Author's note 42 years after publication: Perhaps this paper's most remarkable feature is that it made it to publication with its thesis statement in the third-last paragraph. To save you the trouble of wading through 45 paragraphs to find the thesis, I'll give an informal version of it to you now: Any organization that designs a system (defined more broadly here than just information systems) will inevitably produce a design whose structure is a copy of the organization's communication structure. This turns out to be a principle with much broader utility than in software engineering, where references to it usually occur. I invite you to read the paper, then look around to find applications. My current favorite is the complex of social issues encompassing poverty in America: access to labor markets, housing, education, and health care. After reading the paper, think about how the structures of our various governments affect their approaches to this system.

A pdf file of the paper in its original form is here.

# How Do Committees Invent?

**Melvin E. Conway**

That kind of intellectual activity which creates a whole from its diverse parts may be called the design of a system. Whether the particular activity is the creation of specifications for a major weapon system, the formation of a recommendation to meet a social challenge, or the programming of a computer, the general activity is largely the same.

Typically, the objective of a design organization is the creation and assembly of a document containing a coherently structured body of information. We may name this information the system design. It is typically produced for a sponsor who usually desires to carry out some activity guided by the system design. For example, a public official may wish to propose legislation to avert a recurrence of a recent disaster, so he appoints a team to explain the catastrophe. Or a manufacturer needs a new product and designates a product planning activity to specify what should be introduced.

The design organization may or may not be involved in the construction of the system it designs. Frequently, in public affairs, there are policies which discourage a group's acting upon its own recommendations, whereas, in private industry, quite the opposite situation often prevails.

It seems reasonable to suppose that the knowledge that one will have to carry out one's own recommendations or that this task will fall to others, probably affects some design choices which the individual designer is called upon to make. Most design activity requires continually making choices, Many of these choices may be more than design decisions; they may also be personal decisions the designer makes about his own future. As we shall see later, the incentives which exist in a conventional management environment can motivate choices which subvert the intent of the sponsor.[1]

### Stages of Design

The initial stages of a design effort are concerned more with structuring of the design activity than with the system itself.[2] The full-blown design activity cannot proceed until certain preliminary milestones are passed. These include:

1. Understanding of the boundaries, both on the design activity and on the system to be designed, placed by the sponsor and by the world's realities,
2. Achievement of a preliminary notion of the system's organization so that design task groups can be meaningfully assigned.

We shall see in detail later that the very act of organizing a design team means that certain design decisions have already been made, explicitly or otherwise. Given any design team organization, there is a class of design alternatives which cannot be effectively pursued by such an organization because the necessary communication paths do not exist. Therefore, there is no such thing as a design group which is both organized and unbiased.

Once the organization of the design team is chosen, it is possible to delegate activities to the subgroups of the organization. Every time a delegation is made and somebody's scope of inquiry is narrowed, the class of design alternatives which can be effectively pursued is also narrowed.

Once scopes of activity are defined, a coordination problem is created. Coordination among task groups, although it appears to lower the productivity of the individual in the small group, provides the only possibility that the separate task groups will be able to consolidate their efforts into a unified system design.

Thus the life cycle of a system design effort proceeds through the following general stages.

1. Drawing of boundaries according to the ground rules.
2. Choice of a preliminary system concept.
3. Organization of the design activity and delegation of tasks according to that concept.
4. Coordination among delegated tasks.
5. Consolidation of subdesigns into a single design.

It is possible that a given design activity will not proceed straight through this list. It might conceivably reorganize upon discovery of a new, and obviously superior, design concept; but such an appearance of uncertainty is unflattering, and the very act of voluntarily abandoning a creation is painful and expensive. Of course, from the vantage point of the historian, the process is continually repeating. This point of view has produced the observation that there's never enough time to do something right, but there's always enough time to do it over.

The Designed System

Any system of consequence is structured from smaller *subsystems* which are interconnected. A description of a system, if it is to describe what goes on inside that system, must describe the system's connections to the outside world, and it must delineate each of the subsystems and how they are interconnected. Dropping down one level, we can say the same for each of the subsystems, viewing it as a system. This reduction in scope can continue until we are down to a system which is simple enough to be understood without further subdivision.

*Examples*. A transcontinental public transportation system consists of buses, trains, airplanes, various types of right-of-way, parking lots, taxicabs, terminals, and so on. This is a very heterogeneous system; that is, the subsystems are quite diverse. Dropping down one level, an airplane, for example. may possess subsystems for structure, propulsion, power distribution, communication, and payload packaging. The propulsion subsystem has fuel, ignition, and starting subsystems, to name a few.

