# CONSTRUCTIVE IMAGES AND DIAGRAMS:
# THEIR ROLE IN INFORMATION SYSTEMS DEVELOPMENT

by

**William C. Sasso**
Information Systems Area
New York University
613 Tisch Hall
40 West 4th Street
New York, NY 10003
(212) 998-4203

June 1988

Center for Research on Information Systems
Information Systems Area
Graduate School of Business Administration
New York University

**Working Paper Series**

CRIS #183
GBA #88-51

Center for Digital Economy Research
Stern School of Business
Working Paper IS-88-51

# Constructive Images and Diagrams: Their Role in Information Systems Development

William C. Sasso
Center for Research on Information Systems
Graduate School of Business Administration
New York University

June 23, 1988

*Abstract: In order to promote more creative solutions to Information Systems (IS) design problems, this paper identifies four roles that images or diagrams can play in the IS development process. These roles are characteristics of the interaction between the image and its creator or viewer, rather than of the diagram itself. One of these roles in particular, the constructive role, can do much to support the generation of creative designs, to the benefit of both systems developers and their clients.*

*The goal of constructive systems thinking is to enhance the creative solution of IS development problems, and it cannot be reduced to a specific, highly structured technique. We can, however, outline a general approach to building constructive images:*

1. *Create a set of candidate analogies, elaborate them, and evaluate the degree to which each guides design of the target system.*

2. *Evaluate how completely this working set of analogies informs the important aspects of the target system, and create additional analogies to fill any major gaps.*

3. *Over the relevant scope of each analogy, research its structure and dynamic interactions. Use these as templates within which to model the target system.*

4. *Validate this design and its functional implications with the system client, adjusting it as required.*

5. *Complete the design by removing details specific to the analogous system and adding those relevant to the target system.*

Center for Digital Economy Research
Stern School of Business
Working Paper IS-88-51

## 1. Introduction

Many approaches to and methods for the development of Information Systems (IS) recommend particular types of diagramming techniques. The Yourdon School of Structured Analysis espouses Data Flow Diagrams (DFDs) and decision trees [8, 4]. The popularity of Entity-Relationship Analysis has resulted in the development of a standard format for E-R Diagramming [3]. Both program flowcharts and systems flowcharts are widely understood (if less widely used) diagramming techniques. Other types of diagrams, such as Nassi-Schneiderman charts, Warnier-Orr diagrams, and menu-based system representations abound. But among this surplus of diagramming techniques, we have little, if any, empirical evidence regarding the superiority of one technique over another, or indeed concerning the general contribution of diagramming to successful IS development.

This paper will identify four distinct roles which diagrams can play in the development of Information Systems. Three of these roles, documentation, communication, and validation, are fairly well understood. The fourth role, that of constructing new insights into the problem or possibilities for its solution, has received far less discussion and attention. A diagram serves a constructive role when it depicts a set of problem requirements in such a way as to suggest a possible design for a solution to the problem. This paper's objective, then, is to clarify the concept of constructive diagrams, and to discuss their application in IS development. In doing so, we will also and discuss the use of analogy as a source of the constructive diagram.

The next section will delineate the four roles, or types of interaction, an image or diagram can have with its creators or viewers. Following this, we will discuss constructive systems thinking and illustrate the idea with a detailed example, in which we will show how different images of the financial consolidation process could lead to very different systems being built for that application. The fifth section will present a general approach and several guidelines for the application of constructive systems thinking in IS development.

Center for Digital Economy Research
Stern School of Business
Working Paper IS-88-51

## 2. The Possible Roles of Images and Diagrams

There are at least four roles any diagram can play in the IS development process. First, it can *document* its creator's currently existing knowledge for future reference. Second, it can help *communicate* that knowledge to others. Third, it may help its creator *validate* his or her current understanding of the system, problem, or situation through the development of a logically coherent, complete, and consistent representation. Finally, the diagram may depict the requirements of the situation in such a way as to suggest the design of a possible system -- it can play a *constructive* role.

These various roles reflect the outcome of the diagramming process rather than its initial intention. The analyst may, for example, begin to create a diagram with a purely documentary intent and find that he or she has generated some new insight about the system represented. For example, perhaps he or she notices for the first time that two of its subsystems have very similar hierarchies of interaction between their modules, which in turn suggests that he may be able to use variations of a single control hierarchy to control the execution of their modules. In this case, the diagram has played a constructive role. Alternatively, the analyst may attempt to create a constructive diagram, and fail, but later be able to use his results successfully as a basis for communication.

**The Documentary Role:** Systems developers often need to store the inputs to or the outcomes of a design decision for their own future reference later in the development process. Here the particular system, component, or problem is clearly understood, and the analyst gains no particular insight into it by creating the diagram. The diagram simply provides an easily understood and relatively concise storage mechanism, in that its creation requires less effort than that of a narrative text description would.

**The Communicational Role:** Here, again, the developer or analyst completely understands the situation or system component to be diagrammed, and portrays it in a diagram in order to explain it to others more effectively. Thus, any diagram used to

Center for Digital Economy Research
Stern School of Business
Working Paper IS-88-51

get an idea across to others is playing a communicational role. This includes documentation prepared for use by others, training and presentation materials, and so on.

**The Validating Role:** In this case, the analyst *believes* that he or she understands the idea, situation, or system, and uses a diagram to test the degree to which that understanding can stand up under fire. If he can create a logically complete and consistent diagram, that in itself is a weak demonstration of his understanding. Often such diagrams will assist in generating a stronger type of proof, via the simulation of the activities and cycles of the system [1]. For example, the analyst may be designing a system in which five states are possible. In order to test his understanding of the system, he can diagram the sets of possible progressions between each of the states. Suspicions he holds regarding the possibility of traveling from state 1 to state 3 may crystallize via his mental simulation.

