CANADIAN THESES ON MICROFICHE

I.S.B.N.

THESES CANADIENNES SUR MICROFICHE

National Library of Canada
Collections Development Branch

Canadian Theses on
Microfiche Service

Ottawa, Canada
K1A 0N4

Bibliothèque nationale du Canada
Direction du développement des collections

Service des thèses canadiennes
sur microfiche

NOTICE

The quality of this microfiche is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us a poor photocopy.

Previously copyrighted materials (journal articles, published tests, etc.) are not filmed.

Reproduction in full or in part of this film is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30. Please read the authorization forms which accompany this thesis.

THIS DISSERTATION
HAS BEEN MICROFILMED
EXACTLY AS RECEIVED

AVIS

La qualité de cette microfiche dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de mauvaise qualité.

Les documents qui font déjà l'objet d'un droit d'auteur (articles de revue, examens publiés, etc.) ne sont pas microfilmés.

La reproduction, même partielle, de ce microfilm est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30. Veuillez prendre connaissance des formules d'autorisation qui accompagnent cette thèse.

LA THÈSE A ÉTÉ
MICROFILMÉE TELLE QUE
NOUS L'AVONS REÇUE

NL-339 (r. 82/08)

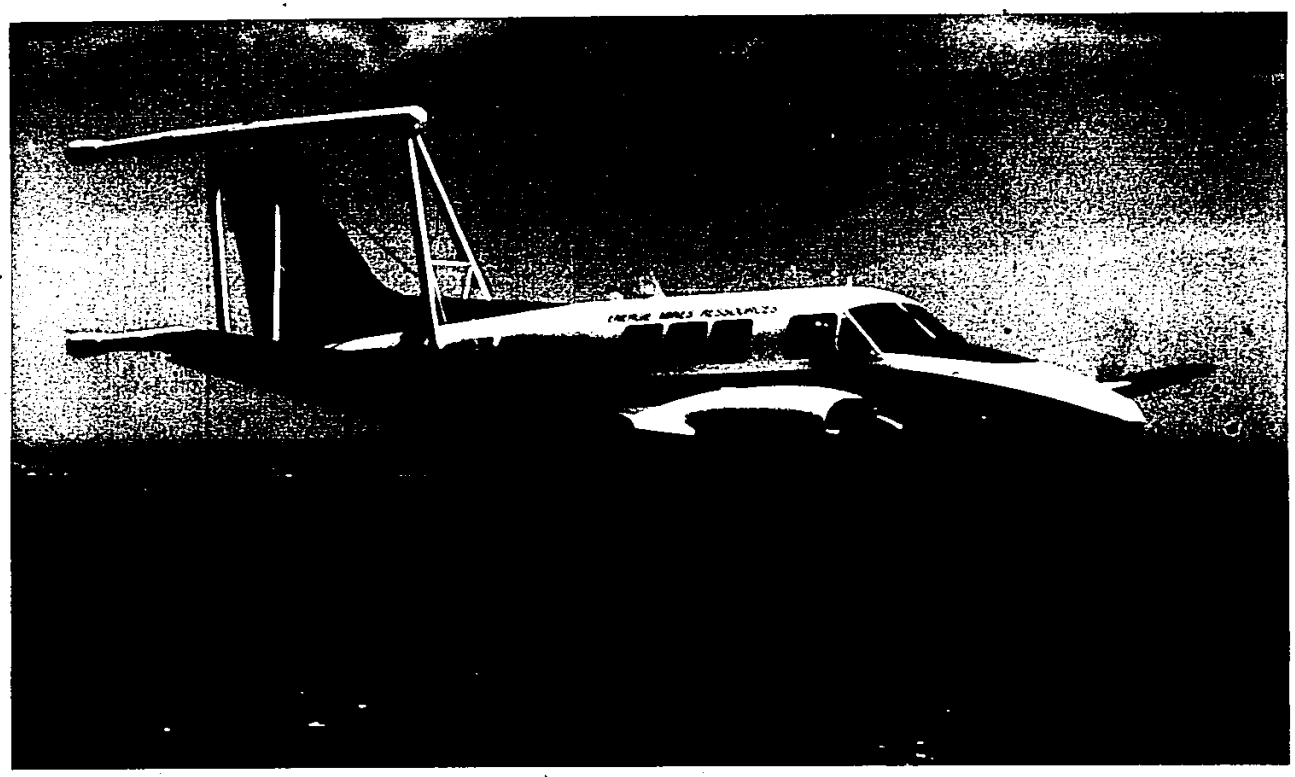
Canada

A SYSTEM FOR AUTOMATED COMPILATION OF EARTH SCIENCE SURVEY DATA WITH SPECIAL PROVISION FOR AEROGEOPHYSICAL DATA

by Michael Thomas Holroyd

Thesis presented to the School of Graduate Studies
in partial fulfillment of the requirements for the
degree of Ph. D. in Geology

UNIVERSITY OF OTTAWA
OTTAWA, CANADA, 1983



FRONTISPIECE. The Geological Survey of Canada
Aeromagnetic Vertical Gradiometer System.
(At the time of writing, the worlds first and only such system.)

Abstract

During the past 15 years earth science survey data compilation has gradually changed from a manual process to one almost entirely computer automated. Computer methods were at first an optional means for improvement of the speed and efficiency of certain phases of the work. They eventually became mandatory as in-flight digital data acquisition devices were introduced and rates of data acquisition increased to the point where manual methods became totally inadequate.

As a consequence of the phased replacement of manual systems with digital ones, and the pressing need to avoid interruptions to the work, the resulting software is highly specialised. Software systems are incompatible between different organisations compiling the same kind of data and even between systems for different kinds of data within one organisation. The result is wastage of manpower in maintaining, learning to use, and improving the great variety of existing systems, and wastage of man and computer time in converting data to a form compatible with the peculiarities of each particular system.

Comparative analysis of highly specialised compilation systems from several different earth science disciplines reveals that processes do exist at the fundamental level which are generally applicable to a range of disciplines. Hence a generalised compilation system capable of eradicating the problems inherent in multiple specialised systems is at least conceptually feasible. The most serious obstacle to its realisation, however, is the diversity of data content and structure among the various disciplines. To surmount this obstacle, a system with a great degree of physical and logical independence of data from software will be necessary.

Existing methods of achieving such independence within data base management systems are found to be largely inapplicable to earth science survey compilation. The principle reasons being the very large quantities of data involved and significant differences between the data retrieval requirements of compilation systems and data base management systems.

It is found that, although the data contents and structures vary greatly between the various compilation systems examined, a general, abstract, model can be constructed which adequately represents all of them and which incorporates the means for achievement of data independence.

The logical data structures of current data base management systems are based variously on relational calculus, set theory and graph theory. The compilation data structure model is based on simple algebra with the addition of some components of vector algebra.

It is further found that simple algebraic manipulation of the model expressions faithfully simulates the actual data manipulation processes which are applied by the various compilation systems.

Furthermore, unlike the logical models of the data base management systems, the logical structure of the compilation data model also represents the actual physical structure exhibited by the data in its most common, sequential form.

The properties of the model permit symbolic simulation of actual compilation systems by which means processes common to more than one system can be more easily recognized. As a result a relatively small number of abstract processes can be identified which when implemented as software modules would, in different combinations, serve the basic compilational needs of a broad variety of earth science disciplines, thus becoming the sought after generalized earth science compilation system.

P R E C I S

Au courant des dernières quinze années, la compilation des données de relevés géoscientifiques a graduellement changé de procédés manuels à des méthodes presqu'entièrement automatisées par ordinateur. Au début, les méthodes par ordinateur ne servaient qu'à augmenter la rapidité et l'efficacité de certaines phases du travail. Elles devinrent bientôt mandataires à mesure que l'acquisition des données numériques en-vol furent introduites et que les taux d'acquisition des données augmentèrent au point où les méthodes manuelles furent rendues totalement inadéquates.

En conséquence du remplacement par stages des systèmes manuels par des systèmes numériques, autant que par le besoin pressant d'éviter les interruptions au travail, le logiciel qui en résulte est hautement spécialisé. Il y a incompatibilité de progiciels entre différentes organisations compilant la même sorte de données et même entre systèmes pour différentes données en dedans d'une même organisation. Il en résulte un gaspillage de main-d'oeuvre dans l'apprentissage à, le maintien et l'amélioration d'une grande variété de systèmes existants. On gaspille aussi le temps de la main-d'oeuvre et de l'ordinateur en convertissant les données à une forme compatible aux particularités de chaque système.

L'analyse comparative de systèmes de compilation hautement spécialisés appliqués à plusieurs disciplines géoscientifiques révèle que des procédés existent qui sont généralement applicables à une variété d'entre elles. Par conséquent, il est possible, au moins conceptuellement, de créer un système de compilation capable d'éliminer les problèmes inhérents aux systèmes spécialisés multiples. Néanmoins, l'obstacle le plus sérieux à sa réalisation est la diver-

sité du contenu et de la structure des données parmi la variété de disciplines. Pour surmonter cet obstacle, il serait nécessaire de créer un système dont les données seraient au plus haut point logiquement et physiquement indépendantes du logiciel.

Les méthodes permettant cette indépendance qui existent déjà en dedans du progiciel de maniement de fichier central se trouvent le plus souvent inapplicables à la compilation des données de relevés géoscientifiques. Les principales raisons en sont l'énorme quantité des données en cause, et les différences importantes entre les exigences de récupération des données de systèmes de compilation et le progiciel de maniement de fichier central.

En dépit de la grande variété du contenu et des structures, on trouve qu'un modèle général et abstrait peut être construit, représentant d'une façon adéquate toute la variété des systèmes de compilation examinés et incorporant les moyens d'arriver à une indépendance des données.

Les structures logiques des données de logiciels de maniement de fichier central courants sont fondées d'une façon variée sur le calcul relationnel, la théorie des ensembles, et la théorie graphique. Le modèle de structure des données de compilation a pour fondement l'algèbre simple avec l'addition de certains composants de l'algèbre vectorielle.

Il se trouve aussi que la simple manipulation algébraique des expressions du modèle simule fidèlement les procédés réels de manipulation des données qui sont appliqués par les systèmes de compilation variés. De plus, au contraire des modèles logiques de progiciels de maniement de fichier central, la structure

logique du modèle de compilation des données représente aussi la structure physique actuelle démontrée par les données dans leur format séquentiel le plus commun. Les propriétés du modèle permettent la simulation symbolique de systèmes de compilation réels grâce à quoi les procédés communs à plus d'un système peuvent être reconnus plus facilement. Il en résulte qu'un relativement petit nombre de procédés abstraits peuvent être identifiés. Utilisés en tant que modules de logiciel, et en différentes combinaisons, ils serviraient à satisfaire les besoins de compilation de base d'une grande variété de disciplines géoscientifiques et deviendraient ainsi le système de compilation géoscientifique généralisé qu'on recherche.

Acknowledgments

The writer wishes to thank the supervisor of this work, Dr. F.P. Agterberg, and the staff of the Ottawa University Department of Geology for their guidance and assistance; Dr. A.G. Darnley, Director, Resource Geophysics and Geochemistry Division, Geological Survey of Canada, whose support permitted this work to be carried out and recieve financial support as an official G.S.C. project; and the staff of the Geological Survey word processing centre who typed this work.

CONTENTS

Page

List of Illustrations

Frontispiece and Figures 2, 5, and 9 reproduced by by permission of the Geological Survey of Canada.

I SUMMARY

Computer automated compilation and cartography of survey data is carried out in all branches of the earth sciences. With the exception of certain graphics software packages, the compilation systems employed are highly specific to the particular discipline. Indeed, most are specific to the particular organisation.

Three significant disadvantages result from this situation.

Firstly, there is an enormous duplication of effort in the creation and maintenance of the many separate systems. Secondly, any scientist engaged in multi-disciplinary work must become familiar with as many different systems as there are disciplines involved. Thirdly, the form and content of digital information is usually radically incompatible between different systems.

A consequence of this last fact is that it is difficult to the point of impossibility to create a general earth science data bank. At a far less ambitious level, simple exchange of one type of information between two organisations generally involves a significant expenditure in time and money to convert the data to a form compatible with the recipient organisation's system.

To solve these problems would require the creation of a generalised software system. Such a system would be capable of applying the common processes of compilation and cartography to a broad range of earth science data types.

As a result of having passed through the same system, the data structures of end results would be mutually compatible, thus permitting the creation of a general data bank. Duplication of effort in creation and maintenance of multiple systems and in learning to use them would be eliminated.

To determine the feasibility of creating such a system, the first questions to be answered are: "Are there such things as 'common processes' of compilation and cartography when referring to a broad range of earth science disciplines? If so, what are they?"

This work begins by examining existing automated compilation systems as applied to five different earth science disciplines. Namely, drift sedimentology, aeromagnetics, airborne gamma-ray spectrometry, regional geochemistry and gravimetry. Each system is described in the same manner as it originally appeared in published literature. Comparison of the descriptions and associated flow diagrams in search of common features reveals a complex pattern of helpful similarities and obstinate differences, e.g. from the fundamental principles involved, gravity and magnetics are closely related; from the operational point of view aeromagnetics is most closely related to airborne gamma-ray spectrometry; yet airborne gamma spectrometric results are essentially geochemistry! Very few truly common features can, however, be found.

It is contended that the reason for the apparent lack of common features is not intrinsic but stems from the point of view of those who created and described the individual systems. They were often created as an automated model of pre-existing manual systems and are described in terms of the earth science practices and objectives.

When regarded from a more general view point, four fundamental categories of process appear which are common to all of the compilation systems. Namely - verification, display, manipulation and development. These categories are shown to be natural rather than artifically imposed at a later stage of the work when a symbolic notation is developed for describing data structures and processes.

On examination of these categories, the first three show themselves to be readily amenable to generalisation. The last category, development, appears at first to be much less amenable but further investigation shows that many development processes are largely made up of a mixture of the other three types of process, and only a small residue remains that is highly specific to a particular discipline and not easily generalised.

Having recognised and surmounted the obstacles to the recognition of generality within autocompilation systems, the next obstacles to be faced are those which lie in the way of implementation of general processes. The three major obstacles are, restrictions imposed by computer capacity, diversity of machines and programming languages, and diversity of data structures. The first problem is being solved automatically as computer capacities continue to increase at an exponential rate. The second problem can be overcome by writing software in the standard version of a widely used language. The third problem presents by far the greatest difficulties. It could perhaps be solved, or rather circumvented, by imposition of a rigid data structure convention. This however would be very difficult to apply to the broad range of data types involved and even if possible would hamper generality and future development rather than promote it.

The problem is that of "data independence". It is a well known problem not restricted to the disciplines considered in this work. The most successful solutions to the problem have been made in the fields of generalised data base management systems. Such systems permit effective retrieval of data from a data base with the actual physical and logical structures of the data base being almost completely independent of the user's concerns. Data base concepts and practices are examined to

determine their relevance to the current problem. It is concluded that data base systems as implemented are not suitable for application to the current problem because of two fundamental differences between the types of data and retrieval requirements of data base systems as compared to earth science survey compilation, i.e. i) earth science compilation systems must deal with much greater volumes of data and ii) although retrieval requirements vary between different compilation systems and between different phases of any one system, within one phase of one system, they are predictable in advance.

In order to surmount this obstacle, a prerequisite is a detailed examination of the fundamentals of data content and structure. Published works exhibit a confusing and conflicting mixture of terminology with regard to data content. This work, therefore, establishes and defines what the writer considers to be a necessary and sufficient set of terms.

The next part of the work is concerned with earth science data structures.

A data structure system can always be simple. If natural systematic relationships exist within the data a more complex structure can be developed.

"Sufficient natural relationships" means that all the records of the data set are not entirely independent and different from one another, i.e. whereas some part of each record must be unique and independent of all other records (otherwise it is just a duplicate) some parts of the record may possess features which are common to many other records, e.g. "all samples have different elemental composition but many have the same gross lithology".

The stated advantages of the more complex structures are improvements in storage and retrieval efficiences. More complex structures are more difficult to

create and maintain, however, hence the best type of structure for a particular need must be an appropriate compromise.

One complicating feature of earth science data which is rarely mentioned in works on data management is the existence of multi-dimensional data. This, plus other factors peculiar to earth science data results in a variety of data structures which are practical and efficient but which are difficult to describe in terms of the standard graphic forms of "flat file" "tree" and "network". They are described in this work by a simple symbolic notation.

With the graphic form, data items are represented as nodes on the tree or network and relationships as lines linking these nodes. The network type of structure appears to be a physical simplification over the flat file, as the number of nodes is less in the former. In the actual data set, however, the relationships must also find some mode of physical expression. For some kinds of earth science data sets, the improvement in storage/retrieval efficiency promised by structural development may be unobtainable in reality.

The drift sedimentological data file is used as a pertinent example. A fully developed network structure for this type of data could, perhaps, produce about a 50% reduction in storage for data items and, in theory, a great improvement in retrieval efficiency.

The practical realities of the retrieval process, however, show that the improvement in retrieval efficiency would be much less than expected.

Many earth science data sets and processing needs appear to be incompatible with the structural concepts of most data base management systems. The reasons for this are that most, if not all, data base management systems were created with the inherent characteristics and processing requirements of business data in mind. The characteristics and processing requirements of earth science data differ significantly from those of business data.

The root of the problem, however, lies much deeper.

Diagrams of networks, through the human sees them as a simplification of the problem, are two dimensional. Even more complex structural relationships are depicted graphically as perspective drawings of three dimensional systems. The computer is, however, one dimensional. All multi-dimensional concepts and structures must be mapped into one dimension by the analyst who creates the algorithm to implement the model. The computer also demands everything in explicit numerical form. The conceptual simplification of the data set attained by a network or other type of diagram is deceptive.

