PROGRAMS AND SYSTEMS

An Evaluation Perspective

GARY D. BORICH
RON P. JEMELKA

Department of Educational Psychology and Instructional Systems Laboratory College of Education The University of Texas *Austin, Texas*

Academic Press 1982
A Subsidiary of Harcourt Brace Jovanovich, Publishers New **York London**
Paris San Diego San Francisco Sao Paulo
Sydney Tokyo Toronto

7

A Modeling Approach to Program Evaluation

If we had a way to conceptualize a program such that we could simultaneously see the program's component parts and view the program as a component part of some larger system, we could perhaps gain a conceptual handle on the complexity inherent in that program. The use of what Suchman (1972) called a "rhetorical program model" can serve this function for evaluation. An on-paper representation of a program and its interrelations would be much less difficult to observe, manipulate, and study, than would the *in vivo* program itself. In this chapter we will introduce some syntax for creating graphic models that will help in understanding and communicating the meaning of a program. We will then use these models to pose questions about the program, particularly questions of the part-whole and value variety.

Describing Programs with Models

The heuristic value of a model is well explicated by Pfieffer (1968):

> Basic to the entire systems notion is the concept of a model, a simplified but controllable version of a real-world situation which serves a function roughly comparable to that of a laboratory experiment in the physical and biological sciences [p. 6].
>
> The problems confronting us today involve incredibly complex mazes or "nests" of interconnections and linkages rather than straight-forward associations and cause-and-effect sequences, and the results of important changes may be extremely difficult to predict. In such cases a good model can help appreciably by supplementing intuition and judgment. It is one thing to rely solely on specula-

> tions about how people will behave under certain real-life circumstances, and something else again when one can objectively investigate possible behavior patterns by simulating those circumstances in a model [p. 6].
>
> A model has an interesting and significant double aspect.... It is an abstraction—a highly simplified version of a fragment of the real world which is too complex for us to deal with directly. At the same time, however, it is one highly effective way of coping with reality [p. 26].
>
> A model is meant to clarify, and to yield information. That depends on how well it is designed. It will certainly be modified or superceded sooner or later in the light of accumulating knowledge, which is the general fate of models. Indeed, from one standpoint the role of a good model is to speed its own absolescence. It cannot provide final answers and is not intended to. It has served its purpose if it provides fresh insights into the working of things [p. 27].

Like all models, including those underlying the use of experimental designs, the program's representation will achieve only a certain degree of fidelity to the real-life situation. Furthermore, there are tradeoffs in the use of models to portray programs. Very simple, easy-to-construct, easy-to-use models, may not be of high enough fidelity to be useful in thinking about programs. Alternatively, models with a high degree of fidelity tend to be very complex and difficult to construct.

Our experiences with various systems analysis and traditional statistical models have suggested to us the use of graphic models for depicting programs. Remembering the old adage that "a picture is worth a thousand words" we have adapted from Ross (1977) a simple box *and arrow* syntax that, we believe, is superior to prose and statistical symbols for communicating some program characteristics, especially (a) program structure and process, (b) the linkages between program activities and outcomes, (c) relationships between values and program outcomes, and (d) relationships between the program and its environment(s). Simply put, some aspects of the way programs are organized make them easy to describe with a graphic methodology.

We will not concern ourselves here with teaching the entirety of this graphic language for describing programs. Our purpose is not to "push" a particular technique. Instead, two simple and basic syntax rules will be introduced to facilitate our present discussion by allowing the substitution of pictures for more cumbersome words. These "pictures" will introduce the notion of communicating conceptually with the aid of graphic heuristics as opposed to relying exclusively on prose.

Here are two simple rules we will use in describing programs:

> ***Rule 1:*** A box will be used to denote any program transaction, that is, any activity, process, or event. The box means nothing in and of itself but is brought to life or "activated" by inputs (which always enter from the

left), constraints (which always press down from the top), and outputs (which always exit the box from the right), as noted in Figure 7.1.

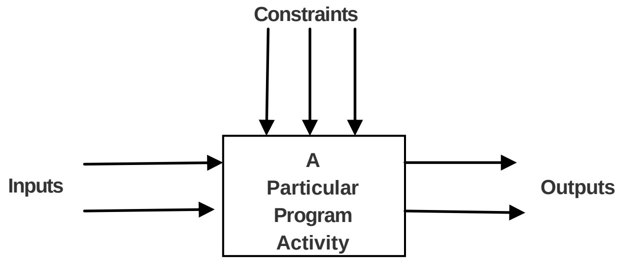

Figure 7.1. Illustration of a program activity.

Inputs—things the activity uses; clients or students, tests, staff, time, classrooms, etc.
Constraints—things that constrain the activity; funding, program priorities, feedback on results, anything that moderates or influences output.
Outputs—things "produced" or resulting from the activity; clients with improved outlook, higher rates of employment, students with a certain skill.

Input, constraint, and outcome designations reveal how transactions within a program are tied together. This is illustrated in Figure 7.2.

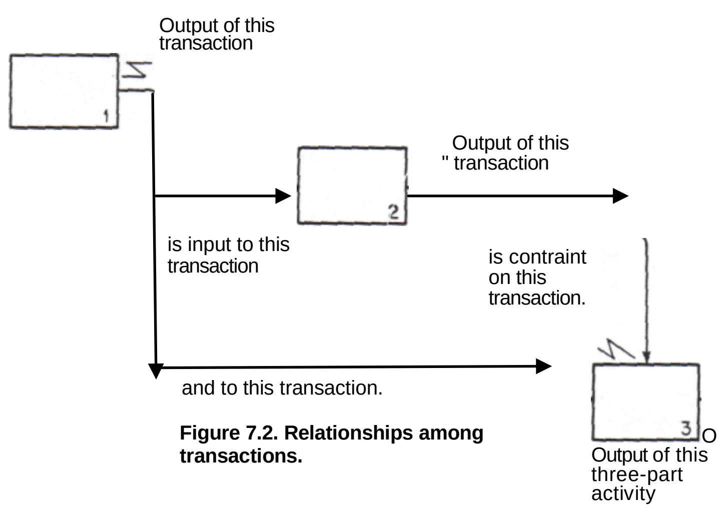

Figure 7.2. Relationships among transactions.