**The Constructive Role:** Here, the developer identifies certain features of the system or its environment, and, working from these important aspects of the problem, selects a candidate analogy. For example, an analyst working on a student-course registration system might choose an airline reservation system as a promising analogy. He can then elaborate the analogy by creating an image of the student-course registration system cast in the form of an airline reservation system. This enables him to validate the analogy as a reasonable and promising one. Further, it helps determine the scope or boundaries of the analogy's promise. A given analogy can inform another IS problem only incompletely: *it is an analogy -- not a complete blueprint*. For example, while airline reservation systems have been instruments of competitive advantage to several airlines, there is little reason to suppose that student-course registration systems can play such a role for a university. At a more detailed level, the smoking versus non-smoking distinction made in airplane seating assignments has no corresponding one in classroom seat assignment.

The constructive image never tells the IS designer "everything he always wanted to

Center for Digital Economy Research
Stern School of Business
Working Paper IS-88-51

know about Application X (but was afraid to ask)." Rather, it provides important but incomplete assistance in bounding the system, identifying environmental constraints, defining required system functionality, and synthesizing a creative and appropriate design solution.

The promising constructive image can be further elaborated into a more complete and consistent set of constructive diagrams, expressing the target system's structure and dynamics in the form of the analogous (base) system.[1] To continue our example, the analyst would articulate how individual aspects of the student-course registration context correspond to the elements of the airline reservation context. Here, flights would correspond to course sections, airplanes to classrooms, flight crews to instructors, airfare to tuition, and so on.

From this picture, the analyst gains new insight about the target system, its corresponding aspects, and their interrelationships. In some cases, he may actually begin to "see" a possible way to configure the system components. This is often a trial and error process. Rather than depicting existing knowledge for his own use (documentation), for that of others (communication), or for the verification of "suspected" knowledge (validation), the act of identifying a constructive analogy and elaborating it into a constructive diagram provides *new* knowledge about the problem or its potential solution.

------------------------

FIGURE 1 ABOUT HERE[2]

------------------------

[1]Following Gentner [5], we will use the term "target system" to refer to the (desired) system being conceptualized in the form of another (existing) system, the "base system." Here, the student-course registration system is our target system, and the airline reservation system our base system.

[2]Figures will be found at the end of the paper, following the References.

Center for Digital Economy Research
Stern School of Business
Working Paper IS-88-51

These four roles are depicted as a hierarchy in Figure 1, with the documentary role as the base of the hierarchy and the constructive role as its pinnacle. Note that each higher-level role includes the interaction effects of those roles below it. In communication, for example, the diagram not only helps the communicator get his ideas across to the audience, but also recalls to his mind the issue to be communicated, serving a documentary role as well. Similarly, a diagram playing a constructive role will also perform the validating, communicational, and documentary ones. For this reason, each higher-level role is pictured as overlapping the lower ones along the right side of the triangle.

Christopher Alexander, a practicing architect and professor of architecture, describes the constructive diagram as follows:

> The constructive diagram can describe the context, and it can describe the form. It offers us a way of probing the context, and a way of searching for form. Because it manages to do both simultaneously, if offers us a bridge between requirements and form, and therefore is a most important tool in the process of design.
>
> In all design tasks the designer has to translate sets of requirements into diagrams which capture their physical implications. In a literal sense these diagrams are no more than stages on the way to a specification of form, like the circulation diagram of a building, or the expected population density map for some region under development. They specify only gross pattern aspects of the form. But the path from these aspects to the final design is a matter of local detail. The form's basic organization is born precisely in the constructive diagrams which precede its design. [2]

## 3. Characteristics of Constructive Systems Thinking

If we use the term "constructive systems thinking" to encompass analogies, images, and diagrams, we can argue that it has several standard characteristics. It projects a structure from the base system to the target system, assists in the identification of alternative solution approaches, provides a meaningful context for client involvement, and generates images that are, by definition, independent and incomplete.

Center for Digital Economy Research
Stern School of Business
Working Paper IS-88-51

The projection of structure from the base system (e.g., the airline reservation system) to the target system (e.g., the student-course registration system) is depicted in some detail in Figure 2.

------------------------------

FIGURE 2 ABOUT HERE

------------------------------

Figure 2 compares, at an admittedly gross level, the entities and relationships involved in airline reservations and student-course registration. The power of the analogy rests only partly on the correspondence between entities, such as passengers/students and flights/sections. More important is the set of similar relationships between the entities, such as BOOK [Passenger, Flight] corresponding to REGISTER [Course,Student] and OPERATE [Aircrew, Flight, Airplane] corresponding to INSTRUCT [Faculty, Section, Classroom]. The identification of these parallel relationships is the means by which the analogy informs our design process. To the extent that the relationships map from the base to the target system, elements and configurations of the target system which will enact those relationships can be patterned after the corresponding elements in the base system. To use Alexander's words, the "... basic organization is born ..." -- what remains is to adjust this organization to include details (e.g., object-attributes) in the target system not present in the base and vice versa. For example, most schools have not yet resorted to overbooking classes, a feature common to many airline reservation systems. Such work, while important, can be characterized accurately as local detail.

How do constructive ideas help identify alternative designs? At early stages of the IS development process, alternative analogies representing different conceptualizations of the problem and its environment, can be elaborated into alternative design problem solutions. In some cases, these solutions will conflict, in that their associated solutions

Center for Digital Economy Research
Stern School of Business
Working Paper IS-88-51

are mutually exclusive. For example, the real-time assignment of passengers to flights in airline reservation systems conflicts with certain other possible analogies to the student-course registration system, such as those where matching/tabulation is performed in a single batch process (e.g., the annual selection of interns by hospitals across the United States in a single batch process). We would term these "competing images."

At the initial stages of the IS development process, then, the identification and elaboration of conflicting or non-intersecting constructive images is valuable. These images help the client and analyst identify alternative approaches to solving their problem. In most cases, however, a single image will predominate later in terms of informing and guiding the actual design and implementation of the system.