The eventual implementation of a model will have more in common with the original unsophisticated flat file data set, than either of them have with the intermediate, multidimensional conceptual model. Hence in certain circumstances, advantage may be gained by keeping the conceptual stage closer to the line between the simple data set and the simple mind of the computer.

The next section of the work concerns itself with creation of an effective but linear model of data structure. A model, furthermore, which is capable of adequately describing those data structures which were difficult to describe as graphic structures.

The model developed allows description of a data set by a simple algebraic expression, which can be written in a highly summarized and condensed form. The relationships between data set components, which were difficult to describe by graphic structures are found to be fully describable by matrix algebra operations. Furthermore, formal algebraic manipulation of these expressions accurately models the changes in data structure that take place within compilation processes. The originally chosen, four basic compilation processes are shown by the model notation to be a natural classification. Display involves no output (in digital form). Verification involves output with no change in physical structure or nominal content from the input. Manipulation involves output in which the physical structure but not the content has changed. Development produces output of new content.

By simple notational conventions, all pertinent characteristics of a data set (level of structural development, attribute content, physical order, access routes, etc.) can be explicitly and concisely expressed. Manipulative processes need no explanatory text as the particular manipulation is evident from the notation.

Each of the compilation systems described at the beginning is described again using the model notation. When described in sufficient detail the existence of many common features becomes clearly evident.

The final part of the work is concerned with implementation of such a system, i.e. the design of a software system based upon the model structure.

II FOREWORD: The need for a generalised, earth science, computer automated compilation system

When seeking to determine the contents and mechanisms of the lithosphere in depth, unlike many other scientific tasks, both direct observation and simulation by experiment are largely ruled out. The former because of the location of the subject matter. The latter because of the enormous physical and temporal scale of the subject and its mechanisms. Hence hypotheses regarding the structures, contents, and mechanisms of the lithosphere have had to rely upon limited direct observations combined with as many indirect observations as can be made.

With the tools of modern science and the advent of automated data acquisition systems the abundance and variety of indirect observations have become almost an embarrassment.

Just as it has become physically impossible to manually compile and map the data as fast as it is acquired, it is becoming intellectually impossible to correlate and reconcile all the different types of observations that may be applied to a given subject.

With information in digital form, all clearly definable repetitive tasks could be handed over to the computer leaving the human largely free to pursue only the essentially intellectual tasks that remain. To accomplish this would, however, require computer programs to perform those tasks suited to automation.

At the current state of development, compilation and mapping software exists within each separate discipline. With the exception of certain graphics software packages, however, the compilation systems employed are highly specific to the particular discipline. Indeed, most are specific to the particular organisation.

Three significant disadvantages result from this situation. Firstly, there is an enormous duplication of effort in the creation and maintenance of the many separate systems. Secondly, any scientist engaged in multi-disciplinary work must become familiar with as many different systems as there are disciplines involved. Thirdly, the form and content of digital information is usually radically incompatible between different systems.

A consequence of this last fact is that it is difficult to the point of impossibility to create a general earth science data bank. At a far less ambitious level, simple exchange of one type of information between two organisations generally involves a significant expenditure in time and money to convert the data to a form compatible with the recipient organisation's system. To solve these problems would require the creation of a generalised software system. Such a system would be capable of applying the common processes of compilation and cartography to a broad range of earth science data types.

Duplication of effort in creation and maintenance of multiple systems, and in learning to use them, would be eliminated. As a result of having passed through the same system, the data structures of end results would be mutually compatible, thus permitting the creation of a general data bank, and facilitating multi-parametric processes.

A generalised computer automated compilation system for the earth sciences is needed. The objectives of this work are to identify and surmount the obstacles that lie in the way of the realisation of such a system.

1.0 Automated Compilation in the general context of the Earth Sciences

Let us begin by defining some of the terms in the title of this work:

i) Automated compilation

This term, in the current context, means:

"Bringing together the separate components of the data once they are in digital form and performing by computer processing methods all the manipulations and computations necessary to create the final, correct, mappable data set."

A "mappable" data set is composed of records each of which contains spatial coordinates defining a point, line or plane within some cartographic reference frame and also data defining one or more earth science parameters which relate to the defined spatial element.

The principle objective of the automated compilation system is the elimination or minimisation of all errors, aberrations and disturbing or irrelevant influences from the spatial coordinates and associated earth science parameters.

In the strict sense "automated compilation" excludes all activities required to create or acquire the initial digital data sets and excludes the subsequent cartography of the final mappable data set. As will be seen, however, automated compilation systems must include cartographic capabilities to produce maps in, or close to, the final desired form as an adjunct to the detection and correction of errors or aberrations.

ii) Earth Science

Those disciplines whose basic objectives are the definition of the current and historical contents and structures of the lithosphere and of the mechanisms of change which resulted in these contents and structures; and

to which survey methods are applicable, i.e. geology, geophysics and geochemistry. Hence, in the current context, those sciences concerned with the definition and interpretation of the features of the biosphere or atmosphere, the physiography of the earth's surface, or the earth's interior at great depth (geography, geodesy, demography, meteorology, planetary physics etc.) are excluded.

Figure 1 illustrates the position of automated compilation with respect to the other components of a complete earth science system.

In the work "computer software for geographic data handling" (Anon, 1975, p. 2), the ADAM system (Holroyd, 1974) is referred to as a "full geographic information system". Figure 1 makes the distinction that the ADAM system, which is for automated compilation of aeromagnetic data, and similar systems are not geographic information system but precursors to such systems. Many, highly complex and sophisticated geographic information systems exist. Several modern systems are described in "Mapping software and cartographic data bases" (Anon, 1979). Blakemore and Lloyd (op. cit. p. 11) describe digitizing and display procedures for spatial data related to industrial planning. Douglas (op. cit. p. 57) critically compares different ways to represent the same or similar spatial data sets. McEwen (op. cit, p. 136) describes the acquisition of digital cartographic data by the U.S. Geological Survey and states that (with respect to the eleven different types of mapped data considered)

"Digitization will be done either from available separates of these maps or during their compilation".

The salient points to be noted about these examples quoted (which accurately represent the general contents of the work cited) is that all begin either with a

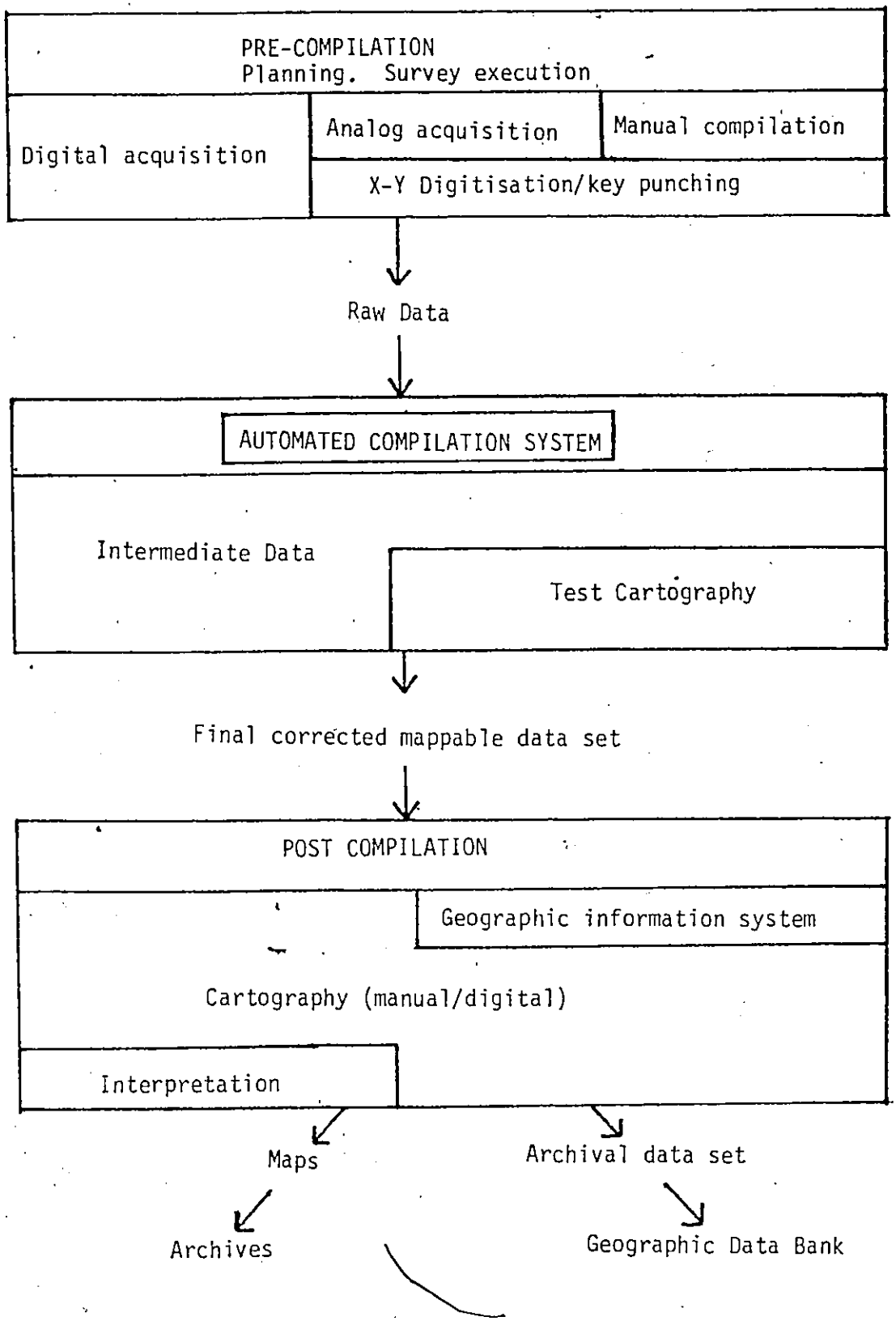


Figure 1. Autocompilation in the context of a complete Earth Science Survey System.

mappable digital data set or by digitisation from a pre-existing map or spatial representation (Photos, etc.)

"Compilation" in these cases consists of acquiring or digitizing the data then correction of errors.

Geographic information systems are not, however, restricted solely to the manipulation and re-mapping of geographic data in different forms. They are often concerned with the integration and interaction of several different types of data over a given region and the development by complex and lengthy computation of new types of data better suited to the objectives of the users and interpreters. The characteristic of all such systems however, is that the significant work begins after the acquisition of the mappable data sets.

The characteristic of an automated earth science compilation system, conversely, is that the significant work takes place between the acquisition of the data and the output of the final, correct, mappable data set.

Not all of the five examples described in the following sections fit exactly into this category. The drift sedimentology system for instance is more closely akin to a geographic information system as its compilational aspects consist almost totally of error detection and correction of key punched information. It is included mainly for comparison and contrast with the other systems and because of the peculiar problems created by the diversity of its data content.

The two aero geophysical examples and the land-gravimetry example, however, fit very well into the auto-compilation category.

All three begin with decidedly un-mappable data. The principle data components of the aerogeophysical systems even begin without spatial coordinates at

all and a great deal of computational effort is expended in deriving coordinates for the in-flight measurements.

A further characteristic of aerogeophysical data – one which creates its own special problems – is the enormous bulk of the data involved. Newly developed devices (see section 3.4) are capable of data acquisition rates of up to 30 megabytes per hour, and are likely to be in use many hours per day throughout the field season.

Even with the more conventional devices very large quantities of data are acquired as a result of flying regional surveys of great areal extent. Problems which would appear to be simple (sorting data into geographical order, transposition of survey network matrices etc.) with moderate amounts of data, require complex and specialized solutions. The Goias survey, for example (Anon, 1977) involved over 300,000 line kilometres of survey flight, recording both aeromagnetic data and four channels of radiometric data. At compilational costs (average computer service bureau rates) in excess of $20 per line kilometre (S.D. Dods, personal communication) for this type of data, it would have cost over $6 million in computer time alone to produce the final mapable data sets. It did not cost this amount in fact, as the volume of work involved in digital compilation makes computer service bureau charges prohibitive. Hence all Canadian aerogeophysical companies now employ in-house computer systems. The equivalent service bureau cost, however, serves to illustrate clearly that the central portion of figure 1 is not disproportionately large compared to the pre- and post-automated compilation phases, and that such systems are a worthy subject for investigation.

1.1 Examples of computer-automated compilation systems for Earth science survey data

Five examples of automated compilation systems are described from various Earth science disciplines. Namely, drift sedimentology, aeromagnetics, airborne gamma-ray spectrometry, gravimetry and regional geochemistry. The scientific background to each discipline is briefly described, followed by a description of the compilational requirements and a flow chart of the compilation systems.

The compilation system descriptions are presented in the same manner as they originally appeared in published literature – or in the way that the workers who created and employed the systems described them in personnel communication.

One of the flow charts – gravimetry – represents a "generalised" system. i.e. one whose intent is to show the features common to all or most gravimetric compilation systems. The remainder represent specific systems actually in use. Moreover, these systems are all in use within the same organisation, – the geological survey of Canada, this fact serves to emphasise the contrasts between the systems.

The examples vary significantly in their contents and complexity and are not presented in any special order. The aeromagnetic compilation system is described in the greatest detail, partly because this example is the most complex of the ones chosen and partly because it contains several components which are similar to, and can therefore be used as illustration of, problems in other disciplines. i.e. the processes of standardisation and merging of in-flight and flight-path data need not be described for both aeromagnetics and airborne gamma spectrometry as they are almost identical.

1.1.1 Airborne gamma ray spectrometry

Gamma ray spectrometry involves the detection, and measurement of the energy of, gamma rays passing through space. Certain ranges of energy or "windows" are selected and the number of gamma rays in each energy range are counted over an interval of time.

The origin of gamma radiation and its relevance to the Earth Sciences is described by Kogan et al (1971, p. 1-47, p. 256). In summary, gamma rays result from radioactive decay of atomic nuclei. The energy of the gamma ray is dictated by the particular decay mechanism and is hence characteristic of the element from which it is emitted. The three elements responsible for the major part of natural gamma radiation are Uranium, Thorium and Potassium. The most common arrangement in a gamma ray spectrometer is therefore three windows covering the characteristic gamma ray energy ranges of these three elements. A fourth window is also employed covering the whole range of natural gamma ray energies. This is known as the "total count" window,

- Airborne gamma ray spectrometry is well known as a valuable reconnaissance tool in the search for uranium deposits (Grasty 1975, p. 503). It is not restricted to such an application however. Grasty ((1). 1976, p. 257) describes its use for water equivalent snow measurement: Gamma rays emanating from radioactive minerals in the ground surface are absorbed in proportion to the mass of water in the snow cover. Comparison of the gamma ray intensity over bare ground and over the same terrain after snow cover allows direct calculation of the water-mass of the snow. Not only is this method more rapid than taking spot samples it also provides a more even coverage and measures water mass directly.

The usage of gamma ray spectrometry most generally related to earth science is direct classsification of outcrop lithology from airborne gamma ray spectrometry data. Newton and Slaney (1977) developed a classification algorithm by examination of the characteristics of gamma ray profiles over several known rock types in the Hearne lake area NWT. When the classification algorithm was applied to other profiles in the region, 63% of the measurement points gave correct identification of the rock type beneath them.

The instrumentation and methods for a typical gamma spectrometry survey are described by Richardson et al. (1972). The instrument consists of an array of "Doped" crystals of sodium iodide. "Doping" means the addition of a small quantity of impurity (usually thallium) to the crystal to modify the quantum energy levels of the latter. When a gamma-ray photon is absorbed by an atom in the crystal, the atom enters an "excited" energy state. It decays to its base state by emission of a photon of visible light. The energy of the emmitted light is proportional to the energy of the original gamma ray. The light is detected and amplified by a photo multiplier tube producing an electrical pulse whose voltage is proportional to the energy of the incident light and hence to the original gamma-ray. Each such pulse constitutes one "count" of specific energy. The usual practice is to compile the counts in each of the four windows (U, Th, K and total count) over successive one-second intervals. The counts and a fiducial are recorded on digital magnetic tape. The aircraft flies at about 200 km per hour at a terrain clearance altitude of approximately 150 metres.

Survey area coverage is by parallel flight lines whose separation distance varies according to the type of Survey (5 to 25 km for regional reconnaissance, 500 metres to 2 km for more detailed local surveys). Either inflight photography or a video system is

used to record pictorially the terrain over which the aircraft passes. A fiducial corresponding to the one recorded on the digital magnetic tape is also recorded on the film or video frames. The film or video record is later compared with a topographic or photomosaic map. Where recognisable terrain features allow correlation between the in-flight record and the map, the points are plotted on the map and annotated with the corresponding fiducial number. The survey track information thus recovered is converted to machine readable form by use of a digitising table.

At this point, all the information necessary for compilation and mapping of the survey data is in digital form ready for submission to an automated compilation system. The data however will have to be submitted to a series of lengthy and complex processes the compilation processes – before it is in mapable form. The ultimate objective is to produce "Equivalent" uranium, potassium and thorium maps of the area surveyed. That is, maps that show ground surface concentrations of these three elements as calculated from the observed gamma ray flux at the flight altitude.

Fig. 2 shows an equivalent uranium map. This map also represents the systematic regional coverage of Canada by airborne gamma spectrometry.

As well as the three equivalent element maps mentioned, maps are also produced of the total count and the U/K, U/Th and Th/K count ratios. A second method of presentation is also employed. "Stacked" profiles of the count rates and ratios are produced to permit detailed examination and comparison of the corrected data. Fig. 3 shows such a stacked profile set.

