DOCUMENT RESUME

ED 050 797 LI 002 860

AUTHOR Korfhage, R. R.; DeLutis, T. G.
TITLE A Basis for Time and Cost Evaluation of Information Systems.
INSTITUTION Purdue Univ., Lafayette, Ind. School of Industrial Engineering.
SPONS AGENCY National Science Foundation, Washington, D.C.
REPORT NO RM-69-6
PUB DATE Jun 69
NOTE 47p.; Presented at the Sixth Annual National Information Retrieval Colloquium, May 8-9, 1969

EDRS PRICE EDRS Price MF-$0.65 HC-$3.29
DESCRIPTORS Computers, Cost Effectiveness, Costs, Evaluation, *Information Retrieval, *Information Storage, *Information Systems, Marketing, *Models

ABSTRACT

A general model for information storage and retrieval (IS&R) systems is proposed. The system is selected from the set of all available IS&R components. These components define the system's users and data sources; the hardware, software, and personnel performing the actual storage and retrieval activities; and the funder who acts as a filter in the selection of the components comprising the system. The outermost level, the ectosystem, includes the funder, and user groups, and the data sources. All other levels belong to the endosystem. Each level within the endosystem consists functionally of processing components and storage components. It is proposed that a measure of the endosystem performance, as a function of the time to complete a requested service can be obtained by specifying the characteristics of these functional components for a specific hardware/software configuration and the characteristics of the user groups. Finally, the selection process for all system components is related to the performance of the endosystem, and to the cost for this level of performance as a function of the ectosystem. (Author/MM)

A BASIS FOR TIME AND COST EVALUATION OF INFORMATION SYSTEMS

R. R. Korfhage *
T. G. DeLutis +*

Research Memorandum No. 69-6
June, 1969

* Department of Computer Sciences
+ School of Industrial Engineering

Purdue University
Lafayette, Indiana 47907

This research supported by the National Science Foundation, Project GN-759 in the School of Industrial Engineering, Purdue University, Lafayette, Indiana 47907.

Presented at the Sixth Annual National Information Retrieval Colloquium, May 8-9, 1969.

ABSTRACT

This paper proposes a general model for information storage and retrieval, IS&R, systems. In the model an IS&R system is visualized as distinct levels of functional components where each level looks upon the totality of all lower levels as a black box which accepts inputs and returns outputs. The system is selected from the set of all available IS&R components. These components define the system's users and data sources; the hardware, software, and personnel performing the actual storage and retrieval activities; and the funder who acts as a filter in the selection of the other components comprising the system. The outermost level, the ectosystem, includes the funder, the user groups, and the data sources. All other levels belong to the endosystem.

Each level within the endosystem consists functionally of processing components and storage components. It is proposed that a measure of the endosystem performance, as a function of the time to complete a requested service, can be obtained by specifying the characteristics of these functional components for a specific hardware/software configuration and the characteristics of the user groups.

Finally, the selection process for all system components is related to the performance of the endosystem, and to the cost for this level of performance as a function of the ectosystem.

Parameters for Evaluation of Information Systems

In all the literature on information retrieval systems, there is one distinct gap. While many writers discuss the evaluation of such systems, in virtually all instances, "evaluation" refers to such concepts as precision, relevance, and recall. Discussions of cost and time are difficult to find.[1,2]

In hopes of filling this gap to a small extent, we present a model of an information system which is designed to bring out the problems of cost and time, and to facilitate evaluation of a system from this economic aspect. The model which we develop is, in fact, suited to almost any computer-based information processing system, whether it is doing retrieval work or scientific data reduction. The particular application of the system is buried in the center of it, within a service module.

It is convenient to divide the universe into three pieces: the endosystem - the combination of computer and communication equipment, programming support, and personnel necessary to accomplish the given storage and retrieval task; ectosystem - that portion of the universe which directly affects the endosystem, including users, data, sources and financial support; and the environment - everything else.

For our purposes, the endosystem must

1) store and maintain data suitable for retrieval by computer,
2) provide a set of services to users inquiring about the stored data or about the services themselves, and
3) establish and maintain interfaces between the endosystem, the user group, and the data sources.

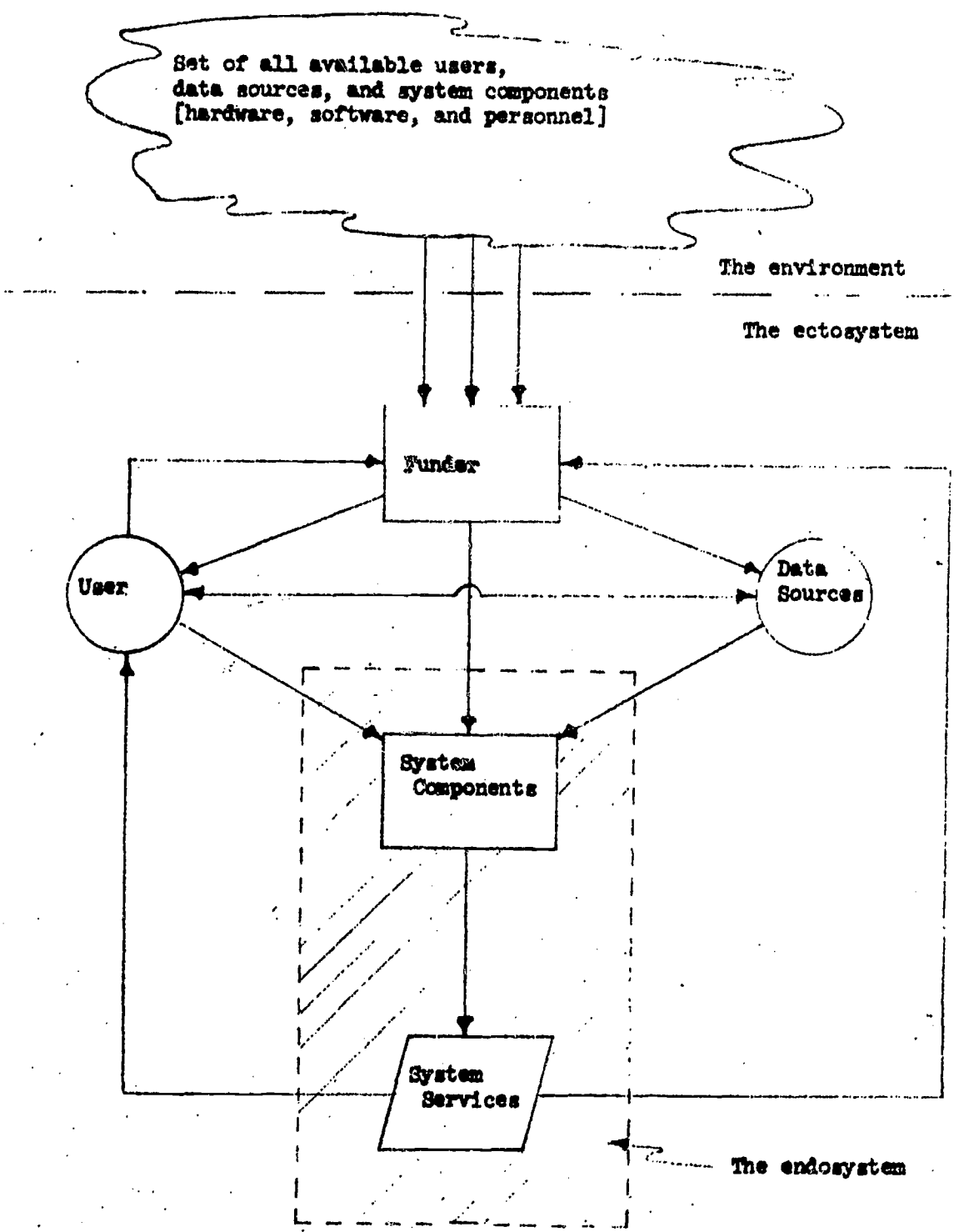


Figure 1. The System

$\underline{s}$ - the set of services provided, $\underline{u}$ - the set of user groups in the ectosystem, and $\underline{d}$ - the set of data sources. Note that the variable P pertains to the endosystem, while the parameters s, u, and d govern the ectosystem.

With the design level properly defined, the system designer should (in theory) be able to determine from the funder his willingness $\underline{W}$ to fund a system operating at any given design level. Presumably the function W(D) is monotone increasing up to some absolute maximum (Fig. 2).

The filtering action of the funder is a result of the interaction between his willingness curve and the cost of endosystems which will meet the various design levels. For the endosystem the optimal cost curve $\underline{C_o(D)}$ will also be monotone increasing, crossing and recrossing the curve W(D) until finally the cost exceeds the absolute maximum of W. The curve $C_o(D)$ will probably not be as smooth as W(D), but will include discrete jumps where new pieces of hardware (or software) must be added to increase the design level (Fig. 3). The funder's filter then passes any design level for which $C_o \leq W$, and rejects any design level for which $C_o > W$.

In practice, the optimal cost curve changes with time, as new hardware and software are developed. At any given time, this curve must be approximated by studying the costs and capabilities of existing systems. Given a design level D, choose an endosystem S which will meet the design level. If the cost $C_S(D) > W(D)$, the system is infeasible at that design level. However, it may be possible to operate the endosystem at higher design levels, D' through D", where $C_S(D') \leq W(D')$. (See Fig. 4a) Similarly if, for another system S', $C_{S'}(D) \leq W(D)$ at the original design level D, then