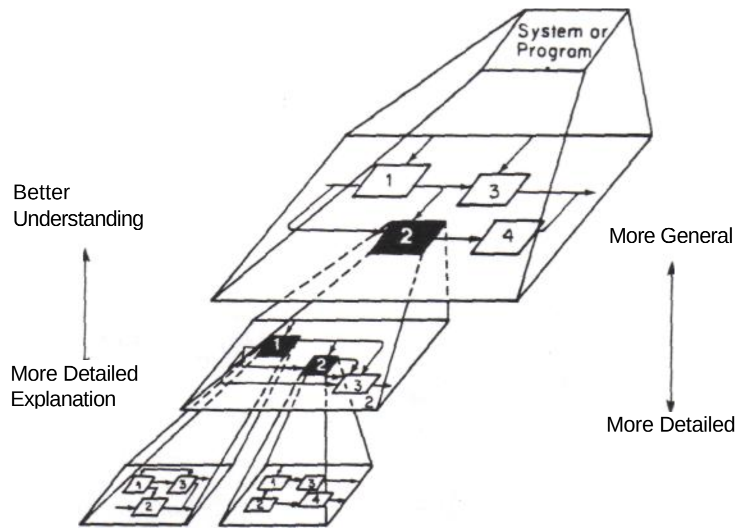

Figure 7.3. **Illustration of a program model.**

Rule 2: Any box, or transaction, can be broken down into its component parts. Alternatively any transaction can be seen as a component part of some larger system. This idea is communicated graphically in Figure 7.3.

The links among program components, or among programs within some larger structure, can be viewed from this graphic perspective in an efficient manner. Considerable prose would be required to communicate all that is communicated graphically, once these two syntax rules are accepted. It is easy with these conventions to imagine (or draw) brief, highly specific transactions within a program component within some program within some overarching national, state, or local policy. In this manner values, the ultimate constraints on what policies will be formulated, are linked to quite specific program details. Of course the model might indicate that aspects of program functioning are not linked to higher-level objectives. Either way, the nature of these linkages is a critical issue for evaluation, but one which has not been addressed because, in part, we have not (as a discipline) had a conceptual framework or methodology for doing so. We are suggesting that these graphic conventions are aids or heuristics for thinking about programs. They give us a way of considering more that is going on at one time than do prose descriptions or statistical models of the same processes.

Models, of course, have some weaknesses too. They make assumptions about the world that may in reality not be true. Models are efforts to simplify and render phenomena understandable, and sometimes they lose fidelity in the process of simplification. Each application of some model will have a corresponding degree of fidelity between the model and the world circumstances it purports to represent. Forcing programs into this graphic systems framework is as liable as any model to these shortcomings.

However, the assumptions behind our program modeling technique differ conceptually from those assumptions underlying the use of our statistical models. For example, the type of program model we propose assumes that (a) *hierarchy* is a highly salient characteristic of programs, and (fa) conceptualizing programs in terms of inputs, outcomes, and constraints as the links between program activities is consistent with the way we implicitly describe phenomena. A review of prose descriptions of programs invariably reveals that inputs, outcomes, and constraints are almost universal concepts for describing a program. Such a conceptualization will have higher fidelity in some settings than in others. But, when thinking about programs, we know of no easily readable model that "lets in" more complexity and makes fewer restrictive assumptions than does this type of program model.

Characteristics of the Graphic Modeling Technique

In the pages that follow we discuss the characteristics of this graphic modeling technique that make the models it produces relevant, credible representations of the programs we evaluate. For readers who want to become more proficient in the use of this graphic language the appendix to this volume provides more information and additional illustrations.

HIERARCHY

Man has a penchant for hierarchy. There can be little doubt of that. Quite simply hierarchy is the way we organize things if we wish to consider more than six or seven things at a time. If we get overloaded we create a few higher order categories, each holding a few cases that are similar on some dimension. A hierarchy is a set of objects, activities, or behaviors, arranged according to some pre-established dimension. The characteristics of hierarchical structure are (a) vertical arrangement of subsystems that comprise the overall system, (fa) definition of higher-level systems determine the meaning of lower-level systems, and (c) dependence of higher-level subsystems upon actual performance of the lower levels.

A hierarchy, then, is a framework that permits complex systems to be built from simpler ones. On the other hand, hierarchies allow complex systems to be broken down (i.e., decomposed into their component parts and subsystems).

As Simon notes, a complex system is

> one made up of a larger number of parts that interact in a nonsimple way. In such systems, the whole is more than the parts not in an ultimate metaphysical sense, but in the important pragmatic sense that given the properties of the parts and the laws of their interaction, it is not a trivial matter to infer the properties of the whole [cited in van Gigch, 1978, p. 375].

Van Gigch (1978) notes that hierarchy implies that:

1. A system is always made up of other systems.
2. Given a certain system, another system can always be found that comprises it, except for the universal system which comprises all others.
3. Given two systems, the one system comprising the other can be called "the high level system" in relation to the system it comprises, which is called the "low level system."
4. A hierarchy of systems exists whereby lower level systems make up higher level systems.
5. The low level systems are in turn made up of other systems and can therefore be considered the higher level system for the lower level systems to be found in it [p. 376].

Simon (1969) explains the nature of hierarchy with the following parable.

> There once were two watchmakers, named Hora and Tempus, who manufactured very fine watches. Both of them were highly regarded, and the phones in their workshops rang frequently—new customers were constantly calling them. However, Hora prospered, while Tempus became poorer and poorer and finally lost his shop. What was the reason?
>
> The watches the men made consisted of about 1,000 parts each. Tempus had so constructed his that if he had one partly assembled and had to put it down—to answer the phone, say—it immediately fell to pieces and had to be reassembled from the elements. The better the customers liked his watches, the more they phoned him and the more difficult it became for him to find enough uninterrupted time to finish a watch.
>
> The watches that Hora made were no less complex than those of Tempus. But he had designed them so that he could put together subassemblies of about ten elements each. Ten of these subassemblies, again, could be put together into a larger subassembly; and a system often of the latter subassemblies constituted the whole watch. Hence, when Hora had to put down a watch in order to answer the phone, he lost only a small part of his work, and he assembled his watches in only a fraction of the man-hours it took Tempus [pp. 90-91; reprinted by permission of the MIT Press].

Simply put, the concept of hierarchy may be used as a natural ordering or reasoning process to bring order and systematization to the activities, behaviors, or events constituting a program. It appears that hierarchy is our basic means of dealing with complexity. Our communication of complexity invariably involves hierarchy, as evidenced by the structure of written communication expressing more than a handful of ideas. Books are broken down by chapters. Within chapters are major subheadings. Within each of these are paragraphs containing some theme or idea, expressed as sentences consisting of phrases that are made up of specific words. Through hierarchical arrangement a large number of disparate ideas and concepts may be woven into a conceptually understandable and meaningful, albeit complex, piece of communication.

Looking at Figure 7.3, it is apparent that our simple syntax embodies these hierarchy principles that seem prevalent in our society. They are probably prevalent for one reason: They reduce complexity into manageable chunks. No more, no less. If the complexity generated by questions posed to evaluation personnel is overloading the tools and expertise available, it seems timely to suggest some complexity-handling mechanisms, such as the graphic syntax we have introduced.

The hierarchy concept is not necessarily the "true" way the world is, but it might help us understand programs. Two points that further substantiate the role of hierarchy in the world, particularly the world of programs, concern the use of hierarchy as a natural extension of reasoning processes and the inherently hierarchical nature of bureaucracy, the roots from which most programs spring.