For a set of diagrams to play a constructive role, they should be independent in the sense that a modificaiton to one diagram should not require corresponding modifications to the others. In Figure 2, for example, we see that a system of important relationships holds for two distinctly dissimilar sets of entities -- one a collection of entities involved with student-course registration and the other a set related to airline reservations. Subordinate diagrams of the structure of student/passenger or instructor/aircrew can change without changing the relationships protrayed in the figure. This independence can be achieved by developing diagrams in a hierarchically-organized fashion, such as that which would be used to develop Data Flow Diagrams [8]. This provides both the necessary linkage between diagrams and their requisite independence.

This independence is important for two reasons, as Alexander describes:

> The idea of a diagram ... is very simple. It is an abstract pattern of physical relationships which resolves a small system of interacting and conflicting forces, and is independent of all other forces, and of all other possible diagrams. The idea that it is possible to create such abstract relationships one at a time, and to create designs which are whole by fusing these relationships ...
>
> I have discovered, since, that these abstract diagrams not only allow you to create a single whole from them, by fusion, but also have other even more

Center for Digital Economy Research
Stern School of Business
Working Paper IS-88-51

> important powers. Because the diagrams are independent of one another, you can study them and improve them one at a time, so that their evolution can be gradual and cumulative. More important still, because they are abstract and independent, you can use them to create not just one design, but an infinite variety of designs, all of them free combinations of the same set of patterns. [2]

Why is it that constructive ideas are incomplete? A single constructive idea is almost necessarily incomplete, in that no analogy is perfect. Therefore, the analyst and client should work together to identify several analogies. In the best case, these will complement each other in two ways. First, each image and its implications will correspond to unique aspects of the target system; they will overlap only minimally in terms of suggesting conflicting solution approaches. Second, the partial solutions suggested by each analogy will fit together in a straightforward manner, implying an identical or congruent system structure and dynamic.

Consider an example of incompleteness. The airline reservation system analogy is a very powerful one for the student-course registration problem, but it is certainly only a partial one. There are, for instance, major distinctions between the airline ticket billing and payment process and the typical Student Accounts Receivable process. One major distinction concerns the multiple and complex airfare structures currently in use, as opposed to the relatively straightforward calculation of tuition charges. A second difference focuses on the scope of services for which the client (passenger or student) is billed, which at universities includes not only tuition (corresponding to airfare) but also student fees, room and board, health insurance, etc. A third major distinction emanates from the fact that many students receive financial aid from their universities, while few (if any) air travelers receive any equivalent compensation from their airline. The image is a valuable and constructive one, but clearly an incomplete one.

What types of insights can a constructive diagram generate? As discussed earlier, it can suggest appropriate control structures for a system, or make explicit the types of interfaces a system must possess, both to its human users and to other information

Center for Digital Economy Research
Stern School of Business
Working Paper IS-88-51

systems. It can clarify the need for interface capabilities within the system, between its hardware, software, and data storage components. It can depict dynamic aspects of system operation, such as the set of possible future states to which the system can move directly from a given current state. Diagrams may present requirements for speed and volume in such a way as to suggest how they may be achieved. They can illustrate the likelihood of system failure and its associated repercussions. Finally, a diagram can merge several of these types of issues, concisely depicting their interrelationships.

## 4. An Example: Constructive Diagrams for Financial Consolidation

Suppose that an analyst is working on the design of a financial consolidation system, which takes the general ledger data from individual reporting units and aggregates it into a consistent set of financial data for an entire firm. One possible way of organizing to perform this function is along the lines of the (abstract) consolidation accounting hierarchy itself, with elements and form corresponding to the corporate structure. Alternatively, the analyst might nominate the "black box" as an analogy. The black box model focuses on inputs and outputs of a system without concerning itself with the system's internals. Here, the general ledger data from the firm's many reporting units is the input, and the stream of various reports produced is the output. Constructive diagrams representing these two (competing) images are shown in Figures 3 and 4 respectively.

-------------------------

FIGURE 3 ABOUT HERE

-------------------------

The corporate consolidation hierarchy as analogy is shown in Figure 3, entitled Stepwise Consolidation. Here, the consolidation process is modeled directly on the organization chart -- divisions consolidate their plants' data, and report their consolidated data directly upwards to their superior entity. Each of these consolidations, for the sake of consistency, the analyst might show in Figure 3 as a triangle. Because they are all shown as triangles, it may occur to him to ask "Do I want to perform each

Center for Digital Economy Research
Stern School of Business
Working Paper IS-88-51

of these consolidation processes with the same software throughout my firm?" If so, he may become quickly aware that ease of installation and machine portability will be very important aspects of this system component. Ideally, he might want to do the same thing with my interface modules, but he can see from the diagram that their inputs are different -- some use yen, while others use pounds, deutschmarks, and dollars, indicating not only currencies but also accounting practices in use. This means that the conversion processes will probably be locally (or regionally) developed and maintained, due to the importance of expertise concerning the local rules of accounting; and that at the same time, they all have to interface with the corporate consolidation system, implying that it forms a binding constraint on all of them. In other words, each one can be different, but they all have to produce directly comparable outputs.

---------------------------

FIGURE 4 ABOUT HERE

---------------------------

In Figure 4, we see a very different picture, derived from the "Black Box" analogy to financial consolidation. Let us call this approach "Direct Consolidation." Since this system consolidates in a single place, the relative importance of portability and ease of installation decline. At the same time, since the size of the single consolidation becomes immense, the internal (machine) efficiency of the consolidation software should assume a greater importance. If the analyst expects the hardware environment and technical support staff to remain stable over the medium-run (e.g., 3-5 years), he may consciously choose to take hardware performance and operation factors heavily into account when designing this software.[3]

The implications drawn concerning the interface modules also differ due to the matrix type treatment they receive in Figure 4. Consider the lower portion of the diagram, in which the individual reporting units are portrayed. The analyst might think of this as a matrix, with each column representing a type of reporting unit (e.g., a sales district,

[3]For a further discussion of the relative advantages and disadvantages of these alternatives, see [7].

Center for Digital Economy Research
Stern School of Business
Working Paper IS-88-51

plant, or research unit), and each row representing the presence of that type of unit in a specific country (e.g., the United States, as represented by the dollar sign). At least two potential approaches, corresponding to the two dimensions of the matrix, suggest themselves. He can define classes of reporting units, e.g., headquarters units, plants, research labs, etc., and develop interface modules intended to serve each of these units. These would then be extended or customized to handle any local variation in accounting practice or currency. Thus, each column in the matrix would be considered a fundamentally homogeneous group.