Airborne gamma radiation measurements are, however, subject to several sources of error or disturbing influences. Furthermore as it is a flux that is being measured, the count rates observed depend on the size and geometry of the detector

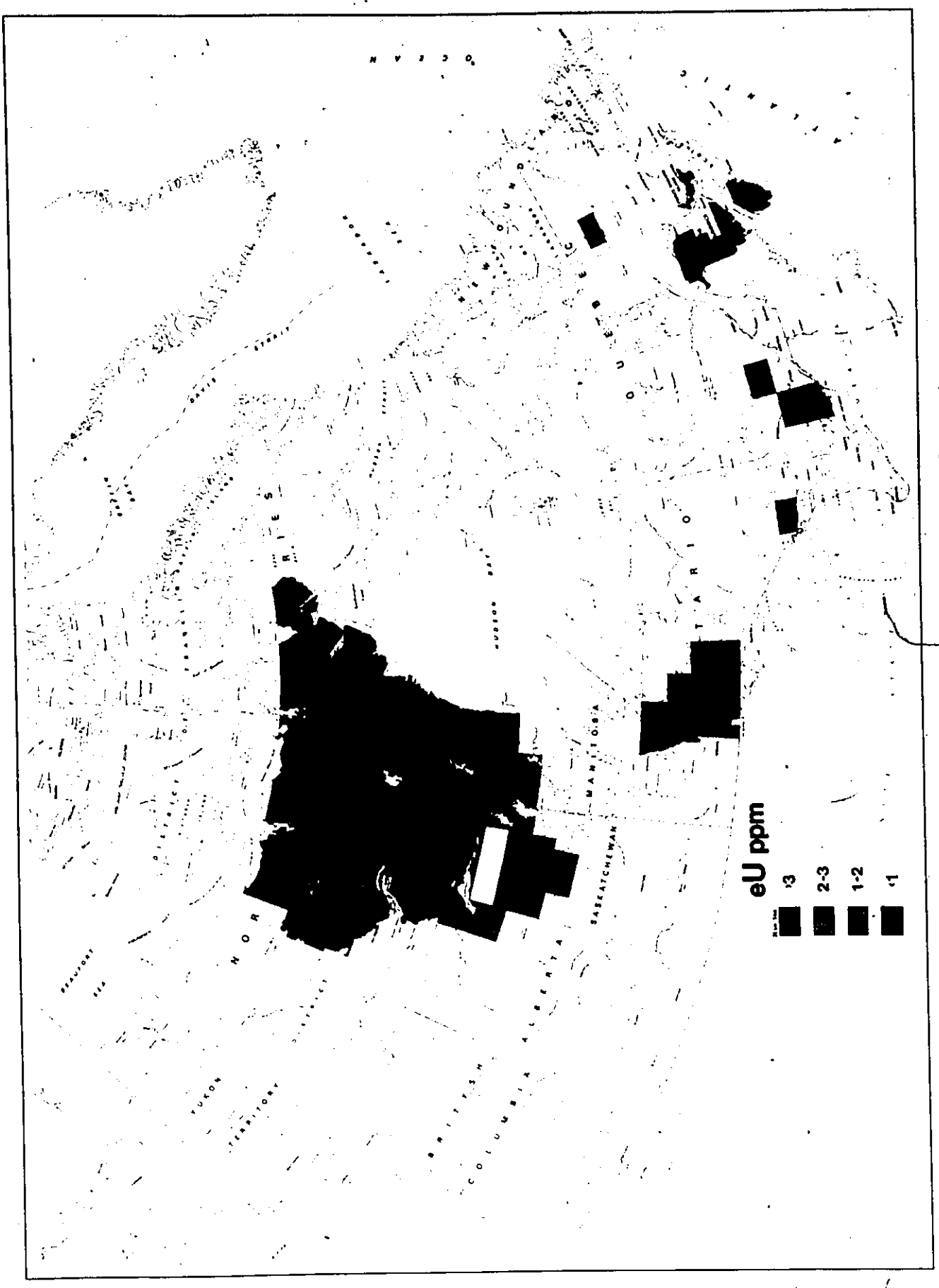


FIG.2: Airborne gamma-ray spectrometric coverage of Canada
equivalent uranium map

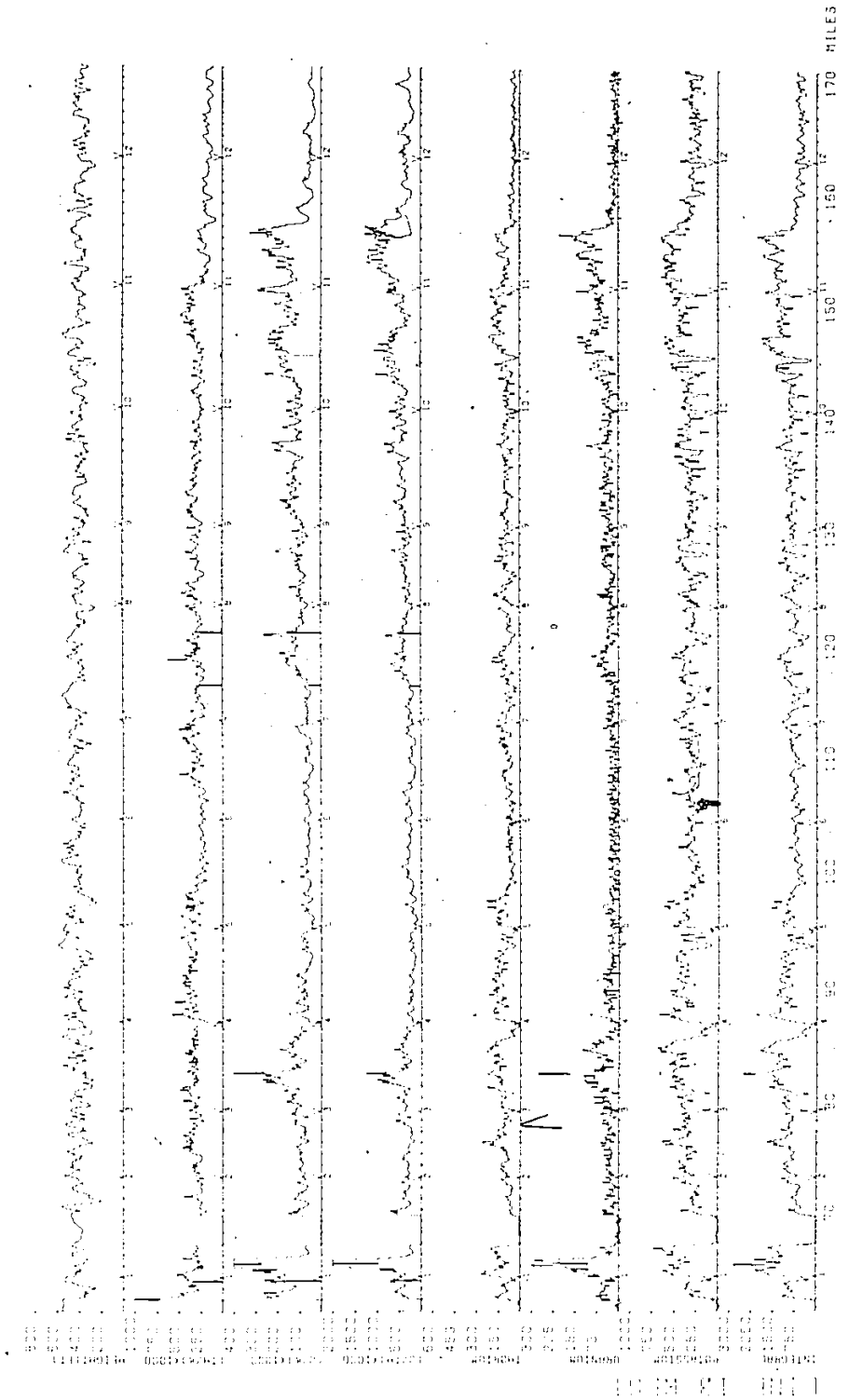


FIG 3: Stacked profiles of airborne gamma spectrometry data

and hence the instrument must be calibrated against a known standard. The disturbing influences and sources of errors of observation are described by Briener et al (1976) and Grasty (1976, 2) describes the calibration procedures.

Three principal disturbing influences exist. The first is distortion of count rates due to "scattering" effects. Gamma rays in their passage through the ground surface, atmosphere and material of the detector may undergo what is known as "inelastic scattering". A gamma ray may lose part of its energy on collision with an atomic nucleus. It will therefore no longer bear the energy characteristic of its parent element and could lose just sufficient energy to appear to have originated from a different element. Thus not only is one count lost from the parent element window but a spurious count could be added to the window of another element. During compilation a theoretically derived "stripping ratio" or "spectral" correction is applied to remove, or at least reduce, this effect.

The second disturbing effect is detection of counts originating from sources other than the ground surface. Sources such as atmospheric particulates, atmospheric radon gas, cosmic rays and trace radioactive elements in the aircraft and instrument fabric. This effect is known as "Background radiation" unlike the stripping ratio corrections, background corrections cannot be calculated from theoretical bases and must be determined by observation. Water is a good absorber of gamma radiation and contains relatively few radiation sources itself. Hence measured radiation above a body of water such as a lake or river should consist mainly of background. Accordingly, during survey flights, whenever the aircraft is passing over a body of water a special code is manually entered into the digital data. During compilation all such background data is extracted and employed to calculate a background correction for the actual survey measurements made in the intervals between over-water flight.

The third disturbing influence is due to variations in the distance between the ground and detector. As distance from a radioactive source increases, the radiation flux density will of course decrease as an inverse square. The Flux will also decrease as distance increases due to increased absorption of gamma rays by the atmosphere. Absorption will depend on the density of the air, which is a function of temperature and pressure. If such variations were not accounted for then each variation in detector – ground distance due to topography or non-level flight would produce a corresponding change in measured count rates. Accordingly "altitude" and "temperature/pressure" corrections are calculated and applied so as to correct the count rates to those that would be observed in level flight over flat ground.

A detailed description of airborne gamma spectrometry compilation is given by Grasty (1974) in the Geological Survey of Canada Airborne gamma ray spectrometry data processing manual. The flow chart provided and the description of the programs, allows construction of a chart showing the general features of automated compilation of airborne, gamma ray spectrometry data (Fig. 4).

Although of scientifically complex origins, and vital to the creation of the mapable data set, the actual calculation and application of the corrections, is a straight forward and simple process. It occupies only a small part of the compilation system as a whole. The other processes on the flow chart, though possessing deceptively simple descriptions in their flow chart boxes tend to be more complex in their implementation than the correction process.

Corrections are applied in the same manner to all members of the appropriate data set. Such processes as extraction and standardisation of data sets, detection and correction of errors and combination of flight path and inflight data require choices between several alternative courses of action depending upon the outcome of an

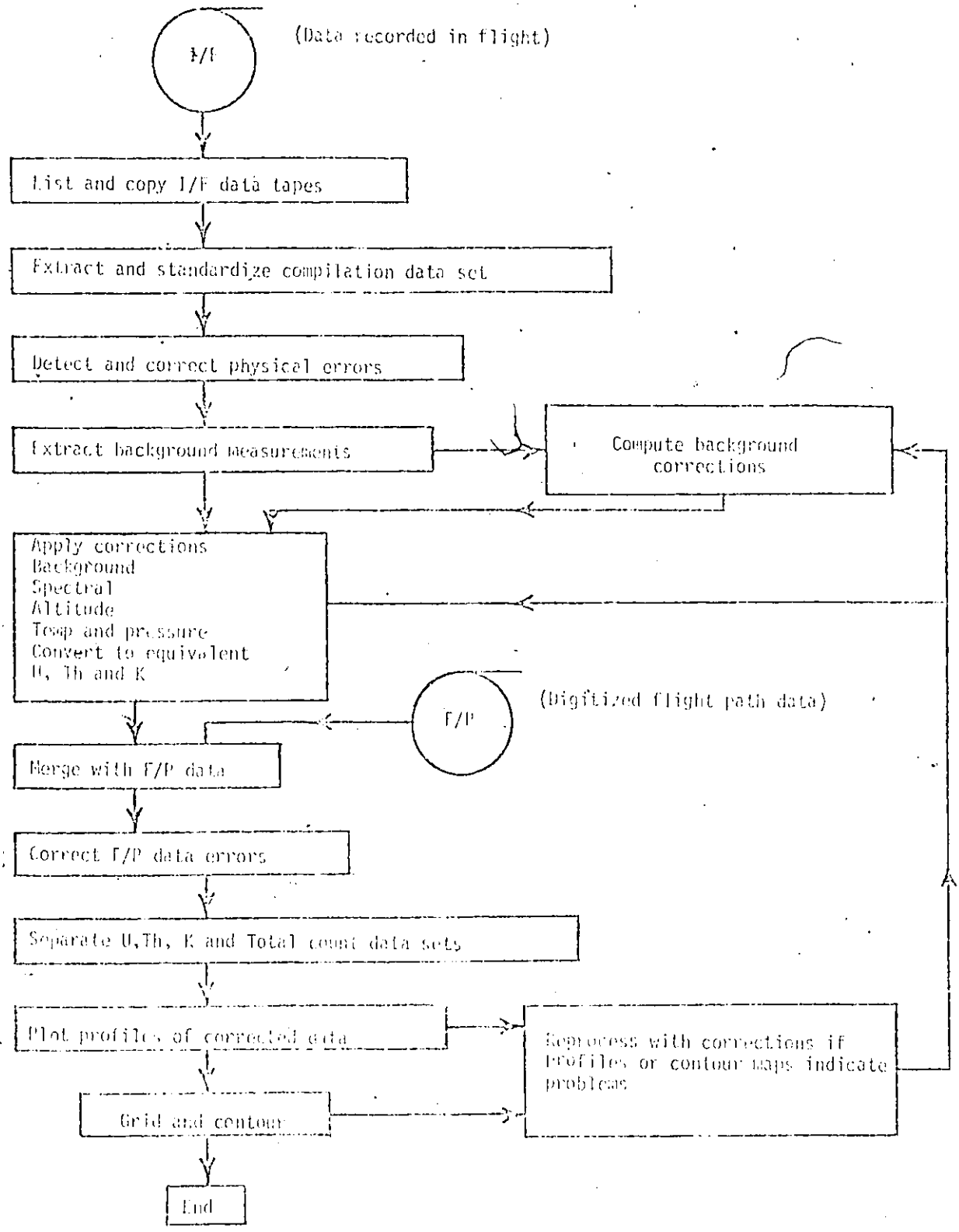


Fig. 4 General flow chart of the gamma spectrometry auto-compilation system as used by the Geological Survey of Canada.

automated logical decision making process. The most complicated of these processes is merging of flight path and in-flight data. As this process is virtually identical to that applied to aeromagnetic data, detailed description of it will be left until the section dealing with aeromagnetics.

1.1.2 Regional geochemical surveys

Mineral exploration directly by geological prospecting may only be applied to exposed material. Drift covered bedrock is inaccessible. Even with abundant outcrop the method has its problems as the size of an exposure of economic mineralisation may be very small in comparison to its non-economic surroundings. In order to improve the chances of discovering economic mineral deposits, means must be applied to determine "targets" – areas with a higher than average probability of economic mineralisation. Debouched ground waters and rain runoff contain minute quantities of dissolved elements representative of the average elemental composition of the rocks with which they have been in contact. As streams merge, their elemental compositions become averaged such that when the main drainage channel is reached, composition of its waters should reflect that of entire watershed. By systematic sampling and analysis of streams and lake waters, areas of higher than average concentration of sought-after elements can be mapped. Thus providing targets for exploration and improving the chances of success by avoidance of potentially unfruitful areas.

As well as providing targets for other exploration methods, the method itself can be applied to regional target areas - sampling at smaller and smaller intervals to "zero in" on the location of an individual deposit.

Less direct and more generally applicable to the Earth Sciences is the statistical analysis of geochemical results. Multivariate analyses can establish

characteristic patterns of trace element composition to distinguish between rocks which are mineralogically similar but of different provenance.

Airborne gamma spectrometry and aeromagnetics with their high rate of data aquisition and in-flight digital recording are obvious candidates for autocompilation. In fact the current rates of survey coverage by these techniques make autocompilation mandatory. One would not, however, normally consider chemical analysis results to fall into the same category, as the data aquisition rate is much slower and the compilational process is much simpler. In the time taken to analyse a suite of stream sediment samples by wet chemistry methods for (say) 10 elements, an airborne multi-channel gamma spectrometer could have covered at least 1000 km of survey traverse and have recorded on magnetic tape over 1 million characters of information. The advantages of computer processing are not, however, solely restricted to the most bulky of data sets. Furthermore, the rates of aquisition and analysis of geochemical data are constantly increasing as helicopter mounted automatic sampling devices and automated analysis methods come into greater use. Bristow (1975) describes a computer controlled geochemical analysis system employing an atomic absorption spectrometer. This device is capable of analysing up to 200 samples in a run, at a rate of approximately 100 samples/hour.

