PERSONNEL RECORD

Employee Name	Education		Employment		Starting Salary
	School	Degree	Employer	Position	
John Jones	Princeton	BA	Co A	Trainee	$3,000
		MA		Coder	
	Columbia	PhD		Programmer	
	Harvard	LLD	Co B	Programmer	$4,000
			Co C	Senior Programmer	$6,000
				Analyst	
			Co D	IR Specialist	$20,000

FIG. 2. Example of output listing of personnel record

In collating systems, once one has a set of records to be displayed, one has the problem of sorting them in some specified manner. If the field being sorted on appears at most once in every record, there is no problem. Suppose, however, each record from which we have extracted our data has as subject a particular apartment house in some city and suppose that among the data in the record are the names of all the tenants and the address of the apartment house. If the interrogation reads "In alphabetical order, list the names and addresses of all people who live in apartment houses in that city", artificial records are created for sorting, duplicating the address of the house for each person who lives in it. Further complications arise if more than one level of sorting is specified at one time.

Display

Display includes preparation of extracted data for tabular presentation using line printing devices. The retrieved information is extracted as subsets of selected subjects. This information therefore has subject structure and must be printed so as to reflect that structure. That is, replications of a particular sub-subject or field are vertically aligned and are vertically spaced so as not to conflict with each other. This also applies to values and repeated fields that exceed alotted column width. Figure 2 shows a personnel record for John Jones. Column 1 consists of the field "Name" that comprises the subject "Employee". Columns 2 and 3 show the fields "School" and "Degree" of the subject "Education". The remaining columns 4–6 show the fields "Employer", "Position", and "Starting salary" for the subject "Employment". No coordination is made among the three subjects "Employee", "Education" and "Employment" within a single record. The subject "Education" permits replications on the field "Degree". These replications are coordinated with the value of school name. Similarly, "Employment" permits replications on the field "Position". These replications are coordinated with the values of "Employer" and "Starting salary".

Summary

In a data organization for an information processing and retrieval system, the recurring problems of structure and scope pervade all aspects of the processing. This is necessary if the system is to permit the variety of semantic relationships required to effect moderately sophisticated solutions for a class of information retrieval problems. These considerations have suggested a flexibility and variability not yet available in any system or computer language. The problems of constructing such programs are considerable, and while such programs are not at hand, they are on the horizon.

REFERENCES

1. SAMS, B. H. Dynamic storage allocation for an information retrieval system. *Comm. ACM 4* (Oct. 1961), 431–435.
2. CHEATHAM, T. E., JR.; COLLINS, G. O., JR.; AND LEONARD, G. F. CL-1, an environment for a compiler. *Comm. ACM 4* (Jan. 1961), 23–28.
3. MINKER, J. M. Implementation of large information retrieval problems. Gordon Research Conference, New Hampton School, New Hampton, N. H. (July, 1961).

An Information System With The Ability To Extract Intelligence From Data*

T. L. Wang

General Electric Company, Syracuse, New York

Introduction

An information system with the ability to extract intelligence or knowledge from a large mass of data has been developed and instituted. This information system was developed under the auspices of the Minuteman High Reliability Component Program, the objective of which is to improve the reliability of the components used in the Minuteman System by two to three orders of magnitude.

To meet this challenge, it is necessary to collect and analyze eventually about 200 million digits of variable data concerning a specific type of transistor alone. This large mass of data, once collected, should contain almost all the answers that evaluation engineers and statisticians must know in order to improve the reliability of this particular transistor.

* Presented at an Open Technical Meeting on "Design, Implementation and Application of IR-Oriented Languages," held by the ACM Computer Language Committee on Information Retrieval on 20–21 October 1961 in Princeton, N. J.

The problem facing the computer people is an obvious one—namely, to design an information system capable of extracting from this large mass of data the needed answers. Because of the tremendous volume and complexity of the data, the answers must be provided in a highly digested form on a fully automatic basis. In other words we want the system to produce not the rearranged or partially summarized results, but rather to extract the ultimate intelligence from the data.