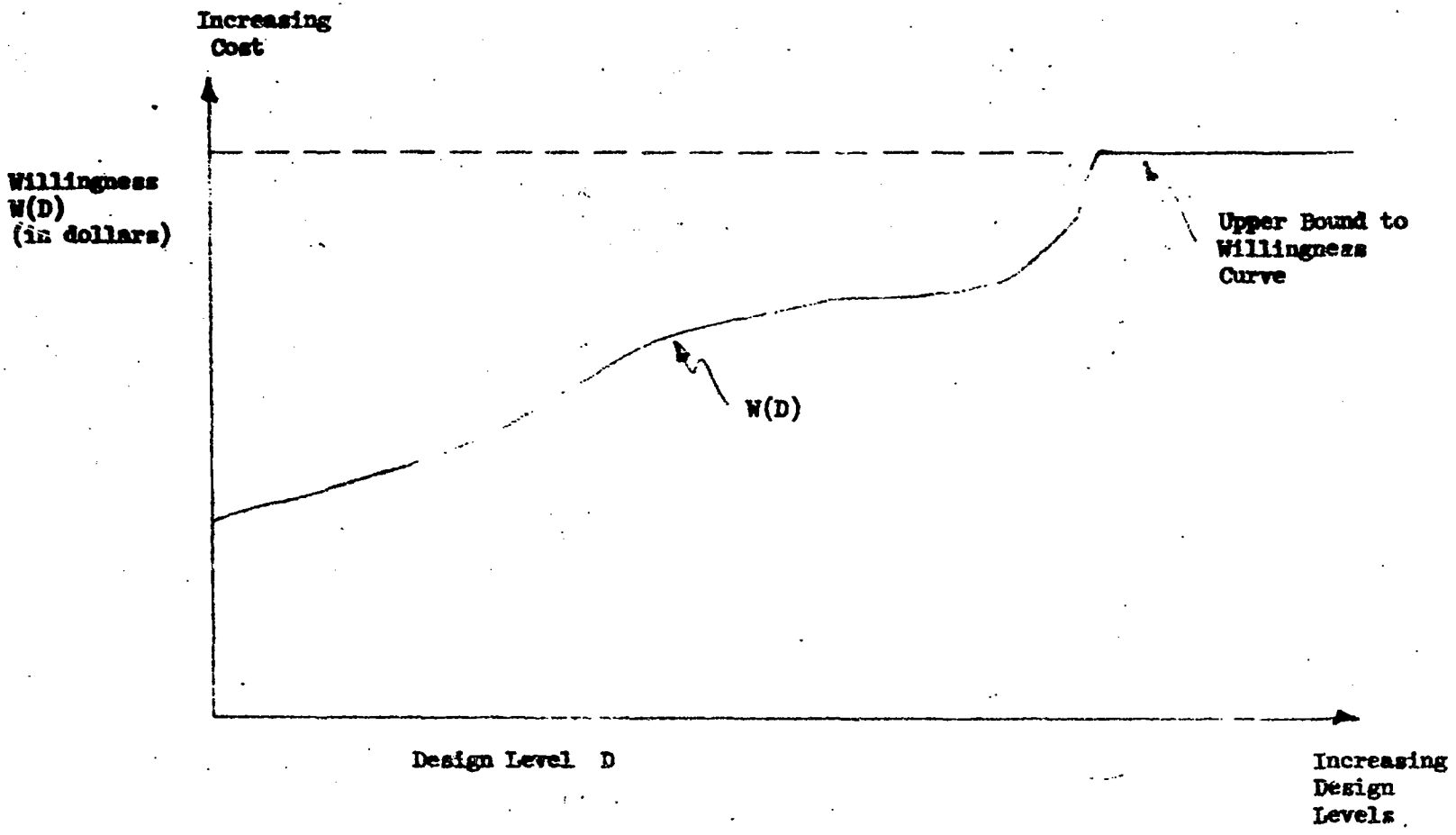


Figure 2. Funder's Willingness to Incur an Expenditure

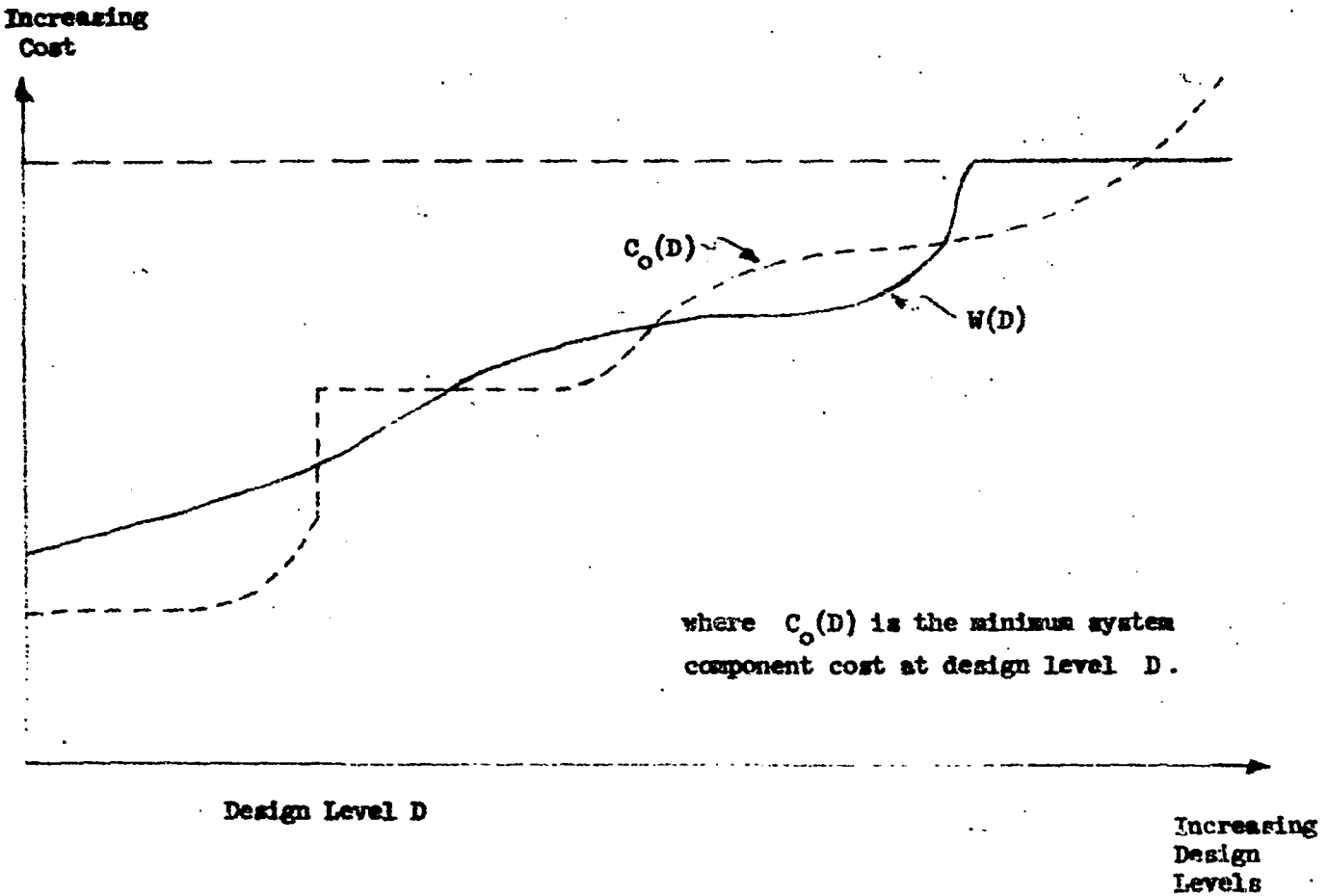
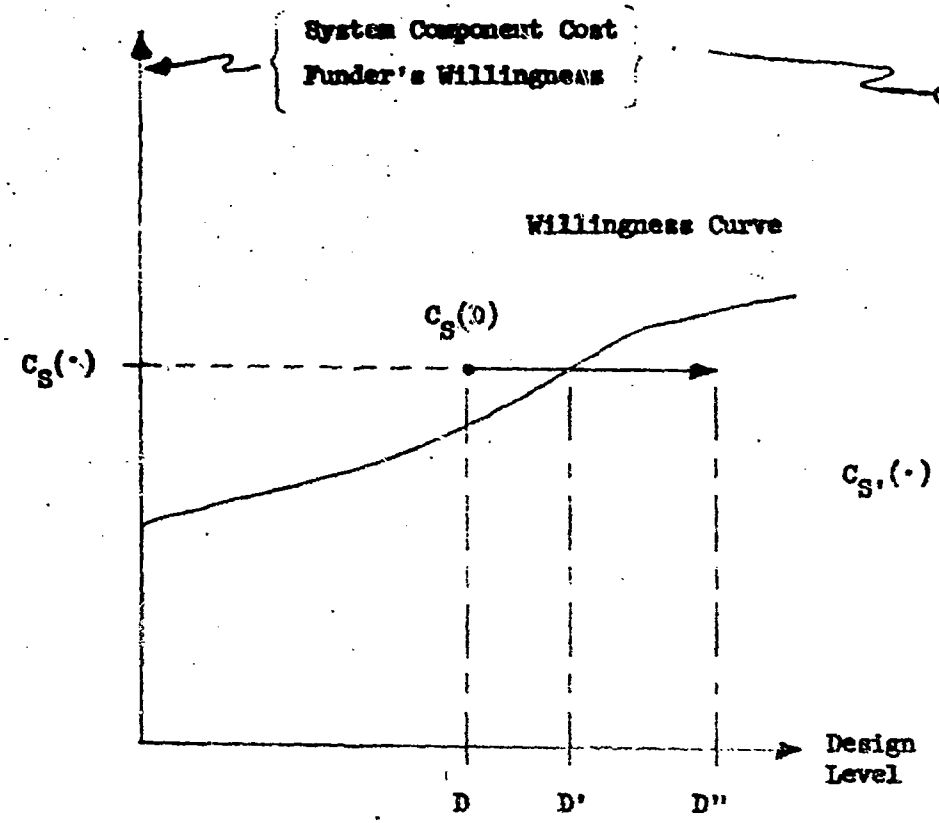


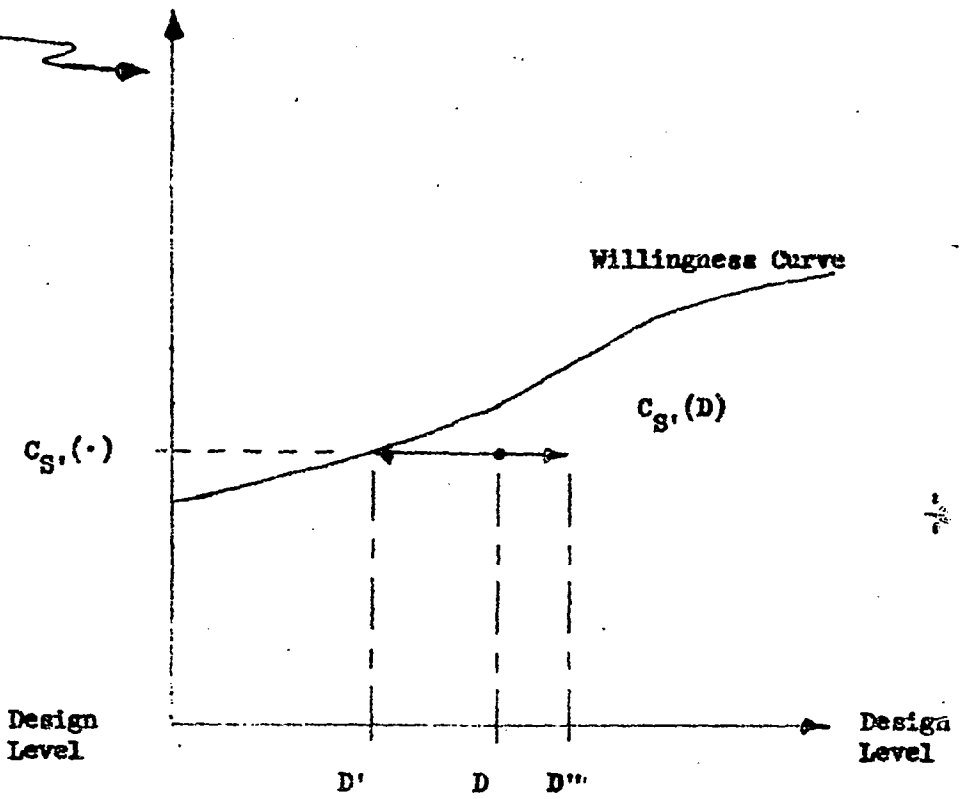
Figure 3. Component Cost and Willingness to Accept the Expenditure