Eleanor Gibson (1969) provides a compelling theory of perceptual development that stresses that our increasing ability to extract invariant features in the environment, label the invariance meaningfully, and ultimately make successively finer discriminations is the zenith of our perceptual development. Certainly the ability to use hierarchical structures in one's thinking fosters the act of bringing order or clarity out of ambiguity and chaos, an intellectually challenging task. To our way of thinking, recognizing and using these hierarchical structures makes one a better thinker and problem solver in some instances. The ability to use hierarchical structures to order concepts, events, activities, or data is essential not only to Gibson's theory of perceptual development but to many current theories of human intelligence.

That the concept of hierarchy is somehow engrained in the way we think is perhaps most evident in our organizing behavior. Hierarchy is usually the most salient feature of our organizational charts and appears to be second only to sequence in commonly used flow charts. We will argue shortly that even flow charts depict a hierarchical arrangement of sorts through constraint and that it is constraint rather than sequence that is usually important in describing an ongoing event or process.

Sjoberg (1975) refers to Weber's view of hierarchical bureaucracy and points out its prominence in our society.

> If we look back over the past several decades, we find that the increased bureaucratization of modern society has been associated with a rising interest in evaluation research. Even today, after substantial criticism on both theoretical and empirical grounds, Weber's conception of bureaucracy, with its emphasis upon rationality, remains a dominant image of how industrial-urban life is and should be organized.
>
> The bureaucratic model rests upon the principle of dominance or hierarchy. Although various positions in the structure are interrelated in complex ways, the overall pattern is one of superordination and subordination. It is within this context of dominance that the notion of universalism is applied in, say, the selection and promotion of personnel. It is within this context that the principle of rationality is applied ... |p. 34].

Sjoberg's and Gibson's views are introduced as examples of the role attributed to hierarchical arrangements in the way we think and organize in the social sciences. In the "harder" sciences, biological and genetic classification tables and the periodic table of elements from chemistry reflect a parallel prominence of this concept. Even within one's own work environment, such characteristics as *lines of authority, chains of command,* and *superior-subordinate* relationships are usually easily discerned. Yet although this organizing principle is pervasive in our society, hierarchy in-and-of-itself has received little attention.

The modeling technique we will describe is designed to reflect the central role hierarchy plays in the way we organize things, particularly programs. It could be argued that this modeling procedure *assumes a* hierarchical structure that may in fact exist only in the mind of the modeler and not in the phenomena being modeled. As with all models, violations of basic assumptions should render a model useless. However, the pervasiveness of hierarchy in our world, particularly human organizing behavior, suggests that this assumption has a high probability of being isomorphic with the real-world situation being modeled and is a safe assumption for most program modeling purposes. The authors' modeling experiences indicate that if hierarchy is not there, you are not going to find it.

TRANSACTIONS

Some readers might question just what the boxes represent in this modeling procedure. These are program activities or transactions. They are the basic building blocks of programs. The word *transaction* denotes the transfer or communication of behavior or skills from one source to another. Thus, a

I

transaction is any program activity in which the program recipient interacts with program components for the purpose of bringing about some change. Within education, a teacher's lecture, a workbook lesson completed alone by the student, or an arithmetic game played by the class may be viewed as program transactions. Each brings the program recipient (student) into contact with various program elements (lecture, workbook, game) and each has some specifiable outcome.

Logically grouping specific transactions creates more general transactions. For example, a series of teacher lectures or presentations on addition, coupled with the students' working through a series of addition exercises in a workbook, and the involvement of the class in arithmetic games and drills constitutes a general level transaction called "teach addition." This general level transaction is itself but one component part or transaction of some yet more general program activity or transaction, perhaps "teach mathematics" or "teach the 3 R's."

We chose an overly simplistic example to illustrate that transactions are program activities conceptualized at varying levels of generality-specificity. In short, transactions are "two-faced." Any transaction is simultaneously a *whole,* or collection of its subcomponents, and a component part of the next more general level of program functioning. Virtually all the programs of which we are aware can be easily conceptualized in terms of their transactions, and further, this conceptualizing can be done at any level of generality-specificity that exists in the organizational hierarchy of a program. Thus, any program component is a transaction that can be viewed interoceptively and exteroceptively.

CONSTRAINTS

We have defined constraints as those arrows that press down on a transaction. We intend to show in the discussion that follows that the concept of constraint is an even more basic and pervasive concept than is hierarchy in the functioning of programs. The shape and form of any given hierarchy is dictated by the constraints and controls that "press" on it. In other words, hierarchy is but a framework indicating how constraints are operating.

Previously we defined constraints as things that affect an activity or transaction, such as funding, program priorities, feedback on results, and other variables that influence or moderate the outcomes of any given transaction. The careful reader probably questioned the definition of a category of events that included "anything that affects output." This is indeed a broad category that demands some explanation. We will first present some of the more mechanistic characteristics of the way constraints operate in the world, and then deal with relationships between constraints, values, and evaluation.

The mechanics of constraint. A constraint does not hold back or limit in the usual sense of the term. Instead, it moderates, influences, or dictates what will happen in one of two ways. *Exteroceptive* constraints indicate "presses" on a program from somewhere higher in the system (hierarchy) than that level at which the program constraint is operating. *Interoceptive* constraints originate at the same level in the hierarchy. For example, funding, which may originate out of policy formulated in Washington, D.C., may have considerable impact on program activities implemented in local settings throughout the country. Consider the example of a federal funding policy in education that provides ample funding for classroom materials, but leaves it to the local districts to supply the personnel to implement the program. Different districts can be expected to be differentially favorable to committing the time of teachers and teacher aides to implement the program. Thus, two exteroceptive constraints that significantly determine what actually happens in the classroom (i.e., constrain the program) would be federal funding restrictions and local staffing patterns.

A real-life example of an exteroceptive constraint comes from the field of bilingual education. Federal law requires school districts to provide bilingual education services to those children with a demonstrated need. Thus in districts with large numbers of this categorical group, the resources available for bilingual education are spread evenly across all those students in need. Some students, however, are particularly handicapped in an exclusively English-speaking environment while others may have equivalent skills in English and Spanish (or French), but still meet the conditions of demonstrated need. The law requires that all be served equally and does not permit local discretion in intensifying the program for some students and attenuating it for others.

Operating simultaneously and independently is the fact that federal desegregation laws require that classrooms reflect the ethnic mix of the community, resulting in classrooms made up of students with tremendous diversity in their language needs. Thus these two *exteroceptive* constraints greatly affect what happens in the bilingual education programs, for they dictate that bilingual education be implemented in classrooms where only some students have a demonstrated need, and even those in need vary considerably in terms of the appropriateness of the bilingual "treatment" being offered.

