NATIONAL BUREAU OF STANDARDS REPORT

NBS PROJECT	September 1, 1965	NBS REPORT
4002415		8987

A Study of the Feasibility of Development of Computer Based Systems for Semi-Automated Comparisons of Building Designs and Regulatory Code Criteria

(A Computer-based Code checking system)

Prepared for
United States Public Health Service
Department of Health, Education & Welfare

NBS

U. S. DEPARTMENT OF COMMERCE
NATIONAL BUREAU OF STANDARDS

A STUDY OF THE FEASIBILITY OF DEVELOPMENT OF COMPUTER BASED SYSTEMS FOR SEMI-AUTOMATED COMPARISONS OF BUILDING DESIGNS AND REGULATORY CODE CRITERIA

(A Computer-based code checking system)

Prepared for:

Mr. August F. Hoenack, Chief
Architectural and Engineering Branch
Division of Hospital and Medical Facilities
United States Public Health Service
Department of Health, Education & Welfare

Prepared by:

The Institute for Applied Technology
National Bureau of Standards
United States Department of Commerce

September 1, 1965

Credits

Project Supervision & Reporting:

Gary K. Stonebraker
Administrative Officer
Technical Analysis Division
Institute for Applied Technology
National Bureau of Standards

Project Staff & Consultants

Douglas Y. Cornog (staff consultant)
Engineering Psychologist
Information Technology Division
National Bureau of Standards

Thomas N. Trump (summer staff)
Mathematician
Technical Analysis Division
National Bureau of Standards

Welden E. Clark, Senior Consultant
Bolt, Beranek & Newman, Inc.
8221 Meltrose Avenue
Los Angeles, California

Earl Isaac
Fair, Isaac & Co., Inc.
1200 Lincoln Avenue
San Rafael, California

Pat Coyle
Computer Applications Consultant
1005 24th Street, N.W.
Washington, D. C.

INDEX

Note to the Reader:

The content of this report is summarized in Section 1, Summary.

Section 4, STATEMENT OF FEASIBILITY, discusses the broad aspects of the system problems in 19 questions and answers concerning feasibility.

Each of these questions and answers is explored in detail in the Appendix.

SUMMARY

This report will state that it appears to be technologically and economically feasible to develop a computer-based system for the comparison of building designs with criteria set forth by regulatory codes.

Such a system can be developed and applied specifically to the problem of comparing hospital designs with requirements set forth in Part 53, Public Health Service Regulations, which criteria apply to the construction of hospitals built with federal support funds provided under the Hill-Burton Act.

A system can be designed to accept a description of a building design and to automatically compare that data to any precisely defined criteria. A full review of the comparison can then be given back to the professional architect employed by the code system, who would examine any exceptions found, and only those exceptions. Thus the professional will be relieved of much of the duty of routine checking of documents. The extent of this relief can be estimated at 50% of the amount of such work that he now does. The remaining 50% of the work involves operations calling for general professional judgments and interpretation, and this remains the responsibility of the code professional.

The system can also act to improve the services of the system by allowing the addition of capability now not possible.

In addition, it will provide detailed files and descriptions of all buildings processed, which should prove useful in program analysis.

The system can be developed through (a) the preparation of a specification for the system and a Request for Proposal directed to industry (b) the acceptance, review and evaluation of bids, and subsequent awarding of contracts (c) the development and installation of the system (d) prototype operational testing and training, and finally (e) acceptance of the system for operational use. The time required could range from 1½ to 3 years, depending upon system complexity.

The exact cost of the system will not be known until bids are received, but estimates of potential systems costs shows them to be commensurate with the cost-value of benefits derived.

INTRODUCTION

The purpose of this feasibility study has been to explore the use of man-machine systems (computer-based) as tools for comparing buildings design with regulatory code criteria, specifically for the Hill-Burton program administered by the United States Public Health Service. The project has been undertaken under the joint sponsorship of this organization and the National Bureau of Standards.

Presently, the Hill-Burton program utilizes a professional staff of about 39 architects and engineers located in 7 regional offices to review proposed designs for hospitals to be constructed using Hill-Burton funds. The purpose of this activity is to assure a minimum standard of planning and construction quality, and to provide continual upgrading of hospital and medical facilities planning.

The criteria for design are set forth in Public Health Service Regulations. Part 53. This document is a combination of minimum property standards, minimum design and facilities standards, and some broad specifications of materials and constructions techniques. Also included are some regulations concerning procedures to be followed in contracting and other aspects of project administration.

Under the load of processing 500 project applications and some 1400 design submissions per year, the professional staff has found it increasingly difficult to spend time in the field in the critical capacity of consulting. The work load is expected to continue growing, and as it is increasingly difficult to obtain qualified personnel and train them, the program in the future may face some losses of effectiveness insofar as its ability to exert a positive influence on the design of medical facilities.

The problem is further complicated by the fact that each project design must be reviewed up to three times during its development. Since in fact the PHS regulations do not have the legal authority of comparable state regulations, there are inevitably compromises which must be worked out between client, local, state and Public Health Service interests, which activities involve much field liaison. The three-phase submission program is designed to resolve the major questions which may arise at the earliest possible times, as well as to provide the opportunity for positive quidance in program development for the facility.

In order to relieve the professional staff for fuller participation in these key decision areas, and in order to assure that program growth may continue without diminished effectiveness, some solution must be sought to relieve the professionals from the routine chores which are normally associated with code inspection activities. The feasibility study was undertaken in order to examine the possible use of computer systems to perform these inspection chores, thus accomplishing this objective. The study has been performed by the staff of the Institute for Applied Technology of the National Bureau of Standards, with consultation by Bolt, Beranek and Newman, Inc., Los Angeles, Calif.; Fair Issaac and Co., San Rafael, California, R. J. Coyle, Computer Applications Consultant, Washington, D.C., and members of the Technical Analysis Division and Information Technology Division of the National Bureau of Standards. The consulting firms involved were asked to examine the problems associated with the development of such systems. Since there is in fact a very large spectrum of possibilities, the consultants were asked to examine the ends of this spectrum, hoping thereby that the feasibility of the system would be bracketed. Bolt, Beranek and Newman explored the use of "interaction system", i.e., those kinds of computer systems which communicate with and enhance the judgment and action of man. More specifically, this involves the use of on-line processing, exotic peripheral input-output equipment, and generally, the extremes in sophistication of equipment and programming techniques. On the other hand, Fair, Isaac & Associates concentrated on the examination of minimal hardware usage, with straight-forward use of tested computer techniques; more specifically, the use of standard key-punch input, batch processing, and conventional output. Mr. R. J. Coyle assisted in assessing potential systems costs and in evaluating the feasibility and need for various forms and concepts of programming.

SOME COMMENTS ABOUT CODE SYSTEMS IN GENERAL

In almost every aspect of the design of buildings, there is a need to have criteria against which designs are compared in order to determine the effectiveness and suitability of the design. Sets of criteria affect the designs of buildings in many ways. There are building codes, property codes, zoning laws, standards of production, design, and practice, professional recommendations, public laws, and many others. The successful resolution of design depends upon the satisfaction of <u>all</u> of the applicable sets of criteria.

Every architect and engineer knows the penalty of not satisfying the criteria of codes. Thousands of man-hours are spent each year, throughout the building design professions, revising endless series of drawings or physically changing construction because one aspect or another of the design did not meet the criteria set by codes.

Failure to satisfy code criteria can be partially attributed to the code itself in that the statements in the code can be ambiguous, meaningless or obsolete. Perhaps equally important is the fact that there are frequently too many criteria for the designer to consider, and he will omit consideration of some inadvertently. Still some other cases of failure occur because the responsible parties, the designer or the reviewer, did not know how to make a decision about the acceptability of a solution, or did not care to try. It is useless to speculate about the motives or intentions of any person or element involved in this activity, as in fact the difficulty are symptoms of problems of systemic origin. Simply stated, the objectives of code systems are often far in advance of the capability of the operational systems to fulfill those objectives.

If we say that codes are obsolete, and then we attempt to revise them, this does little good if the code will shortly fall into obsolescence again. If we resolve code difficulties on one project after failure to design properly, this is not a significant gain if we are going to encounter the very same difficulties again on the next project. And if we make a decision to accept an innovation on a project, it does little good unless continuity of that decision is assured.

The problem of the generation and use of building code criteria must be attacked as a systems problem, and further, it must be viewed for what it really is: a subsystem of the design process which assures consideration of broader social objectives in the design of individual buildings. Viewed as any other function, especially viewed as a police action, the objectives of code systems become badly distorted; for instance, we find that many code-checking activities concentrate more on control during the construction process (which results in costly changes) rather than support of the design activity (which would result in the specification of satisfactory solutions and their subsequent embodiment as conditions of contract).

Speaking in broad terms, a code system involves the following elements:

1. A method of writing (and updating) the code;
2. A method of communicating the code requirements to those who must implement them in buildings, i.e., the designer and builder.
3. A method of testing design and construction against the code criteria to assure compliance.
4. A method of assessing the needs of the users both in particular and general, feeding information back into the system for improvements of codes or for granting exceptions to the general case of the code.

We may refer to these sub-functions as the functions of:

CRITERIA GENERATION

COMMUNICATIONS

TESTING

EVALUATION & IMPROVEMENTS

We also must bear in mind that the code system is a two-way system; while it has as its major objective the promotion of increased quality in buildings, it must also have as its objective continuous improvement of itself to match the needs and opportunities of changing times.

As has been noted, the testing activity runs parallel to the activities of design and construction. The <u>timing</u> of the interaction of testing with these parallel systems is extremely important to the maximum effectiveness of either of the activities. Poor timing of interaction can completely deter reaching the objectives of either.

When the architect writes a program of criteria expressing the client's needs, he is continuously testing his designs against those criteria as he proceeds through the design process; at important decision points in the development, he calls the client to make his own (the client's) tests of criteria against design. This is particularly useful in the events of criteria that are difficult to express, such as those normally involving aesthetic judgment.

Consider, however, the use of codes. While the designer may have a written statement of criteria, (the code itself) and while he may consider them as he forms design decisions, he does not submit his design to any <u>authoritative test system</u> (and in many cases cannot do this) until the design decisions have been substantially completed. Thus properly timed evaluations in the many important design decisions are lacking when they are most needed.

Thus the two most critical systems characteristics in the code system are the proper structuring of the internal communications for self improvement, and the proper structuring and timing of external communications which implement the system and bring back new information to it.

With respect to the design system using the code system, the most important characteristics of the code system are (a) its availability for interaction at the time of decision-making; (b) its consistency and accuracy in rendering judgments; (c) its ability to accept the widest possible variety of solutions which truly satisfy the criteria; (d) its ability to update criteria to new objectives, changing technology and changing performance needs.

Finally, a word should be said about the form of code statements. We must be able to relate a code statement to a proposed solution. We are far more able to make sophisticated criteria statements than we are to use them or test solutions against them. The critical need for improvement of codes is the development of logical structure and methodology which firmly relates criteria characteristics and solution characteristics. Examples of this kind of logical structure are engineering equations which allow precise comparisons of a particular structure and what we think is a "safe" structure. As these methodologies become more precise, they also have tendencies to become more sophisticated and expensive, and thus less "practical" to use. New methodologies such as those allowed by computers, which increase the ease of use of more sophisticated criteria and test systems, are a necessary component of an improved code system. The longer-term objective must be to create a system of codes which satisfies all of these needs and has these requisite systems characteristics. The shorter-term objective, that of creating a test vehicle, can and should be so oriented as to contribute a sound base for this long-term objective.

STATEMENTS OF FEASIBILITY

This section will attempt to answer 19 basic questions which affect the feasibility of developing computer systems for comparing hospital designs and code criteria. While this section of the report is somewhat lengthy, the determination of the feasibility of a complex system is not a simple task. The results cannot be presented simply if full understanding is to be achieved. However, the results of the study are summarized as answers to the "questions" in order to make this section as short as possible.

For those persons who wish to explore any particular point in detail, Appendix I, Support of Statements on Feasibility, is provided. Sections of this Appendix are referenced to the question numbers. The Appendix is designed to provide detailed information for those who are interested, but it is hoped that the basic feasibility of such systems is accurately and fully represented in the following questions and answers:

1. <u>What is the purpose of this report?</u>

(a) This report is intended to discuss the applications of computer systems to the problem of comparing building designs to criteria set forth in regulatory codes.

2. <u>For whom is the report intended?</u>

(a) Specifically, this report is directed to the Architectural and Engineering Branch, Division of Hospital and Medical Facilities, U. S. Public Health Service, who are responsible for enforcing minimum design standards in hospitals and medical facilities constructed under grants-in-aid provided by the Hill-Burton Act of Congress. This agency has contributed in part to support of this study.

(b) Generally, the report is directed to any agency, local, state, or federal, responsible for the conduct of regulatory code systems, and to the building design professions in general.

3. <u>Why should the use of computers be important to any of these persons?</u>

(a) As we upgrade the quality of criteria toward performance standards, increasing demands will be made on the code system and its professionals for sophisticated consulting and interpretation services. The professionals must be freed of "routine" duties in order to provide this important service. Further, interpretation of more sophisticated performance criteria will demand the development of more exacting analytic methods to test building designs. The use of the computer will allow the use of more sophisticated analytic methods which would not otherwise be practical.

*4. <u>What kind of tasks are involved in comparing designs and criteria?</u>

Very basically, the following operations must be performed:

(a) The criterion must be examined

(b) Data needed about the design must be listed

(c) A search is performed

(d) If found, the data is acquired

(e) Design data and criteria are compared

(f) A conclusion is reached and a report issued.

This basic process is diagrammed in Figure A.

*5. <u>How can the computer be applied to this process?</u>

Combinations of the computer and peripheral input-output equipment can do, or aid in doing, the following things:

(a) It can store lists which specify what data is needed to answer questions about criteria,

(b) It can help guide the operator in collecting data,

(c) It can automatically measure and/or compute certain data from graphically presented solutions,

(d) It can store complete and detailed descriptions of buildings,

(e) For all precisely defined criteria, it can retrieve all appropriate data, compare it to the criteria, and issue a report listing all exceptions to criteria.