Alternatively, the analyst could consider the rows, which describe the set of different operational units present in a given country (represented by a currency symbol) as the fundamentally homogeneous units. In that case, he could design interface modules for converting the United States, United Kingdom, Japanese, Philipine, and Australian accounting and currency systems into our corporate one. These would have to cover a wider range of capabilities (corresponding to the wider range of functions performed across the various types of units), but would most likely require less modification and customization for use in each reporting unit than would the conversion modules developed from the reporting unit (or row-based) approach.

These diagrams tell the analyst a great deal, but they do not tell him everything. For example, neither of these diagrams says anything about the specific algorithms or data structures used to process the consolidation. Neither says anything about the number of modules in either the conversion or consolidation processes, and neither makes any statement regarding whether the systems are manual or computerized. We could, for example, use either a systems flowchart or a DFD to describe the "innards" of the conversion and consolidation processes shown in either figure. Moreover, these flowcharts or DFDs could change without the higher-level figures having to change. In Alexander's words, Figures 3 and 4 "... specify only the gross pattern aspects of the form." But that in itself is a major contribution.

Now suppose that the analyst were to use Figure 3 as a basis for discussion with his

Center for Digital Economy Research
Stern School of Business
Working Paper IS-88-51

clients. Upon looking it over, someone might point out such relevant factors as "Actually, our sales district in the UK keeps all their books in US currency anyway" and so forth. This type of interaction would represent micro-validation, i.e., the testing and correction of existing knowledge at the level of individual facts. More troublesome and valuable, of course, are comments like the following:

> Yes, this is very nice, but it only does half the job. It brings us, very nicely, to the point at which the data has been consolidated. But we don't consolidate for consolidation's sake, we consolidate in order to prepare reports of various types for internal management, directors, and external investors, creditors, regulators, and taxing authorities. In other words, once the data gets to the pinnacle [of Figure 3], it has to be broken out again, sometimes in standard pre-defined reports, and sometimes in a flexible, easily manipulable fashion with very rapid turn-around to support our management decision-making processes.

This comment demonstrates that the analyst has achieved a macro-level validation interaction, which in this case involves determining whether the scope of the system is defined similarly by the developers and the clients. Furthermore, this diagram can play communicational roles in both directions, not only from the developer to the client, but from the client to the designer as well. For example, a manager might comment:

> You know what else I don't see here? I manage corporate financial analysis. When we get actual financials, either for a division or for a group, the first thing we do is to compare them with the budgeted and forecast figures for that unit. Now our budgets and forecasts get aggregated in a very similar fashion to actuals, but we do them on a completely separate system with a partially incompatible chart of accounts. So we wind up converting a lot of these budget account figures by hand, in order to have a basis for comparison. Since this proposed system has interfaces to alternative charts of accounts used by our overseas subsidiaries, would it be possible to build an interface module to our budgeting and forecasting system?

This comment also can also be characterized as macro-level validation. These issues, the need for reporting capabilities and the interface to other data sources, both relate to the proposed scope of the system. Both are important. If the analyst has not considered these questions already, the diagram has played a constructive role by eliciting them. If

Center for Digital Economy Research
Stern School of Business
Working Paper IS-88-51

the analyst has seen and addressed them, he or she will have another diagram of the subsystem(s) designed to handle these issues ready at hand. Another possible reaction to the second issue is to broaden the discussion, by introducing the question "Should we build an interface to the Budgeting/Forecasting System, or should we consider extending this system to assume the function of aggregating Budgeting/Forecasting figures as well as actual financials?" The reciprocal issue, which may certainly merit discussion as well, might also emerge: "Is it feasible to extend the Budgeting/Forecasting System to handle the consolidation of financial actuals?" Even if the decision is made to leave the two independent systems (budgeting/forecasting and financials) in place, the consideration given the question may well improve the design indirectly, by indentifying the Budgeting/Forecasting system as a relevant analogy. Its strengths can be emulated or surpassed, and its weaknesses can be considered and avoided.

## 5. An Approach to Constructive Systems Thinking

Here we will outline an approach to constructive systems thinking in IS development. We will sketch a set of preconditions which will facilitate the process, outline the process itself, and then caution the reader against several of its common pitfalls.

**Preconditions for Constructive Systems Thinking:** What has to be true for diagrams to play a constructive role? First, the analyst has to realize that this is a trial and error process, and that often an unsuccessful attempt to create a constructive diagram will indirectly bring him closer to his goal. Secondly, he has to allow a certain amount of free association to occur in his mind as he considers the problem. Rather than abruptly close off thoughts like "This looks like ..." or "You know, that reminds me of ...," he has to remain sufficiently flexible to accord each of these investigation.

Third, the analyst must force himself to generate several constructive images. He should not be content to settle for the single, "obvious" one. Research [6, 1] suggests that experience with an appropriate analogy to the target system may be a critical factor in successful software design. It further suggests that designers are most likely to

Center for Digital Economy Research
Stern School of Business
Working Paper IS-88-51

draw analogies from their own past experience, which is fine so long as that experience informs the current situation in a meaningful fashion. But the most valuable analogy is not necessarily the most obvious one. Rather, its value will be determined by its congruence to the target and our ability to understand and adapt the elements and configuration of its solution.

The fourth important condition for the creation of constructive diagrams concerns the richness and flexibility of the representation itself. DFDs, for example, provide a relatively impoverished set of representational capabilities, allowing only four diagrammatic elements, and depicting each instance of an element (e.g., each process) in exactly the same form. Nonetheless, these capabilities could be enriched fairly easily. For instance, the thickness of a data flow could be used to represent the current or expected volume of data. The size of a process could indicate its frequency of execution, and its shape or color could denote whether it is currently performed manually or has been computerized. Alternatively, different shapes or colors could be used to code processes according to their function or logic -- in a consolidation system, for example, one could use triangles to show the actual consolidation/aggregation activities, and depict conversion activities (e.g., from a subsidiary's chart of accounts to the consolidation chart of accounts) as squares. The height of each consolidation process triangle could then represent its speed of execution, and the length of each data flow could represent either geographical distance or actual transmission time.