It may be less obvious that a theory is a system in the same sense. It relates to the outside world of observed events where it must explain, or at least not contradict, them. It consists of subtheories which must relate to each other in the same way. For example, the investigation of an airplane crash attempts to produce a theory explaining a complex event. It can consist of subtheories describing the path of the aircraft, its radio communications, the manner of its damage, and its relationship to nearby objects at the time of the event. Each of these, in turn, is a story in itself which can be further broken down into finer detail down to the level of individual units of evidence.

*Linear graphs*. Fig. 1 illustrates this view of a system as a linear graph -- a Tinker-Toy structure with branches (the lines) and nodes (the circles). Each node is a subsystem which communicates with other subsystems along the branches. In turn, each subsystem may contain a structure which may be similarly portrayed. The term*interface*, which is becoming popular among systems people, refers to the inter-subsystem communication path or branch represented by a line in Fig. 1. Alternatively, the interface is the plug or flange by which the path coming out of one node couples to the path coming out of another node.

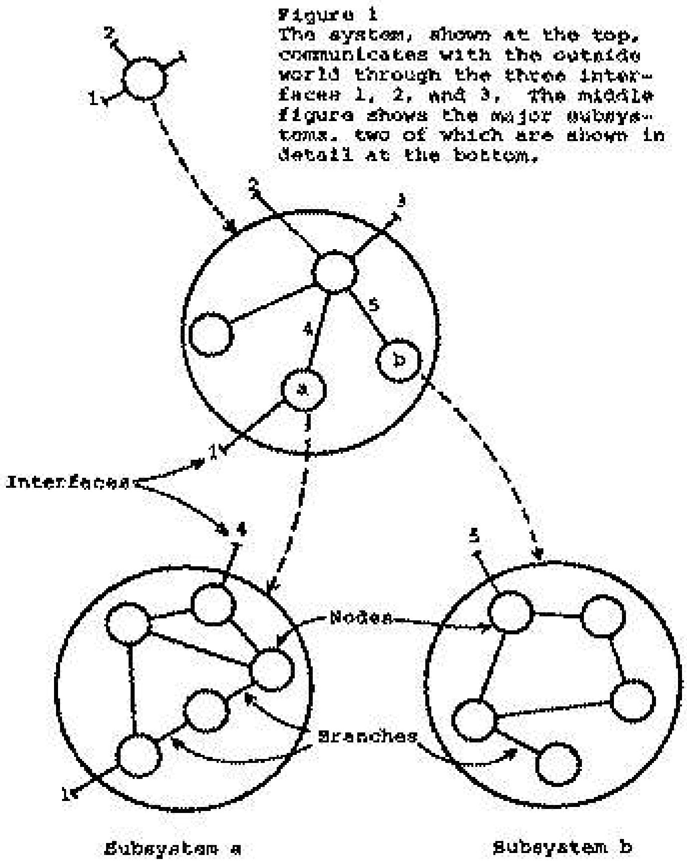


Figure 1
The system, shown at the top, communicates with the outside world through the three interfaces 1, 2, and 3. The middle figure shows the major subsystems. two of which are shown in detail at the bottom.

Figure 1

Relating the Two

The linear-graph notation is useful because it provides an abstraction *which has the same form* for the two entities we are considering: the design organization and the system it designs. This can be illustrated in Fig. 1 by replacing the following words.

1. Replace "system" by "committee."
2. Replace "subsystem" by "subcommittee."
3. Replace "interface" by "coordinator."

Just as with systems, we find that design groups can be viewed at several levels of complication. The Federal Government, for example. is an excellent example of a design organization with enough complexity to satisfy any system engineer. This is a particularly interesting example for showing the similarity of the two concepts being studied here because the Federal Government is both a design organization (designing laws, treaties, and policies) and a designed system (the Constitution being the principal preliminary design document).

*A basic relationship.* We are now in a position to address the fundamental question of this article. Is there any predictable relationship between the graph structure of a design organization and the graph structure of the system it designs? The answer is: Yes, the relationship is so simple that in some cases it is an identity. Consider the following "proof."

Let us choose arbitrarily some system and the organization which designed it, and let us then choose equally arbitrarily some level of complication of the designed system for which we can draw a graph. (Our motivation for this arbitrariness is that if we succeed in demonstrating anything interesting, it will hold true for any design organization and level of complication.) Fig. 2 shows, for illustration purposes only, a structure to which the following statements may be related.

