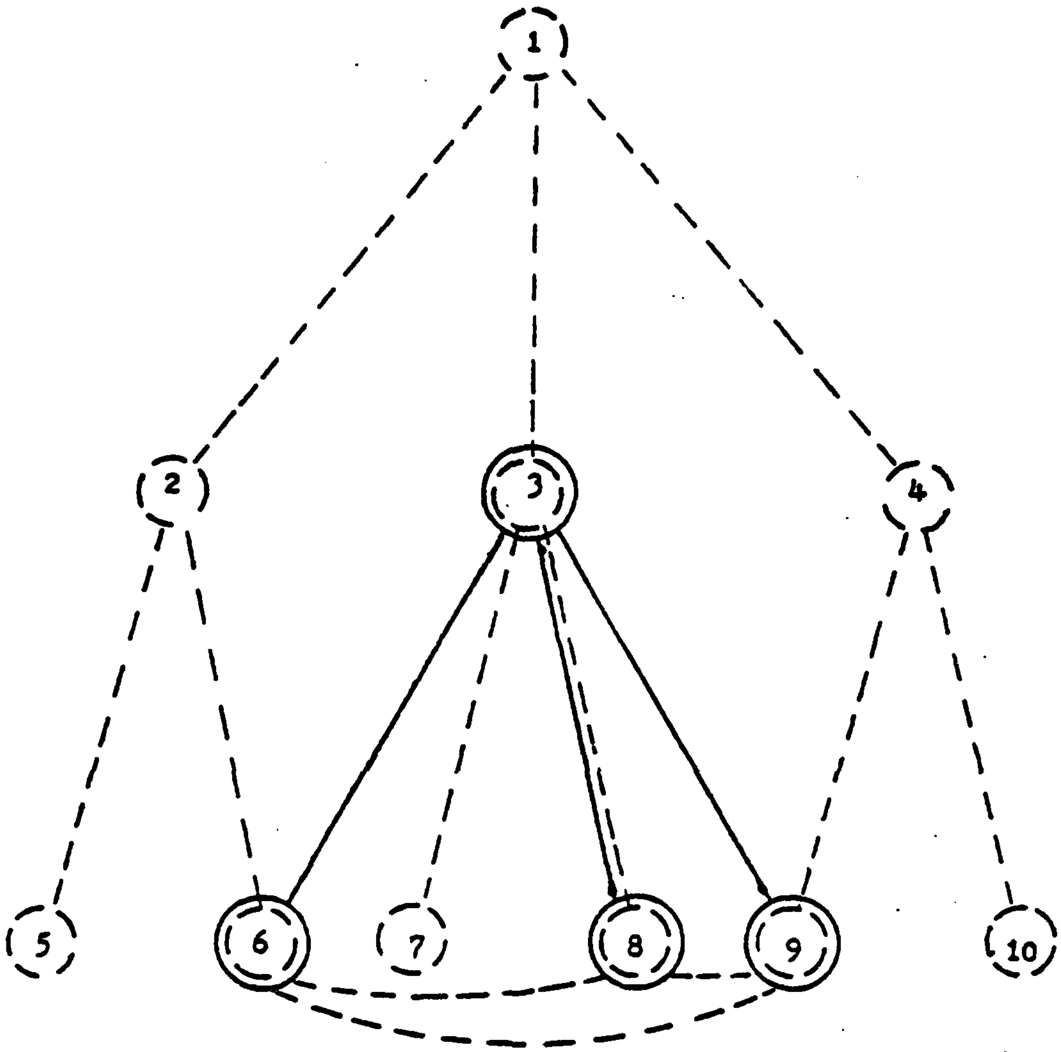


FIGURE 17

TEXAS INSTRUMENTS CST

CONTROL GRAPH MODEL

TABLE 10

TEXAS INSTRUMENTS CST

CONTROL GRAPH MODEL

GRAPH PATH	INFORMATION TRANSMITTED
3,6	$i_{3,6} + i_{6,8} + i_{6,9}$
* 3,8	$i_{3,8} + i_{6,8} + i_{8,9}$
3,9	$i_{3,9} + i_{6,9} + i_{8,9}$

* path already exists from "functional" underlay; hence there is no requirement for an additional direct link (β). However, additional (CST) information is transmitted along the existing link giving rise to α I units of extra effort.