Suppose the hypothetical educational program example we cited earlier was in actuality the third of a three-part accelerated reading program for gifted students and that the three parts were sequential. Then, as Figure 7.4 indicates, the outcome of this program transaction will be constrained by each student's outcomes from parts I (word recognition) and II (vocabulary). In the manner indicated in Figure 7.4, the outcomes of parts I and II of the

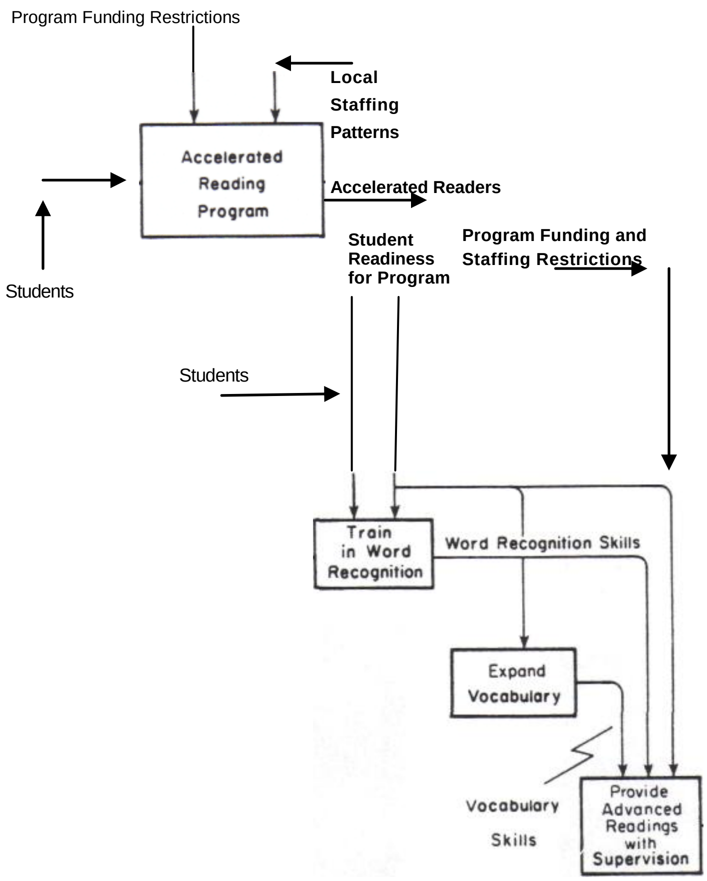

Figure 7.4. A partial decomposition of an accelerated reading program.

program interoceptively constrain or moderate the outcomes of part III, supervised reading practice. These are considered interoceptive constraints because they originate from within the program itself at the same level of generality as the program transaction they are constraining.

Another very important characteristic of constraint embedded in Figure 7.4 concerns the sequence of program components. Figure 7.4 indicates that word recognition and vocabulary training components must precede practice in reading. The figure says nothing about the sequential relationship between word recognition and vocabulary training components themselves. Presumably these could be implemented in either order.

The related but separate issues of sequence and constraint are often confused in our thinking about programs. Consider the general belief that the ability to subtract and multiply must precede the ability to divide. Out of this belief we sequence training such that students can already subtract and multiply before we attempt to teach division. The sequence is thought to be important. It's not, really. What is important are the *constraints* on learning division, two of the most salient of which are the ability to subtract and multiply.

Constraint and sequence so often go hand-in-hand that we are cognizant only of sequence, when actually it is the principle of constraint that is critical. Sequence is so salient to us time-conscious humans that the principle of constraint becomes obfuscated. Our reductionistic cause-and-effect paradigms, which depend on time relations and describe phenomena in terms of if-then propositions, are undoubtedly an outgrowth of our obsession with sequence and our neglect of constraint.

We are not saying that sequence can be ignored or that cause-and-effect models are useless. We are saying that exclusive reliance on and blind acceptance of these constructs can obscure as well as shed light on the way the world works. If we note that X tends to occur before Y we assume a sequential relationship. What about those less frequent instances in which Y precedes X, or they occur simultaneously? Given the sequential set we have towards these phenomena, would we even notice these deviations? If we did notice, how would we classify these events? What if the interactive effect of A, B, C, and D combine to trigger X first and then Y most of the time, but not all the time? Is thinking about these problems with our usual sequence-oriented framework going to help us discern these relationships? Recognizing and describing constraint can be a powerful tool in program evaluation.

A general rule-of-thumb might be that if sequences occur with regularity in programs, we can expect that constraints are at play. If upon careful examination, no constraints are found, order (or sequence) may not be important to the program. The question is how much of the sequencing in the world represents true constraints and how much is artifact, tradition, convenience, and the like. Granted, artifact, tradition, and convenience are constraints in their own right but they are usually conceptually separate from the entities they "constrain." We are interested in those constraints more intimately tied to the program they hold together.

Constraints as values. Thus far, we have stated that interoceptive and exteroceptive constraints determine what a particular program transaction might look like and what outcomes might result from the transaction. The reader is probably wondering just what constraints really are. Enough vague reference to policy, funding, previously attained student skills, and the like; we need to expose constraints for what they really are: *values.*

We stated earlier that constraints represent the "glue" that holds programs together. It is through the mechanism of constraint that any given level of program detail is tied to both the higher-order (more general) transactions occurring "above" it, and to the lower-level (more specific) program transactions "below" it. The concepts of hierarchy and program modeling are ways of identifying and understanding the constraints of a program. If modeling is the pathway to understanding program constraints, then constraints are the avenues toward understanding the values undergirding a program. Exteroceptive constraints, at any level of program detail, are but operationalizations of someone's values.

We mentioned earlier that funding serves as a universal constraint on programs. The particular level of funding available to a program can serve as an operational definition of the amount of value placed on *a* program. The more a particular program is valued, the higher the level of funding and the deployment of other resources is apt to be for that program. Less valued programs are assigned a lower priority and therefore will receive fewer resources than their more highly valued counterparts.

Values usually will not appear in close juxtaposition to the program(s) they spawn, for, as was noted in Chapter 3, values are seldom identified, stated clearly, or even considered, particularly in program evaluation. In the best of all worlds, the way we have organized our society allows and guides us to do and make possible those things that we as a people value. But how do we determine what 200 million people value at any one time on any given issue? And even if those values could be determined, how do we determine the implications of these values?

In our society we accomplish these ends through our political system. Through the legislation of laws and execution of laws (and some jurisprudence about the implications of our actions) we move from a value context to an action context. In this manner values are transformed into constraints upon our federal level programs. These constraints are apparent throughout the system as policies, funding levels, regulations, guidelines, and the like. These higher-level constraints in turn interact with local characteristics to produce the nuances of constraint brought to bear on any program activity implemented at any particular site.

We are not saying that all programs emanate from the federal level. What we are saying is that all programs evolve from value to action contexts through some decision-making process (which may or may not be political, depending on how one defines the term). The guidelines, procedures, structure, goals, and purposes that flow from this decision-making process, in turn, serve as the definition of the program. These constraints or program definitions tie together all program activities, inputs, and outcomes.