Thus a classical DFD is unlikely to be an effective constructive diagram, primarily because of its "narrow bandwidth." It can present only a narrow range of information, and fails to show many of the important dimensions of an information system. On the other hand, a DFD may be an excellent point from which to begin working on constructive diagrams, due to the independence of individual diagrams conferred by the DFD's hierarchical structure. Once the analyst has created a skeletal picture of the system, in terms of its processes, data flows and stores, and external interfaces, he can flesh it out by adding the additional dimensions most critical to the particular situation.

Center for Digital Economy Research
Stern School of Business
Working Paper IS-88-51

**Constructive Systems Thinking -- A Process Outline:** Constructive systems thinking cannot be done in a regimented, algorithmic process -- it requires intelligent, imaginative, and sometimes critical thinking. The steps we describe here are an outline, rather than a hard and fast procedure to be mindlessly executed.

1. Create a set of candidate analogies.
2. For each of them, develop a constructive image.
3. Determine the relevant scope of each analogy.
4. Compare the implications of those analogies whose scopes are similar -- at each level of scope overlap, make a tentative selection.
5. Evaluate the degree of target system coverage provided by the current set of analogies.
6. For each member of the current analogy set, research the structure and dynamic of the base system (over its relevant scope).
7. Form a set of constructive diagrams, each casting the target system in structure of the base system over its relevant scope.
8. Validate the resulting working design and its functional implications with clients, adjusting as required.
9. Complete the design by removing detail local to the base system and adding that required by the target system.

The critical issue in the first step is to generate a good set of analogies from which to choose. This is an excellent place to get client personnel involved, because they probably already use some analogies to describe the system, either as it is currently or as they would like it to be. To the greatest extent possible, try to focus their analogies on man-made systems, which generally provide the most valuable guidelines (as discussed later in this section).

In the second step, the analyst and each client attempt to understand the latter's analogy and to portray it in graphic form. These could, but need not, look like Figures 2, 3, or 4 (above). The form itself should vary according to the salient aspects of the

Center for Digital Economy Research
Stern School of Business
Working Paper IS-88-51

analogy. One critical issue here will be to uncover the systematicity of the analogy, the degree to which the base system's higher-order relations map meaningfully to higher-order relations in the target system. The goal is to articulate the nature of the analogy; its evaluation will come later.

In the next step, which should usually be performed by a group of analysts and users working together, the scope of each analogy is determined. The scope of an analogy represents that portion of the target system which it is congruent to. In earlier discussion, for example, we have established that the airline reservation system analogy's scope informs design of the student-course signup section of that system, but does not inform the student accounts receivable segment. In determining scope, it is important to identify both the problem space over which the analogy maps *and* that over which it does not.

After the analogy discussion and scope determination meeting, the analyst works out which analogies inform the same scope and which have different scopes. The former compete with each other, while the latter complement each other. The analyst, working from his own general knowledge and that of other members of the project team, can then lay out the implications, in terms of development and operating costs versus performance, of competing analogies. Clients can then use these implications to select a working set of complementary analogies.

Once the working set of complementary analogies has been selected, the analyst and interested clients should determine whether it *collectively* informs the target system "sufficiently." In doing so, they can ask, "Are there major gaps in the target system, for which we have identified no analogy?" If so, an appropriate course of action would be to spell these gaps out (e.g., "We need to find a model for the student accounts receivable process") and return to step 1 (above) in order to generate some candidate analogies. Where the gaps are minor, the derivation of analogies to fill them may be unnecessary -- a suitable design, appropriate within the larger framework provided by the analogy set.

Center for Digital Economy Research
Stern School of Business
Working Paper IS-88-51

Finally, the analyst reaches the final three steps, in which the he or she prepares more detailed mappings from the target systems to the form and elements of the base systems. This involves research into the structure and dynamic behavior of the base systems. This reseearch enables the following step, the development of an internally consistent set of constructive diagrams, which form the nucleus of the target system design. The final step, then, refines and elaborates this nucleus. It removes elements which represent the local detail of the base system, and adds those which represent the corresponding detail present in the target system. At this point, the analyst and clients have created a design for the target system.

In applying this approach, the analyst should try to avoid two main pitfalls of constructive systems thinking:

- Wherever possible, avoid the use of living organisms as analogies. In general, their use complicates the template process, because many features of organisms are either not well understood or difficult to recreate artificially.

- Keep in mind that the physical image is the *least important* part of the analogy; the relationships between components are far more important. Do not get sidetracked with drawing pretty pictures -- focus on the correspondence between the important relationships.

## 6. Summary

This paper has suggested that diagrams can play four distinct roles in the development of an information system. It can *document* its creator's currently existing knowledge, or it can act as a basis for *communicate* that knowledge to others. It can help the analyst *validate* his understanding of the systems or situation, or it can help him *construct* such knowledge. These various roles depict the outcome or effect of the diagramming process rather than its initial intention. The analyst may attempt to create a constructive diagram and fail, but later find that his results can be very effective as a basis for communicating my ideas to others.

Constructive systems thinking projects a structure from the base system to the target system, assists in the identification of alternative solution approaches, provides a

Center for Digital Economy Research
Stern School of Business
Working Paper IS-88-51

meaningful context for client involvement, and generates images that are, by definition, independent and incomplete.

The goal of constructive systems thinking is to enhance the creative solution of IS development problems, and it cannot be reduced to a specific, highly structured technique. This paper has, however, outlined a general approach to building constructive images and diagrams:

1. Create a set of candidate analogies, and elaborate and evaluate the degree to which each guides design of the target system.

2. Evaluate how completely this working set of analogies informs the important aspects of the target system, and create additional analogies to fill any major gaps.