6. <u>What are the basic requirements for accomplishing this?</u>

(a) First, all the criteria to be considered by the system must be logically precise statements, free of relative terms, such as "adequate," "near," or "safe."

(b) Second, a way of stating the criteria to the computer in "understandable" form must be developed.

(c) Third, a way of describing the content and geometry of the building to the computer must be developed.

(d) Fourth, a method of processing must be developed;

(e) Finally, a mode of reporting the exceptions must be developed.

7. Are the criteria of Part 53, Public Health Service Regulations, adequate in form to meet these requirements?

(a) About 80% of the number of statements in the Hill-Burton code are statements such as,

> "General storage (required is) 20 square feet per bed, concentrated in one area...."

which are logically precise statements. Such statements can be introduced as equations which the computer can solve, e.g.

Area of storage = number of beds x 20
(true or false?)

(b) About 20% of the statements are logically ambiguous, such as:

"Scrub sinks strategically located in corridor (of contagious disease nursing unit) (are required)"

and cannot be considered without added definition (in this case, what constitutes strategic location?)

8. Can a description of a building be stored in the computer?

(a) Descriptions of spaces can be stored as a set of X-Y coordinates describing the extreme points (e.g., corners) of a space. From these sets of coordinates, distance, area, volume and location can be computed. Names and descriptions of materials and equipment can also be entered and stored, in the same manner as conventional computerized information. The locations of elements or equipment can be described using the coordinates system used to describe the space boundaries.

(b) All of this data can be entered into organized "lists" which can be systematically searched, using any of several existing list-search and list-processing techniques.

(c) Techniques for description of mechanical, structural, & electrical networks may require substantial research.

Components of such networks may be described as equipment or materials and examined, but the behavior of the <u>total system</u> will be difficult and perhaps not practical to consider <u>in detail</u>.

(d) Similarly, the introduction of entire sets of specifications into the computer would require storage of impractical size; and by the time specific information is located and extracted from specifications, it may be more practical to complete consideration of such data external to the machine system.

9. <u>Can a practical system of processing be developed?</u>

(a) At least two have been identified. One alternative is to design a "query language" which asks questions about the data file: it asks for example, "list any bedrooms that have four beds". The data file lists the location of all beds. These are compared, and the number in any location given. As an alternative, a list of all beds is printed out in order of location. This list is scanned visually for exceptions.

(b) An alternative method is that each criteria is stated as an equation or as a symbolic equation to be solved by the machine and tested for its mathematical truth or falsity. For instance, the criteria requiring 20 square feet of general storage per patient bed can be restated as an equation:

(area, general storage) $\geq$ (no. of beds) x 20

This equation is solved by searching the data file for the value associated with each data name. That value is substituted for the name in the equation:

(2000 s.f.) $\geq$ (110 beds) x (21 s.f./bed)

This is solved by the computer

2000 $\geq$ 2200

and tested by comparing both sides of the equation for equality. The conclusion in this case would be that the statement is false.

10. How are the results of this comparison communicated to the operator?

(a) When a false statement is found, the computer can refer to and print out a standard English sentence stored in its memory, such as,

> "inadequate general storage provided; 2000 s.f. vs. required 2200 based on 110 beds x 20 s.f./bed."

The underlined items pertain to the case at hand, being instance values inserted in the standard sentence. As an option, the computer could drive a plotter, reproducing the hospital plan and positioning its error reports at the point in plan where the exception occurred.

11. What major operational considerations must be made in the use of such a system?

(a) First, consideration must be given to the skills required to operate it. Present GS-13 (average) skills can probably be replaced with GS-7 technical skills more readily available. Higher professional skills would still be utilized to examine exceptions reported by the system and to examine criteria which cannot be considered by the system.

(b) Second, special consideration must be given to the operation of transferring data from plans and specifications into the computer. The basic alternatives are two. First, data needed can be transferred to forms, then punched onto standard punch card, paper tape or mark-sense cards. This alternative requires that all data be acquired in a "batch," introduced into the computer, and processed. This is known as batch processing. The major disadvantage is that errors in data may not be discovered, and if discovered, necessitate partial or complete resubmissions. The kind of data handled in this case suggests high probability of this kind of error, and hence a high resubmission rate.

(b) The second of the alternatives involve special on-line equipment which transmits data directly into the machine without key punch or other intermediate translation. Graphic data can be directly traced the use of graphic input devices. Using such devices, the operator can trace the outline of plans and equipment. The computer will automatically assign coordinate values to the critical points of the plan, thus automatically translating and graphic data into computer readable format. Names and other attributes of entities can be entered on on-line typewriters as the input proceeds. Using combinations of these techniques, all data can be entered directly into the machine without the useof punch cards or paper tape. Such an on line system can process data as it receives it, ask for more data and guide the operator in the acquisition of data. This communication, or conversation, between man and machine takes place through the typewriter hooked to the computer or through a Cathode ray tube screen (like a TV screen) which can display words and pictures.

The use of the computer in conversational mode with the operator provides important capability. The computer can organize and guide work (which can reduce the skills required to operate the system) and can redisplay (on the oscilloscope) graphic and text input for an accuracy check by the operator. This minimizes the chances of erroneous or incomplete input. Figure B gives an illustration of these pieces on input-output equipment in a hypothetical configuration. The computer itself is not illustrated.

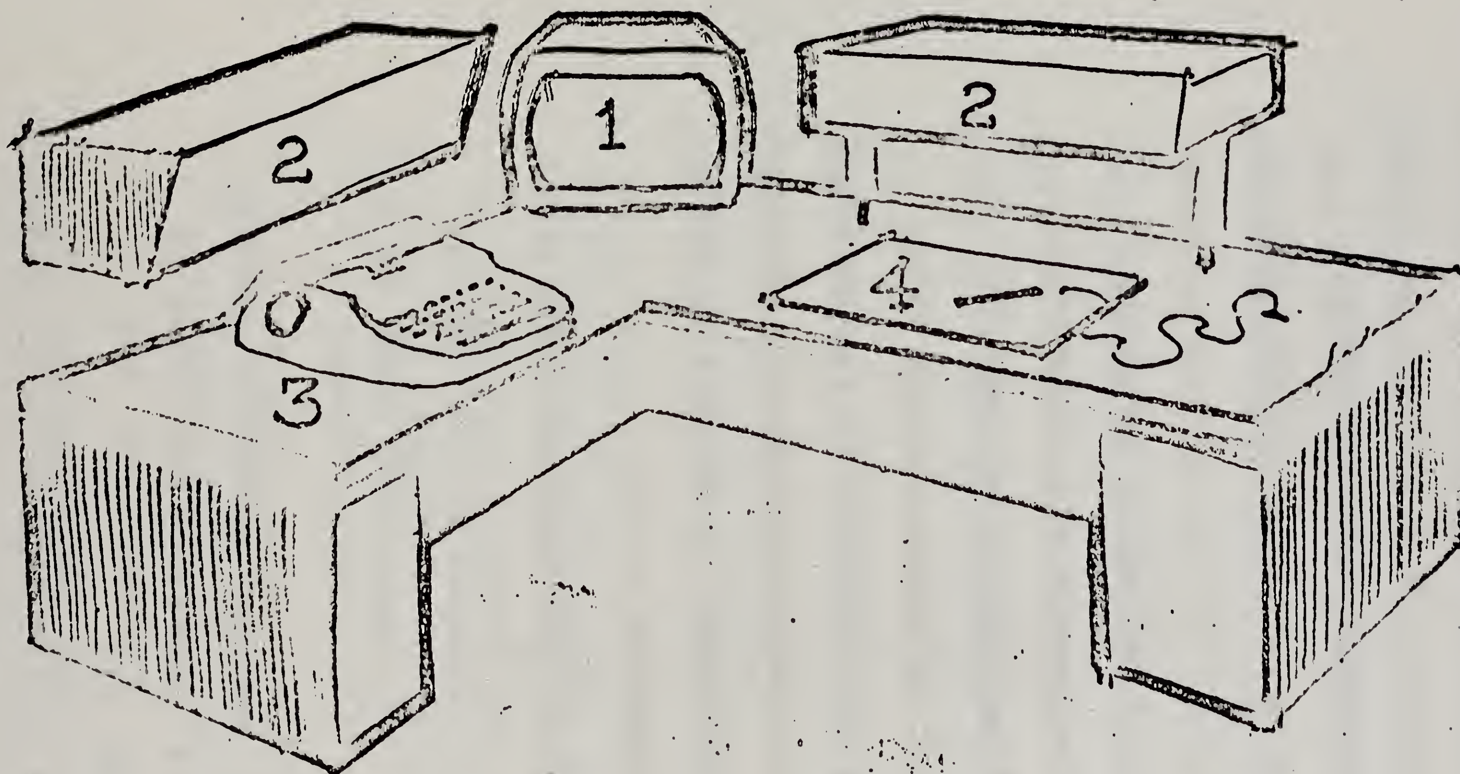
12. <u>Are there more alternatives to be considered?</u>

(a) In fact, many more; however, the best judgment is that these alternatives of (1) batch processing with minimal "hardware", i.e., computer equipment, and (2) on-line processing with maximum use of exotic hardware, represent the extreme of application. Most other technologically feasible systems represent some combination or modification of these alternatives.

(b) One major alternative to be considered is the use of remote "time-sharing" input-output stations, hooked by long-distance communications lines to a central computer. This would allow use of the system by many separate regional offices. At present, long distance data transmission may be prohibitive in cost.

13. <u>How can the best system be selected from these alternatives?</u>

(a) The evaluation should be made on the basis of the costs of each alternative compared to the benefits derived. The major <u>costs</u> of the system would be:



1. Oscilloscope Display Screen
2. Control Consoles
3. Electronic Typewriter
4. Graphic Input Device

Figure B: Hypothetical Input-Output station for On-line Processing System

(1) Costs of purchased equipment (or) Costs of rented equipment and processing time;

(2) Costs of programming the equipment;

(3) Cost of administering the development of the system;

(4) Annual costs of operating personnel;

(5) Overhead on equipment and personnel.

Costs 1,2, and 3 should be discounted over the life of the system, which is assumed to be five years. Cost 4 includes the services of a programmer operating continously to up-date and modify programs.

(c) The benefits derived are harder to measure in units of cost. They should include:

(1) Value of time of professional architects and engineers relieved by the system,

(2) The value of any capability added by the system,

(3) The costs of expected expansion without the system as opposed to the costs of expansion with it.

14. <u>What is the expected value of the benefits of a computer system?</u>

(a) Detailed examination of the form of criteria, the sources of data in plans, and the nature of the comparisons formed have been undertaken and contrasted to reports of time spent in various activities by the professionals in the program. The staff is currently processing 500 projects with some 1200-1400 estimated design submissions per year. The current estimated annual cost of reviewing drawings and contract documents is estimated at $350,000. This study indicates that approximately 50% of these man-hours could be replaced by the machine system; therefore the major benefit would be the relief of skilled manpower valued at $175,000 per year.

(b) Some of the immediate added capability of the system would include:

(1) The provision of detailed data files on each design submitted at each phase of submission, in form suitable for statistical analysis by computer (analytic programs would be a separate cost). This would allow analysis of space usage, regional and building type costs, etc., on a detailed current basis, and other options.

(2) Assurance that all criteria in the computer system were applied to every project, and applied uniformily and objectively

(3) Ability to expand program capability more readily through decreased dependence upon scarce skills.

(4) The ability to use existing professional skills in more critical (and more challenging and attractive) roles.

No attempt has been made to assign a dollar value to this added capability. The discussion of potential future capability is postponed until a later section.

15. What would be the expected costs of a computer system?

(a) This is highly dependent upon the particular configuration and operation selected. For the two extremes examined, the cost of the batch-processing system was estimated to be about $410,000 per year, all costs considered; likewise, full on-line processing systems estimated out at about $300,000 per year. A third alternative examined involved the use of a "small" computer to operate input-output equipment with all the "on-line" features, but which relegated the main processing to a rented time on a larger computer. This kind of system estimated out at about $175,000 per year, which appears to be a feasible cost compared to the value of professional skills relieved. Thus it appears that some configurations are feasible on a benefit-cost basis, without considering the value of added capability. These cost estimates are the best available without full systems design, but are still approximations.

16. <u>Would the installation of a computer system require changes in the internal operating procedures?</u>

(a) Quite likely. You could not expect to hire 300 new personnel with specialized skills and put them to work without changes in procedures. Similarly, the capability of a computer system will demand some changes and <u>allow</u> others.

(b) It is very likely that processing would be done at one location if maximum cost-benefit is to be derived. This would be a major change over the current practice of regional processing of applications and submissions. High-speed communications systems (not necessarily computer-operated) can still link the processing facility and the regional office, and the net effect would be the same as if the regional operative sent a talented clerk into the next room with the plans.

(c) Freeing the professionals for more consulting and liaison work is also likely to induce change in the form of more information and ideas gathered from the field.

There must be a major drawback

(a) While on the last subject - that of change - one "drawback" did emerge. The computer is a robot. It has no self-contained intelligence or judgment. It does only - and precisely - what it is told to do. Therefore, if the criteria entered into it says that a single-bed patient room should have 100 sq. ft. area minimum, it will reject a room with 99.99996 sq. ft., whereas the human would have accepted that. Thus every exception reported - but only the exceptions - needs review by someone capable of judgment, i.e., a professional.

Of course, an alternative is to build tolerance into the machine. We might in fact accept a tolerance of ±5 sq.ft., or 95 sq.ft., and accordingly we program the machine. Is the problem solved? Not necessarily, because now the question is simply moved to whether or not we accept the room of 94.999998 sq. ft. as being within the tolerance limits. This could go on forever. The system would also pass a patient bed room 50' long a 2' wide. The computer is too precise.