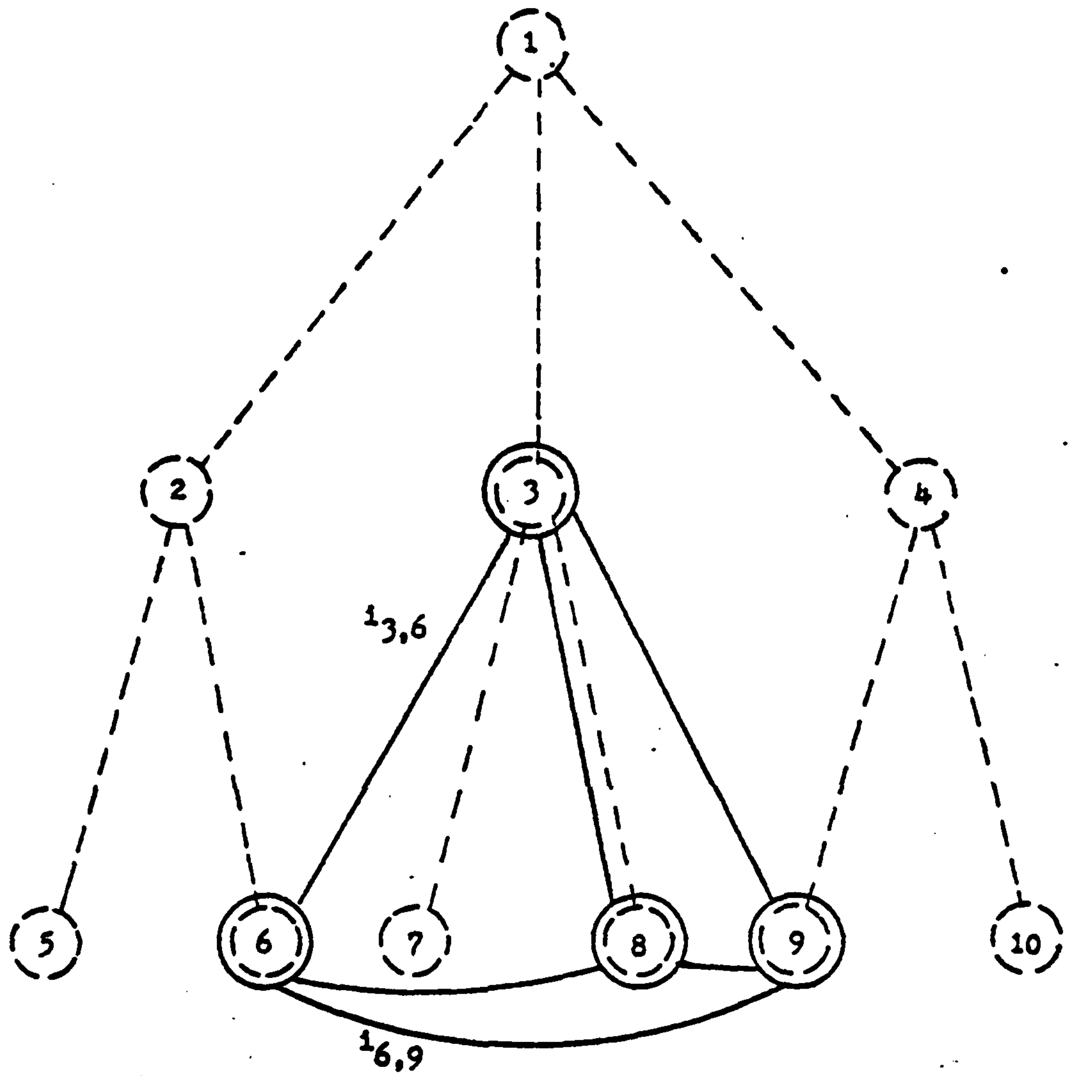


FIGURE 18

TEXAS INSTRUMENT OST

INFORMATION GRAPH MODEL

TABLE 11

TEXAS INSTRUMENT CST
INFORMATION GRAPH MODEL

GRAPH PATH	INFORMATION TRANSMITTED
3,6	$i_{3,6}$
* 3,8	$i_{3,8}$
3,9	$i_{3,9}$
6,8	$i_{6,8}$
6,9	$i_{6,9}$
8,9	$i_{8,9}$

* path already exists from "functional" underlay; hence there is no requirement for an additional direct link (β). However, additional (CST) information is transmitted along the existing link giving rise to α I units of extra effort.

3.3.3 Analysis of the composite organization structure graph models

If we let $i_{a,b} = I$ for all a,b (i.e., constant information transfer), then summing the units of information transmitted over all paths in each of the control and information graphs for the WSM and OST organization structures (Tables 8, 9, 10 and 11) yields the results shown in Table 12.