3. Over the relevant scope of each analogy, research its structure and dynamic interactions. Use these as templates within which to model the target system.

4. Validate this design and its functional implications with the system client, adjusting it as required.

5. Complete the design by removing details specific to the analogous system and adding those relevant to the target system.

Center for Digital Economy Research
Stern School of Business
Working Paper IS-88-51

## References

[1] Adelson, Beth, and Elliot Soloway.
The Role of Domain Experience in Software Design.
*IEEE Transactions on Software Engineering* SE-11(11):1351-1360, November, 1985.

[2] Alexander, Christopher.
*Notes on the Synthesis of Form.*
Harvard University Press, Cambridge, MA, 1971.

[3] Chen, P.
The Entity-Relationship Model -- Towards a Unified View of Data.
*ACM Transactions of Database Systems* 1:9-36, March, 1976.

[4] DeMarco, Tom.
*Structured Analysis and System Specification.*
Yourdon Press, New York, 1978.

[5] Gentner, Dedre.
Structure Mapping: A Theoretical Framework for Analogy.
*Cognitive Science* 7:155-170, 1983.

[6] Guindon, Raymonde, Herb Krasner, and Bill Curtis.
*Breakdowns and Processes During the Early Activities of Software Design by Professionals.*
Technical Report STP-260-87, Microelectronics and Computer Technology Corporation, Austin, TX, 1987.

[7] Sasso, William C.
Architectures for Financial Consolidation: A Comparative Study.
*Journal of Accounting and EDP* 3(2):4-11, Winter, 1988.

[8] Yourdon, T., and L.L. Constantine.
*Structured Design.*
Yourdon Press, New York, 1978.

Center for Digital Economy Research
Stern School of Business
Working Paper IS-88-51

