Data Physics - an Unorthodox View of Data and Its Implications in Data Processors

R. R. Korfhage, W. H. E. Day, L. L. Beck, W. F. Appelbe
Southern Methodist University
Dallas, Texas

Introduction.

From prehistoric days, mankind has been involved with technology - "...the utilization of energy and materials in controlled situations to modify and organize man's physical and social environment."[1] Initially, emphasis was placed on the mastery of materials. Gradually the emphasis shifted toward the mastery of energy, which has played a dominant role in the technology of the early twentieth century. Now, as computers and communication systems come of age, the emphasis is shifting once again - to the mastery of information.

Computers, once regarded as "number crunchers," are more properly viewed as information processors. Past years have seen fundamental advances in information processing - the theoretical work of Salton, Sparck-Jones, and others on the nature of information processing, the development of database concepts, and work on hardware specifically oriented toward information storage and retrieval. These advances have laid solid ground for the advancement of information processing.

It is an appropriate time to step back and examine the foundations of our field - to determine how the various approaches and fragmentary principles fit together. The approach that we currently call "data physics" has this as its objective. While our work clearly is based on that of others in the field, our concepts may seem unusual to the practitioner.

We may draw an analogy between our work and that of the cosmologist. The cosmologist studies the birth, evolution, structure, and death of the universe. He uses concepts from relativity theory and physics that are alien to everyday experience. The man in the street is concerned with daily living, with getting from home to work - and he could care less (in practical terms) about the exact geometry and properties of the universe. Similarly, we propose a broad and esoteric view of the universe of information - a view that must jibe with the everyday world of libraries and databases, but which may ultimately be stated in terms quite different from those that the practitioner normally uses.

We undertake four tasks in this paper: to define the data physics approach, to state the important issues, to propose a model of the information universe, and to discuss the implications of this model.

The Data Physics Approach.

Data physics adopts the view that data represent an underlying reality subject to natural laws and relationships, which can only be inferred by observation and experimentation. Any information system, built for the benefit of its users, should reflect these laws and relationships in its architecture and operation. It is the task of the data physicist to discover these laws and relationships, and to present them in a form useful for the design of information systems. To this end, he observes and experiments.

We are in a veritable sea of data, but the data are observable only on a finite scale - through a moving window, as it were. On the basis of such observations and experiments as this view permits, we formulate a global or cosmological picture of the underlying reality. This picture in turn stimulates and directs further observations and experimentation. More importantly, this picture provides a fundamental gestalt that conditions and limits the practitioner's approach to the data and reality.

We know the data, not the reality. Thus we are required to work with the "space" of data. We must study the characteristics of this space before we can properly understand the laws and design an information system which reflects them.

Does the data space have an inherent organization, or is it subject only to an organization of our choosing? What is the dimension of the data space? What is the aging process for the space? Can a data space be self-organizing? What types of processes can be done on the data space? The answers to these and similar questions provide the basis for a sound theory and the development of information systems which reflect the theory.

Issues.

While a valid cosmological model must fit reality, it is unconstrained by the fetters of conventional concepts. Thus we approach the issues on an "ideal" basis, avoiding implementation concerns for the moment.

It is fundamental to our endeavors to assume that the data space does have an intrinsic organization, which represents exactly the underlying reality. But since this reality is unknown and unknowable, so also is this inherent organization. There are, however, two levels of organization that are important. The first

is that imposed by the data physicist in an effort to model the reality and simultaneously to serve the needs of the users of data. We suppose that this organization is closely related to the inherent organization of the space. It is thus the organization on which theories are based. The other level of organization is that of the user. This organization may neglect the fine points of the theory, and may ignore large portions of the data space - but it enables the user to handle the data he needs rapidly and efficiently.

Dimensionality in a data space is rather complex. It seems most reasonable to regard the data space as finite dimensional, but with two unusual properties. First, the characteristics of the dimensions are not uniform. One dimension may assume an infinite variety of values, while another may have only a small finite number of different values. Comparisons between the dimensions may be quite meaningless. In addition, while the number of dimensions is finite, it is not fixed. As new data are added to the space, the dimensionality of the space is likely to increase. A datum that is a point (that is, all attributes are known) with respect to the current dimensions may become a hyperplane as new dimensions are added to the space - dimensions in which the characteristics of the given datum are not known.