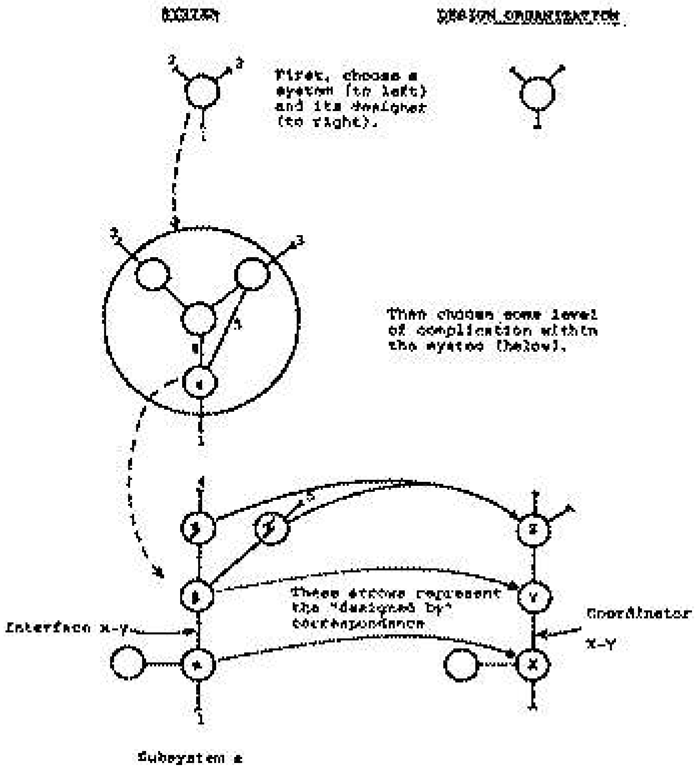

SYSTEM
DESIGN ORGANIZATION
First, choose a system (to left) and its designer (to right).
Then choose some level of complication within the system (below).
These arrows represent the "designed by" correspondence
Interface x-y
Coordinator X-Y
Subsystem x
1
2
3
4
5
x
y
z
X
Y
Z

Figure 2 This is an illustration of the strong relationship between the structure (graph) of a system (left) and the structure of the organization which designed it (right).

Figure 2

For *any* node *x* in the system we can identify a design group of the design organization which designed *x*; call this X. Therefore, by generalization of this process, for *every* node of the system we have a rule for finding a corresponding node of the design organization. Notice that this rule is not necessarily one-to-one; that is, the two subsystems might have been designed by a single design group.

Interestingly, we can make a similar statement about branches. Take any two nodes*x* and *y* of the system. Either they are joined by a branch or they are not. (That is, either they communicate with each other in some way meaningful to the operation of the system or they do not.) If there is a branch, then the two (not necessarily distinct) design groups X and Y which designed the two nodes must have negotiated and agreed upon an interface specification to permit communication between the two corresponding nodes of the design organization. If, on the other hand, there is no branch between *x* and *y*, then the subsystems do not communicate with each other, there was nothing for the two corresponding design groups to negotiate, and therefore there is no branch between X and Y.[3]

What have we just shown? Roughly speaking, we have demonstrated that there is a very close relationship between the structure of a system and the structure of the organization which designed it. In the not unusual case where each subsystem had its own separate design group, we find that the structures (i.e., the linear graphs) of the design group and the system are identical. In the case where some group designed more than one subsystem we find that the structure of the design organization is a collapsed version of the structure of the system, with the subsystems having the same design group collapsing into one node representing that group.

This kind of a structure-preserving relationship between two sets of things is called a *homomorphism*. Speaking as a mathematician might, we would say that there is a homomorphism from the linear graph of a system to the linear graph of its design organization.