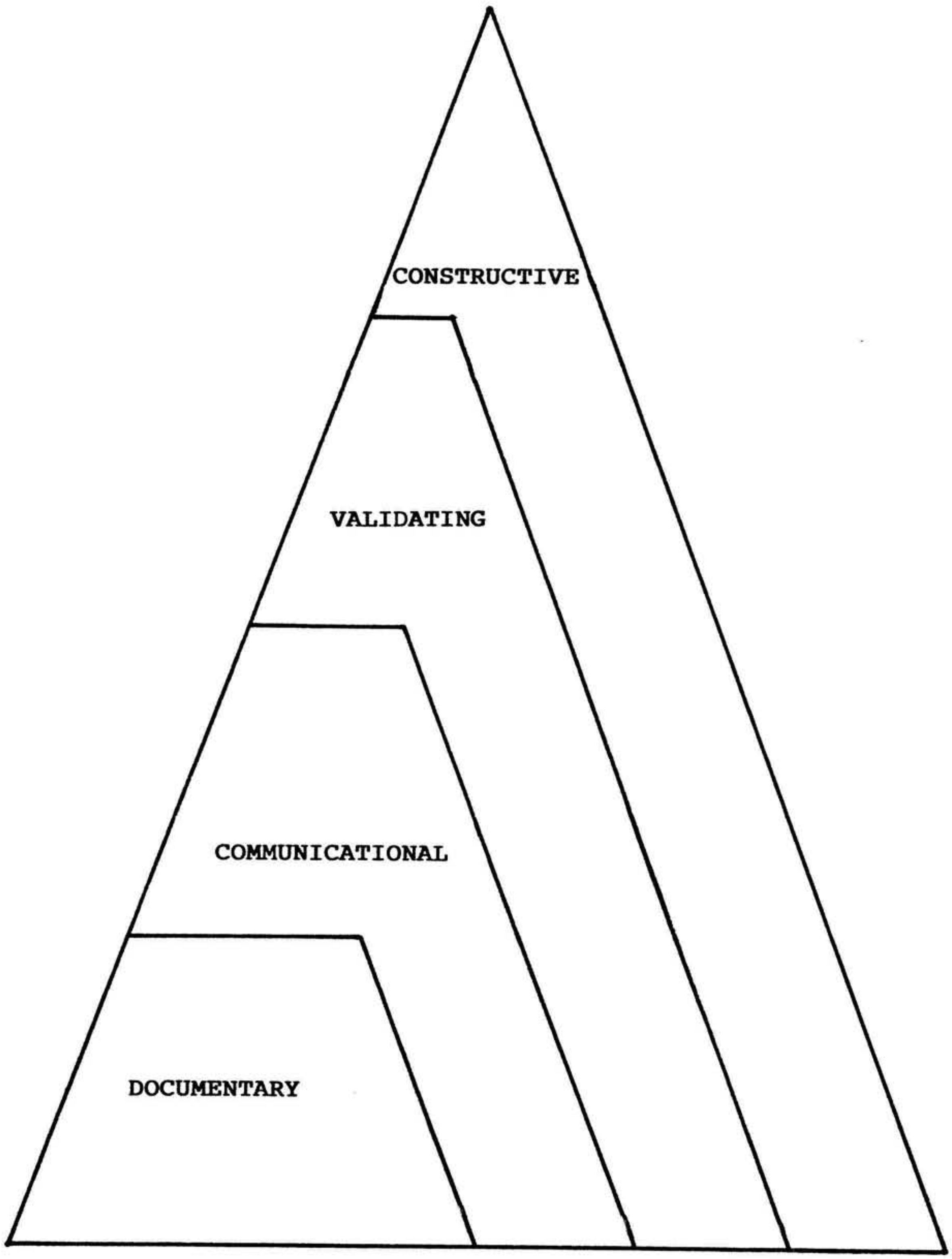


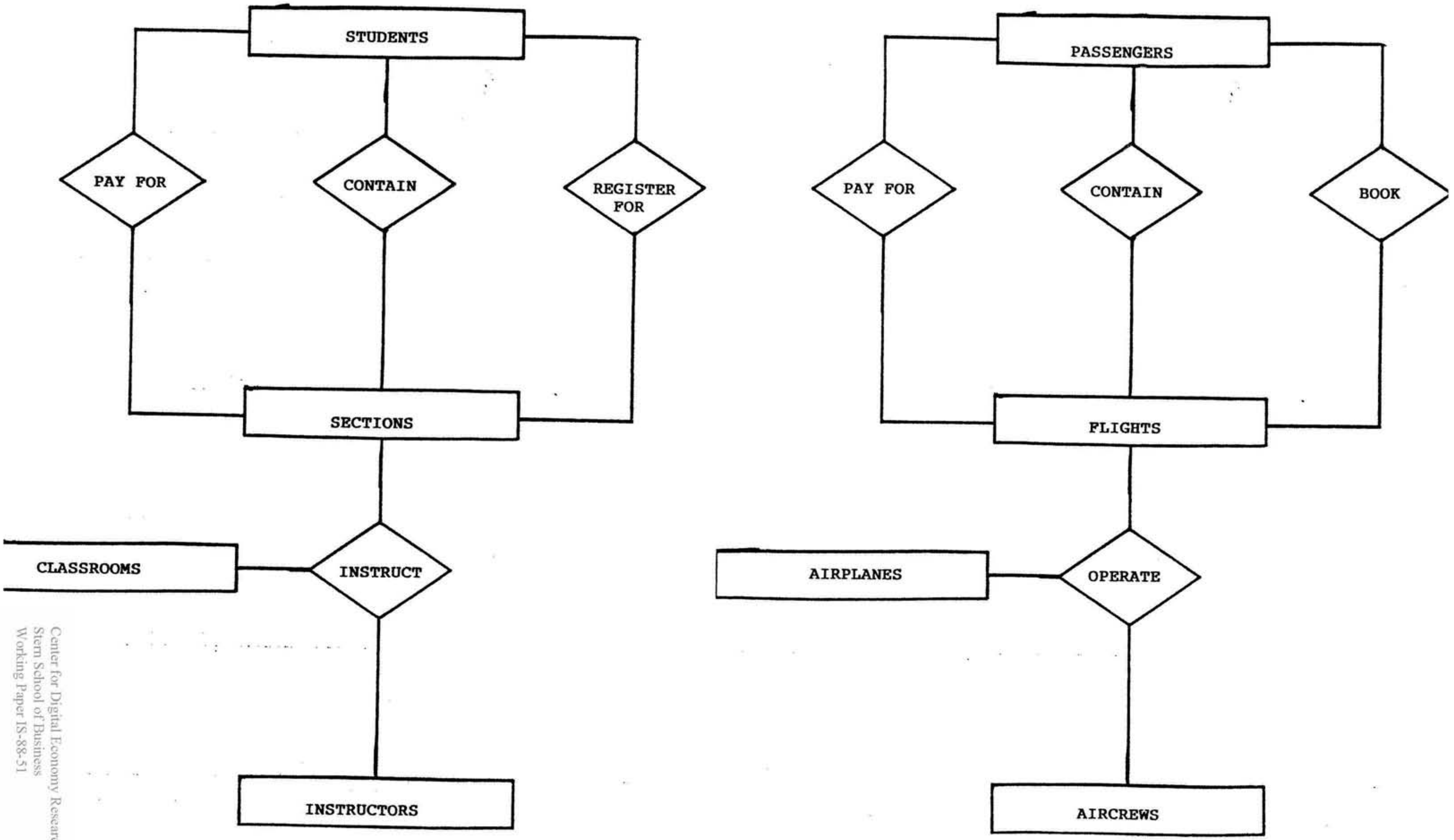
**FIGURE 1**

**THE FOUR ROLES OF DIAGRAMS**

Center for Digital Economy Research
Stern School of Business
Working Paper IS-88-51

FIGURE 2

CORRESPONDENCE BETWEEN BASE (AIRLINE) AND TARGET SYSTEMS



Center for Digital Economy Research
Stern School of Business
Working Paper IS-88-51