Geochemical surveys are principally used as a reconnaissance tool for economic mineral exploration. Cameron and Hornbrook (1976) for example, describe the application of the method to reconaissance for uranium in the Canadian Shield. Figure 5 shows a uranium map of the lake superior region. The data aquisition techniques employed by the Geological Survey of Canada are described in detail by Garrett (1974). They are, in essence, as follows. When samples are collected, a

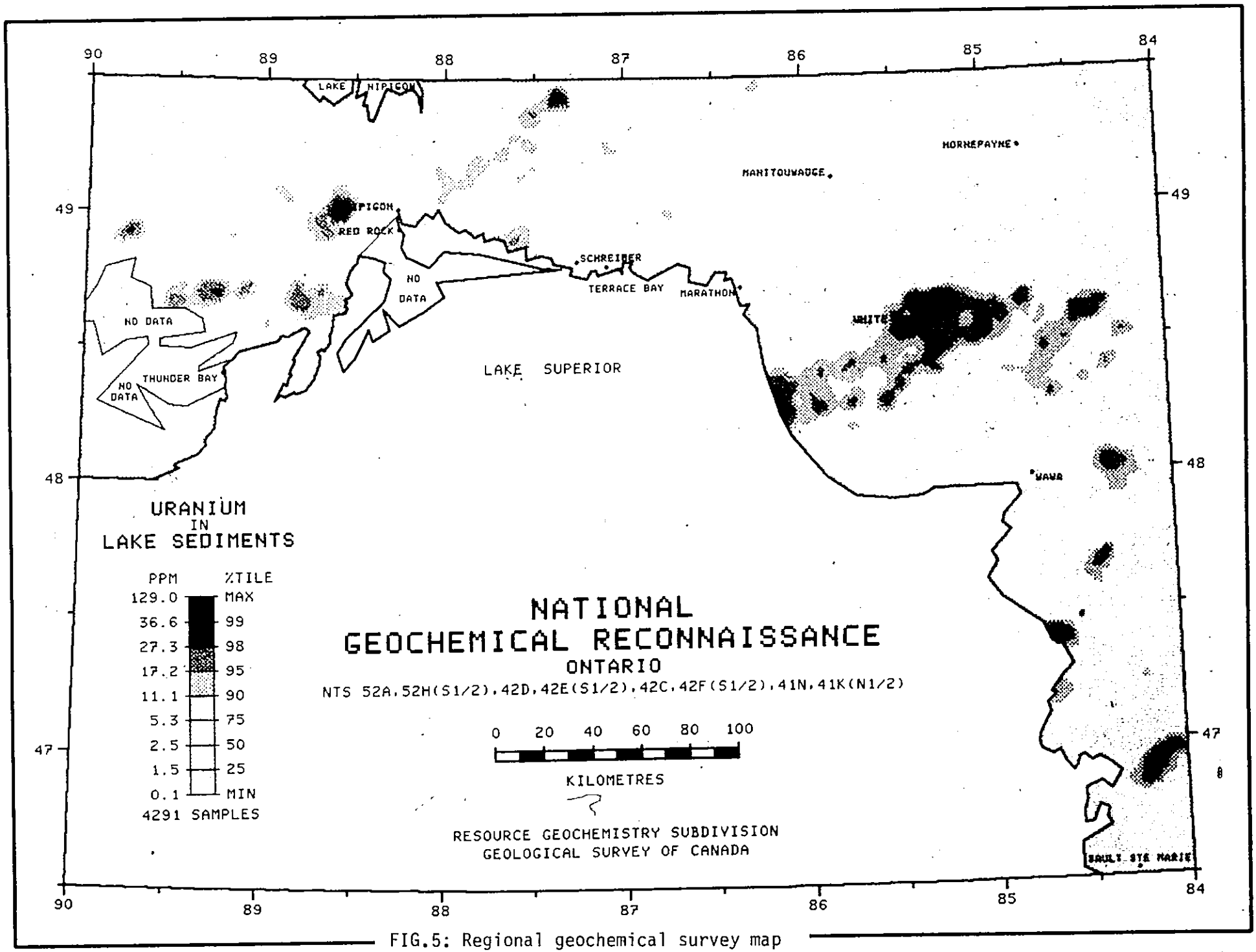


FIG.5: Regional geochemical survey map

"field card" is filled out. This card contains the sample number and such information as the altitude of the sample site and description of the immediate environment of the sample. On return to the office, the field cards are keypunched. Sample sites are plotted on maps of the survey area and their coordinates converted to machine readable form on a digitising table. Wet chemistry analysis results are written onto coding forms and keypunched. Atomic absorption spectrometry analysis results are produced by an automated spectrometer and emerge in machine readable form on digital data cassettes. Thus the data necessary for compilation and mapping is originally resident in four separate data files. Namely the field card, station coordinate, wet chemistry and AA files. These files become available at different times with their contents ordered differently. The components of each file are indexed by sample number which is used as the fiducial to correlate and combine the information.

The compilation process for geochemical data is briefly described at the start of Geological Survey of Canada open-file 506 (Lund, 1977). From this source and D. Elwood (Pers. comm.) it is possible to construct a flow chart of the geochemical compilation system employed by the geological Survey of Canada (Fig. 6). The first stage of the compilation process is to combine the field and the station coordinate files to produce a file containing all information auxilliary to each sample.

The wet chemistry results, though derived by batch analysis for one element at a time, are written on to coding sheets and hence can be arranged by sample number before keypunching.

The atomic absorption spectrometry results on the other hand, are output directly in machine readable form. Like the wet-chemistry analyses, however, they

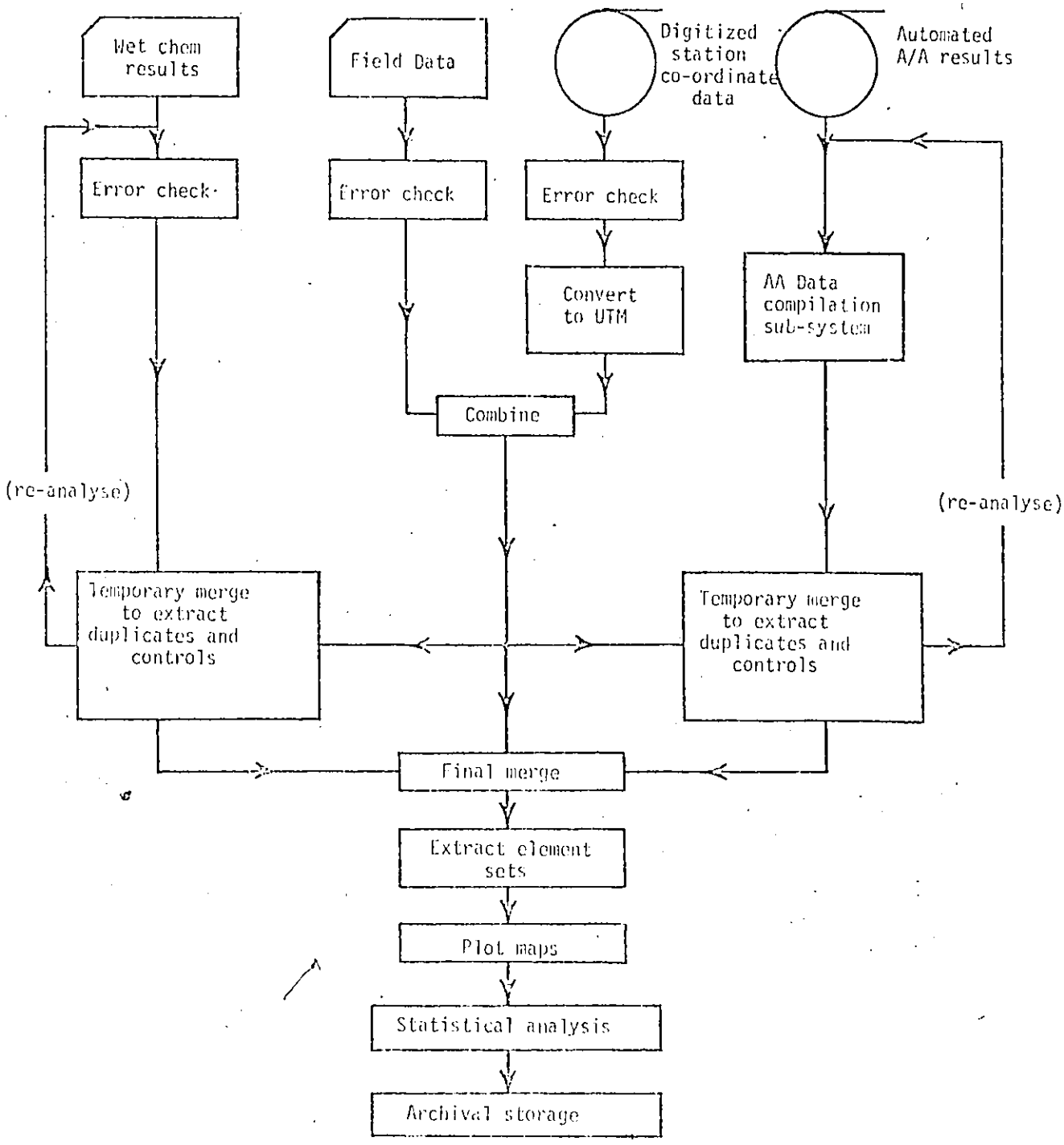


Fig. 6 Flow chart of the automated geochemical compilation systems of the Geological Survey of Canada.

are performed in batches for one element at a time. An automated compilation sub-system is therefore required to verify and rearrange the results in the requisite order. This sub-system is described by Holroyd (1975) and depicted in flow chart shown in Figure 7. A brief description follows.

In the laboratory, the analyst splits some samples into two parts and introduces "control" samples of known content and "Blank" samples which are simply pure water. These additional samples are for quality control and for determining the instrument drift.

The data from each element batch are checked for "physical" errors (miscoded, missing or superfluous characters) and a special index word is created for each result. This index word contains in highly condensed form, the Sample number, the element name and a "flag" which indicates whether the result is routine, control, duplicate, or contained on error. When sorted by the index word a file emerges with routine analyses first, controls second, all duplicates third and all erroneous values last. Within each group the results have been re-arranged by sample number. (i.e. all results for any given sample number occur consecutively) The next stage removes the index word and creates a output file in the standard form required for entry into the main compilation system. Statistical analysis to determine the overall quality of the results may also be made at this stage.

As well as the duplicates and controls inserted in the laboratory, the original samples supplied to the laboratory also contain "blind" duplicates and controls to enable the recipient of the results to assess the quality of the analyses. Every sequence of twenty sample numbers contains only 16 routine samples. The first

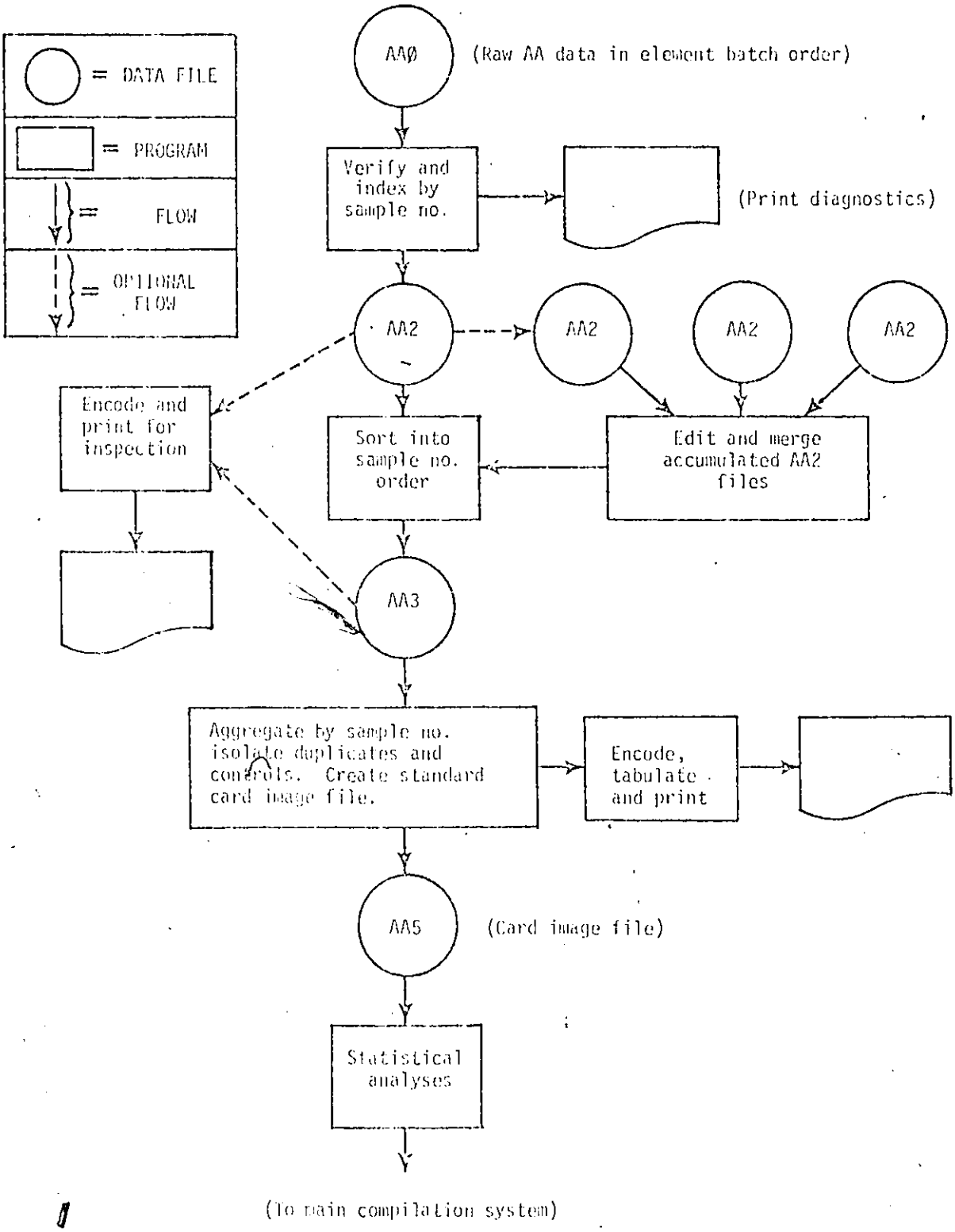


Fig. 7 Auto-compilation sub-system for automated atomic absorption analysis data.

member of the remainder consists of one "field duplicate". (a sample as gathered in the field is put into two separate containers and annotated with different sample numbers). A second, similar, duplicate is made from another sample when the samples are being prepared for analysis. This also receives its own sample number. The last two members which make up the set of twenty sample numbers consist of control samples of known compositions. The identity of the controls in the suite of 20 samples are not discernable from the samples numbers. Only a special anotation on the field card distinguishes between routine and control samples. Hence the analyst is unaware of the identity of the control samples.

The first major task of the compilation system is therefore to extract the analysis results of duplicate and control samples so that duplicates may be compared with each other, and control results compared with the known content of the control sample. The task must be performed rapidly as any unacceptable discrepancies must be reported and the affected samples sent for re-analysis. Extraction of control and duplicate analyses requires a temporary merging of the auxiliary and analysis file to identify these types of sample from the annotation on the field cards.

When all analysis results are received, the three files are merged to produce the final data set. From this file all results for a particular element can be extracted, maps produced and statistical analyses made.

1.1.3 Drift sedimentology

The detailed study of recent unconsolidated sediments as a geological and specifically, resource exploration tool rather than for glaciological, geomorphological, or engineering geological purposes is a comparative newcomer to the Earth sciences.

Dreimanis (1976) describes the origin and properties of glacial tills, Shilts (1976) describes the history, development and current techniques of mineral exploration by investigation of contents and properties of glacial till. Granath (1978) describes a study of Heavy minerals in placer deposits in northern Sweden. Mineral exploration in drift covered areas has generally applied methods to "penetrate" the drift. Aeromagnetic measurements, for example, to which non-magnetic drift is "transparent". The drift cover is hence regarded as an obstacle to be overcome by indirect methods of assessing the likelyhood of bedrock mineralisation. Drift sedimentology has directed its efforts to employing the drift as an active contributor to mineral exploration rather than as an obstacle. Investigations of the processes of transport and deposition of drift and of the physical and chemical changes which take place in the transported material are aimed at providing a mineral exploration tool. If observed properties of drift material can be correlated with the probability of mineralisation of the source rocks and transportation processes sufficiently well understood to allow tracing back to the source, then exploration targets can be established by these means. In order that the method be broadly applicable it is necessary that it not be restricted to the simple, obvious, cases. Very few areas exist where simple observations can quickly lead to economic mineralisation. e.g. a fan of clearly visible and highly mineralised boulders pointing directly to a source a few kilometers away. Hence a great variety and quantity of data must be compiled and correlated for the majority of cases where such fortuitous simplicity is absent.

As with regional geochemistry, drift sedimentological data is presented to the compilational processes already in, or almost in, mapable form unlike the more indirect observations of gravimetrics, magnetics, and radiometrics, which require complex and extensive processing to convert the observed data into mapable form.

Drift sedimentology is, however, intriguing to the computational scientist for the enormous complexity of its information. The paper by Shilts (1978) presenting the results of a detailed sedimentological study of a single stratigraphic section clearly demonstrates the complexity of observations to be made and correlated. The intent of this study was to determine the sampling frequency and distribution and the parameters to be measured from each sample so as to adequately characterise a particular till sheet. The writer concluded that for the particular section studied, twenty one separate parameters were significant. Each parameter however could be made up of many separate observations and measurements. "Contrasts in Ni Content" for example, requires analysis for Ni at many points distributed over the section. Sutterlin and Gwynn (1971) in an assessment of the applicability of computer methods to the analysis of heavy mineral data from tills employed three categories of information with regard to each Sample; namely - "locational/auxiliary", "Bedrock geology" and "heavy minerals". These three categories together contained over 30 data elements descriptive of each sample. The Geological Survey of Canada drift sedimentological data file contains five categories of information (Shilts, pers. com.) – Identity and auxiliary information, geochemical, mineralogical, geotechnical – physical and storage locations. Each of these categories may contain many sub-categories. The complete set of entries describing but a single sample could consist of over 300-data elements. (See section 4.2 below)

All five categories of information are acquired by keypunching from written coding forms and entered into the digital data bank. Fig. 8 shows a flow chart of the compilation process.