The concept of self-organizing is generally taken to mean "self-reorganizing." This represents a process, rather than an entity. If we regard the data space as a collection of entities (or data about entities), then it should have no power to self-reorganize. Indeed, as the inherent organization of the data space corresponds to reality, any reorganization at this level would imply a reorganization of reality. While an evolution of reality is possible, we prefer to believe that a fundamental reorganization is not. However, the organizations imposed on the data space, whether theoretical or practical in nature, can be reorganized. But this is a process which is external to the data space.

While the relativist regards time as another dimension, we must note that it behaves quite differently from other dimensions. In most situations we cannot access the future with any certainty; and we often have only imprefect access to the past. As the dimensionality of the space changes with time, questions that are meaningful for the "present" time may be impossible to answer, or even meaningless, for past times. It appears, thus, that time should be treated in some special way, rather than being included simply as another dimension.

Closely related to time is the aging process within the data space. While a datum is capable of surviving intact throughout time, normally there is some sort of aging - through summarization, updating of data, displacement by totally different data, and other processes. Any data physics model must allow an aging process to occur.

The processes that should take place on a data space are those of location, alteration, evaluation, and retrieval. These processes are quite well understood in conventional information systems, and there seems to be little reason to extend their scope or definition.

A Framework for Information Processing.

The issues that we have defined have led us to an eight-component model. The first component, the underlying reality, plays very little role in the overall system, due to its unknowable nature.

The second component, the data space, has its theoretical properties indicated by the issues that we have raised. This is the "memory" of the overall system. Thus it is appropriate to raise questions about the implementation of this memory. In particular, the multidimensional nature of the data space leads to the problem of implementing an n-dimensional space in a conventional computer system.

Our model postulates two intermediate levels of processors for the data space. The basic level consists of a number of subspace processors, each handling one relatively small portion of the data space. We must question the characteristics of these processors, and the interactions between them. Uniformity of design among these processors is also a fundamental issue. While the number of processors may change during the life of a system, we regard each processor as relatively fixed and limited in its capabilities: it handles all of the processing for its own portion of the space, and it can communicate with other processors about their operations and results.

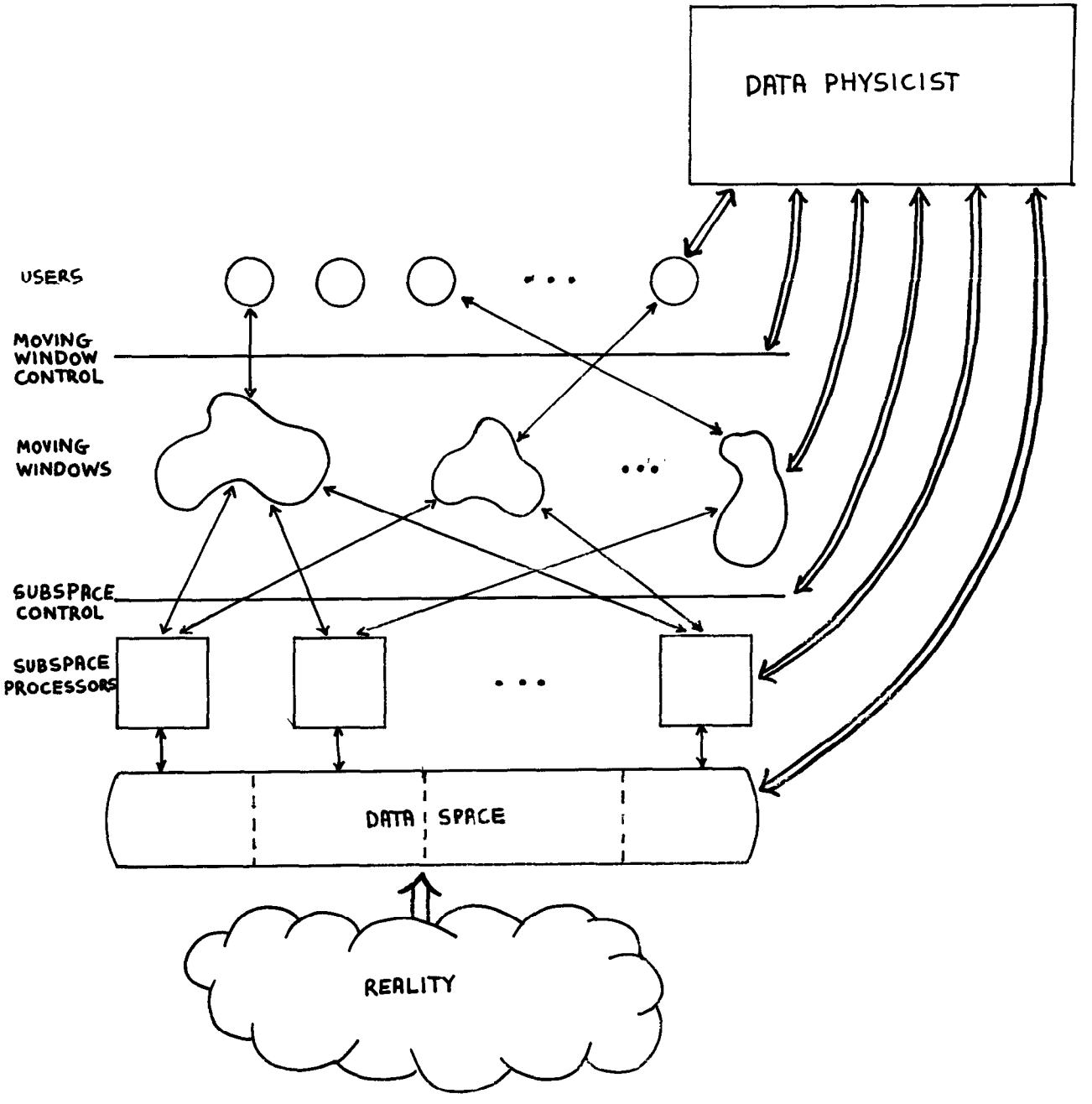
The other intermediate processor is the moving window processor. This is essentially a query processor. While the fundamental units for a moving window may be regarded as primitive, these units may be assembled in a sophisticated manner, allowing a moving window to integrate information from several subspace processors.