Before I go further, let me explain what I mean by intelligence when I say we want to extract intelligence from data. Intelligence, as I have arbitrarily defined it, is a piece or pieces of information that offer us more meaning than the numerical values. For example, if we use the figure $10,000 in computing one's income tax, we are using it as a piece of information. If we use it to describe a man, it tells us something more than the fact that this man's salary is $10,000 a year. For one thing, we know this man is not exactly a bum. If we have enough figures, such as 4 years of college, 5 years of experience, 30 years of age and so on, we will have quite a feeling about what kind of man he is. This kind of feeling derived from a single figure or a group of digested figures is what is here referred to as intelligence.

System Concept

An information system with the ability to extract intelligence from data will essentially possess the ability to provide answers to any questions one might have, as long as the raw data pertaining to the specific questions is available within the system. The first problem in designing such a system, however, is the ever present language problem. To expect the system to take any question one might have in the original form of English and proceed to produce the answer is a little beyond the present state of the art. However, if we were to examine how we humans go about finding the answers to a question from a large mass of data, you will find that we essentially determine first the specific data pertaining to a question. Then we retrieve this pertinent data, and perform some kind of analysis to produce some meaningful answer. The hardest parts to automate seem to be in two areas. The first one is to ascertain the specific pertinent data from the original question. The second area is to determine what kind of analyses need to be performed on the pertinent data in order to produce the answer—whether just taking an average on the data, or performing a correlation coefficient or performing some kind of computation. Now if we were to do these two parts manually, we can well expect an information system to do the rest automatically.

Essentially, instead of feeding into this system a question in the form of English, we specify through the use of a fairly rigid retrieval language the specific data pertaining to a question, and at the same time instruct the system to perform a specific analysis with the retrieved data. Let me give you a very crude example here. Let us assume that we have an information system, in which we have collected a lot of data on the personnel within a company. One of the questions we want to ask is this: "Are married men steadier workers than single men?" After examining the question, we know that the data pertaining to this question is the years of service with the company. We also know that the average years of service of married men versus that of the single men will produce some sort of answer to our original question. To get the answer, we simply instruct this system to retrieve the years of service and group them by married and single men and perform an average on each group.

Basically an information system possessing this capability needs three elements. The first one is the collecting and storing of the raw data. The second element is a retrieval mechanism to retrieve the pertinent data involved in an inquiry. The third element is the ability to perform all the different types of computation required on the retrieval data.

System Description

Based on this concept, we have developed an information system to serve our needs. This system, as I have just mentioned, consists of three parts:

1. Data Collection and File Maintenance. The first part is called data collection and file maintenance. Ninety percent of the data is generated on the automatic test equipment and is fed into the system to be organized and updated into master files. A quite elaborate validity checking scheme to detect test equipment or human errors as well as any malfunction of the data collecting system is incorporated. This of course is due to the fact that the entire system is dealing with the reliability of a type of transistor. Hence the reliability of the input data is of utmost concern.

The major breakdown of the master file is a variable-word-length record. Within each record, the data is further organized into four separated levels with one of the levels containing reference type of information which characterizes the data within each record. This type of structuring offers a great deal of freedom in file organization and at the same time provides a much greater flexibility in data retrieval.

2. Retrieval. The second part of the system is the data retrieval. This is really the heart of the system. As mentioned earlier, to produce an answer to a question posed to this system we have to transform the question into the form of the specific data related to it and define to the system the pertinent data through the use of a retrieval language. Since most of the users of this system do not possess any knowledge of computer programming, the retrieval language used must be simple enough so that it can be mastered by any users in a matter of hours.

The basic element of the retrieval language consists of a six-character alphanumerical call letter designating a field of information and its associated value desired, separated by an equal sign. The selection of the characters is such that they are very close to the terminology used and are readily recognizable. These elements, further

separated by comma signs, constitute the basic retrieval language used in retrieving data. All elements within a retrieval request are considered to have "AND" relationships among them. However "OR" relationships are permissible under certain situations.