$$C_S(\cdot) = \{D:C_S(D) \leq W(D)\}$$

$$= \{D:D' \leq D \leq D'' \mid S\}$$

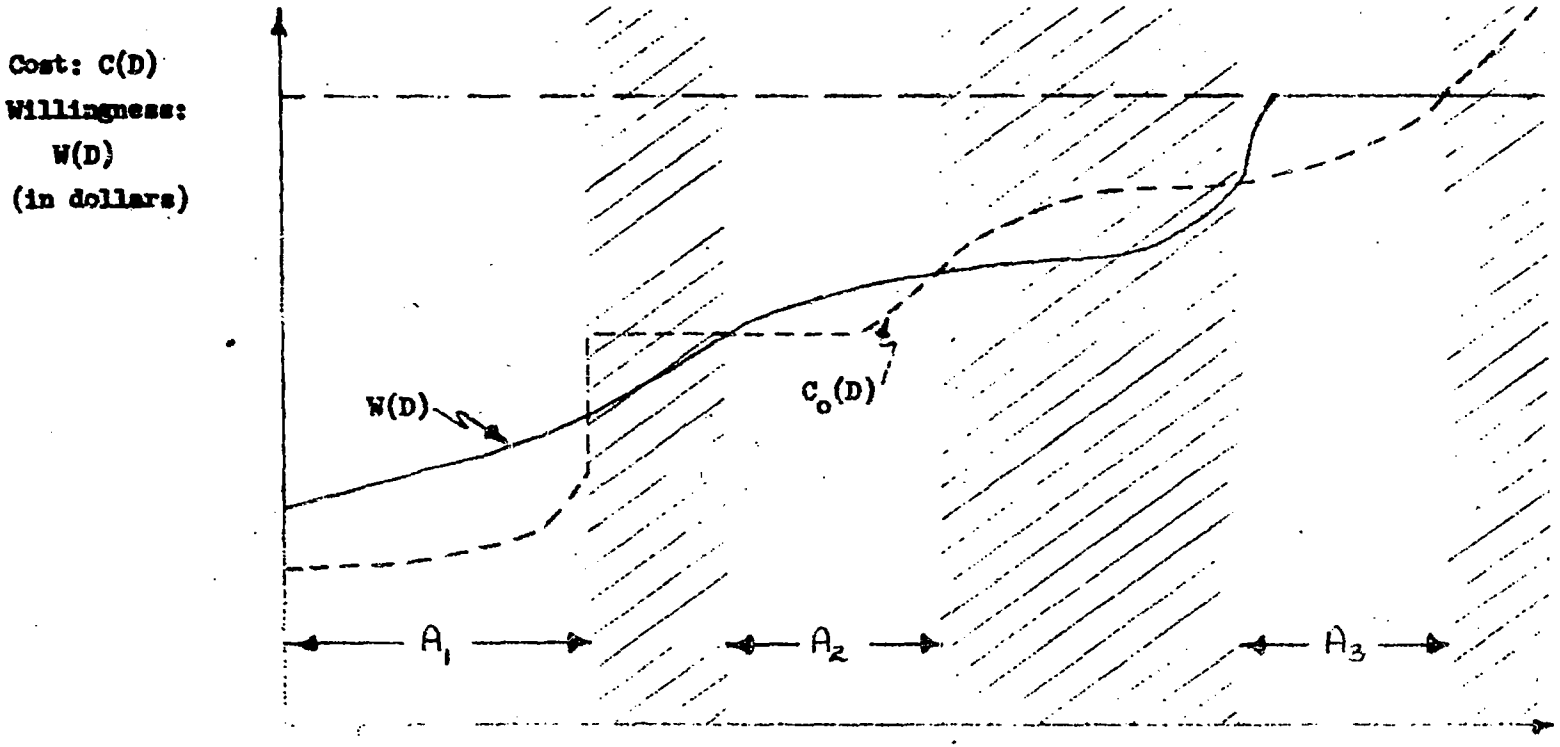
Figure 4a. Acceptable Design Levels for Component Configuration S



$$C_{S'}(\cdot) = \{D:C_{S'}(D) \leq W(D)\}$$

$$= \{D:D' \leq D \leq D'' \quad S'\}$$

Figure 4b. Acceptable Design Levels for Component Configuration S'



Where: P ≡ performance level = $P(S,t_s)$
S ≡ satisfaction independent of time
t_s ≡ time to provide service s
s ≡ services; u users; d data sources
A_i ≡ regions of design levels acceptable to funder

Figure 4c. Acceptable Design Levels

the endosystem S' is feasible at the initial design level. Again in this case, one can determine the range of design levels D' through D", both higher and lower than th initial level, for which the endosystem is feasible (Fig. 4b).

Thus each specific endosystem studied helps to define the feasible ranges of design level. Carried to its limit, this process defines the optimal cost curve, and hence the design levels which the funder is willing to support, given the constraints of available hardware, software, and personnel. (Fig. 4c). In practice, the optimal cost curve is rarely found because of the time and effort involved.

The user. The user set is thought of as a collection of user types rather than the number of persons who will actually use the system. That is, at this stage the variety of users rather than the volume is of prime importance. The design of the endosystem must conform reasonably well to the characteristics of the user set if the entire system is to function efficiently.

The user characteristics will influence the services required of the system, most noticably in the types of services provided, the frequency of use of the system, and the degree of user/endosystem interaction demanded.

Similarly the user set will influence the data store, not only in the type and volume of data stored, but also in the indexing scheme used for the data, and the system effort required to inform the user of this indexing scheme. As far as the user is concerned, the form of the data store is irrelevant, although from the endosystem's viewpoint, the type and volume of data, and the indexing scheme are all related to the hardware and software of the data store. Thus, the user exerts a direct influence on the ectostore of data, and a more indirect influence on the endostore.

In addition, the user strongly influences the performance of the syste through his satisfaction or dissatisfaction with the services provided, the time lapse between query initiation and information receipt, and the cost. This influence, perhaps more than the others, is felt indirectly through the funder, since the funder's willingness to pay for a system is often dependent on the users' acceptance of the system.

The indirect influence of the user set will be felt not only through the funder, but also through the data sources. The users' interests will to a large extent dictate those data sources which are included in the ectosystem. Further, it is reasonable to assume that the user will modify and extend these sources through his use of information from the endosystem.

The Data Sources. The operation of the endosystem is influenced by three characteristics of the data sources: the volume, the difference between the external and internal representation of the data (its ectoform and endoform), and the immediacy of the data source.

The volume of data clearly affects both the storage requirements and the processing time in the endosystem. These two factors are also affected by the data formats. Data sources which provide strictly formatted data demand little processing time, provided that the format is one which the endosystem uses. Similarly, unformatted data will demand little processing time upon entry to the endosystem if the system is designed to make use of such data. However, it will often require more storage, and more processing at the retrieval stage than will formatted data. Finally, data sources which provide unformatted data to an endosystem which uses formatted data demand a considerable amount of system time for data conversion, and often some temporary storage during the conversion process.

These remarks on the effect of data format should also be considered as applying to the process of indexing and analyzing data. Such a process may be thought of as creating a formatted data record to be appended to the given (formatted or unformatted) data.

Data sources may be classified broadly into three types, depending on their immediacy: the batch stream, the on-line stream, and the real-time stream. Each of these streams have characteristics which cause varying degrees of interference between the system services provided for the user set and the data collection functions associated with data stream processing.

In the batch stream, data collection and presentation is periodic. Updating of the information store involves the alteration of a relatively large volume of data, and may be a rather complex process. This typically happens in a document retrieval system. New documents are added (or old ones modified) on a daily, weekly, or monthly basis; and the addition of a new document may require a rather lengthy content analysis. Such a data stream has low immediacy, and hence a low priority among the various claimants for service. However, once initiated, updating consumes essentially all of the system capability until it is completed.

At the other end of the spectrum, the real-time data stream has absolute immediacy. Found typically in monitoring systems (e.g., for industrial processes or hospital patients), the real-time stream consists of data which must be processed immediately or lost forever. The immediate processing which is required may be quite trivial and hence not cause much interruption in the other work of the endosystem. Nevertheless, it takes priority over every other function, and hence materially affects the system performance.

Intermediate between these two types of data streams is the on-line stream found, for example, in banking and reservation systems. In this situation, the entry and updating of data is treated on a par with the user services, and generally occurs thoroughly intermixed with the other system functions. For such an input stream the acquisition and updating of the data can be delayed while another system service is being performed. However, the delay is short when compared to the time lapses involved in processing the batch data streams.

The performance of the endosystem is dependent on the software and hardware effort required to process the data stream, and on the interference of this stream with the other system functions. In the batch stream this interference is minimal since data entry and updating can be scheduled during periods of light user loads. However, the delay in updating data will cause some deterioration in the quality of the system output, which must be taken into account in measuring the performance. For the real-time data stream, the effect on performance is quite different. Accurate up-to-the-minute data enhance the quality of the system output, but at the cost of repeated interruptions of the user functions.

Finally, the actual cost of data entry and updating will depend also on the processing which is necessary to put the data into useable form. As discussed above, if the processing involves much formatting or data analysis, this cost can be quite high; while if the data enters the endosystem in a form which is directly useable, the cost is a minor factor in the overall evaluation of the system.

The Endosystem. We have discussed the funder, the user set, and the data sources - the components of the ectosystem which affect the performance of the endosystem. Since these are elements of the ectosystem, they are not totally under the control of the system designer, who must generally accept assumed values for these elements, and work within the constraints thus imposed.

The endosystem, however, is under control of the designer. The various parameters associated with the hardware, such as cost, timing, capacity, and environmental demands, are well known and must be considered in any system design. To a lesser extent, similar parameters are known for the software and for the operating personnel. The designer must estimate these to the best of his ability, and design the endosystem accordingly. This is simpler for an information retrieval system than for a general multiprogramming system since a fixed, well-defined group of programs will be run.

From the outside, the endosystem can be viewed as a black box accepting inputs generated by the user set and by the data sources, and returning services to the user. The time interval required to provide the services desired of the system is a function of the components contained within the black box. Performance and cost of the system are determined largely by the components selected. By studying the interaction of these components under a variety of assumed activity loadings for the system, the designer can estimate the capabilities of the endosystem. Further information about the endosystem can be obtained by varying the components within it and observing the changes in anticipated performance.

The model which we propose for the endosystem has three classes of components: processors, which carry out the functions of the system; task storage, which is transient and fractionated; and data storage (Fig. 5). Although the task storage may take a number of different forms, we shall refer to them all as queues, and assume that in general there are two task queues associated with each processor - the queue of tasks waiting to be worked on, and queue of tasks being processed but currently in suspension for some reason. It is assumed that each phase of processing takes a fixed time, and that all time delays are accounted for within the various queues.

It is also assumed that a supervisor operating within the central processing facilities exists. Its functions include the maintenance of communication between the processing modules, and the resolution of conflicts between the processors when two or more of them are contending for a particular hardware resource. This supervisor, which is transparent to the user, may be considered as part of the hardware, although some consideration must be given to the overhead penalties incurred with different supervisors.

For the analysis of the endosystem, the model is examined layer-by-layer, at each stage taking into account all interactions between the module currently under examination and those modules previously examined, but considering the unexamined inner layers as a black box. Time and volume distributions are assumed for the inner layers, and the reaction of the portion of the system under investigation is observed.