TABLE 12

INFORMATION TRANSFER EFFORT

FOR THE COMPOSITE ORGANIZATIONAL STRUCTURES

ORGANIZATION STRUCTURE	INFORMATION TRANSFER EFFORT	
	CONTROL MODEL	INFORMATION MODEL
WSM	$44(\alpha I) + 10\beta$	$31(\alpha I) + 15\beta$
OST	$9(\alpha I) + 2\beta$	$6(\alpha I) + 5\beta$

Letting I = 1, we can express relative to I; then equating the total effort expendwd in the control model with the total effort expended in the information model, we obtain the cost-effective cross-over point between direct communication and the use of an intermediary. For the WSM model the cross-over point occurs at

$$31 + 15\beta = 44 + 10\beta$$
$$5\beta = 13$$
$$\beta = 2.6$$

and for the OST model the cross-over point occurs at

$$6 + 5\beta = 9 + 2\beta$$
$$3\beta = 3$$
$$\beta = 1$$

3.4 Analysis of the Basic and Composite organization structures

The information transfer efforts for the basic and composite models over a range of values of $\beta / \alpha I$ are presented in Table 13. Taking the Project organization structure as an example, we see that for a value of $\beta / \alpha I = 1$ (which is determined by the dynamics of a particular organization) the information transfer effort for the control model and the information model is the same and equals 42 units of effort. If the value of $\beta / \alpha I$ is varied up and down by 50% in order to test the sensitivity of the models to the parameters we find that for $\beta / \alpha I = .5$, the information model utilizes 3.5 units of effort less than the control model (34.5 versus 38) while for $\beta / \alpha I = 1.5$ the information model utilizes 3.5 units of effort more than the control model (49.5 versus 46).

TABLE 13

INFORMATION TRANSFER EFFORTS FOR THE BASIC AND COMPOSITE MODELS OVER A RANGE OF VALUES OF β / α I

ORG. STRUCT. (MODEL)	EFFORT FORMULA	β / α I					
		.5	1.0	1.5	2.6	3.3	5
PROJECT (CONTROL)	$34 + 8\beta$	38	42	46			
1.0 (INFORMATION)	$27 + 15\beta$	34.5	42	49.5			
FUNCTIONAL (CONTROL)	$51 + 9\beta$			64.5		80.8	96
3.3 (INFORMATION)	$28 + 16\beta$			52		80.8	108
MATRIX (CONTROL)	$78 + 18\beta$	87	96	105			
1.0 (INFORMATION)	$72 + 24\beta$	84	96	108			
WSM (CONTROL)	$44 + 10\beta$			59	70		94
2.6 (INFORMATION)	$31 + 15\beta$			53.5	70		106
OST (CONTROL)	$9 + 2\beta$	10	11	12			
1.0 (INFORMATION)	$6 + 5\beta$	8.5	11	13.5			
		INFORMATION TRANSFER EFFORT					

○ — represents the cost-effective cross-over points, and their associated transfer efforts.

An examination of Table 13 reveals that for the "Project", "Matrix" and "OST" organizational structures the information model is more effective than the control model for $\beta < \alpha I$ (i.e., the cost of a link is less than the cost of exchanging information). For the "Functional" and the "WSM" organizational structures the information model is more effective for $\beta < 3.3 \alpha I$ and $\beta < 2.6 \alpha I$ respectively.

Figure 19 depicts the cost-effective cross-over points between direct communication links and the use of an intermediary for the basic and composite organization structures.