Constraints define some or all of the following program characteristics: who will be served by the program; how, under what circumstances, and by whom; what is expected to happen; how performance will be evaluated; and so on. In this manner constraints define and draw the boundaries of programs and transactions within programs. If constraints are interpretations of values, then programs *in actuality are operational definitions of values.* They are expressions or reflections of value. Nothing more, nothing less. If one finds a program which aims to facilitate cause X, it is because someone, somewhere, valued cause X. The program might not further cause X as was the original intent; instead, cause Z might be furthered. This might occur as the result of not thinking the program out thoroughly before implementing it. Or, it might occur as a result of the intrusion of someone else's values on the original intents of the program, as in the case of program sabotage by a vindictive program manager or an uncooperative staff, or restrictions placed on program operations or its staff by some governing board or advisory committee.

It can easily be seen that by the time a program is implemented, it may have been altered to the extent that even program *intents* are contradictory to the values originally providing impetus for the program. And we have not even addressed the issue of what the program is actually doing and whether these ends are consistent with program intents or anyone's values. This is, of course, the basis of our objection to solely relying on the decision-oriented approach in which the role and function of program objectives is so prominent.

Similar views have recently been espoused by Morell (1979). Basing his statement on a paper by Brandl (1978), he proposed that

> politicians are not primarily interested in the outcomes of social programs. Rather, politicians are interested in the number and diversity of interests and constituencies which advocate a particular position. If enough people advocate a particular course of action, that is good enough to make a decision. The actual value of that course of action, or its effect on the problem for which it was instituted, is of secondary importance. If there is any truth to this analysis, it is small wonder that information on the effectiveness of programs is not quickly integrated into political decision making. By the time such information is available, politicians are being pressed by other constituencies on other matters. What, then, must the evaluator do in order to be relevant and useful? The answer is that attempts must be made to influence not only politicians, but a/so the constituencies which lobby politicians. That is a much slower process than feeding information to a small body of decision-makers. It involves the process of bringing about a change in the thinking of large numbers of people—of convincing those people that particular programs may bring about a desired result.
>
> It is not enough to convince decision makers that a particular type of compensatory education will help disadvantaged children. One must also convince those who lobby decision makers to institute compensatory education programs. Valid data on the incidence of child abuse did not bring about child abuse programs. Those programs were brought about by the actions of citizen's groups who believed that such programs were needed. Social research might have hastened the advent of anti-child-abuse programs, but that hastening would have taken place only if concerned groups knew of and believed in that research. Given the need to influence a diversity of people and groups, it is certainly not likely that the results of any single evaluation will be immediately useful. The most appropriate time for evaluation seems to be *before* a new program is set up, rather than after it has been implemented. This implies a considerable amount of foresight, patience,

> and planning on the part of evaluators... [pp. 162-163; this quote and subsequent quotations cited to Morell (1979) are reprinted with permission from Program *Evaluation in Social* Research, by J. A. Morell, 1979, Pergamon Press, Ltd.).

Heretofore we have not had any aids for conceptualizing the role of values in evaluation. Because traditional quantitative methodology attempts to be value-free, values have not found a niche in the evaluative enterprise. There is a need for the evaluation community to be sensitive to the variety of values held by those individuals with some direct or indirect interest in the program. Given this perspective, the field as presently conceived may not be capable of playing the role of "multipartisan educator" as suggested by Cronbach et a/. (1980) in their call for the reform of program evaluation.

"Teasing out" the values underlying the constraints on programs would seem a legitimate evaluation activity. If one does not attempt to understand the value context of a program, then regardless of what other activities are pursued, it can be argued that one is not evaluating the program in a very important sense of the term. The modeling methodology and its attendent ideas provide one possible framework for understanding the relationships between values and what a program does. This perspective allows values to assume their proper place in program functioning. Rather than ignore values, we see them as the "guiding light" of programs. Hierarchically decomposing a program reveals its constraints at successively more specific levels of detail. Having identified program constraints, inferring values is a less risky conceptual leap.

Similar views have been put forth by Cooley and Lohnes (1976). They assert that the value statements inherent in programs can themselves "be analyzed into a set of propositions subjectable to empirical investigation and that failure to perform such analyses in evaluation studies is inexecusable [pp. 9-10]." They argue that the values that have guided educational practice have traditionally been determined by politics and custom and that their validity has not been challenged by educational researchers. They find it curious that value propositions have evaded empirical scrutiny despite educational researchers' heavy emphasis on empiricism. Clear thinking about values in education is considered essential because educational practice is greatly influenced by the value attached to desired educational goals. The alternative to rational inquiry into values is the determination of values on the basis of power, which places the educational enterprise "at the mercy of special interest groups who commend values favorable to themselves as universals [p. 10]."

As noted in Chapter 3 a basic premise of Dewey's notions about values and valuations was that values could be mistakenly viewed as absolutes only if they were considered out of context. When considered In context, values lend themselves to elucidation as propositions about real entities and the error of ascribing to them absolute or universal properties is thus avoided. The task of the evaluator becomes one of ascertaining whether value propositions inherent in a program setting reflect only convention or tradition or whether they imply empirically testable relationships between means and ends.

Finally, we want to note that because our perspective is systems-oriented, it is necessarily value-oriented. If one does not consider more generic issues that press on a program, and operates only on the level of specificity at which the program operates, then one is not pursuing a systems-oriented evaluation as we are using the term. Using terms like input, interface, and output does not make something systems-oriented. A concern for part-whole relations is systems-oriented, as when we try to link program activities and outcomes to higher-order constraints and values. In the most immediate sense these constraints and values lie in the larger program, system, or organizational unit of which the program is a part.

MEANS-END

A final but related concept embedded within our hierarchical, systems view of programs is the means-end continuum. We mean the term precisely as we did in Chapter 3 and as did Dewey (1922, 1939). He made no absolute distinction between means and ends. Any program activity or condition can be viewed as occupying space on a continuum such that it is simultaneously an end to those events and conditions that preceded it and a means to those that follow. For example, a therapist encouraging a client to improve his or her appearance is a means to an immediate end, improved appearance. Improved appearance, in turn, may be a means to more self-confidence, leading to more sociable behavior, which in turn might lead to a more positive self-regard, and so on. Dewey (1922) made the assertion that only when an end is conceptualized as a means is it fully understood, appreciated, or even obtainable.

These are some rather profound ideas. To say that some end is not even attainable unless its subsequent role as a means is realized does not bode well for the reductionistic, "here-and-now" conceptualizations of most of our program outcomes, particularly our measures of those outcomes. Does this mean we need to think things through more thoroughly or that we must appreciate the fact that you never do just one thing with any program intervention? The implications of Dewey's reasoning are the same as the examples of the mismatches between program values, intents, and outcomes reviewed in the previous chapter.

The systems modeling methodology we are suggesting embodies this concept of means-end continuum just as it does the concepts of hierarchy, transaction, and constraint. The "higher up" one goes in a program hierarchy the more general are the program intents. For example, a cognitive behavior modification treatment package for social anxiety may have as its ultimate goal, increased self-esteem. A more narrow goal would be increased socializing behavior by its participants. At a more specific level of detail the program might attempt to build self-confidence as one means of achieving this end. Within this self-confidence portion of the program may be nested transactions on improving grooming and appearance, teaching self-talk strategies for controlling anxiety, and providing practice by having participants role-play social situations.