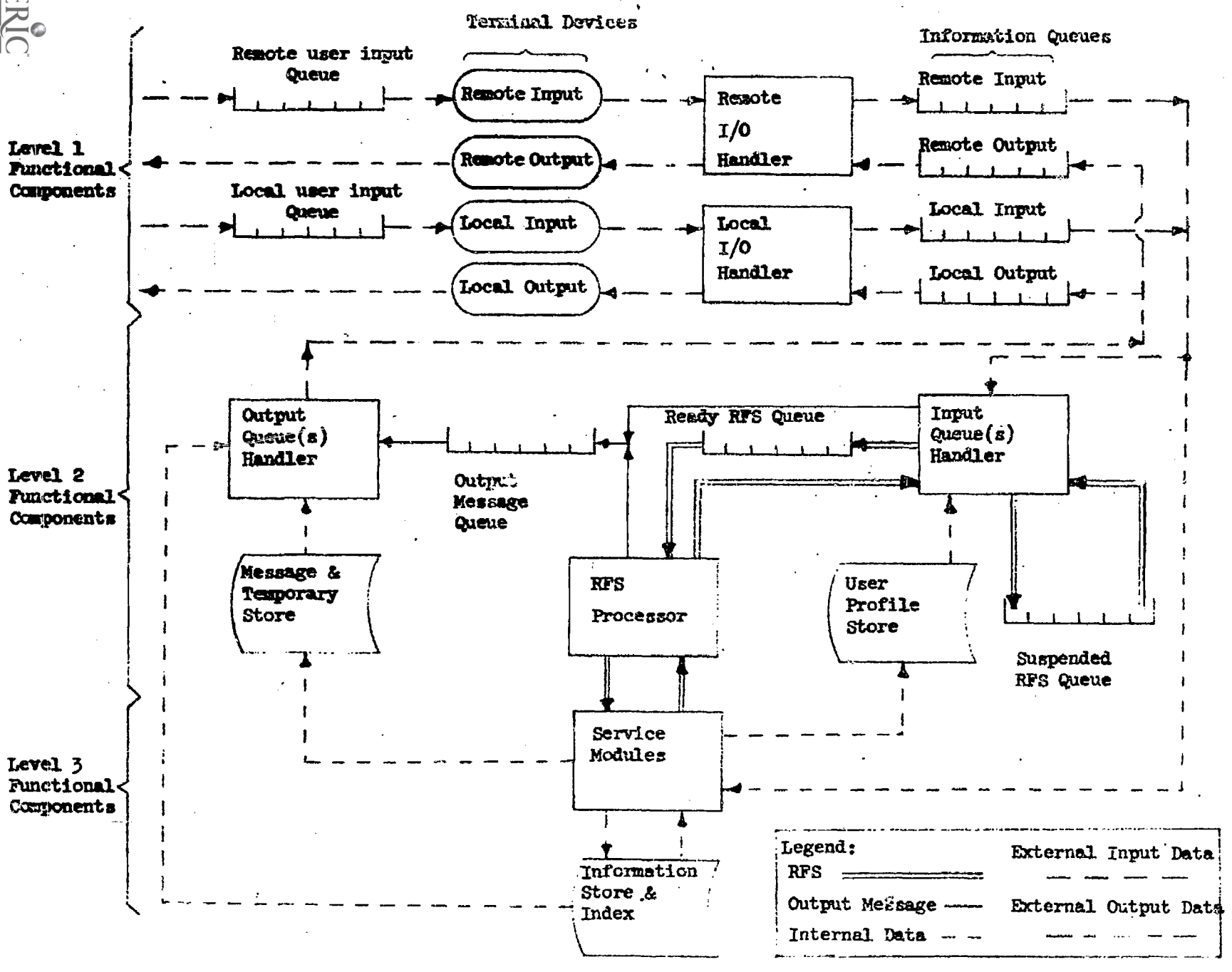


Figure 5. Endosystem Functional Components

Thus, as the first stage in an analysis we consider only the terminals, the I/O processors and their associated queues (Fig. 6). These processors interface between the user and the data source on one side, and the remainder of the endosystem on the other. Thus they must handle activity originating on either side and flowing to the other side.

In the figure, a distinction is made between the processors for local and remote I/O devices. By definition, a remote device is one which requires a modem for attachment to the central processing facilities. The problems of recognizing valid transmission of data to and from a terminal device are more acute with such devices than with the local devices. Hence, it can be assumed that two different types of processing will be necessary. (These may, of course, split into different subtypes.)

Associated with the I/O processors and with the terminals are the usual queues. For activity into the endosystem, the basic input queue is at the interface with the ectosystem - users waiting for a terminal, cards waiting for batch processing, etc. In Figure 6 activity across the interface AA' must be established. This activity is of two types:

1) activity originated by the user and data sources
2) activity originated by the system.

The first type of activity, when user originated, is a request for a service by the endosystem, designated as a RFS. The user may also, once a service request is established within the system, be required to supply additional information before the service can be completed. The other source of input to the endosystem is data for the system's information store. As mentioned previously, the requirements for processing this type of input are dependent on the character of the input stream, i.e. batch, on-line, or real-time.

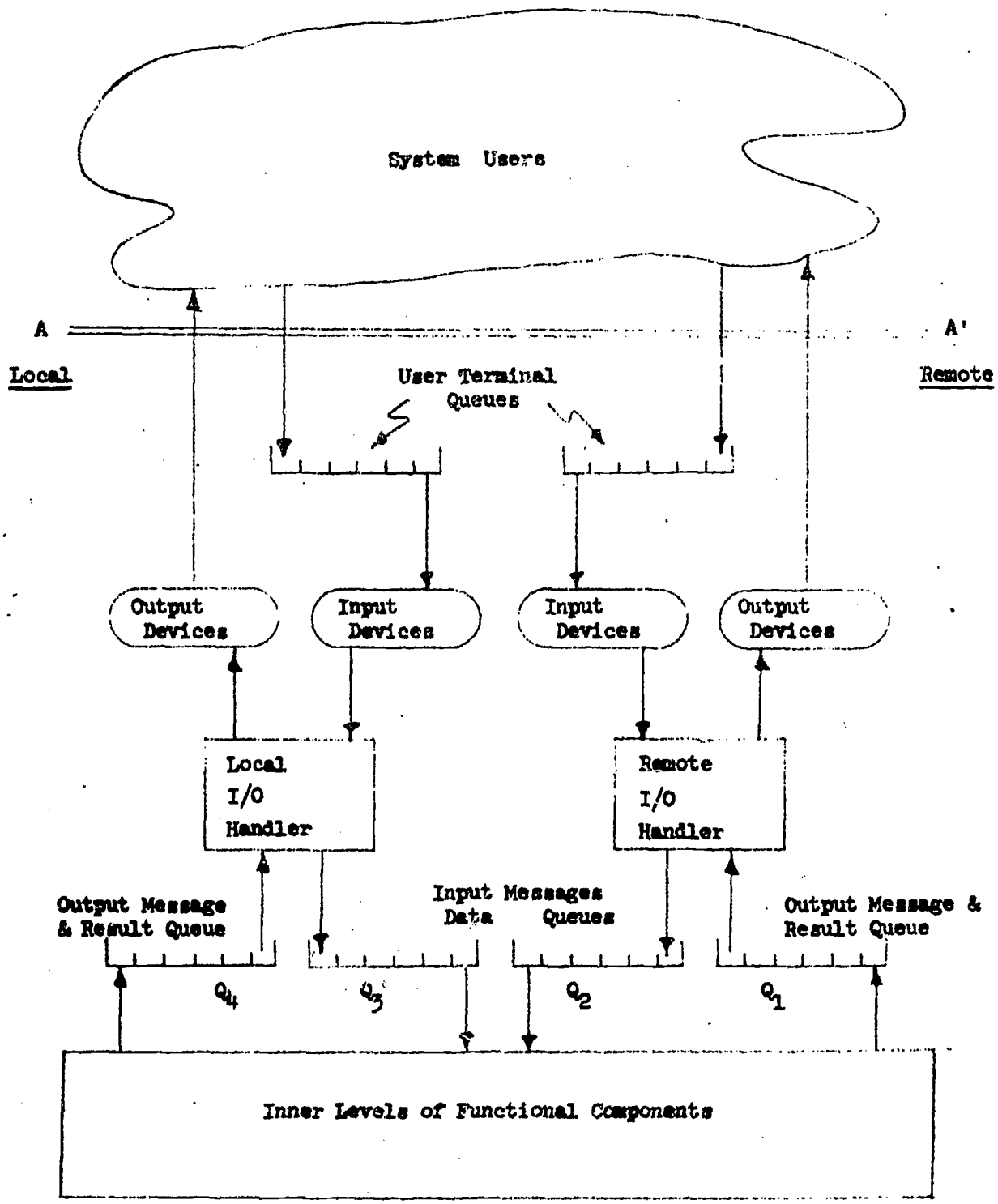


Figure 6. First Level Functional Components

The endosystem output must also be considered. It is generally of two types, either the output from the service requested, or a request by the endosystem for additional information.

In order to evaluate endosystem performance, the input and output activities must be specified. On input these specifications are for the arrival of the user or the data source information at the terminal, not at the central processing facilities. The information concerning the inputs must include for each terminal the distribution of activity with time, the distribution of service type requested, and the distribution of the message size for the inputs.

Whereas the inputs can be described in terms of their arrival rates at the ecto-endosystem interface, outputs must be handled in a slightly different manner. For a given request for service, the arrival of the corresponding results depends on the overall performance of the endosystem. This performance on the other hand is a function of the hardware/software equipment comprising the endosystem and the characteristics of the input stream. What can be described for the outputs is the quantity of output for each RFS processed. The description needs to include distributions for the number of units of output and for the length of the units.

Specifying the input/output activity does not enable one to determine the traffic rate within the endosystem. (See AA' in Figure 6) The flow rate across the interface is dependent on the components and performance of the unanalyzed part of the system. If the communication characteristics are known for the terminals and the connecting lines, and if the capabilities of the I/O handlers is specified, then an upper bound on the input activity

flow rate can be determined as a function of these hardware/software components. Similarly, if one assumes no interaction between the input and output activity at the interface, i.e., parallel communication with the terminal device thus allowing simultaneous transmission in both directions, then an upper bound on the output flow rate can be determined. As with the inputs, this upper bound is a function of the hardware/software components used for the handling of I/O.

The basic functions of the I/O handlers are to accept input from a terminal device and to send messages to a terminal. On input, its functions include the recognition that data is available for transmission, the assembling of messages from sequential transmitting devices, the recognition that data has been transmitted erroneously, etc. A secondary function of the input handler is to place the assembled message onto a queue thus making the message available for further processing. The input handler might also transmit a primary response to the terminal indicating that the character stream has been received (e.g., a fully duplexed system).

The I/O handler on output is responsible for transmission of data to the terminal devices. It receives its data from output queues which connect it to the rest of the system, information being placed in the queues as a result of the processing of the RFS's.

Having looked at the I/O handlers as being the processing modules responsible for communicating with the ectosystem, attention can now be turned to those components which process the input data and hence generate the resultant output from services performed.

2nd Level Functional Components

The primary function of the components in the second layer is to provide an interface between the I/O handlers and the modules actually performing the system services. As one of the functions of the second layer modules, they standardize all information flow between the participating modules and the service modules. A result of the second level interface is the independence of all the service modules from each other and from the rest of the endosystem. They are connected to each other and to the rest of the system via a second level module to be referred to as Request For Service, RFS, module.

The modules and their position in the information flow within the second layer components are shown in Figure 7. These components are briefly described below:

1. User Profile Store - a storage area reserved for user profiles. The profile indicates what services each user or class of users may obtain from the system.

2. Suspended RFS Queue - this queue, or store, is a holding area for service requests which are in a state of suspension due to the lack of sufficient information to transfer the request to a service module.

3. Output Message Queue - messages to the user indicating the status of his request are held in this area until they can be further processed. The format of the messages is such that it can contain the actual message, an address to the actual data to be transmitted, or the address of a vector of addresses to the information in the system's information store which is to be transmitted.