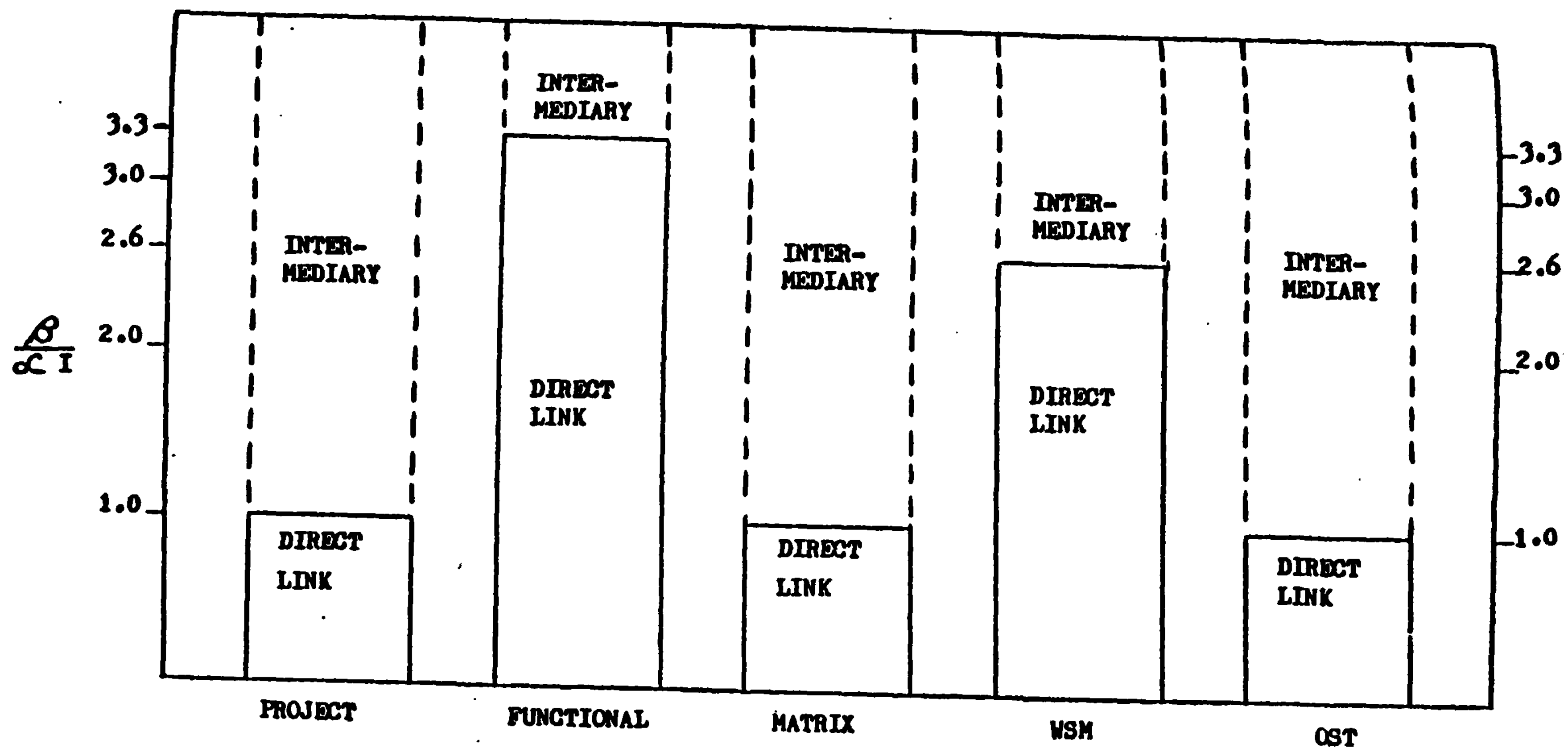


FIGURE 19

VALUES OF β / α I AT WHICH COST-EFFECTIVE CROSS-OVER BETWEEN DIRECT COMMUNICATION LINKS AND THE USE OF AN INTERMEDIARY OCCURS FOR THE BASIC AND COMPOSITE ORGANIZATION STRUCTURES

CHAPTER 4

CONCLUSION AND SUGGESTIONS FOR FUTURE RESEARCH

4.1 Conclusion

This thesis has examined two premises:

a) that the effect of organizational structure on software engineering management can be quantified, and

b) that an organization structure which promotes lateral exchange of information maximizes productivity.

Premise A has been shown to be true. The effort associated with the existence of a direct communication link between two individuals independent of the amount of information transmitted and the effort associated with the transmission of information between these two individuals have been quantified as the parameters β and αI respectively. The quantitative parameters were then applied realistically to organizational structures wherein the constraint on the maximum number of interfaces that an individual can handle effectively was set by the control structure of the organization.

Premise B has been proven true for a subset of values of the organizational parameters β and αI. In Table 13 the value of β relative to αI is varied from 0.5 to 5.0 for both the control and information models of every organizational structure. The information model represents the type of organization advocated by premise B. The effectiveness of lateral exchange of information within an organization is identified as being dependent on the relative value of β with respect to αI, and the cost-effective boundary for its application is determined. Figure 19 graphically depicts the cross-over points for each organization structure.

The fact that the behavior of the innovative "OST" organization structure model can be reduced to that of the basic "project" model shows the innate strength of the quantitative graph models. If we had chosen to analyze (in Figures 18 and 19, and Tables 10 and 11) the underlying operating organization in addition to the OST structure, then the model would have behaved like the basic "Matrix" model. It is therefore concluded that the wide applicability of the quantitative graph models has been demonstrated.