## Systems Image Their Design Groups

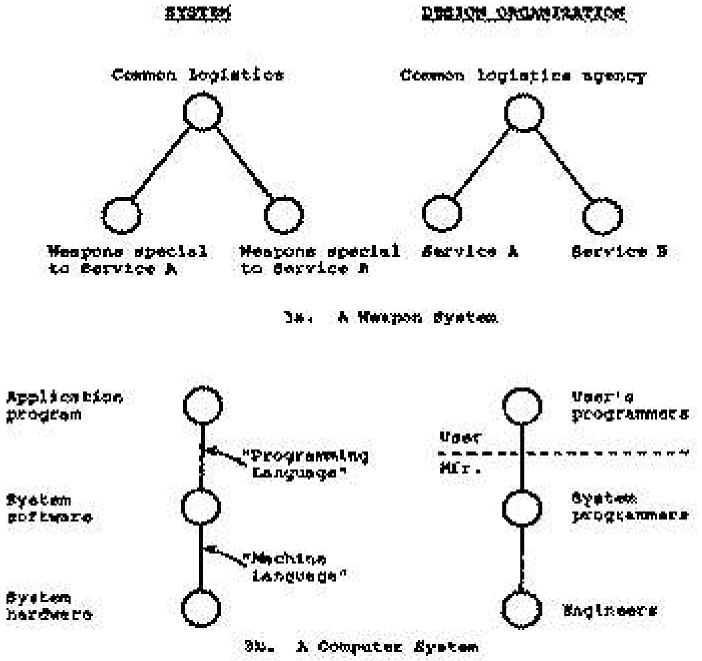
It is an article of faith among experienced, system designers that given any system design, someone someday will find a better one to do the same job. In other words, it is misleading and incorrect to speak of the design for a specific job, unless this is understood in the context of space, time, knowledge, and technology. The humility which this belief should impose on system designers is the only appropriate posture for those who read history or consult their memories

The design progress of computer translators of programming languages such as F0RTRAN and COBOL is a case in point. In the middle fifties, when the prototypes of these languages appeared, their compilers were even more cumbersome objects than the giant (for then) computers which were required for their execution. Today, these translators are only historical curiosities, bearing no resemblance in design to today's compilers. (We should take particular note of the fact that the quantum jumps in compiler design progress were associated with the appearance of new groups of people on territory previously trampled chiefly by computer manufacturers -- first it was the tight little university research team, followed by the independent software house.)

If, then, it is reasonable to assume that for any system requirement there is a *family* of system designs which will meet that requirement, we must also inquire whether the choice of design organization influences the process of selection of a system design from that family. If we believe our homomorphism, then we must agree that it does. To the extent that an organization is not completely flexible in its communication structure, that organization will stamp out an image of itself in every design it produces. The larger an organization is, the less flexibility it has and the more pronounced is the phenomenon.

*Examples*. A contract research organization had eight people who were to produce a COBOL and an ALGOL compiler. After some initial estimates of difficulty and time, five people were assigned to the COBOL job and three to the ALGOL job. The resulting COBOL compiler ran in five phases, the ALG0L compiler ran in three.

Two military services were directed by their Commander-in-Chief to develop a common weapon system to meet their respective needs. After great effort they produced a copy of their organization chart. (See Fig. 3a.)



3a. A Weapon System

3b. A Computer System

Figure 3 Two examples of identity of structure between a system and its design organization.

Figure 3

Consider the operating computer system in use solving a problem. At a high level of examination, it consists of three parts: the hardware, the system software, and the application program. (See Fig. 3b.) Corresponding to these subsystems are their respective designers: the computer manufacturer's engineers, his system programmers, and the user's application programmers. (Those rare instances where the system hardware and software tend to cooperate rather than merely tolerate each other are associated with manufacturers whose programmers and engineers bear a similar relationship.)

System Management

The structures of large systems tend to disintegrate during development, qualitatively more so than with small systems. This observation is strikingly evident when applied to the large military information systems of the last dozen years; These are some of the most complex objects devised by the mind of man. An activity called "system management" has sprung up partially in response to this tendency of systems to disintegrate. Let us examine the utility to system management of the concepts we have developed here.

Why do large systems disintegrate? The process seems to occur in three steps, the first two of which are controllable and the third of which is a direct result of our homomorphism.

- First, the realization by the initial designers that the system will be large, together with certain pressures in their organization, make irresistible the temptation to assign too many people to a design effort.
- Second, application of the conventional wisdom of management to a large design organization causes its communication structure to disintegrate.
- Third, the homomorphism insures that the structure of the system will reflect the disintegration which has occurred in the design

organization.

Let us first examine the tendency to overpopulate a design effort. It is a natural temptation of the initial designer -- the one whose preliminary design concepts influence the organization of the design effort -- to delegate tasks when the apparent complexity of the system approaches his limits of comprehension. This is the turning point in the course of the design. Either he struggles to reduce the system to comprehensibility and wins, or else he loses control of it. The outcome is almost predictable if there is schedule pressure and a budget to be managed.