Thus a subspace processor is stable, normally existing unchanged for the life of a system, whereas the moving window is highly dynamic and transient. Once a query has been processed, its moving window may disassemble, the component parts then becoming available for other use.

The ultimate processor in our model is the user. The purpose of any information system is to provide useful data to a user. Reality and the user form, in a sense, the opposite ends of the system. Whereas the reality supplies the data upon which the system must operate, the user supplies the queries that drive the system. In the abstract, we avoid making any assumptions about the character and abilities of the user, recognizing that in any workable information system, assumptions of this type may be needed.

Corresponding to the three levels of processor in the model there are three levels of control. Subspace control is intermediate between the subspace processors and the moving windows. This control mechanism provides much of the interaction and communication among the subspace processors, as well as security over access to any subspace processor. It also handles certain monitoring and reporting functions.

Window control, intermediate between the user and the moving windows, is the second level of control within the system. This is a more delicate control, since without it the user has the ability to integrate the information obtained, through the moving window, from several different subspace processors,



thus creating security problems. Thus this level of control must be effective at monitoring and limiting such integration of information. While the subspace control is focused on the activities of a subspace processor, the window control has its focus on the activities of the user.

The final level of control is the data physicist himself. In conjunction with his studies of the information system, and the information universe, he designs and authorizes all of the system components that are intermediate between the ends - the subspace and moving window processors, and the subspace and window controls.

This is our framework - on the one end, reality and the data representing it; on the other end, the user and the data physicist; and between the ends, the processors and controls necessary to match the data to the user's needs.

Theoretical models.

Within the framework that we have briefly sketched,

possibilities abound. Since we regard the the ends of our model as relatively fixed, we concentrate on the middle - the two processors and their controllers.

We regard the subspace processor as a relatively limited instrument, capable of manipulating and updating the information in its own segment of the data space, and of communicating with other subscape processors, particularly those "in the neighborhood." Because of the basic uniformity of data, we assume as a base that all of the subspace processors have the same fundamental design. Of course, portions of this design may be more heavily used in one processor than in another; but at present, we assume that the disparity of use is not sufficiently great to lead to several different designs for processors at this level.

The exact capabilities of the subspace processors depend heavily on the characteristics of the subspace. Since we have assumed a multidimensional space, our theoretical processors must be capable of handling such a space. As the data space assumes more definite characteristics, the subspace processors evolve to reflect this. For example, if we assume that the data space is holographic in nature, or that it consists of "nudged quanta" (as in one science fiction story)[2], the processor we postulate must operate on such a space.

We have postulated communication among the subspace processors. This is vital since there are no clear demarkations of the data into neat cells. The question of whether two or more processors can actually access any given cell of data leads to significant decisions in the design of both the processors and the controllers.