Or is it? Did we mean what we said when we said 100 sq. ft.? If so, why were we willing to accept 99 or 97? or 95? or 94.99998? We did so because we recognized that the plan configuration fulfilled the functional need of getting stretchers in and out of the room, providing access on both sides of the patient and providing space for medical equipment that might be needed in the room...and these were the real criteria we had in mind.

(b) In its own seeming inability to do what we ask it to do, the computer will point out these fallacies in our own thought processes. It will force discipline and rigor in thought such that criteria will have to say what they mean, instead of providing the approximate values of the past solutions. Eventually, the statements of criteria must be based on concepts of how we wish the building to perform. These will be complex and difficult statements, such as:

> "the patient bedroom must provide adequate manuevering space for the entry and exit of equipment for transporting non-ambulatory patients."

(c) Having contributed to the forcing of this rigor, the computer also works to allow it, in that it allows us to develop techniques to "simulate" the actual movement of stretchers in and out of rooms, and even complex things, such as the traffic loads on facilities, the evacuation of buildings in emergencies, and the effectiveness of communications systems. Without a computer, these things are not possible.

(d) It is important that code systems have such capabilities, to allow the imaginative designer to seek new solutions which can place patients comfortably in 70 square feet, but which also constrain the irresponsible designer from solutions which do not work.

17. <u>Is this part of the future capability mentioned earlier?</u>

(a) Yes. As experience is acquired through use of the system, there is potential that many capabilities can be added.

(b) Some of these capabilities will be in support of more sophisticated criteria statements such as those mentioned above.

(c) But some of the capabilities which could be added would support the Hill-Burton program in one of its other purposes, which is to act as consultants on hospital design in order to maximize the quality of these facilities nationwide. It is technically and theoretically feasible (although the costs are not estimable) to add programs which would allow simulations of traffic flow, which could test facility capacity, the positioning of stairs and elevators, the location of exits, and so forth. Demographic studies to help select proper hospital sites could be performed. Other services to the architect, such as full dimensional checking, checking for conflicts of subsystems, and the like, can be visualized.

18. Would these changes, like the initial installation, cause changes in future procedures?

(a) They certainly permit speculation about the future organization and operation of code systems. First, we can visualize extension of the system to the point where a limited number of input-output stations could be made available to architects to assist them with the actual design problems. As, and if, this proves feasible, we can begin to think of the architect designing hospital on such input-output stations in his offices, connected by remote lines to the code-system computer-processor. As he designed on-line with the code system, he would receive immediate interpretation of his decisions as being acceptable to or not acceptable to the code criteria. Thus many costly errors and changes in the design phase could be avoided. Such a system, of course, avoids translating the data from plans and specifications into the computer, and in fact the computer can be used to produce such plans and specifications. Presumably the architect would observe the comments of the machine system, and the need of checking finalized designs could be reduced to a minimum, if not altogether eliminated.

(b) Further, if code criteria can be introduced in such a manner, many other client criteria, program criteria, professional practice criteria and the like can be introduced into the machine to interact with the designer. We begin to be well on our way to the development of more sophisticated, accurate and consistent designs.

(c) All of these things are possible; it is essentially a question of whether they are feasible on a cost-benefit basis. As computer technology and the demands for better design continue to grow, it is almost surely only a question of time.

19. <u>How could such a system be developed for PHS use?</u>

(a) The next step would be to create a performance specification for such a system which would request proposals from industry for the design, programming, development, and installation of the system as a package. There are several industry firms which are interested in and capable of developing such proposals.

(b) The performance specification itself would require firm identification of the needs of the program, examination of alternatives, and the creation of a "Request for Proposal" document to be used as a basis for bidding by industry. This document can be prepared by the National Bureau of Standards. It is estimated that this work, combined with receiving and evaluating subsequent proposals, would require from six to nine months.

(c) Pending the receipt of a satisfactory proposal, a contract for development of the system would be awarded to the successful bidder(s). The system would probably have limited prototype operational capability within a year after that time; it could be fully operational and assuming a full work load in about two years.

(d) During the development phase, many other tasks, such as changeover plans, training, hiring, and reorganization must be undertaken as parallel activities within the Hill-Burton program.

(e) This work should be administered by a Joint Project Development Team composed of Hill-Burton Program personnel and staff from appropriate divisions of the National Bureau of Standards.

(f) A separate detailed proposal for this work is being developed by NBS and will be submitted separately.

APPENDIX I

DETAILED ANALYSIS IN SUPPORT OF FEASIBILITY STATEMENTS

SUPPORT OF STATEMENT 4: DEFINITION OF THE PROCESS

It is generally accepted that the process of design can be represented by descriptions of three distinct activities; analysis, synthesis and evaluation. In analysis, the problem is identified, the relevant variables are determined (as well as their interrelationships) and the criteria for the judgment of a solution are fixed or defined. The act of synthesis is the act of forming physical entities which perform the functions in question, assembling these entities, and specifying the solution with words or pictures.

In evaluation, we somehow attempt to simulate the resulting performance of the solution we have listed; this simulation yields answers in the form of qualitative or quantitative values, which are then compared to values expressed in the design criteria. Errors in the design are determined through observation of disparities between the simulated solution values and the design criteria values. When disparities are observed and judged to be unacceptable, the process cycles back to either synthesis or analysis and the process is repeated until a satisfactory solution is reached.

This of course is a highly simplified and idealized description, but it is adequate to handle the general discussion of the case. It also represents accurately major characteristics of the process, namely, that the design process can be described as an iterative (recycling) feed-back system. "Feedback" in this case represents information generated as a result of some previous activities which is used to guide or control subsequent repetitions of that activity.

A simple illustration of feedback in design may be gained by examining the selection of a steel column. Given the load, we may select a column thought to be about the right size (synthesis), submit it to engineering equation checks (evaluation), and find it "too big". We then select a smaller column, and going through the same process, find it "too small". On the next try, the information generated in the past two tries will tell us to select a column between the two sizes tried; in this case, the range of options is becoming successively narrowed as a result of the information generated on previous tries. The previous results are "feeding back" information to the new activity.

Building code inspections are in fact evaluations of decisions made by the Architect or Engineer. They will provide feedback information to the architect or engineer whenever a solution is found to be unsatisfactory, and will require him to re-enter the stages of analysis or synthesis as surely as evaluations performed <u>during</u> the design process would have. The principal differences between the two kinds of evaluation are in timing and purpose.

In the sense of <u>purpose</u> in building code evaluations, it may be construed as two possibilities: (a) to assure the consideration of broader social objectives, such as overall quality and land use in the community, or (b) to assure that the best experience is used in reaching responsible design decisions. To the immediate purpose at hand, it makes no difference which of these motives is used or whether either are correct motives.

On the other hand, the problem of <u>timing</u> does make as substantial a difference as one could imagine. Given client criteria, or personal criteria, or professional criteria, the designer usually has <u>authority to make decisions about the results of evaluation</u>. He can decide whether or not a solution is acceptable, or can call a meeting at any point in the design process to determine the acceptability of any solution in question. Thus decisions of evaluation are closely timed with activities of analysis and synthesis.

With a code system, however, the evaluations are made essentially after the total design is completed. The design, of course, is a series of dependent decisions, in which mistakes in early decisions affect all subsequent and dependent decisions. An incorrect assumption about floor loads, for instance, affects all subsequent engineering calculations, and hence perhaps the bay spacing and the total functional solution of plan. When the designer is forced to proceed through a complex series of decisions without any authoritative evaluation and then submit the results for this evaluation, it can precipitate disastrous (for the designer) results necessitating the re-examination and correction of hundreds of decisions.

This is a serious functional defect in the design evaluation system, which deserves a great deal of attention and analysis concerning its actual cost in efficiency and man-hours lost. It is a defect to which a least part of this study can be directed. The timing of the code evaluation decision has more immediate ramifications to the design of this system, however, which are quite important as constraints to the system.

The fact that the code evaluation follows the completion of an entire design, or at least a major portion of it (some codes, such as the Hill-Burton, review design decisions at two or three points in the process), forces the creation of a rather elaborate system of communications. The designer must record all of his decisions formally, as if they were final, for communication to the code people. The final design document is so elaborate and detailed that it is not economically possible to produce several different forms of document suited to different purposes; convention dictates that this information is given to the code people in a form, namely plans and specifications, which is primarily intended to guide the <u>construction process</u>. What is presented is a set of names, graphic symbols, and descriptive phrases which is rarely directly useful in the evaluation process, and may not even indicate the information necessary for the evaluation in the design of the system: the information for evaluation is presented in the form of plans and specifications, which may or may not be the correct, i.e., most efficient, form for evaluation purposes.

This first constraint immediately imposes a second constraint on the process of code evaluation. The information presented in plans and specifications not only may be in the wrong form, but there is always a great deal of extraneous information which is not required. Thus the first activity in which the code evaluator becomes immersed is the separation of relevant data from irrelevant, and its translation to proper form for usage. Thus there is a substantial added burden on the code evaluator in the performance of complex information search and retrieval. This is an operational requirement, but <u>it will act as a major constraint on how well the code inspection is performed or can be performed</u>.

The presentation of information in the form of plans and specifications also contributes to another major constraint in determination of exactly <u>how</u> design evaluation can be done. The information given on plans and specifications concerns itself only with physical entities, descriptions of their characteristics or attributes, or descriptions of the processes used to assemble them during construction. The function of the code system evaluator, then, is to construct comparisons between this information and the concepts stated in the code criteria.

The code criteria themselves are variable in form. They can be either materials specification statements* or performance specification statements. The former concern themselves with the specification of required physical entities or the specification of required attributes of those entities (such as material, dimension, etc.). The latter specify only the required functional performance of the building or some aspect of it, and never concern themselves with any physical characteristic of the solution. Many criteria do not clearly fall into either category, but occupy some middle ground*.

If the criteria are stated purely in terms of materials

---

*An example of a materials specification criteria: "Ceilings shall be acoustical tile". A performance specification for the same function might read: "Acoustic suppression shall be provided as required to maintain maximum noise level of 15 db within the space." A criteria such as "Acoustical tile shall be provided as required to maintain maximum noise level of 15 db." is neither purely materials, since it does not specify all the material characteristics, nor performance, since it does not provide for other means of acoustical control.

specifications or the attributes of physical entities, the problems of forming a comparison are quite different, and much simpler, that if the criteria are performance-oriented. The criteria of materials specifications can be compared directly, in many cases, with the information on the plans. The process of comparison is limited to establishing a correlation or identity between two pieces of information. For instance, if the specification called for acoustic tile ceilings, the location of that same phrase "acoustic tile" with reference to ceilings specified in the plans is adequate to complete the check.

On the other hand, if the criteria specifies performance, their is no direct relation between the listing of materials entities and the performance concept. The code evaluator is now faced with the problem of determining what elements within the space contribute to acoustic suppression; he will then require some sort of a model, i.e., set of rules or relations, which predict the performance of that configuration. The model must be capable of translating the given set of physical attributes into terms of level of functional performance. An excellent example of such a model is a set of engineering equations which allow us to determine if a beam called 16WF35 possesses a "safe" performance level in supporting a given load. The ability to use performance criteria by and large rests on whether or not such predictive models exist, and on what level of sophistication.

Even if the criteria in question are totally materials specification criteria, some translations of the data as described above are still necessary. For instance, if the criteria specifies that any room could be within a certain distance of an exit, this exit dimension is almost never given on the plans. Execution of the evaluation depends upon the ability of the operator to extract that data from the given information; if he cannot read it directly, he must execute a series of procedures involving compilation, measurement, computation, and/or judgment of the required values. (Compilation consists of reading several values from different sources; measurement the use of standard instruments, such as a ruler, for the derivation of values; computation of course, is the execution of mathematical procedures such as addition, multiplication and the like).

In either of the above cases, the function of procedures used has been to accomplish the translation of data into terms comparable to those of the criteria, allowing comparisons through the matching of identitical values. In some areas, the procedures are rigorously defined (even though they may not be particularly accurate). For instance, one aspect of the safety of a steel beam is defined as the condition that tensile stresses should not exceed x thousand pounds per square inch. There are procedures to predict the real stress in a given beam configuration, which also results in a calculated stress of x thousand pounds per square inch. The two numbers are directly comparable.

Most factors to be judged do not benefit from such rigorously defined procedures. There is in the Hill-Burton Code a criterion which stipulates "adequate" hand washing facilities, with no further definition of the term "adequate" provided. In the plans, there are a number of sinks located at the position in question. The process of comparison obviously depends upon some judgments on the part of the evaluator. He must form some kind of informal model in his mind, bridging the gap between the two concepts "x number of sinks" and "adequate handwashing facilities". The particular answer he may form is not of particular interest to this discussion; rather it is the fact that his "model" is based on his experience and his "intuition" and is highly informal in most cases, defying quantification and precise description. These same kinds of models are used in many areas of design: aesthetic quality, functional adequacy, and so forth. Perhaps they can be formalized and quantified, and perhaps not; the important thing is that they are used in many evaluations, and at the present time they are <u>not</u> defined. Thus the success of the system depends in large part on processes which we know little or nothing about.

Because these informal, intuitive, judgmental models are formed on the basis of personalized experience and capability, they are not always extremely consistent or reliable. But most code systems require consistency and impartiality in rendering judgments. In areas of concern where models capable of predicting performance are lacking or not agreed upon, the requirement for consistency forces the system to rely upon detailed materials specifications which can allow evaluations of unquestionable veracity.

The previous discussion suggests that an important characteristic of criteria is the <u>level of definition</u>. At least three important levels can be defined:

1. <u>Full definition</u>: the specification concepts are measurable in defined units, and logical procedures exist to relate the physical attributes of an entity to the specification concepts.
2. <u>Partial Definition</u>: the specification concepts are defined and measurable, but formal or rational procedures do not exist to relate physical attributes to the concepts.
3. <u>Undefined</u>: the specification concepts are logically ambiguous or imprecise.