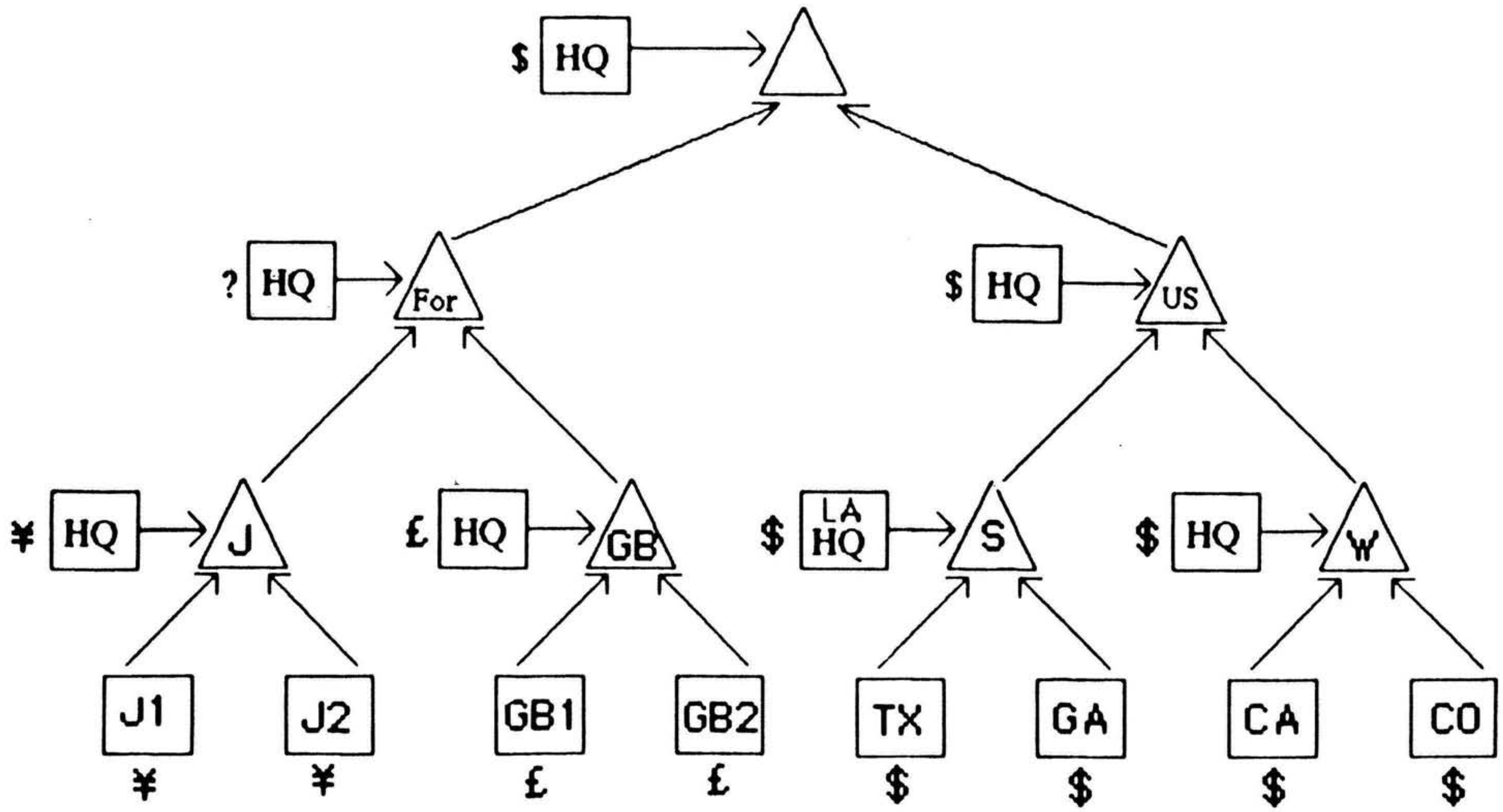


FIGURE 3

A CONSOLIDATION SYSTEM MODEL
BASED ON THE CONSOLIDATION
HIERARCHY

(THE STEPWISE MODEL)

Center for Digital Economy Research
Stern School of Business
Working Paper IS-88-51

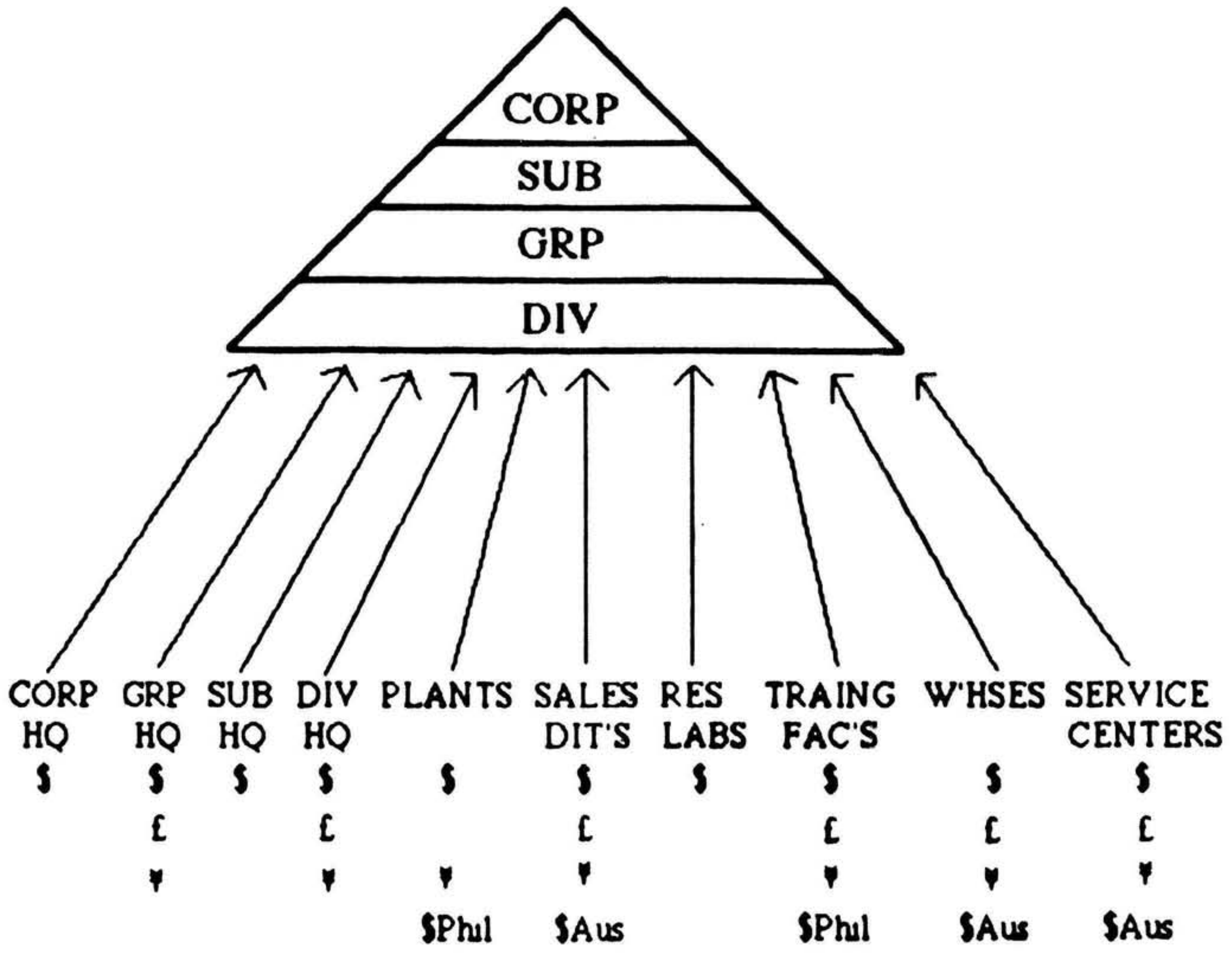


FIGURE 4

A CONSOLIDATION SYSTEM MODEL
BASED ON THE "BLACK BOX" MODEL
(THE DIRECT ARCHITECTURE)

Center for Digital Economy Research
Stern School of Business
Working Paper IS-88-51