Subspace control must be concerned with two distinct problems: the ability of a user (through a moving window) to access a given subspace processor, and communication among the processors themselves. As is well known from networking, different organizations of the subspace processors lead to different problems of control, and different implementations of the control mechanisms.

To indicate the possibilities, consider a data space that is holographic in nature. (Some psychologists suggest that human memory is fundamentally holographic.)[3] One can conceive a mechanism that represents each "dimension" of the space by a different wavelength of light. Thus retrieval of information becomes simple and essentially instantaneous along any one dimension. But for such a space, control and security present formidable problems, since any portion of a hologram replicates (to some accuracy) the entire hologram.

Picture now a moving window that corresponds to a query from the user. While the data space has an inherent structure, and the subspace processors can therefore be standardized, queries are free-form and unpredictable by nature. Thus the "moving window" must be capable of conforming to a query, and of matching this conformation to the data space. Through a moving window, the user can examine an area of the data space which has arbitrarily high dimension, but is restricted to a limited range of values in each dimension. To provide the flixibility necessary to handling queries, the moving window must have these fundamental capabilities:

(1) Motion - The ability to examine different parts of the data space, as required;

(2) Shape changing - The ability to change the dimensionality, or the range of values in each dimension of the area being observed. This includes the ability to expand or contract, to "focus in" on a specific area;

(3) Filtering - The ability to impose a new view of the data as it is observed. This might include reducing the dimensionality of the image presented (much as one does when taking a photograph of a three-dimensional object). It would also include clustering or other operations to "simplify" the view given to the user.

As we suggested above, moving windows may well be constructed of rather simple elements, that work in concert to achieve the desired effect. The mechanisms controlling a moving window as it searches for a response to a query are not well understood. While we tend to think of hierarchical control mechanisms, models drawn from cellular automata, bird flocks, amoebas, or myxomycetes indicate that a more diffuse control mechanism might be appropriate.

Control of access to moving windows by a user is extremely difficult. Basically the problem is that with access to a single subspace processor, the user is limited in the information that he can obtain in response to a query. But since a moving window can, in theory, access several subspace processors simultaneously, a much more integrated response to the query is possible. Worse yet, by utilizing several moving windows, either simultaneously or sequentially, a user may collect information and integrate it outside of the information system. This can be very useful, but can also be extremely hazardous - as illustrated recently by the amateurs who have pulled together the information necessary to construct atomic weapons.

Practical implications.

It is interesting to muse on a giant amoeba, feeding its way through the data space and absorbing information in response to a given question. Whether this, or any other concept, has merit either as a theoretical construct, or as leading to some realizable approach to information system design, must be judged on the basis of realistic criteria.

Within the general model framework that we have postulated, many specific models can be constructed. Translating any one of these models into a practical proposal can be done only if a clear set of goals and objectives for database development can be formulated. Inevitably there will be compromises in any practical design, but identification of the objectives permits the compromises and conflicting requirements to be clearly recognized.

Among the many possible objectives in a data base development proposal, four stand out as of primary importance:

(1) Adaptability. An adaptable system has been defined as one that can be readily modified to meet a wide range of user and system requirements. Although some databases are static, the majority evolve over a considerable period of time, both in response to changes in user needs, and in fundamental system capabilities.

(2) Reliability. A reliable database is one which is secure against failures in hardware and software, and which controls the capabilities provided to a given user. Thus reliability incorporates the two objectives of security and integrity, clearly closely related.

(3) Efficiency. The efficiency of a database can be quantified by performance measures such as response time and throughput, and a design proposal must specify these objectives. Ideally, the efficiency should be independent of the adaptability of the system, but this is rarely so in practice. Usually system performance deteriorates as the flexibility and size of the database increases.

(4) Simplicity. The fundamental cause of the so-called "software crisis" is complexity, or our human inability to cope with it. There are two approaches to achieving the objective of simplicity: modularization and standardization. Modularization implies developing a database in which the interaction among different components of the database is minimized. Standardization implies use of a limited number of clearly defined tools and techniques whenever possible, and duplicating components of the database system which perform similar functions.

These design objectives can be applied both to the overall database model and to components of the model. If the model is carefully designed, then many of the compromises necessary in achieving the overall objectives can be avoided by assigning different objectives to different components of the design. For example, the efficiency of a system can be localized to the subspace processors, rather than requiring a complex file organization.