The means-end continuum is embedded within the hierarchical structure of a systems view of a program. Our traditional methodology conceptualizes our outcomes only as discrete, unitary events. Thus the means-end continuum, so critical to understanding what a program actually does, has not been a focus of evaluation.

The Use of Models in Program Evaluation

What does this system model perspective have to offer? What will it do for us? We introduced the notions of *hierarchy, transaction, constraint,* and means-end as systems-oriented concepts that seem to characterize the way our programs are implicitly organized. We think the modeling we have suggested is useful because models constructed in such a way provide a framework or schema for juxtaposing these basic (perhaps universal) program characteristics with both the content and context of any given program.

A model is a heuristic that applies these generic characteristics to a program, highlighting the program's adherence to each of the concepts. Traditional methods do not focus on these program characteristics, but rather force on the program a reductionistic, determinist model that will be applicable in some instances but not in others. While our model assumes some other, qualitatively different kinds of conceptual and structural relationships, it does not force them on the program. The form of the program will dictate the form of the model, not vice versa.

A part of our systems perspective is a decomposition methodology with the following characteristics.

1. It is a graphic language that in precision is somewhere between prose and mathematical portrayals of an event.

2. It allows us to "see" and conceptualize an increasingly complex and multidimensional world.
3. It gives us a heuristic we need for understanding complex problems in our professions.

By now it should be clear that a decomposition model of a program is not like an organization chart of the type used by large corporations to show the relative control, power, and/or prestige of president, vice-president, division managers, unit managers and so on down the line. The process of decomposition represents a *conceptual* breakdown of the program while the typical organization chart represents a *descriptive* breakdown. The latter is less useful for planning and evaluation in that it can hide important conceptualizations needed for understanding and operating a program just as the chart shown to stockholders can hide the fact that it is actually the vice-president, not the president, who may be "running" the company.

Also, decomposition models are not "event" charts describing the administrative stages that participants of a program go through. Instead, decomposition is the hierarchical arrangement of program components revealing a family of interacting subsystems. Interactions can take place both among component parts within subsystems and between subsystems. These interactions represent the ways in which subsystems and their component parts are coupled, or combined into larger components to produce an outcome more complex and more encompassing than could be produced by any single component.

With enough information about program functioning, decomposition models can be made sufficiently precise and detailed to show the conceptual and psychological underpinnings of the program. Decomposition methods represent a detached view of the program such that, ultimately, one may be led to the conclusion that there is no program at all. Hence, users of decomposition methodology should be aware that decomposition methodology will dig beneath the surface impressions of a program to find its conceptual structure, and finding none, will so reveal this fact, regardless of any preconceived notions of what the program is supposed to be doing.

Figure 7.5 presents a conceptual schematic of our program modeling approach. At the most general level, the program is represented as a single box. Values providing impetus for development and maintenance of the program appear as constraints at this level. At the next level of program detail, the larger box representing the whole program is broken into its major subunits, depicted here as A_1, A_2, A_3. The number of subunits and their labeling designations are arbitrary. At the next level of detail, A_1, A_2, and A_3 could be further decomposed. A hypothetical decomposition of A_2 is illus-

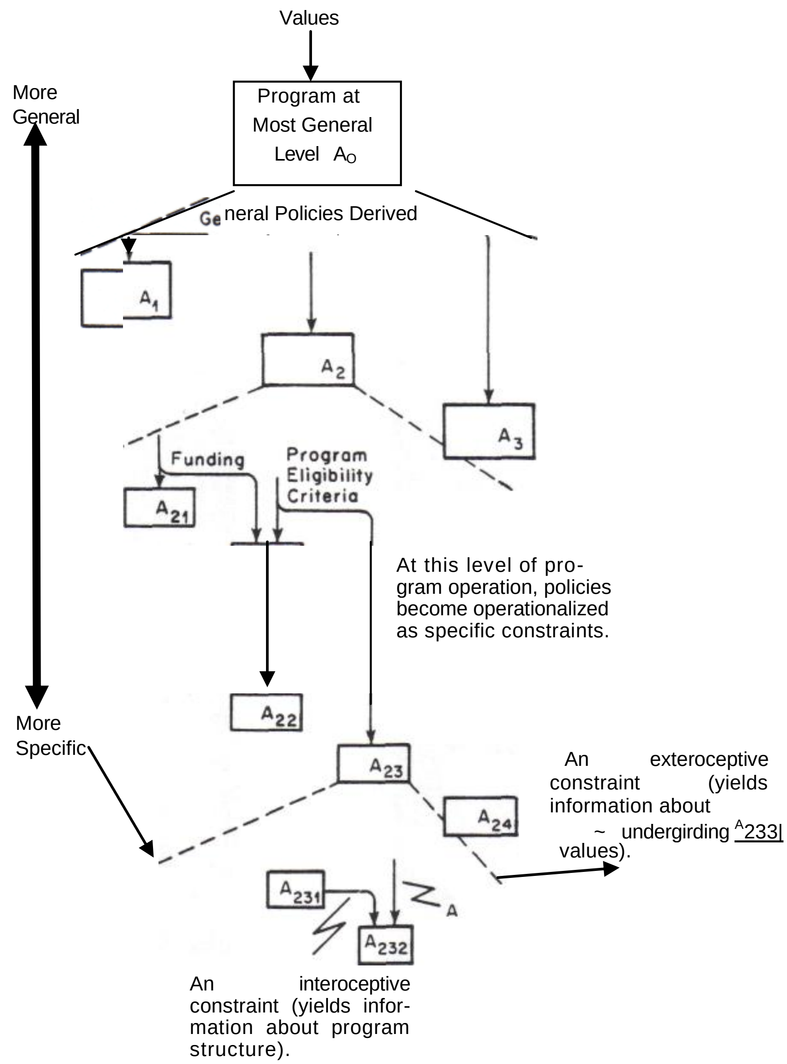

Figure 7.5. A context-free program model.

trated. The four subdivisions of A_2 (A_{21}, A_{22}, A_{23} and A_{24}) could be further decomposed and so on.

The values that appear as constraints at the topmost box become successively more specific as one moves down into program details. Values become operationalized with successive levels of specificity in the hierarchy, thus defining important program parameters such as, "Who will be served?" "At what cost?" "In what ways?" "By whom?" "For what reasons?"