A manager knows that he will be vulnerable to the charge of mismanagement if he misses his schedule without having applied all his resources. This knowledge creates a strong pressure on the initial designer who might prefer to wrestle with the design rather than fragment it by delegation, but he is made to feel that the cost of risk is too high to take the chance. Therefore, he is forced to delegate in order to bring more resources to bear.

The following case illustrates another but related way in which the environment of the manager can be in conflict with the integrity of the system being designed.

A manager must subcontract a crucial and difficult design task. He has a choice of two contractors, a small new organization which proposes an intuitively appealing approach for much less money than is budgeted, and an established but conventional outfit which is asking a more "realistic" fee. He knows that if the bright young organization fails to produce adequate results, he will be accused of mismanagement, whereas if the established outfit fails, it will be evidence that the problem is indeed a difficult one.

What is the difficulty here? A large part of it relates to the kind of reasoning about measurement of resources which arises from conventional accounting theory. According to this theory, the unit of resource is the dollar, and all resources must be measured using units of measurement which are convertible to the dollar. If the resource is human effort, the unit of measurement is the number of hours worked by each man times his hourly cost, summed up for the whole working force.

One fallacy behind this calculation is the property of linearity which says that two men working for a year or one hundred men working for a week (at the same hourly cost per man) are resources of equal value. Assuming that two men and one hundred men cannot work in the same organizational structure (this is intuitively evident and will be discussed below) our homomorphism says that they will not design similar systems; therefore the value of their efforts may not even be comparable. From experience we know that the two men, if they are well chosen and survive the experience, will give us a better system. Assumptions which may be adequate for peeling potatoes and erecting brick walls fail for designing systems.

Parkinson's law[4] plays an important role in the overassignment of design effort. As long as the manager's prestige and power are tied to the size of his budget, he will be motivated to expand his organization. This is an inappropriate motive in the management of a system design activity. Once the organization exists, of course, it will be used. Probably the greatest single common factor behind many poorly designed systems now in existence has been the availability of a design organization in need of work.

The second step in the disintegration of a system design -- the fragmentation of the design organization communication structure -- begins as soon as delegation has started. Elementary probability theory tells us that the number of possible communication paths in an organization is approximately half the square of the number of people in the organization. Even in a moderately small organization it becomes necessary to restrict communication in order that people can get some "work" done. Research which leads to techniques permitting more efficient communication among designers will play an extremely important role in the technology of system management.

Common management practice places certain numerical constraints on the complexity of the linear graph which represents the administrative structure of a military-style organization. Specifically, each individual must have at most one superior and at most approximately seven subordinates. To the extent that organizational protocol restricts communication along lines of command, the communication structure of an organization will resemble its administrative structure. This is one reason why military-style organizations design systems which look like their organization charts.

## Conclusion

The basic thesis of this article is that organizations which design systems (in the broad sense used here) are constrained to produce designs which are copies of the communication structures of these organizations. We have seen that this fact has important implications for the management of system design. Primarily, we have found a criterion for the structuring of design organizations: a design effort should be organized according to the need for communication.

This criterion creates problems because the need to communicate at any time depends on the system concept in effect at that time. Because the design which occurs first is almost never the best possible, the prevailing system concept may need to change. Therefore, flexibility of organization is important to effective design.

Ways must be found to reward design managers for keeping their organizations lean and flexible. There is need for a philosophy of system design management which is not based on the assumption that adding manpower simply adds to productivity. The development of such a philosophy promises to unearth basic questions about value of resources and techniques of communication which will need to be answered before our system-building technology can proceed with confidence.

**Notes**

[1] A related, but much more comprehensive discussion of the behavior of system-designing organizations is found in John Kenneth Galbraith's *The New Industrial State* (Boston, Houghton Mifflin, 1967). See especially Chapter VI, "The Technostructure."

[2] For a discussion of the problems which may arise when the design activity takes the form of a project in a functional environment, see C. J. Middleton, "How to Set Up a Project Organization," *Harvard Business Review*, March-April, 1967, p. 73.

[3] This claim may be viewed several ways. It may be trivial, hinging on the definition of meaningful negotiation. Or, it may be the result of the observation that one design

group almost never will compromise its own design to meet the needs of another group unless [doing so is] absolutely imperative.

[4] C. Northcote Parkinson, *Parkinson's Law and Other Studies in Administration* (Boston, Houghton Mifflin, 1957).