These descriptions apply equally to materials specification or performance criteria. Since we have already discovered that the precision of evaluations depend upon the establishment of logical identities which are unambiguous, it is now possible to observe that the level of performance of an evaluation system will depend more upon the level of definition of the criteria than upon the form (i.e., materials or performance) of the criteria. It is also possible to observe on this basis that a well-defined system of evaluation should be able to handle either form of criteria equally well; the system will depend principally on how well the system of measurement and translation performs.

The importance of the human in the system is also clear at this point. If a system is fully defined, it can be expressed in terms of logical and/or mathematical equations, and a machine such as a computer could make the evaluations as well as - and probably more reliably than - the human being. But as the system becomes more and more undefined, we become more and more dependent upon the human ability to form the necessary intuitive models through his sense of rightness, or through his understanding of the values of society as he has informally, but adequately, come to understand them.

Summary:

Thus far, we have identified the following parts of the process, some of which will be controllable variables in the design of a new system, and some of which will act as constraints:

1. The code evaluation system is a subsystem in the larger design system, acting as an evaluation

mechanism subsequent to analytic and synthetic decisions by the designing architect or engineer;

2. The timing of the code system evaluation forces the designer to summarize all of his decisions in a formal set of plans and specifications;
3. The description of the design solution is therefore presented to the code system as plans and specifications, which may not contain the information in the precise form needed for evaluation, and always contain much more information than is needed;
4. This forces the code system evaluator (inspector) to first search through the documents for information he needs, and then
5. Translate that information into the form required through the use of formal or informal procedures,
6. In order that the final evaluation be made, where-ever possible, in logically equivalent terms forming identities.
7. The results of this evaluation are then returned to the designer, who must re-cycle through his analytic and synthetic phases to correct any errors observed during the evaluation.

This process is described as a flow diagram in Figure 4.1. A more detailed and elaborate model of this process was prepared in the course of this study, but is not here presented due to its complexity and difficulty of presentation*. It is hoped that this model will be available as a separate paper in the future.

---

*The model, in the form of a flow diagram, was prepare jointly by Weldon E. Clark of Bolt, Beranek and Newman, and Gary K. Stonebraker of the Institute. It covers in detail most of the aspect of evaluation discussed in the later sections of this report.

SUPPORT OF STATEMENTS 5 THROUGH 10: The Application of Computer Technology to Code Evaluation Processes

In the main body of the report, statement 5 states in a rather unqualified way that an electronic computer is capable of accepting a description of a set of criteria and a description of the complete geometry and contents of a building, extracting the relevant data from the description of the building, and comparing it to any precisely defined criteria. Taken in the unqulified way, the statements are true, but they deserve modification and qualification.

The principle qualification in the statement of feasibility is that the system can automatically process any precisely defined criterion. (i.e., "fully defined" by the definition in the previous section). The simple fact is that many of the Hill-Burton criteria are not fully defined (about 20% of the number of statements, in fact). Thus the design of a fully automatic system which handles all inspection work is, for this reason if none other, not feasible. The criteria requiring judgmental consideration must be left to the human in the system. Thus we are concerned with the design of a man-machine system, not just a computer system.

This is true for many other reasons as well. First of all, the data is presented to the system as a set of plans and specifications. While it might be desirable to consider revising the form in which the data is presented, it is not now considered practical to consider such alternatives. The reasons for this are presented later in this appendix. For the time being, it will suffice to say that this form of presentation must be accepted as a constraint on the system design. This means that the data must be translated from its combined graphic and alpha-numeric form into "machine-readable" form. This translation, while it might be automated fully in the future, for the present will depend greatly upon a human being.

Finally, the ultimate constraint of any computer system is that the machine possesses only the talent, intelligence and capability with which it is programmed. The rightness of its performance and its answers depends upon the rightness of the questions asked of it, and the human must always be available to judge the quality of the results of a computer operation.

Thus we are clearly concerned with the problem of interaction between man and machine, and the problem will be to intelligently divide the responsibility and the work to optimize the results, that is, to make the best results possible at the least possible costs. (Costs should not be construed in this case as dollars, but rather effort expended, in terms of humans, resources and other considerations.

The problem of designing a man-machine interaction system is significantly different than the problem of simply replacing human functions with a machine. The entry of the machine inevitably causes redesign in the role of man; likewise, the capability of the man to perform tasks must be examined carefully and contrasted to the machine capability available before intelligent decisions to use machine capability can be made.

Functional Problems Requiring Examination

The use of the computer in any evaluation role presupposes that a series of conditions are met:

1. First, that the machine possesses a complete file of all information it requires concerning the building design under consideration;
2. Second, that the machines "understands" how evaluations are made and to what information they are made;
3. Third, that it is capable of executing the evaluation;
4. Fourth, that it can give meaningful and reliable reports of the evaluation.

The first problem is usually described as the problem of data input; the second and third problems are those of processing, and the fourth that of output. Each of these problem areas is discussed in order in the following sections.

Methods of Processing for Code Evaluations

It will be useful to discuss the aspects of automatic evaluation by computer prior to discussing the problems of input and output, since the design of input and output subsystems is in many ways dependent upon the mode of processing selected.

Let us presume for the moment that the necessary data about the building, including descriptions of its contents and geometry, has been stored in the computer memory. (The accomplishment of this will be discussed in the later section on Input). We are now concerned with providing instructions to the computer as to how to retrieve and process this information such that the required evaluations will have been performed.

In their consulting reports, both Fair, Isaac Associates and Bolt, Beranek and Newman suggest that the data file, i.e., the description of the building stored in the computer, be viewed as a body of information about which questions are to be asked. Thus a criterion, such as "Every patient bed room shall contain a lavatory," can also be viewed as a question which may be asked about the data file: "Does every entity named patient bed room contain within its limits an entity named lavatory?" The question is answered by examining the data file, and a yes or no answer indicates whether or not the criterion is satisfied.

There is a rather large number of options as to exactly how this operation may be performed by the computer. The intention of this study has not been to identify the best way to do it, but rather to demonstrate that it can be done. Two different methods have been suggested by the two consulting firms in cooperation with the staff, and a hearing presented only as demonstrations of possible solutions, not as implied solutions. Each method has enjoyed limited operational demonstration on a computer.

Fair, Isaac and Associates selected what can be described as a "query" technique to perform the necessary processing, on the grounds that most criteria question the existence of an entity or an attribute which is specified in the data file. The mode of operation consists of stipulating a question representing a criterion; the computer responds by returning a list of all entities or attributes thereof meeting the condition of the question. Lists can be scanned visually for exceptions to the criterion, or in some cases automatically processed against master lists to determine the exception. For instance, to determine if all bedrooms contained a lavatory, the list of all bedrooms would be compared against a list of rooms in which the lavatory appeared. The techniques employed allow considerable freedom in the questions asked, and the list of questions need not be fixed. This feature is provided through complex list manipulation techniques, which allow great freedom in the structuring of the data in the file at some considerable expense to processing efficiency. The programming language used in the demonstration was LISP.

The techniques suggested by Bolt, Beranek and Newman in contrast to the above is presented in the following excerpt from their report:

> "If the requirement statements are viewed as questions to be asked of a body of data then we need to establish some form in which such questions are written. In the suggested system the requirement statements are considered to be in the form of logical expressions analyogous to those that would be written in a computer programming language. In natural language a requirement statement might be as follows:
>
> "If the space is a patient room and
>
> 1) the number of beds is 1 and the area is greater than or equal to 100 sq. ft., or
>
> 2) the number of beds is greater than 1 but less than or equal to 4 and the area in sq. ft. is greater than or equal to 80 times the number of beds
>
> then the statement is true; otherwise, the requirement is not met and this fact should be reported.
>
> The use of conventional logical and algebraic symbols serves to make such a statement much more concise and to standardize its form. The above example can be restated as:
>
> $$\text{If } (\text{space} = \text{pat rom}) \rightarrow ((\text{beds} = 1 \wedge \text{area} \geq 100) \vee (1 < \text{beds} \leq 4 \wedge \text{area} \geq 80 \times \text{beds}))$$
> $$\text{then statement} \Rightarrow \text{true else go to report}$$
>
> Where the symbols have the following approximate definitions:

$=$ means <u>is equivalent to</u>

$\wedge$ means <u>and</u>

$\vee$ means <u>or</u>

$<$ means <u>less than</u>

$\leq$ means <u>less than or equal to</u>

$\geq$ means <u>greater than or equal to</u>

$\rightsquigarrow$ means <u>is assigned the value</u>

"The parentheses imply that the statement is to be evaluated in parts, starting with the innermost parenthetical expressions.

A comparable statement for materials or equipment might be stated in natural language as:

> All patient rooms shall have lavatories with faucets having goose neck spouts opening above the rim of the fixutre and with wrist action handle.

This might be restated as:

> If (space = pat) $\wedge$ (lavatory = type A) then statement $\rightsquigarrow$ true else go to report

Acceptable lavatory and faucet combinations would be listed in a table labelled as Type A. If no lavatory were found referenced for a patient room or the referenced lavatory and faucet were not listed on the approved table a non-compliance report would be indicated.

Statements in such form are essentially por tions of a computer program. Thus, the collection of statements is equivalent to a program which is intended to operate a file of data which are the stored items of information from the design solution. The implication of viewing the set of statements as a program is that a collection of statements relevant to one code can be gathered together in one program and those relevant to another code in another program. Either of these programs might then be run for the same set of design data. Further, modifications to the set of requirement statements is analogous to modification of the logic of a computer program, which is a common and easily described process."

The actual processing of criteria written in the equation form suggested above is accomplished by examining the statement serially or in parts as suggested by the parentheses in the statement. The names printed in the statement are names of entities or the attributes of entities; the symbols are operators, or instructions to the computer as to which action it is to perform next. The information in the data file is organized to form a list of names and properties (i.e., values) of the names. For instance, the name of an entity might be "patient bed room"; its specific value, used to distinguish it from like entities, might be "room 154"; similarly, the name of an attribute of the patient bed room might be "area"; its value might be "225 s.f.". In processing, the computer will encounter a name; it would search through the file until it found a like name. It would then acquire the value of the name, e.g., "room 154" and would substitute it in the equation. Similarly, it might acquire the area value and all other values named in the equation. When all values had been acquired, the equation can be solved. If the resulting answer is true, the entity examined complied with the requirements; if not, it is tagged as an exception. The exact mode of reporting exceptions will be discussed later under Output.

In order to organize the information properly for such kind of processing, it is necessary to arrive at some conventions, i.e., systems of notation, specifying how the information will be stated to the computer and in what manner it will be stored. Bolt, Beranek and Newman suggest:

> "An attempt to automate the process of checking the design of a building against code regulations must depend on some organized and logical description of the building. Thus, an important problem to be faced is development of a rational scheme of notation. Such notation must provide means for describing all of the elements and networks of services which together make up a modern building. The notation scheme should be equally useful at a more general level for description of groups of buildings in urban developments. A partial development of such a system of notation is presented in this Section.
>
> A. <u>Entities and Nested Sets of Entities</u>
>
> A physical complex of spaces, enclosing elements, networks of services, etc., can be described as a collection of entities. For example, if the complex being described is a college campus the major entity of concern is the entire site. The set of entities which that site includes or <u>owns</u> is comprised of buildings, utility networks, and circulation networks.
>
> Each of this set of entities may in turn own a subset of entities. For example, a building may be comprised of a collection of entities which are the major wings or floor levels, and within each another collection that are the rooms of the building. Each room may further have a subset of entities which includes items of equipment etc.
>
> B. <u>Kinds of Entities</u>
>
> A number of different varieties of entities can be defined that together comprise the important features of a given physical complex. Five different kinds of entities are defined here as sufficient to describe buildings and their environment:
>
> Spaces
> Networks
> Enclosure elements
> Equipment
> Materials

"An entity of one kind may subsume or own entities of another kind. For example, a space entity that is a building may own an airconditioning network entity which in turn owns airhandling equipment, cooling equipment, and diffusers, -- all equipment entities. A particular space entity may include enclosure element entities -- the walls, floor and ceiling of the room.

The distinction between the five kinds of entities are not always precise and clar-cut. As an example, the collection of corridors, stairways and elevators, sidewalks, driveways, parking lots, and so forth which comprise the circulation elements of a medical center complex may be considered as a collection of interconnected space entities. These may alternatively be considered as a circulation network.

C. Attributes of Entities

The five different kinds of entities which we have defined can each be described in terms of a list of attributes. These attributes are the main factors which can be measured to distinguish one entity from another of the same kind. A partial list of attributes for each of the five kinds is presented below with some explanatory comments. A definition of some important properties of the attributes is also presented.

1. Lists of Attributes. The attributes which are defined for each of the kinds of entities are stated in Table 1. Table 1(a) applies to spaces as entities, Table 1(b) to networks, etc

   Unique entities must each be fully described in computer storage. Many times, however, repetitive instances of an entity exist that need not be separately described. In these cases an instance form is used in which only the few attributes necessary to define the existence and the location of the particular instance are necessary. Table 2 illustrates the instance forms.

D. Properties of the Attributes

There are several items of information that are important for each of the attributes listed in Table 1. The principal property of each attribute is, of course, the value that that attribute takes on for a particular instance. If the attribute were area the value might be 100 sq. ft. for instance.

TABLE 1

ATTRIBUTES OF DESIGN DATA ENTITIES

Attribute	Description
1a Attributes of Space Entities	
Identifier	Code for kind of entity
Name	Label for particular entity
Location (x, y, z)	Coordinates of reference point (e.g., corner of space)
Shape	Code for basic geometric form class (e.g., rectangular)
Plan coordinates x_1, y_1 x_2, y_2	Location coordinates (e.g., for corners)*
Height	Vertical dimension of space
Area	
Volume	
Reference to enclosure element entity	In same order as lines described by plan coordinates; then floor, ceiling
Reference to materials entity	References to surface finishes, in same order as enclosure element reference above
Reference to owned entity	Reference to equipment, doors, subunits of space (e.g., toilet enclosures, closets) etc.