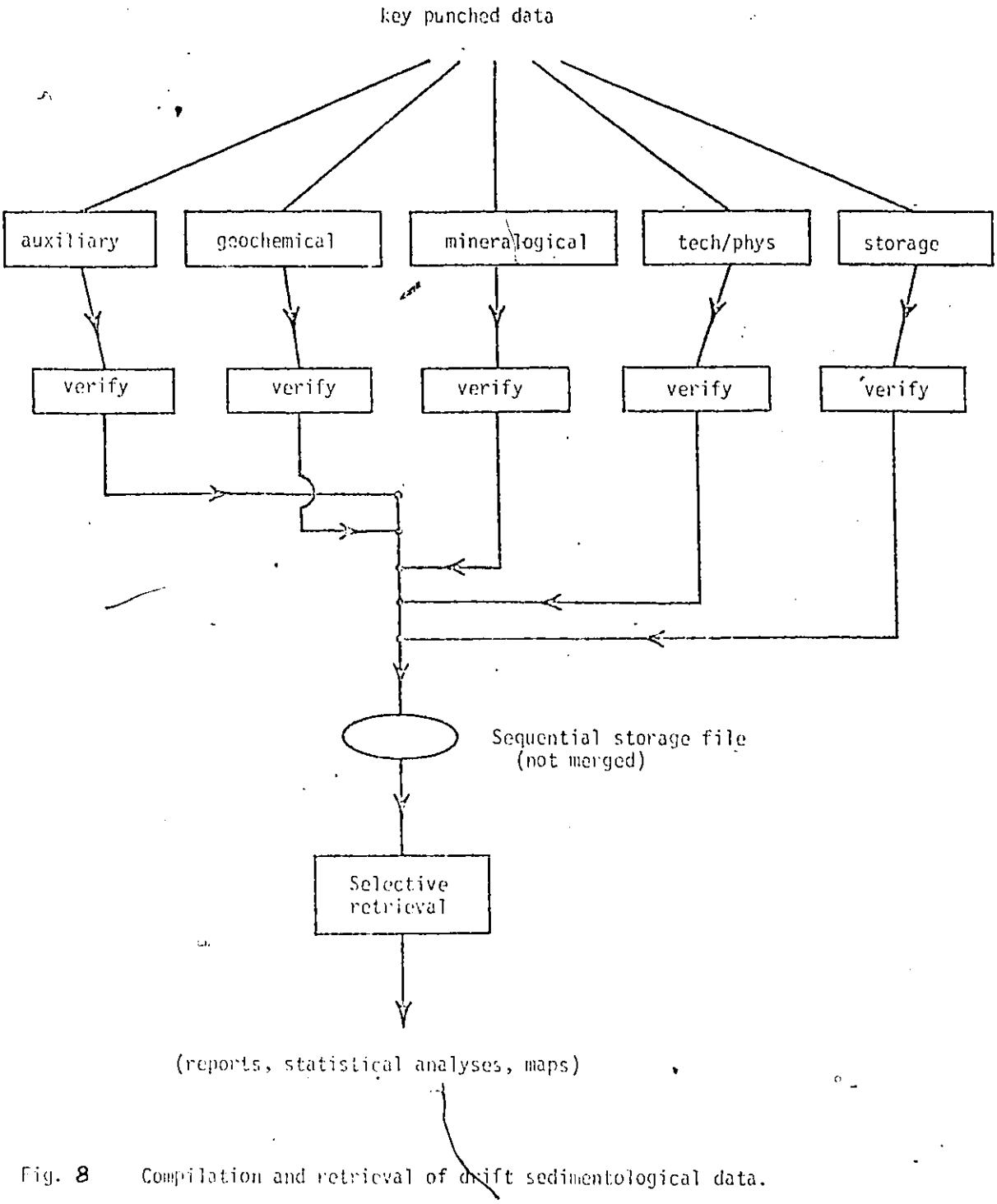


Fig. 8 Compilation and retrieval of drift sedimentological data.

After verification of the correctness of the keypunched data, the five sub-files are stored sequentially. A data retrieval system is employed to retrieve sample data for reports, maps and statistical analyses. The retrieval system must be capable of selection of information according to complex logical conditions. Sutterlin and Gwynn (1971) give an example which requires the system to find all samples whose rock type was gabbro or diabase or andesite or basalt or diorite, and then output a specified selection from the stored information pertinent to these samples.

The problems created by the complexity of the data descriptive of a simple sample are compounded by duplications of samples. As with geochemistry, field samples may be split and the parts analysed separately as a quality control measure. Added complexities arise from the fact that a sample may be re-analysed several times for the same elements if results are of questionable accuracy or different parts of the sample may be subjected to different analyses or tests. Hence, the physical information structure of the digital data files must be sufficiently complex to match the logical complexity of the information.

1,1.4 Aeromagnetics

The earth's magnetic field will induce magnetisation in a ferromagnetic body. The magnetized body then possesses its own magnetic field. Hence the magnetic field strength in the vicinity of the body will consist of two components; the earth's main field and the field of the magnetized body. Field strength is defined as the force exerted on a unit positive pole in the field. Magnetisation of the body consists

essentially of the creation of magnetic dipoles throughout the body, each dipole aligned parallel to the inducing field. In the interior of the body, however, the positive pole at the end of one dipole is nullified by the negative pole at the start of the immediately following dipole, hence only on the surface of the body can poles exist which actively contribute to the body's magnetic field - negative poles on those elements of surface through which field lines enter the body; positive poles on those surface elements through which field lines exit. The body and its magnetic field are therefore dipolar. Hence the total field at a point in the vicinity of the body is the vector sum of the contributions from the earth's field and that of the body.

Thus any deviation of field strength from that due solely to the earth's field is a magnetic "anomaly" - an effect presumed due to the magnetisation of sub-surface bodies.

The purpose of aeromagnetic surveys and compilation is to produce maps of these anomalies.

Interpretation of aeromagnetic data seeks to define the sub-surface lithological contents and structures which are responsible for the anomalies.

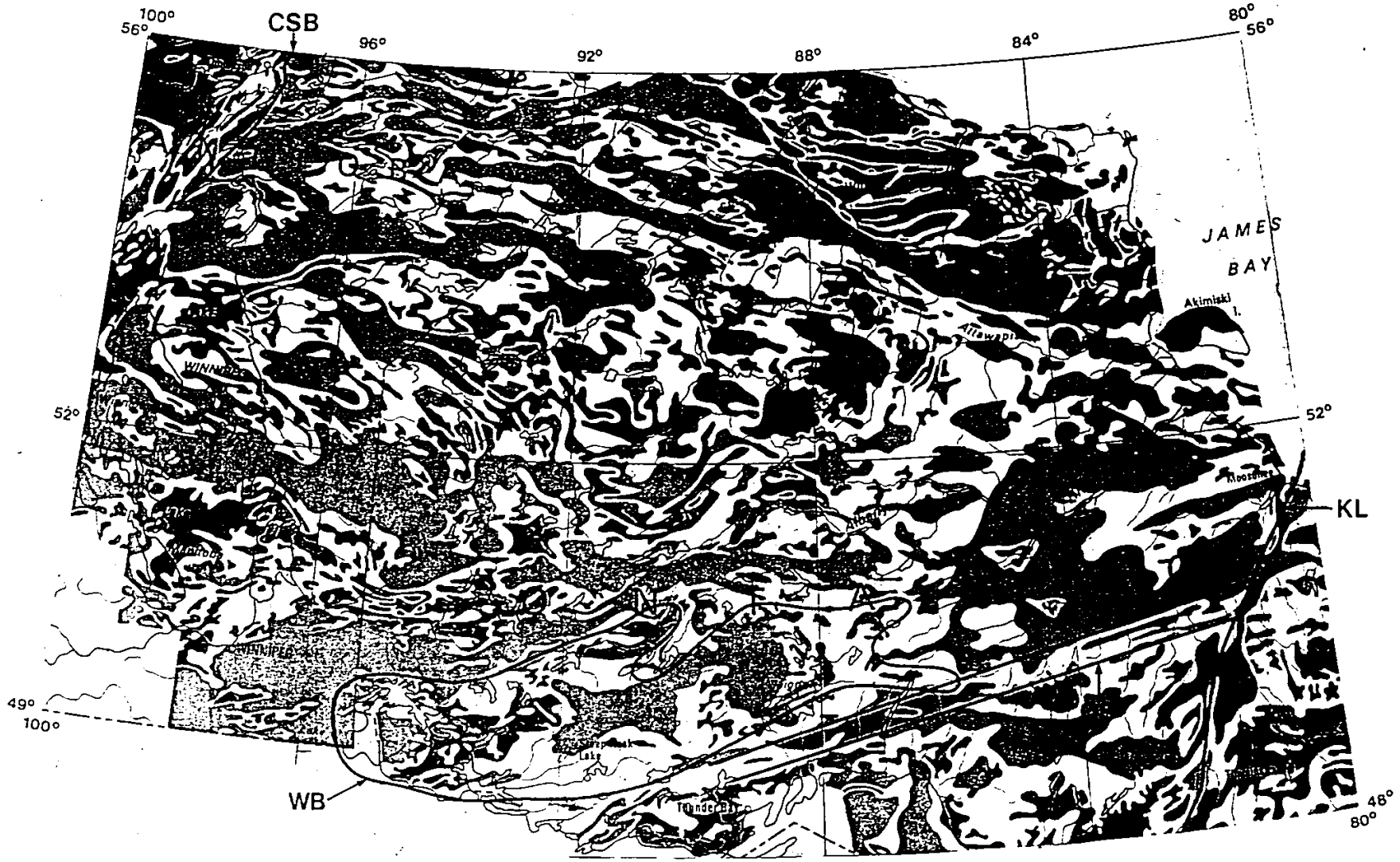
Two basic levels of interpretation are carried out. The first level attempts visual correlation between the patterns visible on the aeromagnetic map and known lithological units or common types of structure, e.g. diabase dykes mapped in a region where they have exposure at the surface may be traced beneath regions of overburden by noting the magnetic "signature" associated with the dykes in the regions where they are exposed and following this signature across the overburden region. Certain characteristic features may be recognized even when no adjacent exposed region

exists for direct correlation - for example, sudden termination of a group of linear anomalies and recommencement of the anomalies after a lateral offset can be interpreted as a fault. Figure 9 shows structural correlations made upon an aeromagnetic map.

The second level attempts to define the size, shape and location of bodies by mathematical treatment of the observed magnetic field values (For example, McGrath and Hood, 1973, p. 349). The results of such methods, however, must be viewed with caution as magnetic (and electrical and gravitational) fields are "non-conservative", i.e. for any distribution of dipoles a unique magnetic field distribution can be calculated. Given a magnetic field distribution however, an infinity of different dipole distributions exists, each of which could produce the observed field. Thus careful boundary conditions must be established and stated in such work, e.g. "If the causative body is a dyke dipping at 75° then the depth to the top surface will be so-and-so."

Igneous and metamorphic basement rocks contain a relatively high proportion of ferromagnetic material, principally magnetite. (The average magnetic susceptibility of such rocks is from 6 to 3000 times greater than that of sedimentary rocks). The igneous and metamorphic basement rocks are a major object of study within the earth sciences. They are, however, mostly covered by drift and/or later sedimentary rocks, but the very processes of their burial beneath overburden also has provided a means whereby they may be observed, albeit indirectly. Although magnetite is highly magnetic, the iron bearing compounds produced from it as a result of weathering processes generally are not. Non-magnetic material is essentially transparent to magnetic fields, hence magnetic measurements mostly reflect the magnetite distribution within the basement rocks regardless of thickness of overburden. As

FIG. 9: Residual magnetic anomalies of central Canada south of Hudson Bay; WB – Wabigoon volcanic belt, CSB – Boundary between Churchill and Superior geological provinces, KL – Kapuskasing Lineament, and QS – Quetico structural zone. Red > +200 γ, yellow 0 to +200 γ, green 0 to -200 γ, blue < -200 γ;



magnetite content generally varies with lithology and a magnetic anomaly occurs only over a change in magnetisation, then magnetometric maps outline significant structural boundaries within basement rocks. Any lithological interface across which the magnetic properties did not change would not, of course, produce a magnetic anomaly. Most lithological differences, however, are also characterised by a difference in magnetic susceptibility. In fact, high resolution magnetometry is often capable of resolving lithological differences not previously detected by other means. Hood et al (1976) interpreting the results of an aeromagnetic gradiometer survey were able to delimit zoning within the White Lake granite in greater detail than previous non-aeromagnetic surveys.

Dobrin (1960, p. 263) describes the instruments and fundamental principles of magnetometry. Prior to the mid 1940's, the only instruments available were the dip-needle and the magnetic field balance. These devices were cumbersome and required a great deal of time and a highly skilled operator to make a reading. As they had to be carefully leveled and oriented they could only be used for land surveys. Aeromagnetic surveying suddenly entered the Earth Sciences after the second world war as a fortuitous offshoot of anti-submarine warfare. Although the fluxgate magnetometer had little success in locating the belligerent magnetic sources for which it was designed, it proved of great value to the earth sciences. Wyckoff (1948) describes the initial development and deployment of this device.

The mid 1950's saw the introduction of the nuclear magnetic resonance magnetometers, the proton magnetometer and later the optically pumped vapour magnetometers. Hood (1971, p. 422) describes the principles and geophysical applications of both of these types of device.

These new devices resulted in the majority of survey coverage becoming airborne. Their lightness, accuracy and ease of operation however also resulted in their extensive use for land surveys. Breiner (1973) provides a detailed description of this application. The most recent development, described by Hood (1975) is the use of twin magnetometers to measure the vertical magnetic gradient as well as the magnetic total field.

Dobrin (op. cit. p. 321) describes the basic operational procedures of aeromagnetic surveying. Coverage for over-the-land surveys, is usually along a set of parallel traverse lines with occasional "control lines" flown at right angles to the traverses. The aircraft's track is recovered post-flight by means of aerial photographs taken continuously during flight. From these photographs, track points are plotted on topographic or photomosaic maps. The in-flight magnetic measurements, originally recorded on a pen-chart, are now almost universally recorded digitally on magnetic tape. The compilational process for aeromagnetics is long and complex. Many sources of error or disturbing influence exist which must be removed or reduced before the data is in mappable form. Aeromagnetics involves absolute measurements of high precision. The high sensitivity magnetometer measures to a precision of about .02 gamma (nanotesla) over a range of about 5000 gamma in a typical survey and up to 20,000 gamma or more in areas of high magnetic relief – a precision of 1 in 10^6. This precision places greater demands upon the compilation system than for other, less precise, measurements. Airborne gamma spectrometry, for example, has a precision of about 1 in 10^2. All corrections and adjustments must be made as precisely as possible so as to honour the precision of the aeromagnetic values. A gradient of 1000 gammas/km is quite common. Hence if the magnetic measurements are taken to

be precise to .1 gamma, then the track recovery should, theoretically speaking, be precise to within 10 cm. This precision is not possible with currently available techniques. Even though the track may be misplaced by an amount greater than the theoretical dictate, relative distribution of the data along the track will, however, be more accurate.

Hood et al (1977, p. 77) describe the basic processes and historical development of digital compilation methods for aeromagnetic data.

The manual compilation process for analogue data was as follows.

i) Plot the flight path onto base maps and verify its correctness by means of a "speed check" (knowing the time interval between plotted flight path points, the aircraft's apparent speed can be calculated. An anomalous variation in apparent speed over a particular track point suggests that the point is misplaced).

ii) Correlating the pen-chart magnetic measurements with the flight path by means of the fiducials they possess in common, pick from the pen-chart the magnetic values on traverses and control lines at their intersections points.

iii) By means of a graphical least-squares method (Dobrin, op. cit. p. 326) determine the level adjustments to be made to the traverses at their point of intersection with the control lines.

iv) Plot the adjustment on the pen-chart and join the plotted points by straight lines so as to establish a corrected chart datum or "base line".

v) Pick off corrected magnetic values from the chart and transcribe these values onto their corresponding points along the traverses on the base map.

vi) Interpolate contour intercepts between transcribed values and draw contours.

Several instances of prolonged repetition of simply defined tasks occur during the compilation. e.g. transcription or picking of values from the pen-chart. Such tasks are idealy suited for computer automation. The initial development of digital compilation systems employed a digitising table to convert the pen-chart data to machine readable form so as to gain the advantages of automatic processing. The logical next step was the introduction of direct in-flight, digital recording to obviate digitisation of pen-charts. Until the advent of the high sensitivity magnetometer, digital compilation was still an option taken because of the advantages it offered. The high sensitivity magnetometer made digital compilation mandatory as manual methods were insufficiently accurate.

Holroyd (1974) describes in detail the "aeromagnetic data automatic mapping" (ADAM) system employed by the Geological Survey of Canada. Fig. 10 shows the flow chart of this System. The in-flight data as recorded, contains elements not required by this compilation system (altitude, Doppler navigator data). It also contains superfluous data such as lines that were subsequently re-flown, or data recorded on the turns between lines. Furthermore, this data was recorded with a structure best suited to the airborne environment rather than the computation centre. The first task of the compilation system is therefore to extract the information sub-set required for compilation and convert it to a more appropriate structure.