At more specific levels of program detail, the transactions represented by boxes become smaller and smaller. In other words, the program "chunks" focused on are smaller and more specific than is the case when one is

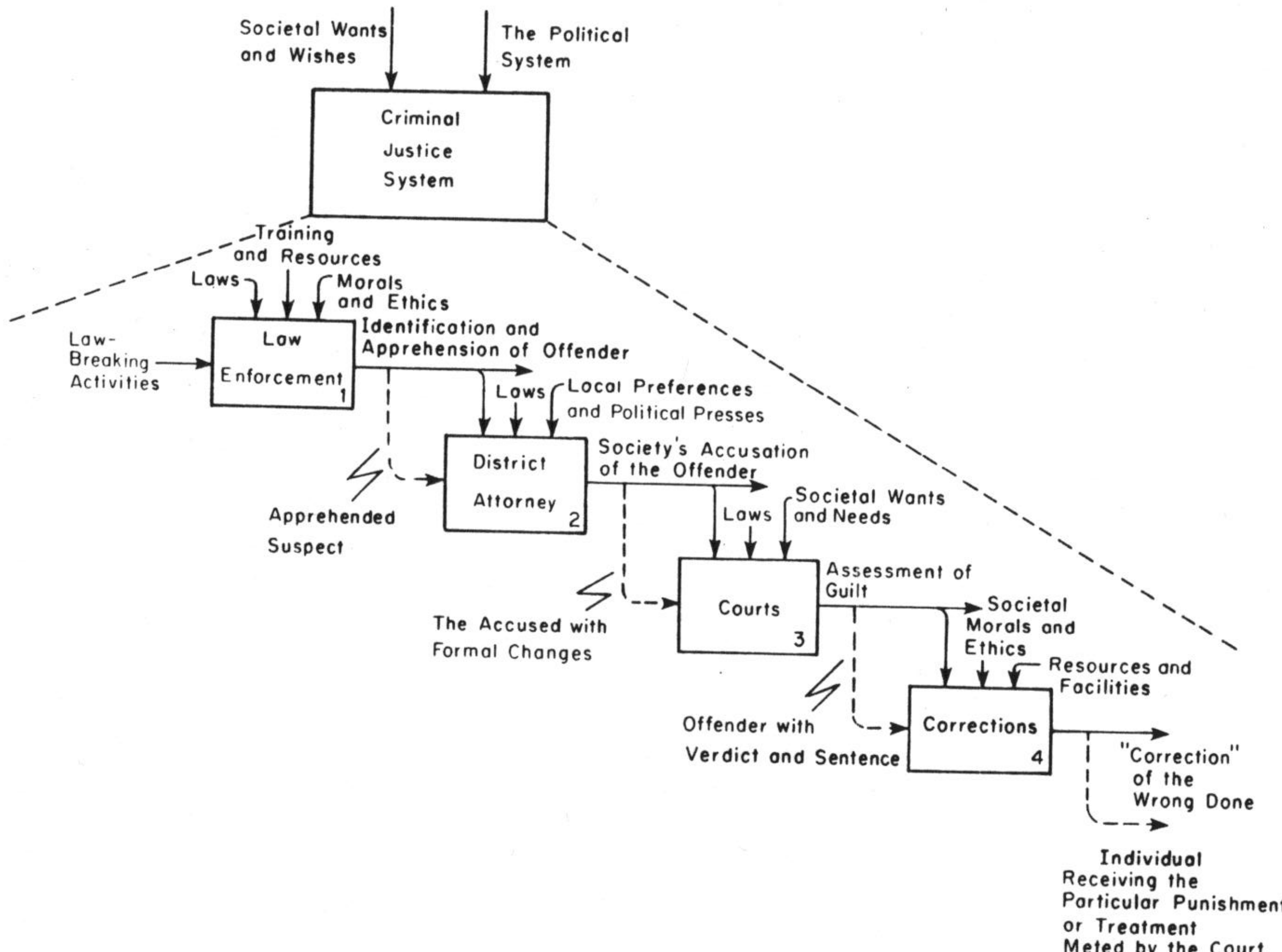

Figure 7.6. **Four basic components of the criminal justice system.**

considering a broader level of program activity. Figure 7.5 illustrates the highly salient role played by hierarchy in this view of programs. Any activity can simultaneously be seen as a program unto itself and as a component part of some larger effort.

The outcomes of transactions at each level of detail should be consistent with and contribute to the attainment of the objectives of the next larger system of which the smaller transactions are a part. It is in this manner that the concept of a means-end continuum is embedded within this modeling perspective. Each outcome or "end" is a "means" to the outcome (or end) at the next higher level of generality in the model. The same is obviously true for constraints. Constraints at a specific level of program detail need to be consistent with higher-order constraints in upper echelons of the hierarchy. To find ends that have no subsequent role as a means to some higher-order end is itself a significant program evaluation finding. The same is true about inconsistencies in constraints at different levels of generality. Such a finding would suggest that very different values are constraining functioning at various levels of program operation, which might hinder overall system performance.

Figure 7.6 presents a high-level view of the criminal justice system. The example is hypothetical but fairly realistic as it is based on personal experience in working with criminal justice systems. In this decomposition the system is subdivided into four primary subcomponents. The primary objective of law enforcement is the identification and apprehension of offenders. This activity is constrained primarily by the laws of the land. These include society's statements about acceptable and unacceptable behavior as well as laws about the treatment of suspects (e.g., use of wiretaps, informing an apprehended suspect of his/her rights). Any law enforcement activity is also constrained or shaped by the training and resources provided to police as well as these individuals' personal morals and ethics (i.e., values) about their work.

Another major activity within the criminal justice system is that process through which formal charges are brought against a suspected offender. !n our society this most often occurs through the office of a District Attorney (D.A.) who is usually an elected official. Constrained by the work of law enforcement personnel, public laws, and political presses, a formal accusation is either made or not made. If the case brought against *a* suspect is weak and without merit or if a suspect's rights are violated in the process of apprehension, charges are likely to be dropped. This is one of the ways the actions of police are an important constraint on the activities of the D.A. In addition, because he is an elected official, the D.A. is pressed to please his constituency. Public concern over a wave of crime or a well-organized effort by a powerful or vested interest group might significantly affect the functioning of the District Attorney's office.

Society's accusation of the offender (formal charges) then serve as one of the primary constraints on another major component of criminal justice, the court system, in addition to formal charges and binding laws, courts are also constrained by societal wants and needs. Encouraging plea-bargaining to clear overloaded court dockets is one example of how societal needs may constrain court activities and avoid a bottlenecked judicial system. Judges who mete out mild sentences for violations of unpopular laws and the judgments and punishments meted by juries are examples of how societal values may function as constraints on the court. Some might argue that personal preferences and predispositions of the judge (i.e., the particular *judge's* values) are an important constraint here as well.

Once adjudicated by the courts, a guilty offender is passed on to the corrections division of the criminal justice system so that the offender can be administered the treatment determined by the courts to be fair and equitable for the offense committed. This "correction" is constrained by societal values about the treatment of offenders and by the resources, training, and facilities available to correctional personnel. It is sometimes argued that the values of correctional personnel themselves are a major determinant of what happens to offenders in the corrections system and some may insist on this constraint appearing in the figure.

While corrections can take many forms, it most commonly occurs as one of the four types suggested in Figure 7.7, fines, probation, parole, and incarceration.