*For other geometric forms other coordinates or dimentsions might be used (e.g., radius for circular shape).

Attribute Name	Description
Ib Attributes of Network Entities	
Identifier	
Name	
Location (x,y,z)	
Configuration type	

Junction node label	One subset of attributes for each node joining branches
Junction node location (x,y,z)	
Reference to connecting branch	
• • •	

• • •	

Terminal node label	One subset of attributes for each node terminating the network
Terminal node location (x,y,z)	
Reference to connecting branch	
Reference to owned network or equipment entity	

• • •	

Branch label	One subset of attributes for each branch
Reference to owned space, material or enclosure entity	
Size or capacity attribute	
• • •	

• • •	

Attribute Name	Description
1c Attributes of Enclosure Element Entities	
Identifier	
Name	
Format code	Reference to list of characteristics pertinent to a category of enclosure elements
(Characteristic) • • •	Description attributes for a category of enclosure elements
Reference to material entity • • •	
1d	
1e Attributes of Materials and Equipment Entities	
Identifier	
Name	Reference to list of characteristics pertinent to a category of materials or equipment
(Characteristic) • • •	Descriptive attributes for a category of materials or equipment

TABLE 2
ATTRIBUTES OF REPLICATED INSTANCES OF DESIGN DATA ENTITIES

Identifier	
Name	
Location (x,y,z)	(Only for space and network entities -- must be for the same relative location as in the original entity)
Reference to original entity	

"Further, if the entity was a unit of space such as a courtyard we might describe an attribute called *shape* for which the value might be *rectangle*, *circle* or *octagon*.

Another property of the attributes that is important in a descriptive process we can label as *mode of determination* of the value. This property does not pertain to the current value of the attribute in question but rather indicates how one evaluates the attribute. We define a scale for this *mode of determination* as follows:

Scale for Mode of Determination

1.	Observe	(Check whether in existence)
2.	Compile	(Derive information from several sources)
3.	Measure	(Compare with some standard unit --a yard stick)
4.	Compute	(Derive a value by some arithmetic operation)
5.	Judge	(Assess the value by taking account of intangible factors, etc.)

The simplest mode is to *observe* whether something exists. Successively more difficult determinations involve deriving a value. If an explicit derivation is impossible only a human judgment can be used to obtain a value."

By the use of such ownership, the total structure of the data file is a structure of five data "trees", one tree being used for each of the five classes of entities, described. The trees are connected at points where the names of the entities owned are common; for instance, the tree of space entities may contain a room; on the ownership list of that room might be an equipment entity such as "air diffuser", with coordinates describing its position. Similarly that same air diffuser will appear in some position on the "network-entity tree". This kind of structure is useful in examining more complex relational problems, such as whether or not rooms in a particular department are serviced by an isolated zoned air conditioning network. But the principal organization of the entity trees serves to relate the entities in the building functionally, i.e., to organize the parts of a system heirarchally. This kind of organization allows maximum freedom in adjusting the level of detail to which any system in the building is described, and allows logical organization of entities into systems and subsystems as they normally function. Thus as heating systems are being discussed, the data is accordingly organized, and need not include, for instance, discussion of space entitites unless required.

A word of caution is justified with regard to the representation of entities as networks. Little or no experience has been gained in the field of describing elements of buildings as networks, and substantial research may be required in the organization of programming with regard to the use of such networks. The tree structure, however, is a common technique in the organization of data, and several tree searching algorithms have been developed and are in use.

Given this kind of a file structure, a complete building or any portion of it, to any desired level of detail, can be described; the structure of the lists provides access to any of the data therein without necessarily searching the entire list of entities. Such a list structure is not actually required. Data could be introduced in non-related packages, if the system designer were willing to accept the necessity of searching _all_ of the data every time he needed a piece of information from the file. The list structure, plus the conventions of the heirarchal organization of the entities, serves much as an indexing system in a file drawer, in that it helps to isolate the location of information desired without searching the entire drawer.

There are some problems associated with such a proposed system of processing, however, which deserve attention.

First, the problem of synonyms is encountered. If the criteria statements specify that each patient room shall contain a _lavatory_, and the data is inadvertently in the file entered under the name _sink_, the computer has no way of knowing that "sink" and a "lavatory" are identical. Similar problems could be encountered with any group of commonly interchangeable words, such as patient bed room, patient room, patient care room; operating room or suite; corridor or hallway, and so forth. One way of overcoming such a problem is to program the machine with a dictionary of synonyms. With such a dictionary, the machine would automatically check for synonyms for names of data it could not locate. Upon finding such a synonym, it would then re-search the file for the data under the synonym name. Without such capability, the machine will tolerate no deviation whatever from naming data exactly as it is called for in the criteria statements. This, of course, imposes an added burden of precision upon the operator, who must remember precisely what name must be attached to each piece of data during the input procedures.

A second problem is related to the exact kind of data required in the criteria as opposed to that given on the plans. For instance, the criterion specifying the required area in patient rooms refers to the area value. Such a value may or may not be given on the plans. If it is not, the operator would have to stop, compute the area, and enter that information. Let us suppose that he did not compute this precise piece of data, however, but rather entered the length and the width of the room into the data file. During processing, the computer would search for "area", but would find no such data. In fact, the existence of the length and the width in the file would allow the determination of the area of the room with a simple computation. But the computer would have no way of knowing this unless it were programmed with derivation procedures which allowed it to attempt to compute needed values from data existing in the file. Without such procedures, the program would have no alternative but to report the data missing and discontinue evaluation of the criterion. Existence of such derivation procedures in the program can greatly simplify the problem of input by reducing the number of manual translations of data necessary to get the proper information into the machine. This point will be discussed again in the discussion of input.

The above problems and proposed solutions are in fact discussions of how to add flexibility to the computer's capability in handling and processing data. The need for such flexibility is in fact dependent upon the ability of the operator to match the machine's need for precision. If there is a presumable correlation between the demand for precision and the level of skill required of the operator, it is generally true that additions of flexibility in the machine operations reduce the skill requirements of the operator.

The precise evaluation of the trade-offs between these factors is extremely complex, and probably must finally be determined by testing of a prototype operational system.

The Requirement for Precision in Criteria Definition

In the opening paragraphs of this section of the report a brief discussion of the level of definition was presented. At this point, the full ramifications of the problem of criteria definition should be discussed.

The logic of any evaluation must be expressable as a series of mathematical or logical operations if it is to be executed by the computer. The computer possesses certain fundamental arithmetic-logic capability with which it can manipulate data; e.g., it can add, subtract, multiply and divide; its logical capabilities include establishing equivalence of two numbers, or determining if one quanity is smaller than or larger than another, among other capabilities. A program is a set of instructions as to which of these capabilities is to be applied to what information, in what precise order.

The "evaluations" which will be performed by the computer will in fact be sequences of these actions performed on numbers or words (the identity of words may constitute a check). Thus each of the concepts in the code criteria must be redefineable as a series of these actions. This is not difficult where the concepts are precise; concepts such as "x shall be equal to y" or "at least equal to y" are obviously and easily handled. Some concepts, such as "does every x contain a y", are more difficult to define but still defineable (in this case by establishing identity between the coordinates describing a reference point on entity y and the coordinates of one of the set of points contained within entity s).

Wherever a criteria statement uses a relative term as its logical operator, it is not possible to define such evaluation processes. What is the comparison of "x shall be adequate..." or "x shall be near to y?" Both adequate and near are undefined operators in the comparison of x and y, dependent upon the existence of informal models of these concepts in the mind of the human evaluator. But the criteria do suggest that certain kinds of information are relevant to the formation of such judgments, and the computer can at least locate and present what quantitative data it does have concerning these evaluations. For instance, given the criteria "handwashing facilities shall be adequate...", there is a clear implication that the number of lavatories is involved in the evaluation, perhaps along with their location, and this information can be retrieved and presented to the human for judgment.

In summary, the computer can play two roles in the act of processing evaluations:

a. Where criteria evaluations are logically defineable procedures, and presuming the existence of necessary data in the file, the computer can fully and automatically perform the required evaluations.

b. Where the concepts of evaluation are undefined, the computer can act as a high speed information retrieval device as well as an editing device, presenting only information appropriate to the evaluation.

Batch Processing and On-Line Processing

The organization of processing in the computer, regardless of the particular processing logic selected, can be done in several basic ways. First, all data can be acquired prior to the start of processing, entered into the computer, and processed continuously in a single run. This is called batch processing. At the other end of the spectrum of possibilities, the computer can accept the data one piece at a time as the operator locates it in the plans and specifications. By continuously monitoring the file of data it builds, the computer can decide when it has enough data to do a particular job. It can then execute that particular job, and return to monitoring the additional data as it comes in. This procedure is continued until all the jobs specified in the program have been completed, or until the operator signals that no more data is forthcoming. This is one form of a procedure called on-line processing.

On the surface, batch processing appears to be more efficient for most jobs, since the computer is actually working all the time it is occupied with the task. With on-line processing, there may be lengthy periods when the computer is idle, waiting for the "slower" human to provide more data. If, however, several operators are working on similar jobs at the same time, techniques of time-sharing allow all of them to feed problems to the computer simultaneously; any number of operators can theoretically be added until the machine capability is saturated.

The computer operating as a time-sharing machine operates effectively by spending a short time (a small fraction of a second) on each problem it is fed. It then stores the work, goes to the next problem, works on it, stores that one, goes to the next, and so on, until it comes back to the first problem. The process is much analagous to placing four sheets of paper on the table, and writing a problem on the top of each sheet. You might spend thirty seconds on that one and so forth, until all the problems were solved. The computer, of course, does this so rapidly that it appears to each of the user that he has the full attention of the machine. As will be discussed later, this technique may have applicability to the Hill-Burton system problem dues to its work load requirements.

The reason that on-line processing systems are potentially of interest is that the computer capability is at hand and available for other tasks related to the input of data. This capability will prove to be important in the design of the system; it is discussed in detail in the section discussing the problems of input. If such capability is desireable, then a third machine configuration should be mentioned. This is the possibility that a small computer can be used to assist in input and to do a limited amount of <u>preprocessing</u>. Preprocessing describes a myriad of small but useful chores, such as the screening of data, accuracy checking, conversion of data from given to required forms (such as the conversion of linear dimensions to area), provision of synonym capability, and other tasks. These specific tasks are also discussed in greater depth in the section on input of data. It might be useful to consider the use of a small computer, interacting with the input operator, to perform such tasks. The output of this small computer system would be a complete data file ready for final processing. A larger machine would be used to batch process the file of data prepared by the smaller machine.

There are many other alternatives to be considered, and the selection of processing schemes and machine configurations is a decision requiring careful and detailed evaluation. This discussion is intended to demonstrate that a large number of options is in fact available, and that there is no single system which can, at this point, be said to be the only system for the job. Subsequent work on the actual development of such systems must accomplish two things. First, it must clearly define criteria of cost, operating efficiency, and other factors by which an actual system selection can be made. Second, it must not reach premature conclusions about the "correct" configuration; it must allow for full exploration of the many complex alternatives to assure that an optimum design is reached.

computer; such a machine is probably feasible only on the rental basis. The main purpose of a central facility would be the support of a large purchased computer, plus whatever efficiencies in work scheduleing could be accrued. The latter point is not important in view of the fact that in the Hill-Buton case each regional office would be able to occupy at least one input operator full time with no difficulty.

II. ON-LINE PROCESSING SYSTEMS, GRAPHIC INPUT

Central Facility: In this instance, the probable work loads could demand the use of a medium sized computer from which (work loads demanding) several IO stations could be timed shared. Submission routines would be approximately the same as above described for the centralized batch processing facility. Processing, in this case, occurs simultaneously with input and output.

Decentralized Facility: There is no technological problem, once time-sharing is accomplished, in the physical removal of IO stations to remote locations. Thus regional offices could be equipped with IO stations linked by long distance lines to a central computer which did both processing and IO control. Long distance lines add considerably

Data Input

The final specification (i.e., design) of a building is a description of physical entities, descriptions of their attributes, and descriptions of the position of those entities in space and relative to one another. The problem of data input is one of providing a machine language description of these physical entities, their attributes, and their position in the total complex of the building. Input is essentially concerns itself with the selection of appropriate data and its translation from words and/or pictures to machine-readable numbers.

Data is presented to the code system in the form of a set of plans and specifications. It is possible to trace it back to earlier forms, such as sketches, calculations, and even thoughts in the designer's mind. In one of these earlier stages, the information may have existed in a more appropriate form for evaluation purposes; ideally, it would be desirable to reexamine all of these previous forms of the data, picking the data up prior to its translation to less convenient forms. To do this, however, implies the insertion of a monitoring mechanism at some point in the design process, or even change in the process itself through the introduction of "design machines" which might be capable of sending data directly to the computer file. The constraints and enormous operational problems implied by such changes are almost immediately overwhelming. Immediately, the problem of diverse, long distance communications systems linking hundreds of design establishments to the code system computer is enough to disqualify such prospects on the grounds of cost. In addition, there would be equipment installation costs, the costs of retraining the users, and the problems of resistance to change in "normal" procedures. It would appear that practical solution of the immediate problem requires acceptance of the data in the form of plans and specifications, so as not to extend the boundaries of the system beyond the scope of Hill-Burton (or any other code system) jurisdiction.

Plans and specifications are a collection of two distinct kinds of information:

1. Alpha-numeric data, which appears on the plans as words, sentences, numbers or abbreviations, and which composes the virtual entirety of the specifications;
2. Graphic data, including plan diagrams, elevations, sections, details, site maps and symbols (such as those used for electrical outlets, columns, doors, windows, etc. or for the indication of materials, such as concrete or aluminum); graphic data uses pictures to name entities or to delineate certain attributes, such as shape (rectangular) or dimension (in the case of scaled drawings).

Information relevant to the execution of evaluations is found in both forms. It is estimated that some 50% of the evaluations will be dependent upon information presented essentially in graphic form. Evaluations dependent upon relative position (is this near that?) upon dimension (200 square feet in area?) and in many cases simple checks of existence (is this provided?) will be answered by information from pictorial displays.