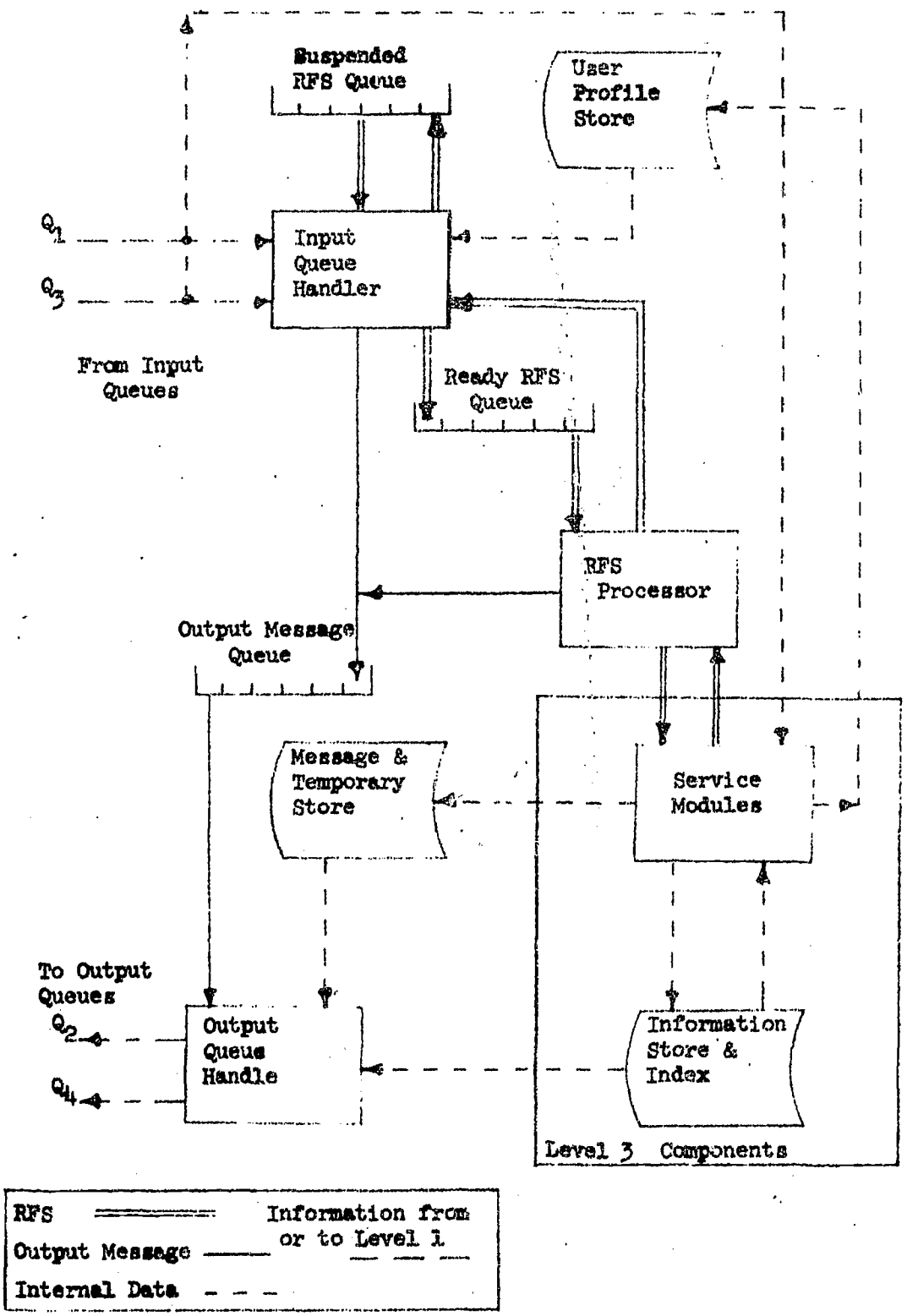


Figure 7. Second Level Functional Components

4. Ready RFS Queue - this queue is a holding area for RFS's which can be further processed. Entries in this area are essentially of two types, new requests for service and requests just removed from the Suspended RFS Queue because additional information has been received.

5. Message & Temporary Store - is a storage area containing two types of information. First, preformed messages which are to be transmitted to the user are resident in this store. Second, the results of the system services are placed here. The service results can be of two types: the actual data, or pointers to data residing in the information store.

6. Input Queue-Handler - is the processing module responsible for the data flow from the input queues established by the I/O handler at level 1 to the RFS processor. The major functions of this module are:

   a) prepare new requests for further processing,
   b) handle suspended RFS's,
   c) provide a path for data from the input stream required by the service modules.

7. Output Queue Handler - this processing module creates the formatted output stream and places it on the output queues for level 1. The amount of processing effort required to generate the formatted stream is dependent on the data content of the output message from the Output Message Queue.

8. The RFS Processor - basically the RFS is the interface between the service modules and other functional components at level 3 and the rest of the system. This processor is responsible for the disposition of the RFS once it is received from the Ready RFS Queue. Additionally, as a result of service by one of the modules, the RFS Processor creates the messages destined for the output stream and establishes the status of the RFS after completion of a service.

In order to visualize the communication between modules within the level 2 functional components, let Figure 8 be representative of the inter-module data formats. Assuming these formats, the processing of service requests within this black box can be described in the following manner.

When a user commences to use the endosystem, he signs on or identifies himself in some appropriate manner. At the same time that he signs on, he could also present to the system information indicating the services which he desires, and data required by a service module to provide the desired services. This data is represented in Figure 8 by format 1 data, i.e., the input to the Input Queue Handler. Assume that the Input Queue Handler standardizes the input as a first function in request processing. Once standardization has been accomplished, this processor must determine whether the incoming message is supplementary data to a previous request, or whether it is a new request for service. A possible method of making this determination is by checking the Suspended Services Queue to see if this particular user resides there. If not, a new RFS is created (represented by format 2) by inserting the user's profile into the service request. Otherwise, the new information

Type	Where Used	Type of Fields Needed			
1	Input to Q-handler	ID	Service Request	Data or Data Pointer	
2	Initial Entry in Ready RFS Queue	ID	User Profile	Service Request	Data or Data Pointer
3	Entry in Suspended Services Queue	ID	User Profile (Updated)	Service Request	Data, Pointer, or Statistical Info. (via Service Module(s))
4	Entry in Output Message Queue	ID	User Profile	Message Profile	Data or Address of Data in Message & Temporary Store
5	Continuing Entry in Ready RFS Queue	ID	User Profile (Updated)	Service Request (Updated)	New or Old Data or Data Pointer

Figure 8. Information Formats Internal to Level 2 and [illegible]

is appended to the existing RFS which has been retrieved from the Suspended Services Queue (format 3). In either case the data is placed onto the Ready RFS Queue thus making it available to the RFS processor. There is the possibility that a valid user ID cannot be found in the User Profile Store, due to bad transmission from a terminal, an invalid user, or some other cause. In this situation, the Input Queue Handler generates an output message and places it onto the Output Message Queue. The input data may or may not be subsequently destroyed.

Once placed on the Ready RFS Queue, the data can be further processed. The RFS Processor obtains new requests from this queue, most likely on a priority basis. The RFS processor analyzes the RFS and transfers it to the proper service module if sufficient information is contained in the RFS. If further information is required before service module selection can be performed then the RFS is returned to the Input Queue Handler for placing onto the Suspended Services Queue. Simultaneously, a message is created and placed onto the Output Message Queue for transmittal to the user.

Upon completion of processing by one of the service modules, the RFS is returned to the RFS processor. Again, the RFS Processor must determine whether further processing is required and can be scheduled immediately, whether more information is needed from the user and hence the RFS returned to the Suspended Services Queue, or whether the request is to be terminated. If the RFS must be returned to the Suspended Services Queue, then the RFS Processor generates a message to the user indicating what further information is needed. In fact, the RFS Processor may generate a message for the user indicating that a service phase has been completed. [Note: the information

for these created messages may have been provided by the service module having just completed processing on the RFS.]

It may be that all requested services have been completed. Hence, a message is generated by the RFS Processor indicating this fact. Also, the message may contain the output information or a pointer to further information about the results. It is assumed that the pointer within the results message refers to data resident on the Message & Temporary Store or to the System's Information Store proper. Within the request termination process, the RFS could be completely destroyed, thus leaving no trace of the service for accounting purposes and user profile updating, or it could be transferred to a service module to perform either or both of the above functions, or it could be returned to the Suspended Services Queue for the accounting functions at some later date.

The Output Queue Handler provides the reverse function of the Input Queue Handler. That is, it supplies information to the output queues at level 1 from information generated at level 2 and at level 3. The output stream creation is triggered by messages in the Output Messages Queue. These messages indicate what data must be placed by the processor into the output queues (Figure 8, format 4). Each message may contain the actual information or it may point to an area in the Message & Temporary Store. If there are a large number of message types which could be sent to the user, or if they are lengthy, then it would be reasonable to expect that these user replies could be stored and only their addresses transmitted via the Output Message records (i.e., the records received by the Output Queue Handler).

Secondly, the output of a service module, take the document retrieval processor as an example, could be lengthy. Hence, the service module may pass the data or a vector of pointers to relevant data directly to the Temporary Store and indicate where these results are to be found in the RFS returned after service completion. In these cases, the processing requirements of the Output Queue Handler are increased by the retrieval functions from Temporary Store of the system's Information Store. Its final function is to format the output data, including sufficient information for routing purposes, and to place this information onto the level 1 output queues, i.e., the remote or the local output queue.

3rd Layer Functional Components: The Service Modules and the System's Information Store

At the bottom layer of the system reside the service modules and the system's information store and index. Figure 9 is a schematic of these modules and their communication links. The schematic indicates one of the basic assumptions made at this level, that the service modules do not communicate directly with each other. Intermodular activity must first pass through the RFS Processor. As previously mentioned in the description of the level 2 functional components, the information received from the RFS Processor is the service request. When a service module has processed a request as far as possible without additional user-supplied data, the RFS is returned to level 2 via the RFS Processor. The data supplied to the service module most likely is not returned (Figure 8, format 5). The service modules may receive additional inputs required to perform the desired services. These inputs are not RFS's but data, and are supplied to the service module either via the Input Queue

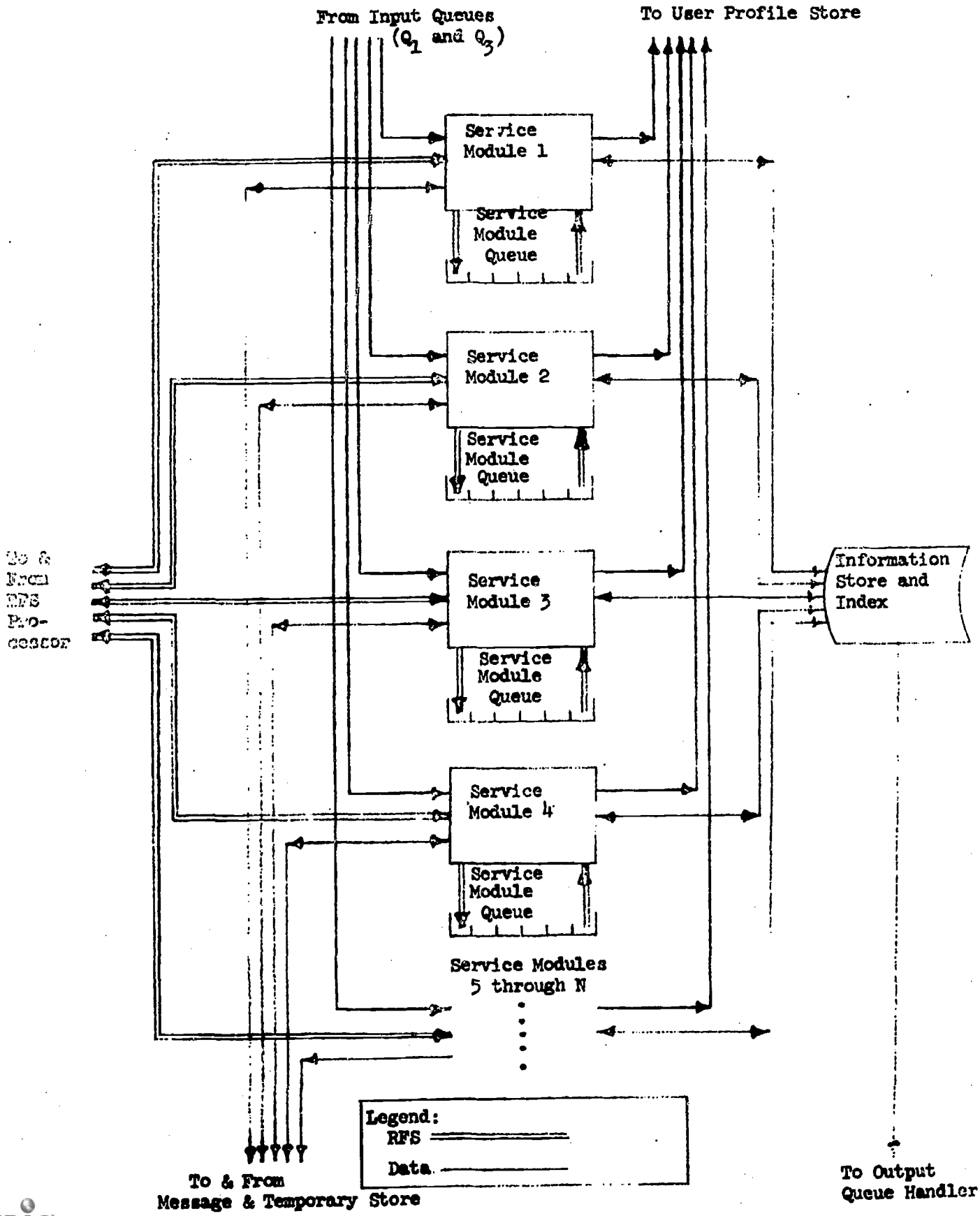


Figure 9. Third Level Functional Components

Handler, or from Temporary Store, or both. In the first case the data is generally user supplied; in the second, the data has been system generated as the result of previous service module processing.