4.2 Suggestions for future research

Certain assumptions and simplifications have been made in this thesis to provide a framework upon which the quantitative graph models could be examined. Two such assumptions are:

a) the relationship between management philosophy, management policy and organization structure depicted in Figure 1; and

b) the linearity of the α's and β's.

Possible future research topics include

a) the distortion of management philosophy when applied as policy, and the distortion of policy when implemented in organization structure,

b) a study of the quantitative graph model behavior when α and β are non-linear, and

c) a comparison of Shooman's and Tausworthe's productivity models.

BIBLIOGRAPHY

1. Brooks, F.P. Jr., "The Mythical Man-Month", Addison-Wesley Pub. Co., Reading, Mass., 1975, pp. 83-90.

2. Daly, Edmund B., "Organizing for successful software development", Datamation, December 1979, pp. 107 to 120.

3. Encyclopedia Britannica, "Engineering" 15th ed., Vol. 6 (Macropaedia), p. 860.

4. Greenleaf, Robert K., "Servant Leadership; A Journey into the Nature of Legitimate Power and Greatness", Paulist Press, New York, N.Y., 1977.

5. Hsia, P., "Software Engineering and Computer Science," Computer magazine Vol. 12, No. 10, Oct. 1979, pp. 87-88.

6. Kershenbaum, A., "Software Management Models: A Graph Theoretic Approach to System Morphology", Unpublished summary, presented at the SOFTY research meeting, Feb. 5, 1979, Polytechnic Institute of New York, Farmingdale, N.Y.

7. Knuth, D., "Computer Programming as an Art" (1974 ACM Turing Award Lecture), CACM, Vol. 17, No. 12, Dec. 1974, pp. 667-673.

8. Musa, J.D., and F.N. Woomer, Jr., "Safeguard Data Processing System: Software Project Management," Bell System Technical Journal, 1975, pp. S245-S259.

9. Schleh, Edward C., "Managing for Success: Capitalizing on each individual" IEEE Engineering Management Review, Volume 7 Number 4, December 1979, pp. 33-41.

10. Shepherd, Mark Jr., and Fred Bucy, Texas Instruments, "Innovation at Texas Instruments" Computer magazine Vol. 12, No. 9, Sept. 79, pp. 82-90.

11. Shooman, M.L., "Software Engineering: Reliability, Design, Management", McGraw-Hill, New York, 1980."

12. Shooman, M.L., and H. Ruston, "Final Report: Software Modelling Studies," Program in Software Engineering Poly-EE-77-C42, SRS112, Sept. 30, 1977, pp. 23 and 24.

13. Synnott, J.B., III, (Bell Laboratories), "Managing Software Development - Requirements to Delivery", Proceedings, Computer Software and Application Conference, 78, Palmer House, Chicago, p. 19.

BIBLIOGRAPHY

14. Tausworthe, R.C., "Standardized Development of Computer Software", Prentice Hall, New Jersey, 1977, Chapter 10.

APPENDIX A

Optimization of decomposition of Problem P (Figure 7), utilizing trade-offs between parameters α and β