Adams (1975) points out that the primary goals of incarceration are (a) offender rehabilitation, (b) punishment, and (c) incapacitation or detainment. Someone's values will ultimately determine whether the *major* thrust of a correctional agency should be custodial, rehabilitation, or punishment. Most prison systems contain elements of all three. This conclusion brings to light some interesting questions. Can we expect a correctional program to simultaneously incapacitate, rehabilitate, and punish an individual? Whose values are the most powerful constraints on an activity? Are lower-level outcomes consistent with those higher in the system? These are but a few of the evaluation issues that are brought to light by this program modeling approach.

Understanding of a program increases by crossing strata. In moving up the hierarchy one obtains a deeper understanding of its significance. One finds broader concepts and longer periods of time, thus allowing the meaning of the program to be more accurately and more easily interpreted. In moving down the hierarchy one obtains a more detailed explanation of the program. One finds narrower concepts and shorter periods of time, allowing

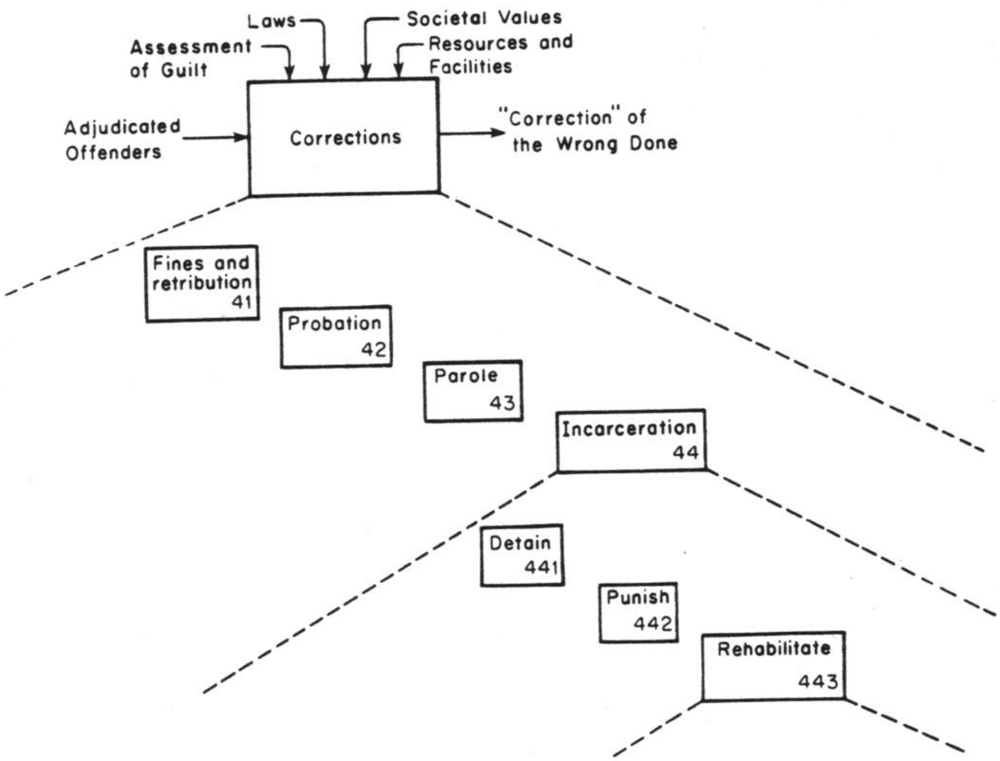

Figure 7.7. **Further decomposition of the criminal justice system.**

for a more precise and detailed explanation of how the system functions to carry out its objectives. Specialization, division of labor, and coordination are three characteristics revealed by a program decomposition. In summary, there are several common features of decomposition.

1. A higher level unit is concerned with a larger portion or broader aspects of the overall systems behavior.
2. The decision period of a higher level unit is longer than that of lower units.
3. A higher level unit is concerned with slower, more time consuming aspects of the overall systems behavior.
4. Descriptions and problems on higher levels are less structured, with more uncertainties, and are more difficult to formalize quantitatively [Mesarovic, Mako, and Takahara, 1970, pp. 54-56].

Summary

Use of the models we are proposing allows us to see the relationships that may exist in program functioning, not necessarily in a mathematical, quantitative sense but in a conceptual sense. Their use provides a way of seeing more "things" at a time with our limited intellectual capacities. They serve as advance organizers, allowing us to use the information we do possess in a more efficient and productive manner. Use of such models might also allow us to use our traditional statistical methodology more efficiently by teasing out issues and variables conceptually and suggesting ways they may interact to produce the outcomes that do accrue. More appropriate and efficient use of statistics would increase the power of evaluations, not power in the statistical sense (the probability that a statistical procedure will detect a difference that really exists) but power in the sense that the results obtained are important and useful to know.

This modeling perspective represents an attempt to fully understand the context of a program or evaluation problem. We firmly believe that having a conceptual grasp of the context and being aware of the more generic aspects of program functioning makes one a "smarter" evaluator in some ways.

The description of program modeling presented in this chapter represents a rather roundabout way of talking about common sense, intuitive judgment, and evaluation. We have made much of the idea that one should devote some effort to understanding the context in which one toils; by so doing, one is more apt to recognize the role he or she is playing in some larger means-end chain. Knowing this allows one to either better fill the role prescribed or to interject one's own wishes or needs into the work context. Most experienced evaluators delight in pointing out incidents in which they were particularly insightful in applying their expertise. Our experience has been that these instances are often concerned with helping the client or employer to understand needs or appreciate the context in which a program or some personnel operate. Only rarely are these incidents concerned with traditional measurement and experimental design issues.

Although some evaluators already have a systems conceptualization of programs, there are many who do not. Training for evaluators sometimes has consisted exclusively of traditional statistical and measurement methodology. We are advocating the formal inclusion of qualitative or commonsense knowing (Campbell, 1978), clear thinking about issues and events from perspectives other than the cause-and-effect paradigm, and the use of alternative heuristics, such as modeling, as significant aspects of evaluation methodology.

Some would say that we are only talking about a pre-evaluative stage in which one logically thinks through that which will be evaluated. Although they would be right, most evaluators operate from a narrower perspective that does not always provide the evaluator the opportunity of a pre-evaluative stage either because of the myopic view the evaluator has of his or her own role vis-a-vis some program or organization, or because of limitations or constraints placed on the evaluator by the environment. For whatever reason, these constraints limit evaluative insight.

As Adams (1975) points out, a systems or "common sense" perspective can take several forms. Thus, such an approach might be a highly formalized systems analysis methodology that may or may not include a graphics syntax, mathematical models, and the like. Alternatively (and more comfortable to us), this systems perspective might be pursued more informally. Rather than a highly visible process, one's "systems analysis" might be performed by the systems-oriented evaluator within the privacy of his or her own thinking about a program. Regardless of the mode pursued, the critical issue is whether the program and its evaluation are understood, particularly within its larger enveloping contexts. We offer our modeling perspective as one means for achieving such an understanding. For readers who wish to learn more about this modeling approach, we recommend the appendix to this volume.