Each of the service modules also contains its own queue. In these holding areas are RFS's which cannot be further processed. The reason for their suspension is not for lack of user-supplied data, but because a hardware/software component required to further process the request is presently unavailable. Examples of these situations are suspension waiting for the necessary access mechanism, or suspension because the service module is processing another request. Providing this queueing facility allows for the existence of a multi-programming environment, therefore, the existence of several partially processed requests within the service module.

The service modules are responsible for updating the RFS to show its new status after processing. The requested service might not have been completed at the time the RFS is returned to the RFS Processor because of insufficient user-supplied data. This situation would be indicated in the RFS, and the request would be placed onto the Suspended Services Queue by the Input Queue Processor via the RFS Processor.

If the system personnel can be considered as a special user class, then the services provided by the system can be considered to include both the external user-oriented services and the internal user-oriented services. An example of the first service type is Information Retrieval. The services directed towards evaluation of the system and maintenance of the system's information store and index are examples of services oriented to the internal users. It is not implied by this classification of services that one type

of service should be excluded from use by the other class of users. Some of the envisioned services provided by the system are

Services oriented towards the external user:

1. Information Retrieval,
2. Content analysis of queries,
3. Instructions on how to use the system,
4. Information on available services.

Services oriented toward the internal user:

1. Information Storage,
2. Content analysis of documental information,
3. Management of the endosystem's information stores,
4. Endosystem performance evaluation and accounting.

The primary endosystem services are assumed to be Information Storage and Retrieval. Figure 10 depicts the inter-component flow to and from the Retrieval service module. Shown also in Figure 10 are the level 2 functional components that interact directly with this service module. Let the retrieval process consist of three distinct activities:

1. Content analysis of a user query - resulting in the query vector,
2. Content analysis of documental information - resulting in a document vector (or index record),
3. The retrieval process - generating a response vector indicating the relevance of each document in the information store to the query.

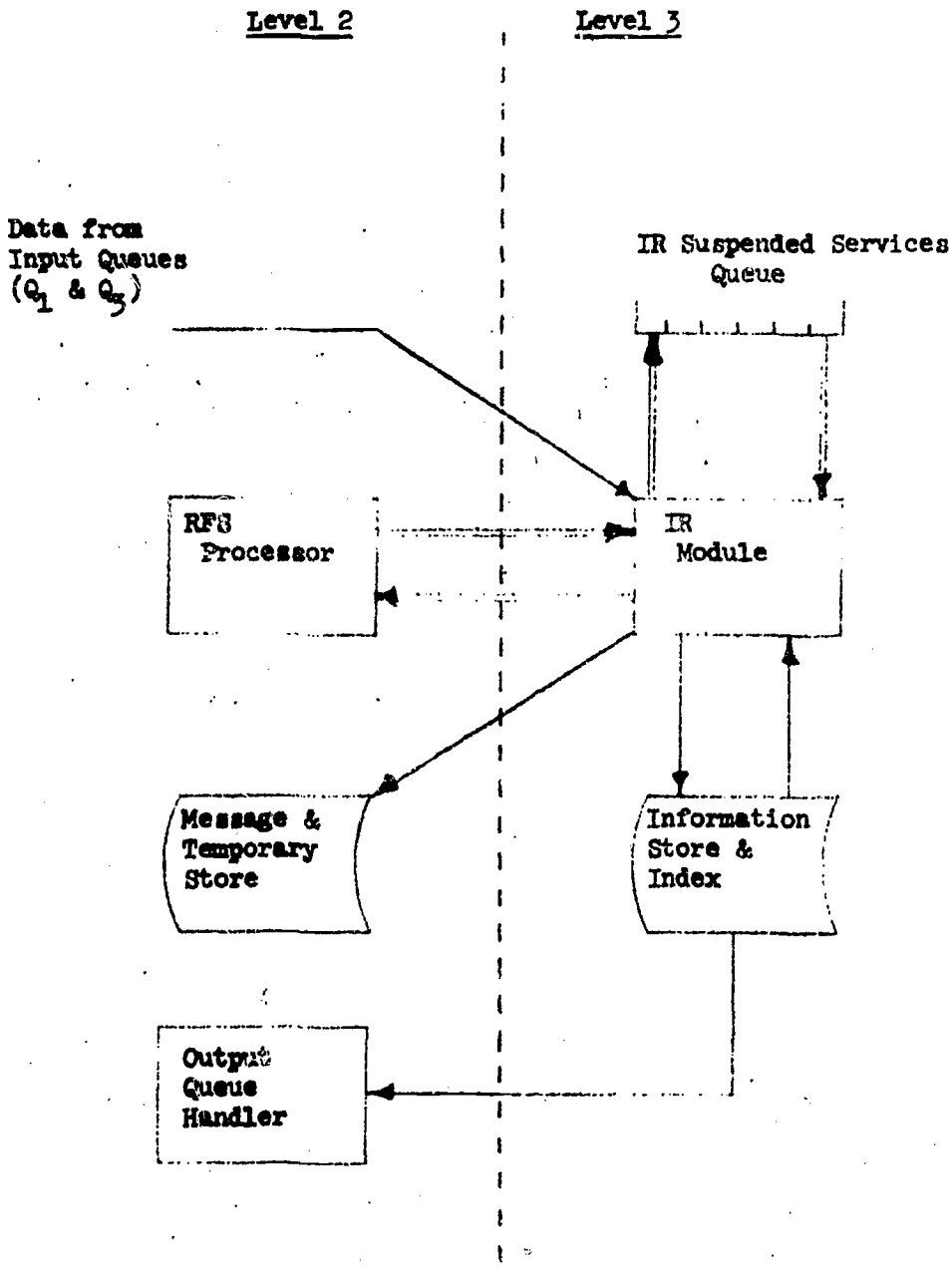


Figure 10. Information Retrieval Service Module

The content analysis function for both documents and queries may be performed by this service module if it is essentially the same process for both types of data. It is not always true content analysis is the same process for both documents and queries. The ectosystem may be composed in a manner such that documental information is highly structured and queries very unstructured or vice versa. In such cases, the content analysis could be done by separate modules, one for each type of input. For discussion purposes, assume that other than expected differences in length between a query and a document, the major difference between these two inputs occurs in their disposition after processing. In the case of a query, the resulting query vector is used as input to the retrieval process. With the document, the document vector and possibly the input is retained by the system in Temporary Store, this data becoming the input to another service module, e.g., the Store Update Service module.

Depending on the design philosophy underlying the services, the query vector may be immediately used, or it may be sent to Temporary Store and the RFS to the suspended Services Queue at level 2. One reason for suspension of the request could be user verification of the suitability of the analysis results with respect to his desires. The RFS is retrieved from the Suspended Services Queue when the user responds, and is again directed to the appropriate service module. [When the RFS is returned to the RFS Processor, the processor (RFS) signals the Output Queue Handler that output is required by creating and placing a message in the Output Message Queue.]

As with query analysis, document analysis may require supplemental user action before it can be completed. There is also the possibility that a single RFS may call for analyzing more than one document. In this case the input stream to the Retrieval module, other than the RFS, would come directly from the Input Queue Handler. Processed documents and the resultant document vectors are placed in Temporary Store. A possibility arises that when the processing for a particular document is completed user intervention is required, hence the RFS needs be returned to the Suspended Services Queue. When no user intervention is required, the data used by the Retrieval module and the output from the processing still is placed in Temporary Store and the RFS returned to the RFS Processor.

There are several variations of the content analysis processing for queues and documents. A simple one is that of Selective Dissemination of Information (SDI). In this case a complete set of queries (profiles of users interests) are passed against new documents and possibly the whole information store if the user would so desire. Additionally, the profiles must be stored in the information store.

The third activity of the Retrieval module is the retrieval process itself. It is assumed that the output of this processing is a vector indicating the relevancy of all the documents in the store to the query. This vector is called the response vector. Additional processing may be required on this response vector by other service modules, or a user response may be needed before any results can be transmitted to him. The RFS is returned to the RFS Processor and the response vector to Temporary Store. The RFS Processor decides on what type of output message to enter into the Output Message Queue.

It must also decide on the disposition of the RFS. Finally, the response vector can be retained in Temporary Store or destroyed when output is completed. It should probably be retained so that the system's Evaluation service modules can use it as data for accounting and statistical purposes.

Establishment of Feasible Operating Regions

The performance of the system, when measured by the time to provide a service, is directly affected by the information flow through the functional components within the system. Delays can occur due to such factors as:

1. the size and type of storage media used for the queues and other storage areas in the system,
2. the speed and power of the hardware/software components selected to perform the processing requirements of the modules,
3. the ability of the I/O handlers, the communication and the terminal devices to cope with the input/output demands of the system environment,
4. the amount of contention by the functional components for the shared facilities.

Delays due to any of the above cause congestion to occur within the information network of the endosystem.

In order to determine the time performance of the system, the total information network must be analyzed. The effect of each functional component on the information flow needs to be determined. The behavior of these components is a function of the hardware/software components selected and the amount and type of system activity. The goal of analysis is to select hardware/software components so as to balance the flow and thereby increase throughput or reduce system cost.

It was mentioned previously that the user plays an important role in the performance of the system, particularly in the types of services he demands for his continued use of the system and the frequency at which he requests these services. Additionally, he must also be satisfied with the performance of the system with respect to his requests. User satisfaction causes constraints to be placed on the acceptable design levels. Other than satisfaction with the provided services, his acceptance of the system is influenced by the time he is willing to wait for results and the cost for the services. These two factors are examined further below.

Let a measure of frequency of system use be the information flow in either direction across the interface between the endosystem and the ectosystem (cross section AA' in Figure 6) over a fixed time period, say ΔT. The ΔT may correspond to a month's activity, a week's activity, or possibly to a financial accounting period. Assume that the latter is the case. The distribution function of this activity as a function of time and service over ΔT is also required for each terminal, or at least each terminal type. Other parameters required of the input specifications are the message lengths and the method of inputting the information, e.g. whether the user types the message himself at a terminal or transmits a prepunched card(s).

Associated with each service requested is the system response. This response is in two categories: messages requiring additional inputs from the user, and the results of services performed. Response activity needs specifying also. These specifications include the number of messages required before service request can be performed or the request terminated. They also include the volume of results to be created for the service request, the size of the message, say in terms of number of lines of output, the number of

characters per line, and the routing of the output. Probability functions can be used to supply the necessary information. The above information in conjunction with the input specifications defines the total output activity over ΔT.

Having specified both the input and output, the information flow across the A-A' interface is known for ΔT. Also required for a system time performance evaluation are the following:

1. the time for each functional component to perform its assigned task(s),
2. the assignment of the functional components to the hardware/software units selected to comprise a particular implementation,
3. the specification of queue and other storage area sizes.

Let t_s be the system response time for service s. It will be defined as the elapsed time from initiation of a request by a user until the last of the results from the requested service is returned by the system, i.e., the last of the results crosses the endosystem-ectosystem interface. A probability function can be used to express t_s over the accounting period ΔT. The values assumed for this probability function depend on the hardware/software components selected, the input activity of the ectosystem, and the output activity of the endosystem. A probability function, expressing the total system response time, t, is defined as the weighted average of the functions for each individual service. From this latter function, the expected response time, E[t], for all services over ΔT is attainable.

In Figure 11, the anticipated behavior of E[t] is shown as the mean input volume demand $\bar{V}$ (assuming the same distributions for the volume demands) increases. The curve would apply only to a single hardware/software component selection. Different components would result in different curves. Also, the curve is dependent on the distribution of the I/O activity over ΔT and not just the mean volume over ΔT. This time-performance curve is the basis for the establishing of the feasible operating regions.