The next task is the detection and correction of physical errors within the data-such as spikes or noise produced by electromagnetic interference with the data aquisition system. The flight path data is converted to machine readable form by means of a digitising table. The maps upon which track points were marked are placed on the table and a cursor placed over each track point in succession. The operator

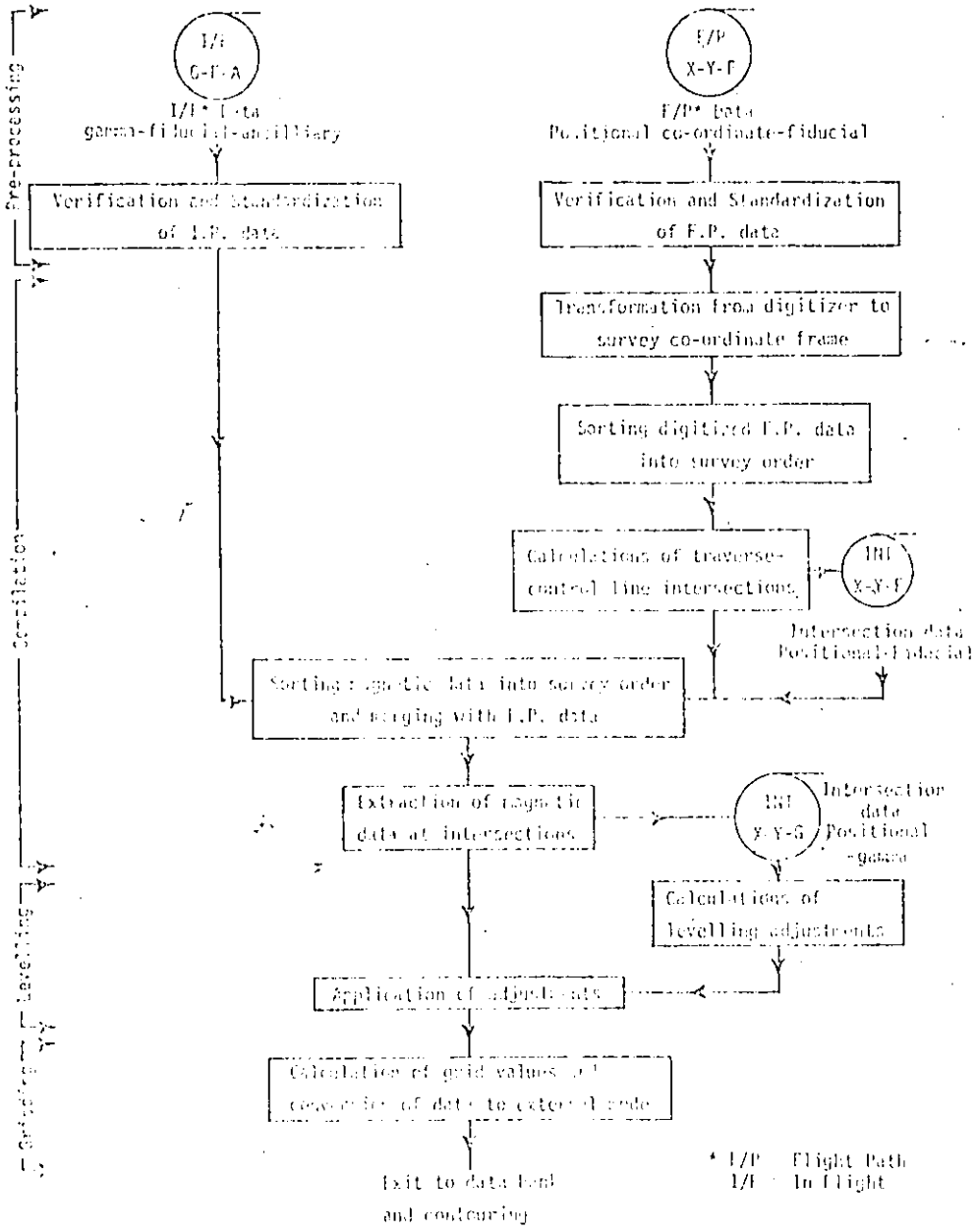


Fig. 10 Aeromagnetic compilation system. (after Holroyd)

key-punches the line and point number whilst the digitiser automatically senses and records the spatial coordinates of the point beneath the cursor. This data must also be standardised and checked for physical errors.

The Cartesian Coordinates of each digitised point are defined with respect to the axes, and in the units, of the digitising table. They are not true geographical coordinates. Furthermore the entire survey at the map scale is usually much too large to fit on the digitising table in one piece. It is therefore digitised map sheet by map sheet. This means that the flight lines, which are usually continuous across the entire survey, are digitised in dislocated segments. Unless the digitising table is of an advanced design and the operator very skilled and diligent, a further problem arises, namely that the coordinate reference frames of each individual map will be independent. These problems have no parallel in the manual system and their resolution occupies a substantial part of the automated compilation system. They are overcome in the following manner: -

i) The coordinates of the corners of each map are digitised and their associated latitudes and longitudes entered via the keyboard before the actual flight path points are digitised. These coordinates are therefore defined with respect to the same frame as those of the flight path points.

ii) The system computes the cartographic coordinates (Usually Universal Transverse Mercator) from the latitude-longitude values. Thus the coordinates of these corner points are known with respect to two different frames. This permits calculation of the rotation, translation and scaling factors that relate the two frames. These factors are then applied to transform the track points on each map from the digitiser to the cartographic reference frame.

iii) The new cartographic coordinates possess the same frame and scale for all maps hence the segments of lines can be sorted, first by line number, then by spatial position, and re-joined to produce a continuous data set of each line.

iv) This process allows detailed verification of the quality of the base maps employed and the accuracy of digitisation. Although the digitiser reference frame is arbitrary, it should be rectilinear and isotropic, as is the cartographic Frame. Hence the distances between pairs of points in one frame should bear a fixed proportional relationship with the distances between the same set of point pairs in the other frame. If this is not so then one (or both) frames is non-rectilinear or non-isotropic. The cartographic frame is mathematically rectilinear and isotropic, hence any discrepency indicates faults in the base map (unstable material or misplaced corner points.) The system calculates the lengths of the map sides and their two diagonals in both frames, compares the lengths between the two frames, and prints diagnostics if discrepancies occur. If the discrepancy is simple (such as an apparent difference in X and Y direction scale factors on the base map due to differential shrinkage) the system automatically corrects the discrepancies.

A further automatic verification routine is made possible by the fact that each base map is constructed with an overlap onto adjacent maps and flight path points are plotted into this overlap region on each sheet. Hence, flight path points falling in this zone appear in two line segments as digitised. When the line segments are sorted and joined, the system compares the coordinates of duplicated points. This allows further assessment of the quality of the basemaps and estimation of the digitising accuracy.

After the above described process, the next stage is to combine the flight path and in-flight data sets. A process complicated by two facts. Firstly the in-flight data is not in the same order as the flight path data. Flying a survey in the most economical manner - economical of time and fuel - results in about half the lines being flown in the opposite direction to the other half, and the lines being flown in an irregular order. i.e. not in the sequence in which they are laid out on the survey base map. The second complication is that the flight path and in-flight data points do not correspond one-to-one. There are usually about fifty in-flight measurements between each pair of recovered flight path points. Thus the following processes are necessary.

i) Each line of in-flight data must be input and stored on a random-access, mass-storage device and an index compiled to the storage location of each line.

ii) The flight path lines are input, one at a time, and the corresponding in-flight data set recalled from storage via this index.

iii) The order of the flight path fiducials indicates whether or not the order of an in-flight data line must be reversed. Lines so indicated are reversed.

iv) In-flight data points are searched to find a fiducial corresponding to the first flight path point. The search then continues along the in-flight data for a fiducial matching that of the second flight-path point. When both are found, a set of spatial coordinates is interpolated between the two flight path points for each intervening in-flight data point. The second and third flight-path points are then treated in the same manner, and so on, to the end of the line.

Automatic verification takes place during this process. The in-flight data should extend beyond the ends of the flight-path data, if not, a diagnostic message is printed and the two data sets reserved for later processing after the problems have been corrected.

When this stage of the process is complete, every measured aeromagnetic value should possess corresponding spatial coordinates. The compilation process is, however, far from completion. The next stage deals with leveling. The problem is similar to ones encountered in several other survey disciplines, such as gravimetry, and topographic surveys. In essence, as one travels around a closed loop formed by segments of control lines and traverses, the difference in the measured parameter values along each segment should add up to zero when one returns to the starting point. (e.g. If one walks around a closed polygonal path one would expect that on return to the starting point, it would have the same elevation as it did when one first left it, regardless of the elevation differences encountered along each side of the polygon.) This however is usually not the case due to errors of measurement. The method employed in the ADAM System depends upon double control lines having been flown. i.e. Each control line was flown in one direction, then the aircraft immediately turned around and re-flew the same line in the opposite direction. The stages of the leveling process are as follows: -

i) The control line data sets are read in and transfered to random-access mass-storage and an index to their storage locations is retained.

ii) The traverse data sets are read in one at a time. For each traverse, each segment of the flight path data (the segments of flight path between successive pairs of recovered track points) is then taken in turn.

iii) All control lines are recalled one at a time, and for each control line each flight-path segment is taken in turn in exactly the same manner as for the traverse segments.

iv) Hence each individual traverse segment can be tested against all segments of all control lines. The test is to determine if an intersection exists between given traverse and control line segments. If an intersection exists, its spatial coordinates are calculated and the magnetic measurement on both the traverse and control lines at the intersection point, are extracted. The final data set produced by this process is a matrix of the form: -

$$(TR, CL, X, Y, GTR, GCL)_{i,j}$$

Where: TR is a traverse number, CL is a control line number,

X,Y are the coordinates of the intersection point between TR and CL, and

CTR, GCL are the aeromagnetic total field values on the traverse and control lines respectively.

i is an index ranging from 1 to NCL (No. of control lines.)

j is an index ranging from 1 to NTR (no. of traverses.)

This matrix must be output to some sequential storage device, and as such must be in linear and not two dimensional form. The order is that of the sequence of operations described above.

v) The matrix is transposed from traverse to control line order. Each consecutive pair of control lines are in fact the two components of a double control line. The control line values are corrected for deviations from exact coincidence of their respective flight paths and a single "true value" control line is created as the mean of the two corrected control lines. The data on

this control line is adjusted to minimise the mean difference between the control line values and the traverse values at all intersection points. The residual CL-TR differences are then taken as correction factors for the traverses at the intersection points.

vi) The intersection data matrix, now containing traverse adjustments, is once more transposed back into traverse order.

vii) Each complete traverse data set is read-in in conjunction with the leveling data set for the same traverse. Values are interpolated between the adjustment points to provide an adjustment factor for every traverse data point.

The data set is now in a state from which a usable map can be made. The automated contouring system for aeromagnetic data requires a "numerical surface" i.e. data values situated at the nodes of a square grid. In order to produce smooth contours, the grid cell size must be significant smaller than the areal extent of the smallest anomaly likely to be present in the data. The grid is generated in the following manner: -

i) Each leveled traverse data set is read-in in turn. Magnetic field values are interpolated between the traverse data points at regular intervals along the whole traverse. The interval is the same as the required grid cell size. Each set of interpolated values for one traverse is output. After all traverses have been so processed there has been produced a matrix of values. "G_{ij}" where "i" is an index from 1 to NGY (No. of grid cells in the y dimensions of the survey), and "j" is an index from 1 to NTR (No. of traverses) This matrix is in column (traverse) order.

ii) the matrix is transposed to row order. Each row is a "Transverse data section" i.e. a set of values at the points where a straight line across the survey intersects all traverses.

iii) Each such row is read in and values interpolated along the row at the grid interval. Thus producing a total survey grid matrix in row order "G_{ki}". Where k is an index from 1 to NGX (NGX is the number of grid cells in the X dimension of the survey). Such a grid is suitable for submission to a contouring program or to other processes such as two dimensional anomaly interpretation routines or digital filters.

This completes the description of the stages of aeromagnetic auto-compilation. In practice, the first time that the last stage was reached would probably reveal flaws in the contoured data resulting from errors or aberrations that had evaded detection at earlier stages. Hence the first "end products" are usually regarded as verification tools to identify such errors. The final end products only appear after the errors are traced back to their source and corrected, and the necessary reprocessing has been carried out.

1.1.5 Gravimetry

Heiskanen and Vening Meinesz (1958) describe in detail the gravity field of the Earth and the application of gravimetry to the Earth Sciences. For any given volume of matter, the strength of its gravitational field is proportional to its mass, and hence, to its density. Gravitational "anomalies" – local variations in the observed gravitational field of the earth – result from inhomogeneity in the density of the sub-surface rocks. As density differences generally reflect lithological differences, measurement and mapping of the gravitational field can be used to shed light on the

sub-surface structure of the lithosphere. As with magnetics and gamma spectrometry, gravity measurements are affected by a variety of disturbing influences which must be corrected for during the compilation process.

Gravity anomalies are very weak in comparison with magnetic anomalies. The Bouguer anomaly map of Canada (Fig. 11) produced by Nagy (1977, p. 59) shows a country wide variation from about - 250 to + 350 milligals. The earth's main field on the surface is about 1000 gals. Hence the gravity anomalies are usually of the order of less than 0.1% of the earth's main field. Local anomalies are measured and mapped which are of the order of 1 in 10^6 of the earth's main field. In comparison, magnetic anomalies may have a magnitude of more than 30% of the earth's main magnetic field although local small anomalies of about 0.01% of the earth's field are measured and mapped.

Heiskanen and Vening Meinesz (op. cit. p. 84) describe the instruments used for the measurement of gravity. The two principal types are the pendulum and the spring balance. The pendulum was the first instrument employed for gravimetry. The strength of the gravitational field can be calculated from the physical characteristics (mass distribution) of the pendulum and its period of oscillation. Pendulum measurements take a relatively long period of time and must be carried out under carefully monitored conditions. Hence such measurements are made today only to establish the absolute value of gravity at principal reference points. The second type, the spring balance, employs the effect that as gravity varies so does the weight of a given mass and so also will the extension of a spring from which the mass is suspended. The properties of such a device vary slowly with time and they can not be

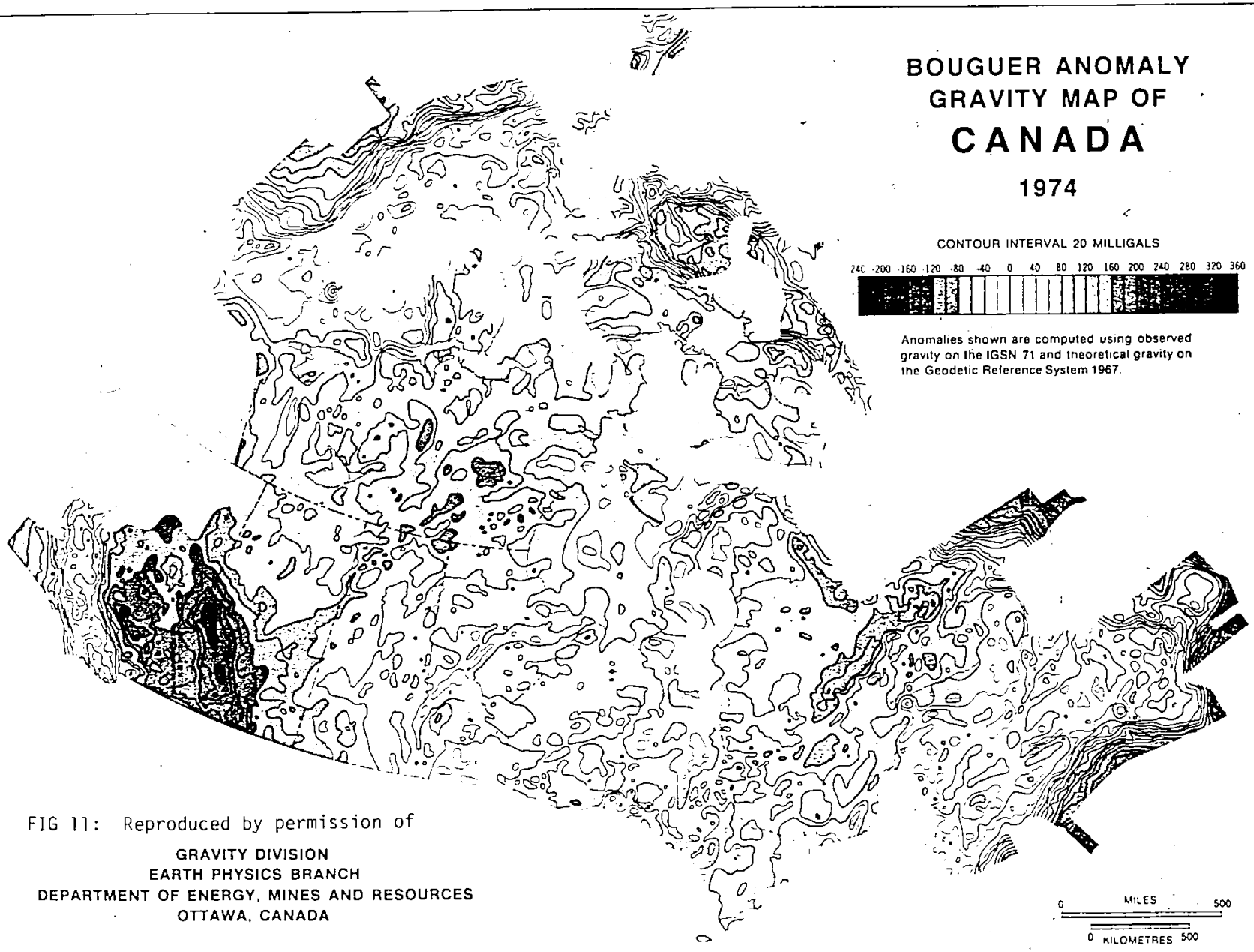


FIG 11: Reproduced by permission of

GRAVITY DIVISION
EARTH PHYSICS BRANCH
DEPARTMENT OF ENERGY, MINES AND RESOURCES
OTTAWA, CANADA

permanently calibrated to permit absolute measurements. Hence only relative measurements of the difference between points are made and absolute values calculated from a measurement made at a principal reference point (a pendulum station).

A gravitational field is indistinguishable from accelerated motion. Therefore measurements made by these two types of instrument can only be made while static or at least while in linear motion at a constant speed. Their use in simple form is therefore restricted to land surveys. Research into inertial navigation systems, led in the early 1960's, to the development of stable platforms and sensitive accelerometers which would allow the use – albeit with diminished precision – of a gravimeter within a ship at sea. Worzel (1966) describes the operation of such a device. In essence, the stable platform keeps the gravimeter level (i.e. its axis parallel with the direction of the gravitational field). Whilst the accelerometers measure the accelerations in this direction. Although changes in the gravity field also affect the accelerometers, this effect varies much more slowly than the up-and-down movements of the ship. This frequency differentiation allows the "noise" caused by the ships motion to be filtered out. A remarkable achievement as the noise is about 10^5 times greater than the signal – the gravity.