Within the model developed from data physics, it is appropriate to apply these design criteria to each component, as each impacts the overall system performance. While the criteria always apply, they have different degrees of importance in the various components, and hence focus our attention on different aspects of design, as we discuss the various components.

Let us first examine these objectives as they apply to the that is, to the data space, the subspace processors, and their controls. In the representation of an individual datum, we regard simplicity and reliability to be of primary importance. Adaptability and efficiency are of relatively little consequence in this representation, assuming more importance in the processing.

As we impose organizations on the data space, particularly practical organizations, the importance of the objectives changes. We find that adaptability becomes quite important, so that the organization can be modified as the system develops in time. Simplicity of organization is also important, although some simplicity may be sacrificed for improved adaptability. The reliability of a particular organization is relatively unimportant, assuming that the processors can overcome any deficiencies in this respect. Finally, the importance of having an efficient organization is heavily dependant on the particular applications for the database. As the applications tend to be more "real time," efficiency of organization assumes increasing importance.

In contrast, for both the subspace processors and their control mechanisms, efficiency seems always to be a key objective, with simplicity and reliability close behind. With the standard design that we have proposed for the subspace processors, adaptability becomes quite unimportant.

Turning now to the user end of the system, we must apply these same objectives to query definition and formulation (that is, to query languages), and to the moving window processors and their control mechanisms. Our knowledge of users suggests that in a query language the key to success is simplicity. The user must be able to state the query simply. Any query language must be adaptable to the individual styles of the users, but efficiency and reliability assume relatively little importance. Most users are rather inefficient at formulating queries, and tend not to ask the questions that provide accurate answers the first time around.

We suppose that any query system that is adaptable to individual user styles will need some form of query analysis and reformulation before it can attempt to provide an answer to the user. For the reformulation process, adaptability becomes of primary importance, with simplicity second. Since the reformulation process will be invoked frequently, it needs to be a relatively efficient process. But because of the general possiblity of interaction with the user, the reliability of this process, in the sense that it must exactly mirror the user's query, need not be very strict.

The final component of our model that we subject to these objectives is the moving window processor. Here, we must consider two levels, since we have postulated that moving windows are aggregates of simple processors. At the unit processor level, simplicity ranks highest as a design criterion, and adaptability ranks lowest. We expect the unit processors to be of a few specialized types, easily replicated, but without any expectation that one of the unit processors will be required to be flexible in its operation. At the aggregate level, however, adaptability is the chief criterion, with simplicity at the bottom of the scale. Not only do we expect to construct rather complex moving windows from the unit processors, but we also expect to require a moving window to adapt to its environment, even to the extent of altering its own configuration. At

both levels, efficiency and reliability are of intermediate importance as design criteria.

In summary, as we see the system design process, simplicity is the key criterion at the basic levels - datum representation, query definition language, and unit processors for the moving windows. Adaptability is vital at the points where the system interfaces with the user's concepts - any imposed data space organizations, query reformulation into terms suitable for system use, and the moving window aggregate. Efficiency considerations are concentrated in the design of the subspace processors, since this is the part of the system that must continually access the data space. And, partly because of the inability of the user to precisely specify queries or the organization of data, the fundamental focus of reliability in the system must be at the level of datum representation. (While we have not specifically singled the control mechanisms, it is clear that on each level reliability, particularly as it relates to system security, is an overriding design consideration.)

Conclusions and prospects.

If the model that we propose is to be useful as a base for information systems design, then it must be capable of reflecting, on the one hand, n-dimensional holographic memories and information-digesting amoebas, and on the other hand, current information systems. Its immediate utility is best shown by considering how current information systems fit into the model.

We have suggested that the data space is multidimensional, and that each datum is a point or a hyperplane in the space. The closeness of this picture to the mathematical concept of a relation suggests that relational database concepts and their extensions are the "natural" ones to use in future work. There are three immediate types of extension to relational concepts. One is full generalization to n-ary relations. To argue that binary relations suffice is akin to arguing that Turing machines are adequate for all computing. True, but painfully irrelevant. The second is the type of extension afforded by the DIAM work of Senko et al[4], into which relational concepts can be nicely embedded.[5] The third extension is to "fuzzy" relations, growing out of the work of Zadeh and others on fuzzy sets and logic.[6] Such an extension might well provide a superior context for retrieval in the face of the users' often fuzzy inquiries.