It is assumed that for low values of $\bar{V}$ the expected response time E[t] will remain constant. As long as there is sufficient excess system capacity to keep all queues and temporary stores at their minimum levels, the contributions to each t_s will be the time requirements of the processing modules, including the retrieval module. As the queue sizes increase and more modules contend for shared facilities, the response times should increase. Finally, if one assumes an arbitrarily long queue of users waiting at each terminal, then response times could become arbitrarily long as volume increases. For practical situations, there would be an upper limit to the actual activity the endosystem would handle in the time interval ΔT.

Associated with a particular time-performance curve (i.e., a particular system) are the costs to provide the services. For a given value of $\bar{V}$, the associated cost, $C_S(\bar{V})$, includes both the overhead costs and the direct costs. Figure 12 represents the relationship between cost, $C_S(\bar{V})$, mean volume over ΔT, $\bar{V}$ and the expected response time, E[t]. The $C_S(\bar{V})$ associated with a particular time-performance curve is not a surface in the cost dimension, but a thread. Figure 13 depicts the cost thread on the cylindrical surface generated by

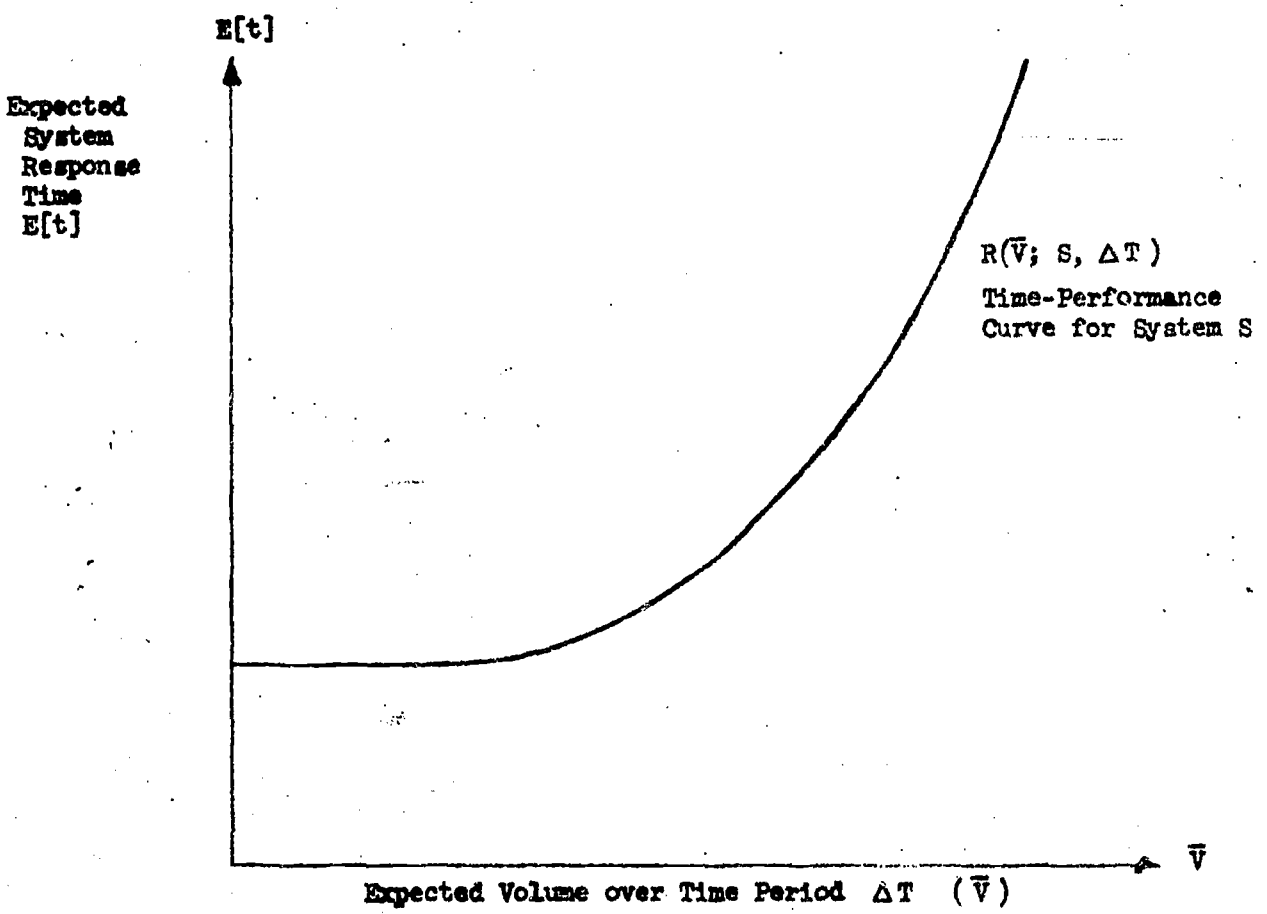


Figure 11. Time-Performance Curve

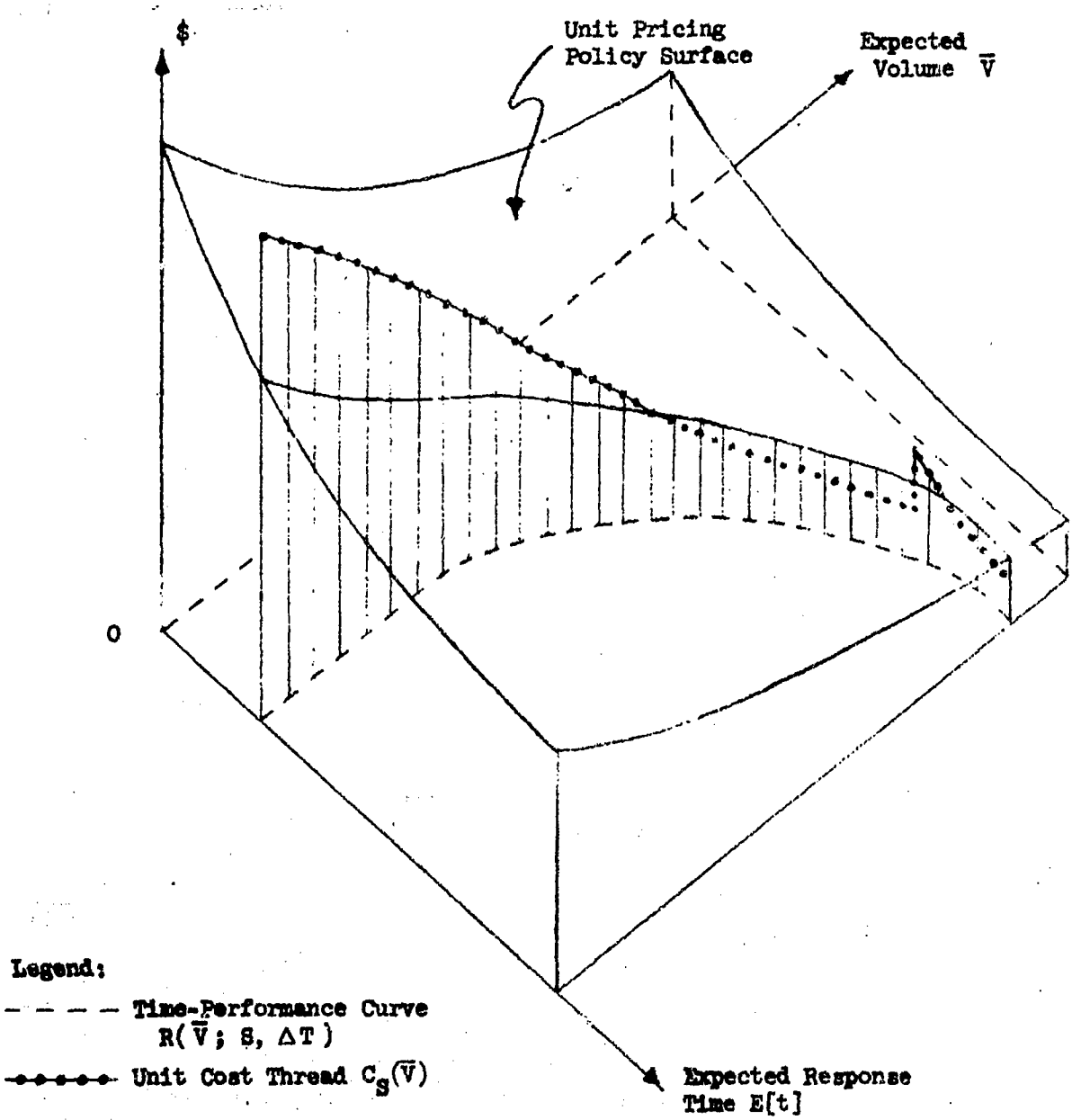


Figure 12. System Cost and Pricing Policy

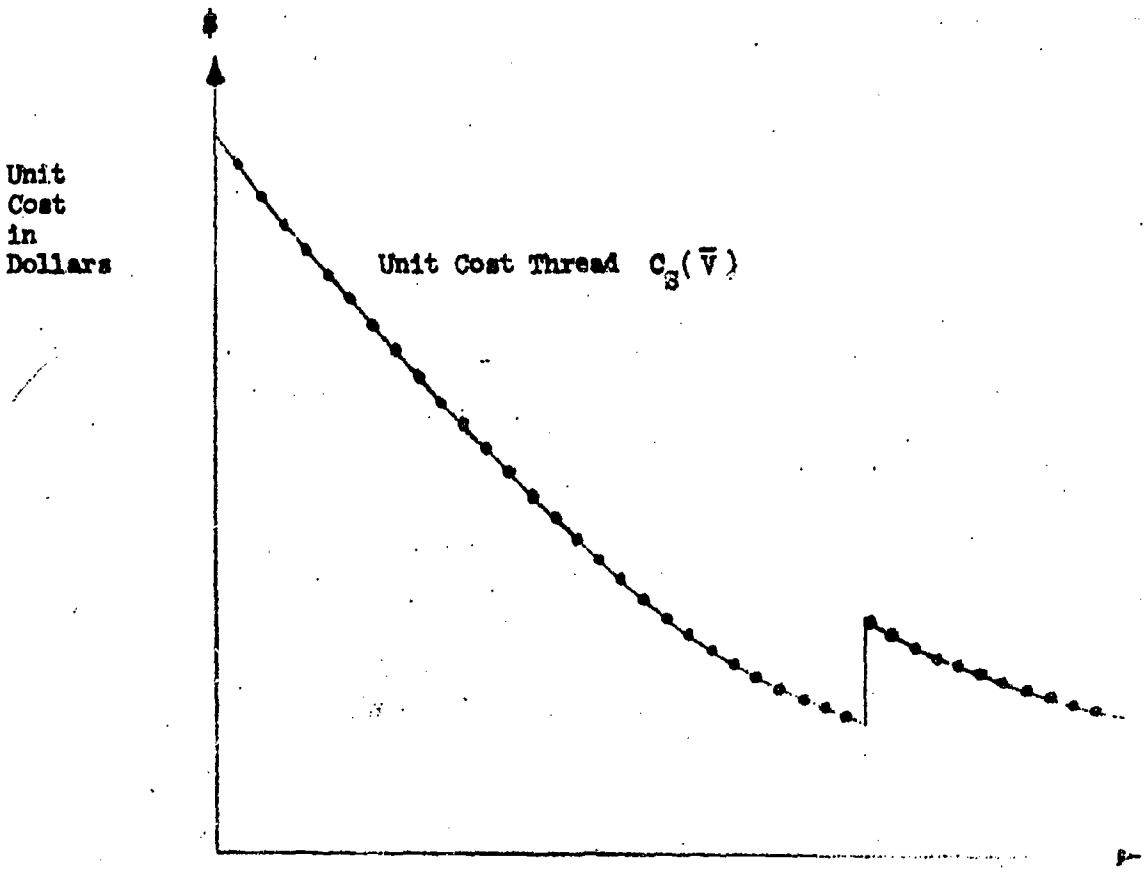
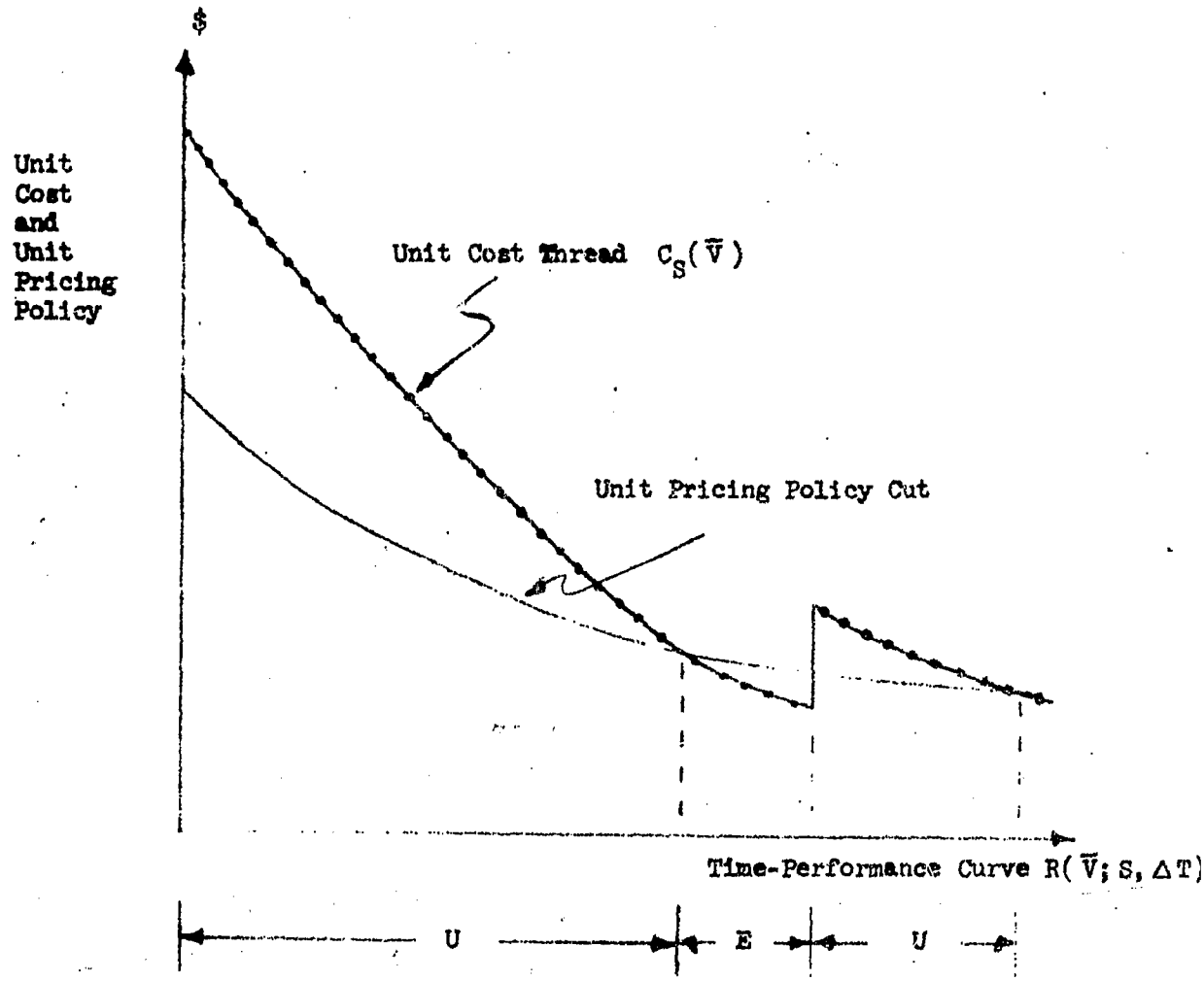


Figure 13. Unit Cost Thread for Specified System

the time-response curve. The cost thread is given as the unit cost for volume $\bar{V}$. A particular point on the unit cost curve represents the expected cost per unit of user request activity when $\bar{V}$ is the expected input demand over ΔT and for a specific configuration of system components [hardware/software/personnel]. Jumps in this thread occur, for example, as second or third shift operations become necessary.

When the role of the funder was discussed, one aspect of his role was that of bearing the cost for the system. He must decide how to allocate this cost to the system users. His cost allocation or pricing policy will take into account the expected activity of the system and the time required to provide the requested services. This pricing policy must consider the user's willingness to pay for a service when these services are provided at a performance level P. The user views the system as a black box and is not concerned with the hardware/software/personnel components reflected in the operating costs of the system. Furthermore, he is not concerned with activity other than his own. Therefore, pricing policy is independent of the configuration and is a surface in the cost dimension instead of a thread as was the case with the cost curve. In Figure 12, the pricing policy surface is shown as the price per request, and is the weighted mean of the pricing for all services to be provided.

Figure 14 shows the pricing policy surface as it intersects the cylindrical surface generated by a time-performance curve. Also shown in Figure 14 is the associated unit cost curve, $C_S(\bar{V})$. Feasible operating conditions, from the time-volume standpoint, exist when the unit price (income) exceeds the unit cost, that is, where $C_S(\bar{V}) < P(\bar{V},t)$. Once the volumes, $\bar{V}$, are determined where favorable operations are possible, the funder



U = Region of Uneconomic Operation

E = Region of Economic Operation

Figure 14, Economic Operating Region

(management) must decide whether or not these volumes are attainable in the actual operations. If these volume ranges are not expected, then profitable operation is not possible with the selected component configuration. It is also worth noting that if the expected input activity, $\bar{V}$, causes the response time for requested services to exceed a desired level, then the configuration is unsatisfactory to the requestor. This again creates an unacceptable component configuration.

Other useful information is available from time-performance evaluations, for example, the cost of future growth in the mean activity, $\bar{V}$, while maintaining the same expected system response time, $E[t]$. Required are the time-performance curves for a family of system configurations. A pseudo unit cost thread results for a constant $E[t]$ in the cost dimension. Plotting this unit cost thread and the unit price cut, curves similar to those in Figure 14 occur. Operations are economically feasible with no degrading of the expected system response time for those values of $\bar{V}$ such that $C(\bar{V}) < P(\bar{V}, E[t])$.

If volume, $\bar{V}$, is held constant, the resulting unit cost thread shows the unit cost variation for a corresponding change in response-time requirements for system services. Again, if the unit cost thread values are compared with the corresponding unit price curve, then response times for economically feasible operations are obtained.