If terrain permits and the objectives of the survey require, land survey coverage is similar to that of aerogeophysical surveys. Measurement stations are located at regular intervals along parallel traverses. If the terrain does not permit such access or if the survey is of a regional or reconnaissance nature, then measurement stations will be spaced out along lines of least resistance –e.g. roads. The slow variation of the instrument sensitivity – referred to as "drift" – must be compensated for. To do so requires that the surveyor establish "base stations" and return to these stations to

remeasure at intervals of about one or two hours. In populated areas these base stations tend to be located in the parking lots of public houses. To allow reduction of measurements to an absolute datum, one or more of the base stations must be linked to a pendulum station. At each survey station the observed gravity reading, the time and the altitude are recorded – written onto a record sheet.

Heiskanen and Vening Meinesz (op. cit. p. 147) describe the theoretical bases of gravity data reduction. As the objective is to determine the sub-surface structure, then the final maps must be of anomaly values. That is, the amount by which gravity varies from that which would be observed if the lithosphere had a uniform density. To produce this data requires that disturbing influences be removed. The disturbing influences that must be corrected for during the compilation process, and the correction procedures are as follows.

i) Drift. The amount by which the instrument reading has drifted between successive measurements at a base station is linearly distributed among, and subtracted from, the readings taken in the intervening period.

ii) Free air correction. The gravitational field decreases as the inverse square of the distance between the measurement station and the earth's centre of gravity – by about 0.3 miligals/metre. Unless gravimetric measurements were corrected for this effect, the resultant maps would reflect a confusing admixture of lithospheric mass distribution and topographic relief. As the change of gravitational attraction with distance from the earth's centre of gravity is essentially linear over the range of elevations encountered by surficial gravimetry, a linear correction is applied. Observed measurements are corrected by a factor proportional to their vertical distance from the geoid. The geoid is the surface obtained by interpolation of mean sea level across land areas.

iii) Bouguer correction. The observed gravity at a point above the geoid is disturbed not only due to its elevation but also by the gravitational attraction of the mass of material that lies between itself and the geoid. To correct readings to the geoid requires that this effect be removed. Unlike the Free-air correction this one can not be based solely on theoretical constructs. An estimate or assumption must be made of the mean density of the material that lies between the observation point and the geoid. The gravitational attraction of the estimated mass is calculated and subtracted from the reading.

iv) Earth tide correction. The gravity fields of the moon and sun cause deformation of the solid earth as well as the hydrosphere. The peak of this deformation follows the moon's rotation about the earth. It therefore causes a cyclic variation of gravity with time at any point on the earth. The magnitude of this effect is calculated or read from tables and subtracted from the observed reading.

v) Terrain correction. The Bouguer correction was made on the assumption that the material between the measurement elevation and the geoid consisted of an infinite sheet of constant thickness. Which is not, of course, the real case. Consider, for example, measurements made at the brink of an escarpement between a plateau and a lower plain at geoid level. In this case the sheet approximates to a semi-infinite one. The gravimeter therefore receives only half the effect that the Bouguer correction dictates. A measurement on a mountain peak results in even greater inaccuracy in the Bouguer correction. In highly accidented terrain it is therefore necessary to assess and compensate for this effect.

vi) Removal of Earth's main field. Even if the Earth was of homogeneous composition, it gravity field would not be constant over its surface for two reasons. The centrifugal force of the earth's rotation acts against the gravitational force. This effect however is maximum around the equator and diminishes to zero at the poles. The centrifugal force also distorts the elastic earth into an oblate spheroid. The equatorial radius being greater than the polar one, as gravity diminishes with distance from the center of mass this causes a reduction in gravity at the equator as compared with gravity at the poles. To compensate for this effect, an equation is employed which can be evaluated to define the earth's main field at any lattitude.

Tanner and Buck (1964) give the equations from which correction factors are derived in their paper describing a "computer oriented system for the reduction of gravity data". The system they describe, though advanced for its time suffered from the limitations of the computing machinery available. Crain (1972) provides a flow chart for a more advanced, generalised, system. Fig. 12 reproduces this flow chart. A highly advanced compilation and management system for gravimetric data, which fully exploits modern computation machinery and techniques is described by McConnell (1977). In this system, the compilation and reduction of gravity data from observed to absolute corrected values is merely one component in an integrated data base system. The data base contains the data from many surveys and other information such as the location and gravity value of base stations. As new surveys are carried out, the new data is entered into the system which calls upon its store of base station and network information to assist in the reduction. When the survey data is reduced and compiled, it is added to the data base.

GENERALIZED GRAVITY
DATA PROCESSING

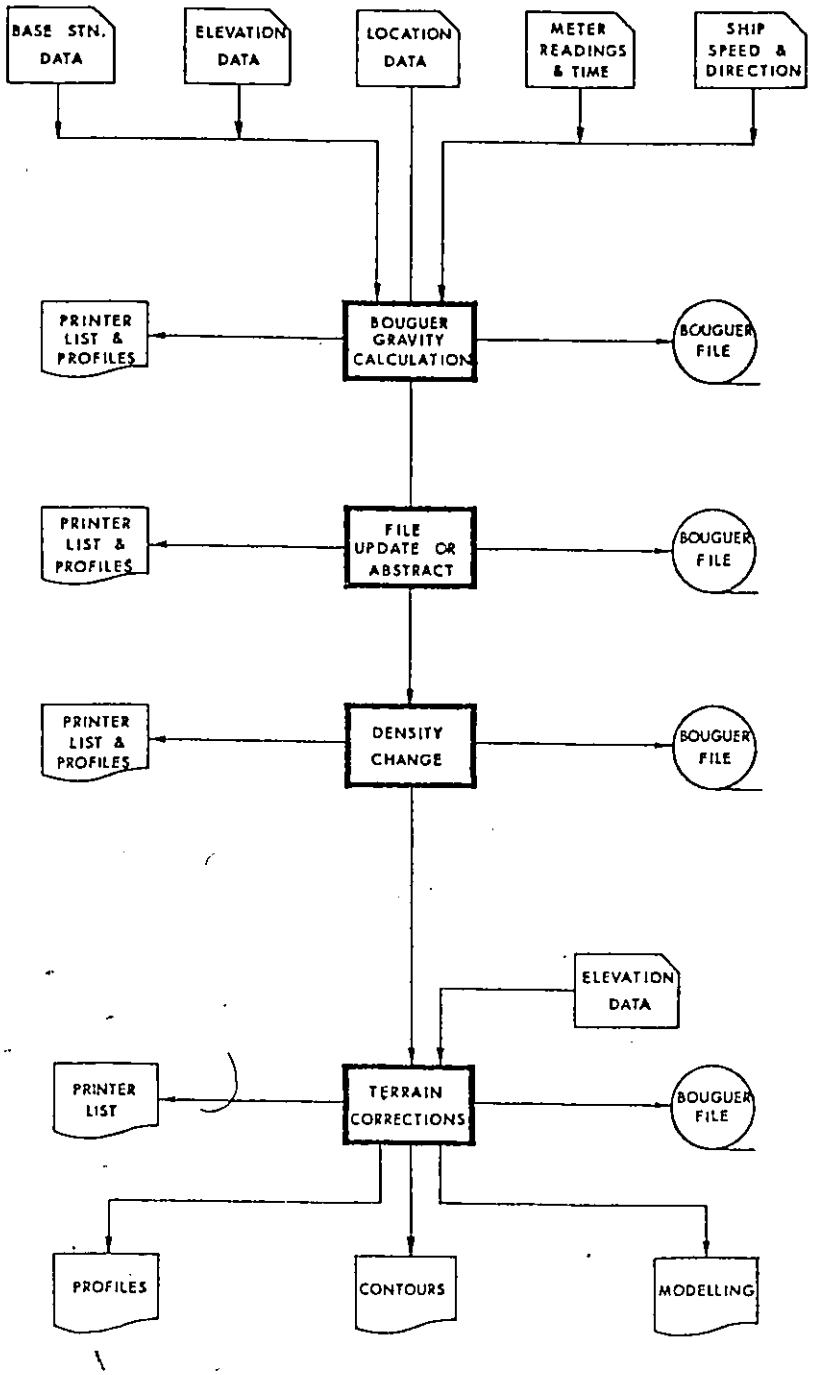


Fig. 12 Generalized gravity data processing system (after Crain).

The basic stages of auto-compilation and reduction of land-gravity data are as follows.

i) The field data records each contain at least the station number, the time and the gravimeter reading. The elevation and location of the station may be recorded in the field or recovered later in a manner similar to track recovery for airborne surveys. The first task of the compilation system, after the data has been converted to machine readable form, is to seek for detectable errors in the data. i.e. such errors as obvious key punch mistakes (alphabetic characters in numeric fields etc., and numerical values obviously out of their permissible range.) The next task is to merge with the basic field data, any spatial coordinates recovered separately. A process similar to, but generally simpler than, that as described for aeromagnetics.

ii) Several measurements of the gravimeter reading differences between each base station and a pendulum station have been made to allow removal of instrument drift. These differences are corrected for drift and multiplied by the calibration constant of the instrument to produce a true milligal difference. This difference, plus the absolute value at the pendulum station, produces the raw absolute value at each base station.

iii) The field data records consist of many sequences of ordinary survey station readings. Each sequence begins and ends with a reading at a particular base station. The next compilation task is to input this data file, and for each sequence to find the drift between the initial and terminal base station readings. Drift values are interpolated for, and subtracted from each survey station reading.

iv) The free air correction is simply derived by multiplication of the station elevation by a constant factor.

v) The Bouguer correction requires the estimation and input of the mean density of the rock between the measurement station and the geoid, after which the correction is derived from a simple calculation involving the station elevation and this mean density.

vi) The Earth Tide corrections can be calculated from the time and date of the measurement by means of an Earth tide equation. Or it can be read from tables and entered.

vii) The terrain correction for each reading is the most complex to derive. A separate program is usually employed into which is input a digital terrain model. The locational coordinates of each station are then input and the gravitation effect of the terrain irregularities surrounding each station is calculated.

viii) These corrections are applied to the base station raw gravity values to provide true absolute gravity values at these bases.

ix) The drift corrected gravimeter reading difference between a survey station and a base station,. multiplied by the calibration constant plus the absolute gravity value at the base station gives raw absolute gravity value at each survey station. Application of the correction factors to this raw value gives the true absolute gravity value at each survey station.

x) The magnitude of the earth's main field at each station is calculated from the main field equations and subtracted to produce the Bouguer anomaly value of the gravitational field at these points

xi) as with aeromagnetics, errors of measurement persist which require a leveling procedure for their distribution. In gravimetric compilation this is usually done by establishment of a network of measurement loops. The closure error of each loop is distributed around the loop periphery. As each loop error is removed it can re-create an error in a previously leveled, contiguous loop. With successive passes, however, the loop errors will eventually diminish to zero.

2.0 The basic processes of compilation and their amenability to generalisation

2.1 An obstacle to the recognition of generality

When we examine the examples given in Chapter 1 in search of common features which might point the way to generalisation of compilational methods, a bewildering network of helpful similarities and obstinate differences reveals itself.

Gravity and magnetics respond to the bulk physical properties of the lithosphere. Airborne radiometrics responds to the average elemental composition of the top few centimetres of the ground surface. Geochemical water analyses, though made upon small individual samples reflect the bulk elemental composition of those parts of the lithosphere, over which or through which, the waters have passed. Gravity and magnetics are linked in that they are passive measurements of natural potential fields, and accordingly possess some similarities in their data reduction and compilation processes. They differ though in that it is the interface between lithological units which causes the magnetic anomaly and the bulk of the unit which causes the gravity anomaly.

The biggest difference between these two is that magnetic surveys are mostly airborne, whilst airborne gravity is not yet practicable. This, plus the fact that aeromagnetics is usually digitally recorded while terrestrial gravity data is usually keypunched from hand written coding sheets, causes the major differences between the compilation systems. Airborne gamma ray spectrometry is closely similar to aeromagnetics in the surveying technique. So similar, in fact, that the two are often carried out simultaneously from a single aircraft. This results in similarities in the initial data structures and the compilational procedures concerned with track recovery, etc. Radiometrics is however, geochemistry by other means. Geochemistry, drift sedimentology and gravimetry have in common the fact that they

are all point-station measurements, unlike aeromagnetics and radiometrics which are pseudo-continuous linear measurements. Drift-sedimentology incorporates geochemical analyses but not the geochemical compilation system as it requires validated, final, results.

The flow charts of the systems also reflect similarities and differences. At first glance, however, only a few common features appear which could point the way to generalisation of compilational methods.

It is the contention of this writer that the sparsity of common features is not an intrinsic property but results from the point of view of those who have created and described these systems.

All the examples given were taken from works published by, or personal communication with, scientists working in the specific discipline concerned. The processes were thought of and described in terms of the practices and objectives of the scientific discipline. Furthermore the basic practices were established long before the intrusion of the computer and many concepts persist which reflect the manual origins of the compilation processes rather than the automated present.

It is not sufficient to blame this situation on resistance to change. Many sound practical reasons can be found to account for its existence. Aeromagnetic compilation provides a good example.

The computer began to intrude significantly upon the hitherto manual processes of aeromagnetic compilation about 15 years ago. Its share of the work increased steadily, keeping pace with the increasing power and decreasing costs of the computer. The advent of effective and affordable digital data aquisition systems and graphics output of cartographic quality gave the final boost; resulting in the last few years, in the development of many, almost fully automated, compilation systems.

Most of the early development work was carried out by private industry and was subject to strict cost-effectiveness standards and production deadlines. It was neither feasible to cease production until a complete digital system could be created, nor to develop a duplicate, but digital, system in parallel with the manual system. Hence the practical compromise was taken of gradually phasing in digital processes to replace manual ones as the digital processes were developed and more sophisticated and efficient devices became available. This method also gave a valuable opportunity to compare manual with automated processes at each stage thereby ensuring that the digital system functioned as required.

Phased replacement of manual processes however, required that the digital processes closely mimic their predecessors. The resulting systems, while far superior to the manual systems in rate of throughput and repeatability, and easy to understand from the traditional view point, tend to be inflexible and hard to maintain in the computer environment with its rapid rates of change and development.

Subsequent software development decreased this disadvantage by increased generalisation within individual processes and by the development of general utility packages, but always within the confines of one type of survey and one basic system plan.

Twenty years ago manual compilation methods for aeromagnetic surveys were largely standardized across the industry. One company could take over from another at almost any stage of the work with little delay. The consequence of ad-hoc software development is that there are about as many auto-compilation systems today as there are organisations engaged in aeromagnetics. No two of which could be considered fully compatible. To transfer digital data from one organisation to another in mid-stream would require at best substantial effort to create "Interface" software. At worst it would require much of the previous work to be completely re-done.

2.2 Abstraction of compilational processes

Whatever its causes, the viewpoint from which Earth Science autocompilation systems tend to be conceived of and described, presents an obstacle to the detection of intrinsic similarities between different systems. To overcome this obstacle it is therefore necessary to adopt a more abstract viewpoint. That is, one less closely bound to any one particular Earth Science discipline and more appropriate to the study as a whole.

One thing that all the systems described do have in common, is their use of the computer. From this viewpoint one can say that all autocompilation systems consist of various combinations of the basic computer processes - namely input/output, separation-sorting-merging of data sets and subsets, and computation. This viewpoint is no more helpful than that of any one specific earth science discipline as it provides no hints as to how to generalize compilation processes. It is, in fact, too general.

The preferred viewpoint lies somewhere in-between these extremes – a viewpoint which remains within the specialised realm of Earth science survey compilation but which permits the compilation processes to be viewed in the abstract. That is, not tied to any one specific discipline.

Examination of the systems described in chapter one, from the viewpoint of compilation processes in general, reveals four basic categories of process. None of which are restricted to any particular discipline. Namely:-

i) Verification – the detection and correction of errors.

ii) Manipulation – the re-structuring of data sets to put them in the form required for some subsequent process.

iii) Development – the derivation of a new data set by computation involving one or more pre-existing data sets.

iv) Display – the presentation of data in a form intelligible to the human.

The boundaries between these processes may in some cases be indistinct. For example verification may involve display. Furthermore their relative importance and juxtaposition will vary significantly between different systems. They do however serve their intended purpose - they are independent of individual disciplines and all of the systems described in chapter one could be adequately described in their terms.

Having established these categories we may now examine them individually, in greater detail to determine their respective amenabilities to generalisation.

2.2.1 Verification

This process involves ensuring, at all stages, that the data is as it should be; that no errors or observations were carried through from a previous stage or were introduced during a stage.

The very features which make automated compilation systems advantageous - speed, repeatability and objectivity, also give rise to their principal disadvantage. The processes are invisible, and even if they were visible, would progress too fast for the human eye to follow. Their objectivity and repeatability is due to the fact that they follow pre-defined instructions to the letter. No deviation from the pre-programmed instructions can take place, regardless of the situation. The consequence of these facts is that unforeseen situations can produce aberrations which propagate, invisibly and rapidly, through the system. The best that can be hoped for in such a case is that the aberration is severe enough to quickly cause a program "crash" or abnormal termination, after which the cause must be determined,

rectified and the corrected data reprocessed. The worst that can happen is that the aberration remains undetected and is passed on through subsequent stages to the end product. The more automated systems become, the more likely it is that such can occur. In manual systems, the information and processes were continually under human scrutiny. Human beings possess the capability to detect and respond to unforseen situations and hence the likelyhood was small of a serious error propagating unseen through the entire manual process. To minimise reprocessing and to maintain the standards of the end product it is therefore necessary to put into the automated system, as far as possible, an automated form of constant scrutiny. i.e. verification processes.