On the other hand, specification of materials, dimension, the names of entities, or similar information will be encountered in alpha-numeric form in notes, printed text, or schedules, combined in many cases with graphic data.

The input of alpha-numeric data, i.e., words and numbers, can be handled in a number of ways conventional to computer systems. The translation of such data to computer-readable format can be accomplished through the use of machines with alphabetical keyboards, like a typewriter, which can produce punch cards or paper tape (which is then read into the computer by mechanical reading devices; these devices sense an aperture in the tape or the card to have a standard alpha-numeric meaning*) Devices such as electronic typewriters bypass this intermediate processing (the preparation of cards and tapes) and feed the computer directly or through intermediate magnetic tapes.

---

* See footnote on next page.

Any of these systems is well suited to the transmission of alpha-numeric data to the machine.

Graphic data presents a somewhat different problem. In order to make "pictures" readable to the machine, they must be essentially redescribed as a series of words and/or numbers, to be stored in the computer memory by the same notation conventions as is the alpha-numeric data. It is therefore necessary to invent a scheme of notation for translating the graphic data into words or numbers.

Graphic data carries in it information regarding the name of the configuration (in the case of a symbol, the symbol is logically equivalent to a name, e.g., "light switch"), and information regarding position, dimension, geometry and/or orientation. There are several ways that each of these kinds of information can be translated into alpha-numeric form:

Names: Where the name of the graphic data is a symbol representing a name, the english name can be substituted and entered as conventional alpha-numeric information. For

---

*The computer does not actually store words or numbers in the sense we know them. The computer "reads" only an electronic pulse, or the absence of one; the presence or absence of current stands symbolically for the number 1 or 0. A string of such numbers (or current and no current), such as 1000100, stands by convention for a standard alphabetic character or decimal number; a word of information is a string of such binary numbers. The apertures in a punch card or tape allow the passage of a beam of light or the closure of a pair of contacts on each side of the card, which triggers a pulse representing a 1; the absence of a pulse is construed as a zero.

instance, if the picture represented indicated "room" or "corridor" or "column" or "light switch", and we wished to convey that information to the computer, those precise words could be used as a label for subsequent information.

Position: By and large, only rectangular spaces or composites thereof will be encountered in a conventional plan. It is convenient to think in terms of describing any point of space in the entire complex, including the site, as a set of cartesian coordinates (x, y, and z values) referenced to some convenient point, such as a corner of the site or a bench mark. Thus the precise position of any point can be determined. From such information, it is also possible to compute distances between points, thus deriving any desired linear measure.

Geometry: All entities within the building are volumes, and as such cannot be represented as a single point, obviously. Thus when the "distance between a and b" is discussed, it usually implies the distance between some points of reference on the volumes a and b, or distances between the parallel planes a and b. Thus "the distance between the room and the exit" is perhaps measured from the center of the door to the room (a point in the room volume) to the center of the exit door, or by other convention. Similar conventions may prove convenient in describing the geometry of any plane or volume in the building complex. For instance, a room might be described by selecting one corner as the reference point, and describing successive points around the room with reference to that point. This could be done in two manners. Fair, Isaac & Associates suggest that, using the top of the plan sheet as arbitrary north, one could begin in the lower left corner of the room, giving the reference coordinates of that point. Successive points would be described proceeding clockwise around the figure as north 10', west 12'6", south...and so forth, until the figure closed. Circular figures or portions of circles might similarly be described as a center of revolution, a radius, and the coordinate points of start and finish. Due to the fact that it might be more difficult to describe non-rectangular figures, or portion of plans with directional changes such that all parts of the building do not line up on the same set of right axes, other alternatives may be more practical. Bolt, Beranek and Newman suggest that all "critical points" in the geometry of any shape be described as sets of coordinates. For instance, the corners of a rectangular space would be listed as a series of x-y coordinates referenced to a single point. Any convenient or necessary number of reference points could be set up; for instance, the rooms on a floor might be referenced to one point on that floor; that point in turn referenced to the main reference point on the site. This method can also be used to describe virtually any shape of space encountered. Irregular figures composed of straight lines can be described by fixing the line intersections points of the space. Curvilinear figures can be approximated by giving successive points along the curve. In the case of shapes for which defined mathematical functions exist (such as a parabola) the space can be generated by a special program, given the critical variables. In the case of describing a shape by fixing critical points, e.g., corners or vertices, the computer can be programmed to assume and generate straight lines between the points described.

By either method, advantage can be taken of the fact that many shapes, especially equipment such as beds and sinks, are repetitive. The shape can be described to the machine once, along with a reference point (such as the center of the head of the bed). Repeat instances of the shape can be entered into the machine by locating the new reference point and asking the computer to regenerate the shape described to it before.

Dimension: Entry of data on dimension would require the description of both the origin and destination points, plus the value of the dimension. The position of the origin must be given by the methods describing position; the remainder of the data can consist of either a length and a direction (such as north) for the dimension, or else the length and the coordinate values for the second point. If the entire building geometry has already been entered in the computer as a coordinate system, substantial advantage can be gained, as the computer can be programmed to automatically compute any dimension between given points (or between points that can be generated as functions of stored points, such as the centroid of a planar figure or the center of a line). Thus, if the building geometry is stored as a set of coordinates, computations of area, distances between points, and checks of dimensions is greatly facilitated. Since in fact many criteria are concerned with such attributes of space, this facility is important in the selection of modes of geometric description.

Orientation: In some instances, changes in the direction of plan, or angular offsets of the conventional right angled grid used in planning, make it desirable to describe the orientation of various parts relative to one another. This can be accomplished by referencing subgrids to major grids or reference grids; once again, such data can be computed automatically from sets of cartesian coordinates outlining the parts in question, and determination of orientation changes may not require the addition of special information.

The above summary lists a few of many alternative schemes, but these are now in use and being demonstrated in several operative systems, and are known to be practical from the operational standpoint.

Requirements for Selective Input

Obviously, not all the information in a set of plans and specifications will be relevant to the execution of evaluations required by the code. For instance, much of the specifications is concerned with legalities controlling the performance of the building contractors; a considerable additional amount specifies particular construction procedures to be used. A large number of entities specified for the building are of no concern at all to the code system. It is therefore inefficient, and certainly impractical in view of the limitations of computer storage capacity, to think in terms of entering all the information found on a set of plans and specifications. In order to limit the size of the resulting computer file and to provide for the efficiency of processing, it will be necessary to restrict the input of data to that which is actually useful and/or required in the evaluation procedures.

As mentioned before, we shall by and large be concerned with the entry of lists of selected entities along with descriptions of certain attributes of those entities, such as dimension, color, position, materials, relation to other entities, and so forth. In some cases, the code is only concerned with the existence of an entity (is room x provided?), in which case the existence of only the name of the entity in the file may be adequate for evaluation.

In other cases, the attributes of the entity are subject to evaluation (is x washable? does x contain y area?); in still other cases, it is the relation of entities in question (is there an x in every room y? is x near y?). Thus the amount and kind of information which must be entered about different entities is variable. The major question which faces the person developing input for the computer is therefore "which entities must be described, and which attribures of those entities must be included in the description?"

An indication of what information is required is contained in the criteria statements themselves. For instance, if the criteria asks, "Is there a lavatory in every patient bed room?", we can deduce that the information we will require will be a list of all lavatories and their locations, and a list of all patient bed rooms and their locations. The processing could be accomplished for instance, by assuring that each area representing a patient bed room contained within it another area representing a lavatory. Using similar procedures, the complete list of criteria can be examined to yield a list of all information required. Such a list can be used by the operator as a check list for the entry of data.

Options for the Organization of Input Procedures

In this discussion, let us assume that a system of notation has been selected which is approximately like the one described in the previous section, and that data shall be entered from the plans and specifications selectively as previously discusssed. In order to assure that the proper amount of data is entered, and only that amount, we will specify that some sort of check list will be provided to guide the operator in developing input. The check list will consist of names of information required, for which values of data are to be supplied.

The check list itself can be organized in many ways; the particular method used will be dependent upon which kind of mechanical device is to be used in transmitting the data into the computer. Thus the organization of input procedures is dependent upon the mechanical system selected for data transmission, and vice versa.

Most data processing systems in use today depend upon the punch card or paper tape as a data input vehicle. Systems tend to employ these techniques for batch processing of data, as the time necessary to punch cards or tapes, and the probability of errors, makes card or tape systems impractical for on-line processing systems. (For on-line work, other devices such as typewriters linked directly to the computer are used). Therefore we tend to think of the use of cards or paper tapes in association with batch processing techniques.

In the event that a batch-processing system proved desirable, the check list for the data could be provided through coding forms, which specified the information desired and the exact format in which it is to be written. An operator would be equipped with a set of such forms at the start of the evaluation process. He would acquire the data from the plans and enter it on these forms. The forms would then be given to a keypunch operator, who would then transfer the data to punch cards or paper tapes. These would be "verified" (checked) and then fed to the computer at any convenient time.

A major constraint of such a system is the way in which graphically presented information must be handled. Given the fact that a great deal of information of position, dimension and relations of spaces (or groups of spaces) must be provided to the data file, and given the fact that a coordinate system appears to be the immediately practical way to describe the geometry of a building, we are immediately faced with the prospect of writing down at least two, and perhaps three, numbers to describe every point in space needed. Defining a rectangular space would require measuring each corner with respect to a reference point, determination of an x and y value for that point, and the manual entry of that data on the form. This information would then be keypunched, checked, and read into the machine. Such a process seems at best cumbersome and inefficient. (There are techniques available to read data directly, or semi-directly, from forms, such as mark-sense readers, which read sensitized marks made with special pencils in a particular position of the form; this would at least eliminate the keypunching aspect, but does not overcome the burdens placed on the data gatherer).

Since in fact graphic representations are the source of at least one-half of the data needed in the evaluations performed in the Hill-Burton procedures, this particular aspect of input is quite important. Thus it was felt that the study could profitably explore the uses of more exotic input equipment, and especially devices which were designed to handle graphic data.

The most sophisticated of such devices is the optical scanner. This device can "read" pictures recorded on microfilm, and could be programmed in this instance to automatically assign coordinate values to all the points picked up on the scanner. The major problem in the use of scanners relates to the state of the art in their development. At the present time, it is extremely difficult to instruct a scanner to discriminate information on the film (for instance, differentiation of lines representing a sink and those representing a room). Extraneous data on a plan would probably have to be screened prior to submission to the scanner.

It might be feasible to think in terms of such preprocessing of the plans before microfilming. For instance, if it were desired to use a scanner to pick up the plan configuration, the plan might be outline in red pencil, photographed through filters to pick up the portions outlined in red, and then submitted to the scanner. These, however, are hypothetical possibilities, and must be taken cautiously in view of the high costs of such equipment and the limited state of the art.

More suited to the specialized purposes at hand are other kinds of "graphic input" devices. Two major categories of devices are now in use on limited prototype bases. The first is an analog-digital conversion device consisting of a pen attached to an arm, which is in turn attached to a base. Angular movement of the arm rotates a rotary potentiometer located in the base, and extension or retraction of the arm operates a linear potentiometer located in the arm itself. Each different position of the pen produces a slightly different pair of voltages through their potentiometers. If one of these voltage pairs is correlated with a coordinate pair, and the scale of movement defined, the computer can automatically compute the correct coordinates for all other positions of the pen within the limits of its movement. Thus if a plan were presented to the device, the operator could place the pen at one corner, assign a pair of coordinates to that point, give the scale of the drawing, and proceed to trace the rest of the plan. The device would automatically compute proper coordinates for all other required points in the plan.

The plan need not be traced in its entirety, depending upon the mode of information storage. Most likely only the corners or other critical points of space would be located with the pen. The device does not automatically store <u>all</u> points traced, but only those for which a specific signal is given by the operator.

A similar device in purpose is the Rand tablet* which is a flat surface 10" square. Imbedded in the surface is a matrix of fine wires, located about 1/100th of an inch apart, so that there are approximately 1000 wires running horizontally across the face of the tablet and another 1000 or so running vertically. These wires are linked to a special device which transmits a coded pulse through each wire. The pulse passing through each wire is different in character. A special sensor pen held on the surface of the tablet at the intersection of a horizontal and vertical wire will pick up the codes passing through those wires. If a plan image were projected on the face of the tablet, the sensor pen could be placed at the points desired, and the coordinate values of that point would automatically be interpreted. While the device may be somewhat limited in size, it has two distinct advantages over the previously described device. First, Its rate of data transmission is considerably faster, thus allowing the operator to work faster; second, the operating pen is not attached to an arm (which might restrict the operator somewhat).

These devices, if employed in the system, would be used principally to automate the translation of graphic data to coded information required by the computer. They eliminate the need completely for the manual interpretation and entry of geometric and dimensional data, by virtue of the fact that auxiliary programs can compute such data from a set of coordinate descriptions. The following kinds of information can either be entered directly, with no intermediate translation or processing, or computed from data entered:

location
boundary position
length)
width)of a space or piece of equipment
height)
area
volume
distance

The automation of the measurement function can mean substantial differences in the time required to develop input required, as will be discussed in the later section on systems costs and benefits.

An important aspect of graphic input operations is the provision of feedback information on the accuracy of input, which can be provided through electronic redisplay on an oscilloscope screen driven by a computer. (An oscilloscope screen is similar to a conventional television screen in appearance). As the operator develops input, it can be instantly redisplayed to him, thus providing him with a visual check on the accuracy and completeness of the information he is providing. Corrections can be made instantly, and thus the possibility of bad data being used in a run is considerably less than in the card systems previously described. In order to provide this added capability, the devices are usually run on-line with the computer. With the addition of an electronic typewriter, alpha-numeric data may also be entered on-line. This, of course, completely eliminates the need for intermediate processing (card punching and related form-writing, as suggested earlier). As will be shown later, this on-line capability adds considerably to equipment and programming costs, but drastically reduces the time necessary to develop input.