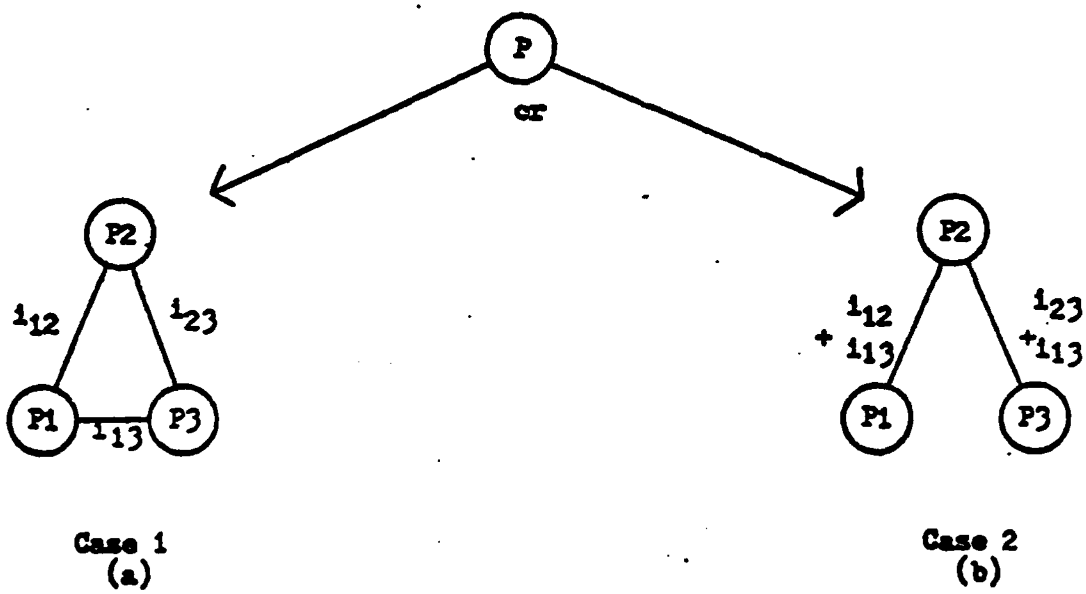


FIGURE 7

(REPRODUCED FROM PAGE 17)

Parameters:

i_{ab} = the amount of information exchange between a and b

α = the constant amount of effort per unit of information exchange.

β = the constant amount of effort associated with the existance of a direct interface between any pair of persons, independent of the amount of information exchanged.

Considering initially only the effort required to exchange information, if all persons communicate directly, as in Figure 7 (a), the total communication effort is minimized and equals,

$$\sum_{a,b} \alpha\, i_{ab} \qquad \text{(A-1)}$$

If, however, some communication takes place through an intermediate point, as in Figure 7 (b), the effort is increased to

$$\sum_{a,b} \alpha\, i_{ab} + \sum_{a,b,c} \alpha\, i'_{ab}(c) \qquad \text{(A-2)}$$

where $i'_{ab}(c) = i_{ab}$ if point c is an intermediary for traffic between a and b, and is zero otherwise. Thus, in the simple case where the effort to communicate is linear in the magnitude of the requirement for information, direct communication is optimal.

However, the cost of the existence of a direct communication link independent of the amount of information exchange (β), must be considered.

Given N persons, the cost associated with the existance of a direct interface between every pair is

$$\tfrac{1}{2}(N)(N-1)\beta \qquad \text{(A-3)}$$

If intermediaries are used to perform some of the interfaces, this effort is reduced to

$$\tfrac{1}{2}(N)(N-1)\beta - \sum_{a,b,c} U(i'_{ab}(c))\beta \qquad \text{(A-4)}$$

where U(x) is the unit step function (i.e., U(x) = 1 for X>0 and U(x) = 0 otherwise).

Comparing expressions (A-2) and (A-4) above, in particular, comparing the last term in each expression we observe a basic tradeoff i.e., the use of intermediaries increases the effort which is linear in the amount of information transferred and correspondingly decreases the effort required to maintain interfaces independent of the amount of information transferred. Combining equations (A-2) and (A-4) leads to a more general (and realistic model of the overall effort (E):

$$E = \sum_{a,b} \alpha i_{ab} + \sum_{a,b,c} \alpha\, i'_{ab}(c) + \tfrac{1}{2}(n)(n-1)\beta - \sum_{a,b,c} U(i'_{ab}(c))\beta \qquad \text{(A-5)}$$

where summation of the first term is over a<b, since i_{ab} has been defined as the total requirement between a and b. We let $i_{ab} = 0$ for a>b because we do not want to count requirements twice.

Rearranging terms in expression (A-5) we get:

$$E = \sum_{\substack{a,b \\ a<b}} \left[\alpha\, i_{ab} + \beta\right] + \sum_{\substack{a,b,c \\ a<b}} \left[\alpha\, i'_{ab}(c) - \beta\, U(i'_{ab}(c))\right] \qquad \text{(A-6)}$$

The first sum in (A-6) is the total effort associated with direct communication between persons. The second sum is the additional effort (possibly negative) of using intermediaries. Thus, a basic optimization problem exists with respect to minimizing the second term.

For example, in the above, suppose $i_{ab} = I$ for all $a < b$. Then, in case 1 (Figure 7 (a))

$$E = 3(\alpha I + \beta) \qquad (A-7)$$

and, in case 2 (Figure 7 (b))

$$E = 3(\alpha I + \beta) + (\alpha I - \beta) \qquad (A-8)$$

Analyzing the last term of (A-8), we see that direct communication is better if $\beta < \alpha I$, and the use of an intermediary is better otherwise.[1]

---

1 A. Kershenbaum, op. cit.,

END

DATE
FILMED

7–83

DTIC