In addition to these basic relationships, the retrieval language has other capabilities:

(a) It has the capability of grouping the desired output by the magnitudes of any field or any combination of fields. Within each retrieval request, it can group the output in three levels of 8 groups each for a total of 512 individual groupings. In other words, it can construct a three-dimensional table. In the event more than a three-dimensional array is needed, it can do so through the use of connecting retrieval requests.

(b) It also has the simple capability of counting the number of times any specified restrictions are met. This again can produce output counts in three-dimensional arrays. This feature, simple as it might seem, turns out to be the most powerful feature of the retrieval. There are almost unlimited ways this feature can be used in the initial aspect toward producing answers to some highly complex questions.

3. Analysis Monitor. The third and last part of the information system is the part that will take the retrieved data and perform the type of analysis specified to produce the answers desired. It essentially consists of many different analyses in the subroutine form with a control to direct all the retrieved data from many different requests through whatever various analyses each request calls for. The analyses available consist of the basic statistical analyses such as frequency distribution, correlation, regression, analysis of variance, and discriminent analysis, plus some special type of analyses. Additional ones, of course, can be readily added to the system if needed.

Interesting Facts

This information system has been in actual use for almost a year now. The frequency of usage is about twice weekly. About 15 to 20 questions have been processed at each run. Some interesting facts begin to emerge.

1. Generalized Character. This information system turns out to be far more generalized than we originally intended it to be. At the initial system design stage, we were concerned with the design of a system to serve a specific need—namely, to provide intelligence to those who need it in improving the reliability of a specific type of transistor. We had no intention of designing a generalized system that could be used in many other types of applications. However, since we were fully aware of the constant changes that we would have to make, we have designed into the system features that permit us to make any file format or other minor retrieval logic changes without major reprogramming. These change features cover such a wide range that they transform the overall system into an almost generalized system capable of many types of applications where extracting intelligence from a large mass of data is involved.

2. Flexibility of Retrieval. Another interesting hindsight is in the retrieval mechanism. When we first tried to spell out the basic features of the retrieval, we thought if we managed to handle 80 percent of the questions posed to the system, we would be lucky. There are bound to be questions so complex that the retrieval language would not be able to retrieve the pertinent data. However, after processing over 1000 inquiries, some of which were quite complex, we have not yet encountered any questions to which we are not able to retrieve the pertinent data. After examining the retrieval mechanism in the light of all different types of requests, we found that the basic concept of simple elimination used in the retrieval scheme could very well account for its flexibility. Each element in our retrieval language can be considered as a restriction which eliminates all data outside its restricted boundaries. When a series of these elements are linked together with AND/OR logic, they turn out to be an extremely flexible yet powerful retrieval scheme.

3. Analysis Aspect. In the analysis aspect of the system, we also encountered an interesting situation. Again at the initial design stage, we thought incorrectly that to provide meaningful answers to any questions would involve all kinds of computational and analytical capability on the part of the analysis portion of the system. We were thinking that eventually we would have to collect at least over 100 various kinds of analytical or computational subroutines in order to serve the need. We started out to use the system with only 10 basic subroutines, and so far these basic ones seem to have served adequately more than 90 per cent of our need. This we believe is due to the nature of the questions one most likely would want to ask concerning a large mass of data. For instance: "Is a certain fact related to another fact?" "Which is better or the best?" "Which of the many variables that could contribute to this is the real controlling factor?" All these types of questions can be readily answered through the use of the basic statistical analyses. This information system cannot of course handle the "why?" type of question directly. You can, however, get the answer by being specific. By asking a series of "Does this or that have anything to do with a situation?" type of questions, you could eventually get the reason why.

Conclusion

It took us a little over a year's time from the initial system design to the actual usage of the system on a limited basis. Now that the initial rounds of fire-fighting work to keep the system going are over, we are beginning to work on more sophisticated features to expand the total system capability. We know there will be questions that are beyond the system's capability to answer. However, we are protected in this respect. We claim that this information system has the capability of answering any intelligent question the users might have concerning the accumulated data. If we encounter any question to which the system cannot produce some meaningful answer, we can always say it's because the question is stupid.