To summarize, the generation of system performance time measures from functional component analysis provides a method of determining system characteristics, economic feasibility of implementation, implications of future growth in system use, and the effects of alternative designs.

RESEARCH MEMORANDUM SERIES

No.

69-1 "Computer Planning and Optimization of Automated Manufacturing Processes", P. B. Berra and M. Barash, May 1969.

69-2 "The Nature of the Distribution of the Life of HSS Tools and Its Significance in Manufacturing", J. G. Wager and M. Barash, May 1969.

69-3 "Some Studies of the Effect of Service Time Variability on the Design of In-Process Storage Areas and the Dynamic Operation of Production Lines", D. R. Anderson and C. L. Moodie, May 1969.

69-4 "Optimal Harvest Policies for Natural Animal Populations in Which Members are Indistinguishable as to Sex at the Time of Harvest", S. H. Mann, May 1969.

69-5 "On Information Storage Models", F. F. Leimkuhler, June 1969.

69-6 "A Basis for Time and Cost Evaluation of Information Systems", R. R. Korfhage and T. G. DeLutis, June 1969.

69-7 "Comparative Design Considerations of the Engineering Plastics and Die Casting Alloys", R. F. Adams, June 1969.

69-8 "Storage Policies for Information Systems", F. F. Leimkuhler, June 1969.

69-9 "A Mathematical Theory for the Harvest of Natural Animal Population in the Case of Male and Female Dependent Birth Rates", S. H. Mann, July 1969.

69-10 "A Mathematical Theory for the Harvest of Natural Animal Populations when Birth Rates are Dependent on Total Population Size", S. H. Mann, July 1969.

69-11 "A Mathematical Theory for the Control of Pest Populations", S. H. Mann, August 1969.

69-12 "To Formulate a Prediction Equation for the Life of Ceramic Tool Materials", Y. I. ElGomayel, October 1969.

69-13 Y. I. ElGomayel (to be completed).

69-14 "A Heuristic Schedule Generator for Parallel Processors which Process Jobs with Precedence Constraints", C. L. Moodie and J. M. Walter, November 1969.

69-15 A Reject Allowance Problem in a Recurrent Production Facility", B. P. Layton and S. H. Mann, December 1969.

70-1 "The Bandwidth of Cubic Graphs", R. R. Korfhage and N. E. Gibbs, March 1970.

70-2 "A Branch and Bound Algorithm for Minimizing Cost in Project Scheduling", A. T. Mason and C. L. Moodie, May 1970.

70-3 "Differences of Perception in the Professional-Organizational Interaction", D. H. Kraft, May 1970.

70-4 "Optimal Inventory Policies in Contagious Demand Models", A. Ravindran, June 1970.

70-5 "Computer Design of School Bus Routes", T. G. Godfrey, June 1970.

70-6 "On the Determination of Optimal Storage Block Configurations", S. D. Roberts and R. Reed, Jr., June 1970.

70-7 "A Paradigm of Commitment: Toward Professional Identity for Librarians", L. C. DeWeese III, June 1970.

70-8 "An Application of LaGrange Optimization of Response Surfaces in Pharmaceutical Product and Process Design", J. R. Buck, June 1970.

70-9 "A Comparison of the Primal-Simplex and Complementary Pivot Methods for Linear Programming", A. Ravindran, July 1970.

70-10 "Management of Seasonal Style-Goods Inventories", A. Ravindran, July 1970.

70-11 "Laplace Transforms in the Economic Analysis of Deterministic Decisions", J. R. Buck, July 1970.

70-12 "Elementary Difference Models in Economic Analysis", James R. Buck, September 1970.

70-13 "The Effects of Junction-Cost Allocation and Physical-Economic Environments on Production Policies which Maximize Profits", James R. Buck and Jorge A. Incer, September 1970.

70-14 "On Compact Book Storage in Libraries", Arunachalam Ravindran, December, 1970.

70-15 "An Annotated Bibliography on Transportation and Communication Networks", K. Ram Mohan Rao, November 1970.

71-1 "An M/M/c Queue for the Distance Priority Machine Interference Problem", Gary H. Reynolds, January 1971

71-2 "Network Flow Optimization with the Out-of-Kilter Algorithm: Part I - Theory", Don T. Phillips and Paul A. Jensen, Fegruary 1971.

71-3 "Network Flow Optimization with the Out-of-Kilter Algorithm: Part II - Applications", Don T. Phillips and Paul A Jensen, February 1971.

71-4 "An Engineering Economic Analysis of Bulk Storage in Tomato Processing", James R. Buck, G. H. Sullivan, P. E. Nelson, Y. E. ElGomayel, and J. C. Pereira, April 1971.

71-5 "Formulation of the Dynamic Deterministic Inventory Model as a Branch and Bound Programming Problem", Richard H. Dean, April 1971.

71-6 "A Three Variable Analysis of Cost Growth", Richard S. Sapp, April 1971.

71-7 "Social Indicators and Second-Order Consequences: Measuring the Impact of Innovative Health and Medical Care Delivery Systems", James G. Anderson, May 1971.

71-8 "New GERT Concepts and a GERT Network Simulation Program", A. Alan B. Pritsker, May 1971.

71-9 "On Programming with Absolute-Value Functions", Thomas W Hill, Jr. and Arunachalam Ravindran, May 1971.

71-10 "Statistical Correlation of Tool Wear Parameters", Y. I. ElGomayel and A. A. Zakaria, June 1971

71-11 "The Application of the Statistical Multiple Linear Regression Analysis to Formulate a Prediction Equation for the Life of a Cutting Tool", Y. E. ElGomayel, June 1971.