The addition of on-line capability also means that the computer is available for many other duties in the course of the input routine. As an alternative to the use of a written check-list for data required, the check list may be stored in the computer. The list can be displayed on the display screen mentioned above; as data is given to the machine, the computer can automatically edit the list, thus providing a positive guide to the operator in the acquisition of data.

In addition, the machine can communicate with the operator in many other ways, even guiding the operator step-by-step through the entire input sequence. The computer and its operator can actually hold a "conversation," the computer asking for data, the operator answering with appropriate responses. A computer operating in such a fashion is said to be in conversational mode with the operator.

The addition of this capability can have a most important effect on the total man-machine configuration, for as the machine acquires the capability to organize and assist with the work, the skills required of the operator can be sharply reduced. Virtually the only capability required of the operator would be basic skill in reading blueprints and specifications, plus a basic ability to operate the machine console. The need for professional judgment or sophisticated ability in interpretation of designs is nearly eliminated from the input operation. Thus it is practical to think of the entire input operation being conducted by a junior draftsman at about the GS-7 grade level.

The addition of the conversational capability changes the role of the machine from a passive role to an active role of communication with the operator. In the batch processing system, no such communication exists except in a very indirect way. Effectively, the machine passively waits for the operator to collect and input all data, and then gives him a diagnosis of format and/or certain kinds of keypunching errors. While such diagnostic programs are useful in batch processing, they do not provide help to the operator at the time he is acquiring data; rather, they point out certain kinds of errors after he has made them. The event of an error forces the cancellation of the run, and forces the operator to return to the data to recheck and correct the error.

In contrast to this, the on-line system utilizes one of the computers major capabilities - consistency and accuracy - in a dialogue with the operator to assure the quality of input data. In addition, it performs the translation of graphic data (with the help of graphic input equipment) automatically eliminating the necessity for tedious hand translation, and, as a final boost to system operations, the on-line capability completely eliminates the need for all intermediate processing.

Systems of Output: The Communication of Results

Once the processing has been completed, the computer must also be instructed as to how to communicate the results back to the operator, in what order, and in what format. This is a vital link in the total system, as the computer is capable of producing literally reams of information. The art of designing output concerns itself with knowing what is needed and to whom it must be communicated.

The usual practice in reporting the results of a code evaluation is the reporting of exceptions to the requirements; this is probably an acceptable practice oriented toward the minimization of communication required. In the computer system, a similar practice can be followed to minimize the output of the machine. As pointed out before, however, the computer will probably produce some exceptions to the strict rule which are in fact acceptable design solutions. These exceptions should be reviewed in detail by the code professional to assure the quality of the evaluation. Thus the output of the computer system should not only report the fact of the exception, but should provide relevant data in a well-organized format which minimizes the need for referral to the plans and specifications. This can help save the professional work in searching the plans and specifications for data relevant to the exceptions, and can thus reduce the time spent in this final evaluation.

Forms of Output Available

The computer can produce both words and pictures as output. Words (and sentences) can be printed on paper either on the on-line typewriter or on high speed printers operating on or off-line. The format of the output is virtually limitless; it can appear as groups of sentences, lists, tables, graphs, or any other convenient form. Thus each response can be tailored in format to the requirements of the criteria. It might be convenient, for instance, to present evaluations which are likely to involve a number of exceptions as a list:

> "Following Patient Bed Rooms do not meet minimum area requirements:

Room #	Area Given	No Beds	Area Reqd
156	95	1	100
163	144	2	160
191	144	2	160

Other kinds of criteria may be best reported as a sentence:

> "Incinerator proposed does not have sufficient capacity to meet requirements proposed in section ......"

These formats can be combined in any conventional way for the production of diagnostic comments.

The diagnostic statements are referred to whenever an exception is discovered during processing. They are stored in the computers memory as standard senteces or formats, and can be modified or changed without affecting major reprogramming in most cases.

In the case of on-line processing, these responses might also appear on the display screen for action by the operator; they might also be printed out immediately upon the completion of processing, assuming that processing is occurring simultaneously with input. This could be an important factor in reducing the total workload. In the event that a major error was discovered, such as the absence of an entire department or an inadequate number of patient rooms, it would probably not be useful to continue the examination, as correction of that error has major effect on the total design. If the operator were made aware of that error at the earliest possible time, he might elect not to process further, thus saving much needless input and processing time.

If the computer has enough information in memory, it can also produce diagrams and drawings on a device known as an x-y plotter. This device consists of a pen moving over paper on a travelling arm; the pen moves to positions corresponding to x and y coordinate values given by the controlling computer. Thus, if the coordinate composition of an entire floor plan is stored, that floor plan can be reproduced using the x-y plotter. Since the position of most entities will also be stored, that position could be used as a locator for comments concerning entities not meeting requirements. Thus the form of the output could assume that of a computer-produced plan with diagnostic comments printed at the point of the exception. The added cost of the plotter might be offset by reducing the amount of time that the professional would otherwise spend locating exceptions on the plans.

The form of output may therefore assume virtually any combination of words and pictures desired and economically feasible, and is not a severe constrain on systems design. It is, however, the important link between man and machine at the conclusion of processing, and therefore deserving of careful evaluation during the actual system design.

Another aspect of output deserving discussion is the creation of permanent records of the data file and the resultant diagnosis. For these purposes, the machine can produce permanent magnetic tapes, paper tapes, or punched cards, any of which can be rerun or reinterpreted at any time in the future. If the data files can be used for other kinds of statistical analysis, their permanent storage on magnetic tapes may be desirable. This would leave the files in a form most efficient for reintroduction to the computer system, since the machine can "read" magnetic tapes many times faster than cards or paper tapes. In the event that a certain run was questioned, it would be a relatively simple matter to put the tape on the machine and repeat the total processing run.

If the probable future need was for referral to the results of the run, card or paper tape storage might prove more convenient, since the results can be reprinted without the use of a computer.

Systems Configurations

Prior to the advent of computer communications systems, it was necessary to think of a computer installation as a single entity, located in a single space in a single place. Current technology has removed this constraint; input and output devices can be any distance from the machine, in any location or configuration; computers of many varied sizes and purposes can now "talk" to each other, trade problems, share storage, and do many other things at any distance and in many combinations. The new technology of remote computer communications opens the way to a virtually limitless number of ways in which systems may be physically combined. This, of course, is an important consideration to any code system (such as Hill-Burton) which has decentralized activity, whether that decentralization involves installations across a city or a nation.

The particular combination of machines, Input-Output (IO) stations, and communications systems is dictated principally by operational requirements and economics. Also involved is time-sharing technology; single installations which could not support a computer can be equipped with IO stations linked by long-distance lines to a central computer, the cost of the computer then being shared by the several stations. (Such a system is the Project MAC system at MIT, where some thirty stations across the campus are time-shared from a central computer). The implications are that by such systems, computer capability is made available to persons who could not otherwise afford it. This potential capability is a necessary part of any system that would eventually link the designer in his office to a code system computer capability; it may in fact be the key to the introduction of computer technology to the entire design and engineering profession on an economically feasible basis.

While it is not possible to determine which configuration is economically feasible in the Hill-Burton case without detailed study, at least the following alternative appear likely:

I. BATCH PROCESSING SYSTEMS, Card Input

Central Facility: A central processing facility might be established, with computer processing time purchased or rented as economics dictated. Plans could be submitted indirectly or through regional offices, and results similarly returned directly or through regional offices.

Local Facility: Since this kind of a system requires little equipment investment, the input could be prepared locally and processed on a rented computer in the locale. There is not any particular advantage in a central facility, since batch processing systems would probably use a more efficient large to the cost, however, and are occupied the full time any station is "on-line", i,e., operating. There are also problems of equipment utilization. The batch processing system, using card input, is slow enough to provide considerable work loads for each regional office. The graphic-input systems are considerably faster; three of four stations would probably handle the entire work load. Thus if each of the seven regional offices were equipped with an IO station, many of them would not be fully utilized, and the systems costs would increase harshly.

III. Combination Systems: The cost of a medium-sized computer which both processes and provide IO control simultaneously appears to be somewhat prohibitive (see Systems Costs). This is principally because the equipment may be more than is needed for IO control, but something less than desirable for optimum processing efficiency. An alternative to a system which links the processing computer directly to the IO station is a system which employs a small intermediary, or satellite, computer. The function of the satellite computer is to provide IO control and perhaps some preprocessing capability; its output is a well-organized data file which is then presented to a larger machine for processing (probably on a rental-time basis). A single satellite computer could probably control up to four IO stations. (There are some problems which may be anticipated in such a system. While it is technologically feasible, there are no systems in use today which drive multiple display stations on a time-sharing basis; this programming would constitute a major piece of development). These stations could be remote from the satellite computer, as described above, economics permitting.

This latter kind of system appears, on the basis of this study, to be configuration deserving careful attention. It minimizes the investment in hardware which is certain to be obsolete in the future, but it still provides the important machine-operator communications of the on-line system. In addition, the preliminary estimates of cost in the following section indicate that such a system possesses inherent economies relative to the problem at hand.

SUPPORT OF STATEMENT 14: ESTIMATED SYSTEM BENEFITS

At the present time, the Hill-Burton Program supports a staff of 14 architects, 15 mechanical engineers, and 11 civil engineers. The annual cost of supporting these personnel in direct salary is $445,000. The overhead rate is $380,000 per annum for a total expenditure of $825,000 per annum.

A study conducted in 1959 by the Division of Hospital and Medical Facilities indicated 47% of the total hours spent by personnel were spent in review of drawings, specifications, and contract documents (Table 14-1). This would place the salary and overhead value of such work, on a pro-rated basis, at $355,000 per year.

Effectiveness of a computer-based system: Detailed analysis of the criteria inspected by architects, civil engineers and mechanical engineers have been made. Each of these subsets of the criteria have been broken apart into those which could probably be profitably done by the computer, and those which could not. For each profession, the number of hours spent in each kind of review work, was listed and correlated with the criteria which would be considered at that phase of submission. This procedure allowed estimates of how much of the work load might be transferred to the machine, at each phase of inspection, for each of the professionals involved. The results are tabulated in Table 14-2. The conclusions are, (a) the architect will benefit more than the engineer in relief of work loads (b) more of the work of preliminary and schematic inspections can be done by machine than for working drawings. This is primarily due to the fact that specifications and detailed mechanical drawings are not included in earlier submissions, and as the breakdown indicates (for reasons already stated), it is expected that it will be difficult to handle input of these kinds of data.

It must be kept in mind that these are highly approximate breakdowns made on the basis of the best information at hand.

If they are reasonably accurate, however, we can expect that one half of the inspection work load can be transferred to the machine system. This would relieve professional time valued at one half of the total time spent in reviewing drawings, i.e. ½ of $355,000, or approximately $175,000. This is the value of the major calculable benefit of the system.

TABLE 14-1

ESTIMATED MAN HOURS PER

PROJECT PER ACTIVITY, BY SKILLS

Source: Division of Hospital & Medical Facilities, Public Health Service

KIND OF WORK	SKILL			TOTAL HRS. KIND OF WORK
	ARCH	CE	ME	
Review Schematics	6	-	2	8
Review Preliminar.	6	2	4	12
Rev. Working Dwgs	12	3	12	27
Review Contracts	5	8	8	21
Subtotal	29	13	26	68
All Other Work	18	33	25	76
Total Hrs/Proj.	47	46	51	144
% Total Spent in Drawing Review	61	28	50	47

TABLE 14-2

ESTIMATES OF AMOUNT OF WORK

TRANSFERABLE TO MACHINE SYSTEM, BY ACTIVITY

	PRESENT SYSTEM	MAN-MACHINE SYSTEM		
WORK DONE	Man-hours invested per project	Profess. man-hrs. req'd	Equivalent man hrs. transferred to machine	% or work trans. to machine
REVIEW OF SCHEMATIC DWGS.				
ARCH.	6	1	5	
CE.	0	0	0	
ME.	2	1	1	
Subtotal	8	2	6	75%
REVIEW OF PRELIMINARY DWGS				
ARCH.	6	1	5	
CE.	2	1	1	
ME.	4	2	2	
Subtotal	12	4	8	66%
REVIEW OF WORKING DRAWINGS				
ARCH.	12	2	10	
CE.	3	1	2	
ME.	12	9	3	
Subtotal	27	12	15	55%
REVIEW OF CONTRACT DOCUMENT				
ARCH.	5	3	2	
CE.	8	6	2	
ME.	8	7	1	
Subtotal	21	16	5	23%
TOTALS	68	34	34	50%

Many of the potential systems benefits are not measureable in terms of dollars value received by the system. These benefits include:

1. <u>Accuracy and Completeness</u>: Whenever the system is used, criteria are checked completely and in a consistent and accurate manner.

2. <u>Improvement in Analytic Capability</u>: More sophisticated analytic programs, such as traffic simulations, are possible with the computer in the system. Further, the use of performance-type criteria, which may require the use of complex predictive models in evaluation, will be facilitated by the computer's ability to handle such complex evaluations on a production basis.

3. <u>Improvement in Records</u>: Complete data files on every hospital design submitted are accessible for computerized analysis; further, computerized storage systems can provide quick access to data on particular designs for use in operational or consulting services.

4. <u>Relief in skill shortage:</u> The system becomes less dependent upon the availability and quality of professional persons as the limiting factor on growth and performance of the system.

These benefits, which are not measurable in terms of dollars, provide insight into the true benefits of any computer system.

The computer rarely saves money; it creates its place not by direct economy but by changing the potential of any operation. By opening new opportunities, the computer simply makes it possible to do the job in a different way; frequently better than it has ever been done before, not just more cheaply.