While the concept of a holographic data space may be a bit futuristic, novel data space management techniques utilizing presently available system components have not been thoroughly investigated. Consider, for example, the microprocessor as a data space organization tool. One might picture a data set, such as a full manuscript, as residing on some computer storage medium, complete with its own microprocessor. When this unit is plugged into an existing system, the microprocessor has the task of deciding where the data belong, and integrating them into the existing data space properly. The microprocessor might then be available for other work within the system, or might be detached and reloaded with new data.

Associative memories, one of the motivating factors behind this series of workshops, have still not been widely implemented in information systems. At the same time, memory technology is evolving rapidly - bubble memories, charge-coupled devices, laser storage, video disks - with relatively little attention being paid to applications of these technologies in information systems.

The subspace processor system readily suggests a minicomputer network, operating on a distributed database. Each processor need access only a portion of the entire data space, but it may be required to do relatively sophisticated operations on that portion of the space. Once we postulate a network system at this level, all of the varieties of network design are available for consideration. Taking into account the security and accountability considerations for the system, certain network configurations will be more attractive for any given information system application. A key issue at this level is that of interaction among the subspace processors. The degree and form of interaction will do much to shape the topology of the network.

One imminent problem is that of interfacing a private information system with a public information network. This interface may take place at either the window control level, or the subspace control level, and in each case the attendant problems of communication, protocol, and security must be solved, along with such mundane problems as charges for service. We give just two examples. A business may query one or more networks for economic and technical information. This is done now, of course, with terminals; and there is no technical reason why the information could not be fed directly into the business's computer for use in its own system. The problem here, other than economic and legal ones, is to create a "logical" network, so that the business does not need to know several different protocols to access information. As a second example, consider the home computer. When the fad fades and the games are gone, these devices, together with the next generation of pocket calculators, will be ready tools for accessing information networks.[7] Despite differences of detail, the same general economic, legal, and technical problems must be saved for this situation as for the business example.

As we have proposed them, the moving window processors take on the appearance of aggregates of microcomputers - devices of limited and perhaps specialized capabilities, but interconnected in a complex and shifting manner. The key issue here is designing a suitable control mechanism. Associative concepts, normally related to memories, find another natural application at this level. It would seem that within the constraints imposed by a particular query, the various portions of a moving window processor enjoy considerable autonomy, perhaps with the controlling influences obeying some sort of inverse square law.

The functional separation of the query process from data space generation and updating clearly shows that the languages used for these processes need not be the same. The possibilities for query languages are myriad, and have been under investigation for some time. We might suggest, as an interactive possibility, that an ELIZA-like[8] system could be used to aid in query definition and reformulation.

There are two principal methods of developing new systems. One is to begin with available systems, modify and enhance them, and thus gradually improve their capabilities. This method offers the possibility of rather quick rewards, but suffers from a limitation of outlook. The other method is to ask, "What would we really like the system to be and do?" This method opens wider vistas. We have suggested some of these here, but there are many others, such as query anticipation, processing of non-literal information, and the many facets of artificial intelligence. However, this approach tends to delay the development of improved systems, since many of the concepts are difficult or impossible to implement with current technology. Nevertheless, we have chosen this latter approach for our work, feeling that if, as we develop our ideas more fully, we can relate them to current technology, they will provide a goal for the development of superior, user-oriented information systems.

References

1. A. Mowshowitz, The Conquest of Will, Addison-Wesley, Reading, MA, 1976. p. 10.

2. H. Draper, "Ms Fnd in a Lbry," Mercury Press, Inc., 1961. Reprinted in A. Mowskowitz, Inside Information, Addison-Wesley, Reading, Ma,. 1977. pp. 178-183.

3. M. Ferguson, "Karl Pribram's Changing Reality," Human Behavior, May 1978, 28-33.

4. M. Senko, E. Altman, M. Astrahan, P. Fehder. "Data Structures and Accessing in Data Base Systems." IBM Systems J., 12, 1, 1973, 30-93.

5. L. Schneider. "A Relational View of the Data Independent Accessing Model." Proceedings of the International Conference on Management of Data (SIGMOD), Washington, D.C., June 2-4, 1976, 57-90.

6. L. Zadeh et al, Fuzzy Sets and Their Applications to Cognitive and Decision Processes. Academic Press, New York, 1975.

7. R. Korfhage. "The Impact of Personal Computers on Library-based Information Systems." SIGIR Forum, 12, 4, Spring 1978, 10-13.

8. J. Weizenbaum, "Contextual Understanding by Computers," Comm. A.C.M., 10, 8, Aug. 1967, 474-480.