SUPPORT OF STATEMENT 15: SYSTEMS COSTS

It is extremely hard to develop reliable cost estimates for computer systems, since (a) the cost of alternative hardware configurations is highly variable for comparable machine systems, and (b) the cost of programming is difficult to predict. This study nonetheless will attempt to develop the best possible estimates for the range of configurations which might be proposed. This is done with the realization that decisions to support the development of such systems must be made with some idea of the potential costs and benefits. While these estimates are provided to satisfy that need, they cannot be in any way considered as actual costs or proposed costs for the systems. Such costs can only be determined in fact through the submission of development proposals from industry in response to detailed system specifications.

The basis of the estimates herein provided are estimates developed by the Institute staff and confirmed by Mr. Pat Coyle, computer applications consultant, Washington, D. C. These estimates were developed for two systems. The first system is a batch processing system which minimizes the use of computer equipment and its purchase. This sytem presumes all input is developed on cards, through the use of languages and format as described in the report of Fair, Isaac and Associates, San Rafael, California*. The hypothetical system

*CLASP: Computer Language for Architectural Specifications, a report to the IAT by Fair, Isaac & Associates, 1400 Lincoln Ave., San Rafael, California.

uses rented processing on an IBM 7094 or equivalent machine (The 7094 rates are herein used due to the fact that the FI work is based on LISP, a processing language currently available only for the 7094 and a few other machines). Output from the system is in simple printed form; no graphic input equipment is used.

In contrast to this, estimates were developed for a second system employing man-machine interaction. This second hypothetical system employs graphic input-output stations operating in conversational mode with the operator; these station are time-shared off a small satellite computer, such as the DEC PDP-8, CDC 1700, IBM 1800 or similar machines. Processing is done from both files prepared by the satellite computer on a medium or large rented computer facility. The system presumes a basic operating configuration such as that suggested by Bolt, Beranek and Newman in their report to the Institute *.

Both systems represent basic machine configurations capable of accomplishing the amount of work described in the previous section. They dod not include consideration of remote

*Report #1260, Computer Aided Checking of Design Documents for Compliance with Regulatory Codes, Bolt, Beranek and Newman, Los Angleles, California installations.

The major costs of the systems were identified as:

1. Program development, including changeover, training, administration of development contracts, preparation of proposals and so forth.

2. Development of computer programs.

3. <u>Acquisition of computer installation</u> on a purchase basis.

4. <u>Annual operating costs</u>

   (a) rented processing

   (b) operating staff

   (c) overhead.

1. <u>Program development costs</u>: A sophisticated system of input-output hardware would be considerably more difficult to secure than a system with conventional hardware. First of all, the problem of writing system specifications is considerably more difficult; second, there is no existing system with the capabilities described operating on a production basis, and some additional development and design will be required; third, the changeover and training requirements will be more difficult, due to the need to train a new kind of operative; fourth, because the system is more complex, a somewhat longer time may be required to achieve full operational capabilities. Program development should range from a minimum of $40,000 to a maximum of $60,000.

2. <u>Development of Computer Programs</u>: If the more conventional batch processing approach with none of the on-line programming options discussed is selected, the programming would consist of adapting one of many existing list-processing languages to the problem at hand. On the other hand, the development of on-line time-sharing graphic input-output systems has not yet been fully accomplished,

and a great deal of additional and new programming must be done. The best expert opinion says that it can be done, however, and it is simply a matter of time and opportunity to assemble the required pieces. The major concepts of the approach have been demonstrated in single or combined applications; they have simply not been assembled into a single operative system. Programming a standard list-processor might be accomplished for as little as $10,000; programming an on-line time sharing system could run to $50,000 for basic capabilities.

3. Acquisition of Hardware: The batch-processing techniques with minimal equipment, i.e., everything translated to and entered on punch cards or paper tape, would require only the acquisition of simple and inexpensive keypunch machines. The volume of processing is not such to warrent the purchase of a computer. All processing would probably be done on rented or shared equipment. On the other hand, the more sophisticated system requires the acquisition of from two to four input-output stations at substantial costs (these are estimated at about $26,000 per station) plus the acquisition of a small computer to perform online-input-output functions and some preprocessing* (such

* Such as conversion of coordinate point data to required areas, construction of lists, etc. The main processing in this system would also be accomplished on a rented-time basis.

computers might cost about 60-80 thousand dollars). Thus an investment of about $160,000 in hardware could be expected for the second system, as contrasted to no investment in the first system. The cost of this hardware should be amortized over its expected life; we have chosen to estimate this life span at five years, at which time the equipment may be expected to be obsolete and replaceable.

4. <u>Operating Costs</u>: It is here that major differences in the systems appears, primarily due to handling of input operations. The process of writing down series of numbers to describe the name, location and sizes of rooms and/or equipment, and then keypunching the results does not, by any stretch of the imagination, come close to the efficiency of the on-line direct entry of graphic and textual information. Detailed estimates of the input times have been prepared by undertaking practice exercises on each type of system, i.e., simulations of the input operations.

Fair, Isaac and Company report that data can be entered on appropriate forms at a rate of entry of 30 items per man-hour; this would presume an "item" of information could be located, read or measured, and written down in an average time of about 90 seconds. Each item would represent one punched card input. It is estimated that use of the proposed system would require 1100 punched cards to describe the information on schematic drawings; 1500 cards for preliminary drawings, and 2100 cards for working drawings.

The current submission rate for drawings is 300 schematics, 400 preliminary, and 500 sets of working drawings per year. We must presume the worst case, that is, that each submission must be input as a completely separate item, disregarding what came before in previous submissions. This is due to the fact that it is expected to be as difficult to update a file reflecting changes as it is to completely resubmit the entire project.

From these estimates about rates and volumes, it was estimated that the total man-hours required for writing up the required forms would be:

schematics:	4.2	man-years/yr.
preliminaries:	7.6	" " "
working drawings:	13.4	" " "
total	24.2	" " "

This same volume would demand a card punching work load of some 9.9 man-years/year, given the rather optimistic keypunching rate of 100 cards per hour.

With regard to the skills required to operate the system, it was estimated that a minimum GS-7 skill level would be required to read and write the forms (the current GS-7 rate is approximately $6000 per annum). This assessment was based on some detailed analysis of the skill required to perform basic required functions, such as computations, and the skill level required to interpret architectural drawings accurately, without making judgments about them. The standard rate for keypunchers is approximately GS-3 at $4000 per annum. The total labor costs in this system for developing input appear, on this basis, to be approximately $190,000 per year.

Contrasted to this, the on-line graphic input system allows a substantially higher input rate. It is estimated, based on experienced data transmission times, that the total time spent by a single operator at a console to input the data required would be:

schematics:	8 hours
preliminary drawings:	12 hours
working drawings:	16 hours

The resultant work load, based on an equal number of submissions, appears to be as follows:

schematics input:	1.2
preliminaries input:	2.5
working drawings input:	4.1
total ..	7.8

or approximately 8 man-years/year. This is assuming the same quantity of data transmitted, and all other factors equal, except the mode of data transmission. It is interesting to note that, even given major errors in the estimates, there would still be substantial margins between the two methods.

It is also evident that while not as much knowledge of plans and specifications would be required by the on-line system, due to its potential ability to instruct the operator, that some increased sophistication would be required to operate the console and to know its capability. Therefore, it is assumed that a GS-7 skill level would also be required to operate this system. Of course, additional card punch operators are not required, since the data is entered on-line. Thus the manpower costs of such a system appear to be based on about 8 men at $6000 per year, or about $48,000.

From calculations of this input load, we can also estimate that if the above estimates are correct, a minimum of 4 input-output stations, operating two shifts, would be required. This in turn would necessitate the addition of "time-sharing" capability on the small computer operating the stations; that is, the computer, by dividing its attention among the four stations, services them all at the cost of a margin of speed at any one of them. (This would be much like a mathematician sitting in a room with four persons who are each continuously feeding him problems; he works on one for a short time, stores it, and goes to the next, and so on, rotating through the problems at hand in the order that he receives them.) This can be accomplished on a computer with speeds such that it appears to the individual operator that he has the full attention of the machine. This time-sharing capability constitutes one of the major portions of the programming to be done in developing such a system.

Both systems will have operational problems which will require changes, improvements and modifications in the programs in the course of the use of the system, and in order to provide for such changes, the cost of a programmer has been included in the costs of both systems. A full time programmer is allowed for the on-line system, since there will be more opportunity for change and expansion in the system; a one-quarter time programmer has been provided for the batch processing approach, which will perhaps require less service once fully operational. The total of these estimates are summarized in Table 15-1. It should be emphasized that these estimates are not firm, and have been formulated on a very tentative basis in order to attempt to fix some costs for the purposes of the feasibility study. They should not be considered as having been confirmed in any way. The final figures of any bid proposal may vary greatly from these figures, although they are considered reasonable target costs.

Overhead is added at 50% of the items noted. This is based on the assumption that the direct addition of overhead will not equal the average organizational rate, since the equipment needed by the new workers is by and large included in the basic costs.

The costs for rented processing are based on estimates of volume, time and efficiency of various programs on what was considered to be an optimum machine for each usage.

TABLE 15-1: SUMMARY AND COMPARISON OF SYSTEM COSTS

ELEMENT OF COSTS	BATCH PROCESSING BASIC SYSTEM		ON-LINE GRAPHIC INPUT SYSTEM	
	TOTAL COST	YEARLY COST	TOTAL COST	YEARLY COST
1. Program Development	30,000	6,000	60,000	12,000
2. Software for Computer	10,000	2,000	50,000	10,000
3. Computer Equipment			170,000	34,000*
4. Labor				
a. input operators @ GS-7, $6050pa	25 req'd	151,300*	8 req'd	48,400*
b. keypunch @ GS-3 $4005pa	9 req'd	40,000*		
c. program service a GS-12, $10,250pa	¼ time	2,500*	1 req'd	10,250*
5. Rented Processing		109,800		19,600
6. Overhead on starred items @50%		96,900		46,300
TOTALS	40,000	408,500	280,000	190,500

SUPPORT OF STATEMENT 16: CHANGES IN OPERATIONS

The major implication of change is that the processing of plans could conceivably shift from a regional basis to a centralized basis. The need for such change is by and large dependent upon the costs of long-distance data transmission, and the corresponding premium the program would be willing to pay to maintain regional processing and have the use of computer systems. One feasible way of organizing the system, however, is to consider the submission of duplicate sets of plans to the regional office and to the central processor. Processing completed, the diagnosis could be transmitted through a teletype to the local office, thus assuring the fastest possible reporting of results. If it were desired that the plotter be used to print graphic diagnoses as discussed in earlier sections, a plotter could be located in the regional office, driven by the computer over LDC lines, or else the plotted results could be transmitted by facsimile device to the regional office.

A major constraint in operations will be the loss of the professionals "direct contact" with the inspection process. The effectiveness of the entire program will depend upon his ability to trust the machine to perform an effective diagnosis. If he minimizes his activity to basic familiarization with plans and the inspection of reported errors, the most effective relations will be maintained between man and machine. If however, he intuitively distrusts the machine, or resents the loss of prerogative, or for any other reason decides to recheck what the machine is supposed to have checked, the benefits of the system can be lost entirely. Thus the effective operation of the system is in large part dependent upon the professionals understanding and acceptance of the machine system. The professionals in the program should be given the opportunity to work with the computer system directly in order to understand what it does and what it does not do, and in order to build confidence in it and in the relation. This required development of attitude can be as important to operations as any other operational change in policy or program.

Some professionals will probably have to be assigned to superintendance of the system on a full-time basis. These persons will be required to augment the judgment of the operators, to assist them with difficult problems, and to assist in the operations of special cases. The machine may encounter difficulties in examining some plan configurations, such as free-form designs, or it may require the insertion of a specially programmed criteria for special instances. The professional skills will be most critical at this point. Further, if the system capability is to be expanded and fully realized, a professional who understands the intention of the organization must be called upon for opinions and decisions as to which options should be taken in which order of priority. Another important operational relationship is that of the program to the state health agencies and to the local architect and client. Expansion of the system to its fullest theoretical potential will depend in large measure on the willingness of these two other sets of persons to accept changes in procedure and practice.

The operations and organization of this total system (architect, state and Hill-Burton) were considered, at the outset, to be the total system amenable to systems design. In fact, the whole problem of data input, which looms so large as a factor of feasibility in this study, could be avoided if the system (i.e., the larger system just mentioned above) could be considered as open to redesign. The proper design of such a total system would clearly consider capturing the information at its point of origin and in its original form, instead of involving itself with complex and wasteful translations of information at some intermediate point. In this case, the point of origin of information is when the architect decides to record a decision which will at some time interact with the criteria. If a total system could be designed so that the computer system intercepted this information during the process of recording it, it could be routed directly into the code system for consideration. The whole data translation problem simply would not exist.

This in effect requires "slipping the computer under the drafting table" on which designs are made, or, in back of the typewriter on which specifications are written. No immediate capability to do this in an economic way seems to exist.

Even if architects could be induced to use peripheral equipment in certain work, e.g., the writing of specifications or selection of equipment, we have not yet solved enough problems of basic capability to allow the designer to do all the things he needs to do, and frequently the use of such equipment would be limited and inefficient. In other areas, the process of design is simply not well enough defined to use the major capabilities, e.g., simulation and optimization routines, of the computer system. Even the graphic capability of computer systems is not yet well enough developed to afford capabilities to the architect which pay for themselves.

Thus the insertion of such equipment into the architects routine would by and large have to be subsidized by the interested party until such time as it demonstrated a payoff to the architect. At that time, the architect might be willing to purchase equipment of his own which could be connected to the code system. But until then, the costs of subsidy would be rather astronomical when considerations of training and logistics are included. Nonetheless, there is a recommendation in the main body of the report that experimentation could be promoted with the architect at a later date. Such experimentation might make the code system equipment available to him on a spare time basis to explore its capability to help him with design work, and this could in turn be useful in speeding the development of general purpose design systems.

All of these problems of present and future operational patterns must be fully and carefully considered in the preparation of a specification for a system. For this reason, as well as for others, it is important that the team actually in charge of developing the project include key personnel from the Hill-Burton program who can advise on the logistics and operations, and be prepared to institute and justify the changes in operations which will be required.