Verification can be broken down into two sub-stages - inspection and correction. The former consists of detecting errors or aberrations, the latter, removing them or replacing them with correct values. Inspection may involve presenting the data in a form suitable for human inspection or defining the inspection criteria digitally and letting the computer apply an inspection algorithm. Correction may involve simple editing to remove or replace a few data values, the complete reprocessing of one phase of the data or the rejection and re-acquisition of a part of the initial data set, depending on the nature and severity of the problem.

The types of error or aberration encountered also fall into two categories - physical and logical, equivalent to syntactic and semantic errors.

i) Physical verification

Physical verification involves the detection and correction of physical errors. That is; miscoded, absent, or supernumerary information, from individual characters to large data blocks.

Mistakes frequently are made when writing information on coding forms and during the subsequent keypunching. They are also made by the electromechanical card readers. Thus geochemistry, drift sedimentology and gravimetry whose primary data is almost all aquired by these means, require stringent error checking routines. This can be done manually by comparing a print-out of the punched card file with the original field notes, but errors will still evade this checking procedure. It is therefore necessary to submit the data to some form of automated checking procedure before allowing it to enter the mainstream of the compilation process. Digitisation of track-data for aeromagnetics and gamma spectrometry involves manual entry of fiducial numbers which is as error prone as keypunching from coding forms. Automated data aquisition systems, such as are employed for the position coordinates on the digitisation system and the airborne magnetometer and spectrometer, are less prone to error. In these last two devices however, although the frequency of errors is less than for manual data entry, the enormous bulk of the data results in significant numbers of errors being present. The bulk of the data also prohibits manual checking of the data values and hence an automated verification process is mandatory.

From the point of view of the abstract numerical process, all the exemplified disciplines share the same physical verification requirements. Namely the facilities to examine the contents of a data set at all levels from the overall gross contents ("is every map sheet, flight line or sample suite of the intended data set present?") to the specific details of individual data elements ("are there any illegal characters present within any of the coded data elements, such as an NTS 1:50,000 scale map sheet reference with a code letter beyond "p" in the alphabet or an alphabetic character in a numeric element?).

The physical verification process could therefore be generalised as the ability to:-

i) describe the contents of a data set,

ii) compare individual data elements with criteria laid down by the user and identify elements which failed the tests.

iii) Insert or delete data items or subsets of data items.

ii) Logical verification

"Logical verification" means the detection and correction of physically correct but logically aberrent data.- i.e. data which would pass the physical verification tests but whose logical significance ("meaning" or "implication") is at fault. Detection of logical errors requires a deeper knowledge of the context of the information than does detection of physical errors. It may require a knowledge of the scientific bases of the discipline concerned or the intuitive ability to detect a flaw as a result of long experience of survey compilation.

For example, an aeromagnetic anomaly of half-wavelength much smaller than the flight altitude is clearly aberrant only to those familiar with the relevant geophysical theory. If one writes down a sample number as "12X" rather than "123" this is detectable as a physical error as it contains a non-numeric character. If the worker wrote down "122" instead of "123" this could also be detected if the system noted the absence of "123" and the duplication of "122" in the list of sample numbers. If the writer however, transposed "123" and "122" the data could pass throughout the system without detection as no evident physical error exists.

The only way that such an error would be likely to be detected would be if a geochemist noted some incongruity in the final results or an incompatibility between the reported elemental abundance of the sample and the purported rock type of its origin.

As can be expected, automated logical verification requires far greater subtlety of method than physical verification. It is however, still amenable to generalisation as it can, at least in part, be carried out by processes similar to those used for physical verification. That is, by comparison not only of individual data elements with acceptability criteria but also by comparing the evaluation of some logical or mathematical function of the data elements against stated criteria. Hood, for example, (1975) describes a method for detecting "noise" in aeromagnetic data by comparison of a function of the fourth differences of successive data values against an acceptability criterion.

If the characteristics of logical errors are too complex or too subtle to be pre-defined by a mathematical function, an alternative course, still amenable to generalisation, exists. That is to display the data in the manner most suited to the detection of errors by visual inspection e.g. by plotting geochemical analysis results in the form of their deviation from the average values in the surrounding neighborhood, or some such.

2.2.2 Manipulation

"Manipulation" refers to processes which essentially consist of the movement and re-arrangement of data so as to form a data set whose content and order are as required for some subsequent verification, development or display process. Many

examples of this type of process can be found within the systems described in chapter one. In every case components necessary for the final data set become available from different sources at different times and usually are in different order than that which is, or will be, required for the compilation process.

The following examples illustrate this process.

The first geochemical data set to be produced is one labeled "field data" on Fig. 6. Such information as the sample number, NTS sheet no., rock type, date, and parameters describing the environs of the sample and other auxilliary data are written onto a coding sheet for later keypunching. For both wet chemistry and atomic absorption analysis, a given suite of samples is analysed for one element at a time. Hence the data is originally derived as "element (- several samples -), Next element (- several samples -)" etc. The order of the data required for the compilation system is the inverse, i.e. "Sample No. (- several elements -), next sample no. (- several elements -)" etc. The wet chemistry results are recorded manually and hence the rearrangement of the data takes place on the coding form on which the results are written. i.e. the results are written in columns and read-off and keypunched by rows. Atomic absorption spectrometer results, on the other hand, are recorded digitally in the order of analyses and must be re-ordered by a pre-compilation system before merging with the other data sets. When compilation is complete, the users of the data generally require data subsets subdivided by geographical area and selected element. Thus requiring a further re-arrangement of the data file.

A second illustrative example concerns aeromagnetics. The geophysical parameter(s) concerned are measured in flight at points along a traverse line crossing from survey boundary to survey boundary. The order in which the traverses are flown is dependent upon economic and aeronautical considerations. The spatial coordinates

associated with each geophysical measurement are usually recovered later, from flight path plots made on photo-mosaics whose boundaries conform to topographic map sheets. Thus, when the two data sets are first brought together, they are radically incompatible in their order. The first task therefore, is to sort each data set to a common order, then to merge the two sets so that each geophysical measurement is grouped with its associated spatial coordinates.

The general requirements for data manipulation are:-

i) Separation of a given data set into two or more data subsets.

ii) Re-arrangement of the order of data within a set.

iii) Combination of two or more data sets.

These types of operation are readily carried out by "Sort-merge" packages which are, to a large degree, data independent. Hence prospects for the generalisation of data manipulation are favorable.

2.2.3 Display

This process involves conversion of data from its machine form to a form intelligible to the human being, namely printed listings and graphics. Data display is not merely the means by which final results - tables, graphs and maps - are presented to the end-user, it is required throughout all stages of the work as a vital adjunct to the verification processes.

Appropriate display is often the most efficient means of solving the more difficult problems of logical verification. Not all error-recognition processes can be defined in terms of computational algorithms. Many others exist which, though they can be so defined, produce inefficient and ineffective results as compared to non-computational methods. This is one of the most striking examples of the limitations

of computer vis-a-vis human capabilities. Namely the pattern recognition process. Failure to exploit such a capability to its fullest would be highly detrimental to the efficiency of the compilation system. Accordingly, the interests of generality, simplicity and efficiency of many of the error recognition processes are best served by providing the means to display the information as appropriate "patterns". In compilation these take the form of listings, graphs, charts, diagrams, maps, etc. e.g. the initial error checking process for keypunched data is to print out the file as possessed by the computer for comparison with the original field notes or coding forms.

An example demonstrating the effectiveness of appropriate display for the logical verification is shown in Fig. 13. This is a listing with graphical features produced by a "speed check" program (Dods, 1975). Such programs are used to detect positional errors in the recovered flight path points of airborne surveys. The apparent ground speed of the aircraft along each leg between recovered track points is calculated and listed in the column headed "GSP-MPH" this is also listed in units of "Map speed-inches/sec" (column headed MPS-IPS). The next column (PCDIF) lists as a percentage, the apparent change in speed across a recovered point. Excessive change in apparent speed could be due to a misplaced track point rather than an actual speed change. To assist the compiler in detecting anomalous values the speed is also displayed in graphic form at the left hand side of the printout. Sudden fluctuations stand out clearly and detection is made far easier than by reading the columns of figures. Three blank columns are printed, headed by "new fid, new X, new Y" into which the compiler can insert corrected values. In this figure the compiler has written "delete" opposite the two anomalous features. This "display" thus combines graphics, listing and coding form for correction. A highly effective and user-efficient combination.

LINE NO.	FIC	GSP-MPH	PSP-IPS	PC-IP	DIF-IPS	NEW FIC	NEW X	NEW Y	CLC Y	CLC Y
									49.427	17.948
1240	40009								49.334	19.356
	40028 X	91.76	.0744	-25.296	.356	DELETE			49.326	21.157
	40041 X	170.24	.1381	38.597	-1.146				49.469	25.157
	40081 X	123.85	.1002	.665	1.521				49.427	27.539
	40105 X	122.45	.0993	-.313	.023				49.520	29.614
	40127 X	116.46	.0944	-5.186	.101				49.489	31.685
	40149 X	116.13	.0941	-5.453	.016				49.424	34.396
	40179 X	111.58	.0905	-9.156	.100				49.434	35.774
	40189 X	120.49	.0976	-1.950	-.070				49.388	38.917
	40225 X	121.41	.0984	-1.157	-.029				49.228	41.817
	40245 X	117.55	.0953	-4.300	.066				49.200	44.576
	40287 X	110.40	.0895	-10.117	.219				49.225	47.261
	40317 X	110.49	.0896	-10.049	-.002				49.192	49.490
	40341 X	114.48	.0928	-6.797	-.072				49.181	51.479
	40361 X	122.67	.0995	-.126	-.133				49.111	54.545
	40391 X	126.26	.1024	2.795	-.090				49.068	55.935
	40403 X	132.27	.1072	7.682	-.063				49.048	58.039
	40425 X	123.59	.1002	.616	.156	DELETE			49.062	61.111
	40459 X	111.45	.0904	-9.261	.363				49.059	63.545
	40483 X	125.10	.1014	1.848	-.270				48.997	67.554
	40519 X	138.75	.1125	12.963	-.450				48.994	72.582
	40565 X	133.76	.1084	8.896	.203				49.025	74.699
	40585 X	130.03	.1054	5.859	.064				49.087	81.278
	40645 X	135.45	.1098	10.271	-.291					

207.45 FEET PER SECOND OR 122.83 M.P.H (NAUTICAL) IS THE AVERAGE GROUND SPEED OF THIS LINE
.0996 INCH PER SECOND IS THE AVERAGE MAP SPEED

Fig. 13 Combined listing, printer graphics and coding form for aeromagnetic "speed check".

Most graphics applications can be met by graphics software "packages" which already exist in a generalised form hence generalisation of display processes seems quite feasible.

2.2.4 Development

"Development" refers to that category of processes which derive a new data set by computation applied to one or more pre-existing data sets. It is the broadest category of the four established and the one whose amenability to generalisation is the least obvious.

Examination of the compilation systems described in chapter one reveals several subcategories of the developmental process which are common to more than one, if not all, disciplines described and which may be generalised to some degree. For example:-

i) Statistical analyses. Both drift sedimentological and geochemical compilation systems include the statistical analysis of final results. A large part of this process consists merely of presenting the data to the analysis. This is essentially a manipulation process which has already been noted as amenable to generalisation. The remaining part, calculation of the statistical parameters, could be carried out by a generally applicable statistics "package" an example of such a package is "SPSS" (Nie et al., 1975) though initially designed for use within the social sciences (Statistical Package for the Social Sciences) its generality allows its use in a wide range of other disciplines.

ii) Interpolation.

The removal of instrument drift (Atomic absorption spectrometer, gravimeter).

the removal of diurnal variation (aeromagnetics),

the removal of background radiation (airborne gamma spectrometry),

the distribution of measurement points between recovered track points (aeromagnetics, airborne gamma spectrometry),

are widespread examples of one dimensional interpolation processes.

All the types of data mentioned in chapter one are acquired as discrete point observations. Some, however, are pseudo-continuous in one dimension (i.e. along the measurement track, the point separation is much less than the natural wavelengths of variation of the data in this direction and hence a simple linear interpolation between data points produces an apparently smooth and continuous curve). None of these measurements are pseudo-continuous in two dimensions. The map plane is however, two dimensional and many processes, such as certain display forms and numerical analyses, require the data to be pseudo-continuous or at least evenly distributed over the plane. (e.g. The contour mapping process, two dimensional spatial filtering, the application of two dimension anomaly routines for gravity and magnetic interpretation, the estimation of areal extent or bulk volume of sediments, the terrain correction for gravimetrics).

To produce pseudo-continuous two dimensional data requires the application of two dimensional interpolation routines. As with statistical analyses, a large part of what is thought of as an interpolation package in fact consists of data manipulation. In the two-dimensional grid interpolation program in use in the ADAM system (Holroyd, 1974, ibid), the actual interpolation routine takes up less than 20% of the program's FORTRAN statements. The remainder consists mainly of verification and manipulation routines.

Different types of interpolation algorithms, however, are required for different types of data. "Piecewise Continuous" polynomials are ideal for gravity and magnetics (Bhattacharyya, 1969, p. 402), but not for less densely sampled, or inherently more "rough" data such as geochemical data. Among the methods reviewed by Crain (1970, p. 71) is "neighborhood" or "weighted" averaging, a method more suited to geochemical data.

Hence some bases for generalisation can be found within interpolation processes. The key lies in separation of the inherently more general manipulation processes from the interpolation per-se, and then the provision of a selection of interpolation methods, each of which may be applicable to two or more types of data.

iii) Removal of extraneous influences.

For most types of Earth science data, development includes the process of removing from measured quantities, interference from sources other than the geological structures that the survey seeks to investigate. e.g. for gravity this includes the removal of time dependent instrument drift and earth tide effect, topography dependent effects, and the altitude dependent free air correction; for radiometrics this includes the removal of altitude and meteorologically dependent background and scatter effects; for aeromagnetics this includes the removal of time dependent diurnal variations and other effects introduced by altitude variations.

As noted above, some of these processes fall partially into the sub-categories of interpolation. Others, though they may require interpolation at some stage, still remain largely distinct from this category. But once again, manipulation routines still form a significant part of such processes thus allowing for partial generalisation at the outset.

2.2.5 Levels of complexity

The amenability to generalisation of these processes - or rather, the weight of specialised process that would be required, depends upon their structural complexity which varies significantly.

The lowest level of complexity is where each data record contains all the information required for the subsequent process independently of other data records or other data sets. An example of this type is the "Free air" correction for a gravity measurement. Assuming that the altitude of the station is one of the elements of the record, each record can be taken in isolation and the conversion factor calculated by a simple formula. Further examples are the stripping ratio correction for radiometric data and an altitude correction for aeromagnetic gradiometer data.

The next level of complexity involves interaction between two or more "co-incident" data sets. "Co-incident" means that there is a one-to-one correspondence between members of the data sets. i.e. the N^{th} result is derived by computation involving the N^{th} member of each of the data sets concerned. Such a situation results from the interpolation processes described above. e.g. after interpolation of the drift values between gravimetric drift measurements, there exists a drift value for each gravity measurement. After a manipulation process wherein the data sets are merged to bring together corresponding elements, this process becomes equivalent in complexity to the previous described one. i.e. each new data record now contains all the information necessary to compute the correction factor.

The highest level of complexity concerns those cases where interactions are required not only between non-coincident data sets but for each result, between many members of one or more other data sets. The "terrain" correction for gravity data and the "levelling" correction for gravity, magnetics or radiometrics all fall into this category.

All of the developmental processes described contain some feature not easily generalised. At the first level of complexity however this feature is likely to be merely a simple function evaluation. At the second level, the major part of the work involves interpolation and manipulation routines previously assessed as suitable for generalisation. After which, once again, the only specialised feature is a simple function evaluation. At the third and highest level of complexity a significant proportion of the process will be unavoidably specialised. But careful construction of the routines would permit an equally significant part to be carried out by more general manipulation and development sub-processes.

To demonstrate that this is the case let us consider the example of aeromagnetic "levelling"- a quite complex, ostensibly highly specialised, adjustment processes.

The magnetic field values along the traverses contain a long wavelength component of non-geological origin. This disturbing influence stems from a combination of several factors. Mainly altitude variation by the survey aircraft, errors in flight path recovery, and discrepancies between the actual diurnal variation along the traverse lines and the diurnal variation as recorded at the ground monitor station some distance away. Were a contour map to be made of the data as it stands, the erroneous amplitude shifts between adjacent traverses would result in an oscillation of the contours known as a "herring-bone" effect. The control lines perpendicular to the traverses were flown expressly for the purpose of leveling.

Many different techniques exist for the derivation of the leveling corrections but all employ in some way the magnetic values on both the traverses and control lines at the points where the two sets of lines intersect. To calculate if and where a

segment of traverse intersects a segment of control line, and to extract the magnetic values at this point, requires only a simple coordinate geometry routine. In order to do this for every possible traverse - control line intersection point requires considerable data manipulation.

After the intersection data set is derived there follows substantial data manipulation to re-arrange the data set in the form required for calculation of the adjustments to be made to the traverses. The adjustment calculation routine though specialised is usually not overly large or complex.

The adjustment to the traverse are calculated only for those points where the traverses intersect control lines. The final phase of adjustment therefore requires further data manipulation followed by interpolation and application of adjustment values at each and every magnetic measurement along the traverse.

Thus when aeromagnetic leveling is viewed in greater detail it is shown to contain many components which belong to other classes of compilation process more amenable to generalisation.

In conclusion, we may state that although development processes appear to be highly specialised - restricted to one or two different types of data they still can largely be carried out by a specialised combination of less specialised sub-processes.

What specialised sub-processes remain could be attached to the outside of a framework of more general processes.

2.3 Obstacles to implementation of generalisation

The preceding section identified, and attempted to surmount, the obstacle that lay in the way of recognition of the generality of many compilation processes. That is, the creation and description of compilation systems from the narrow viewpoint of the discipline concerned rather than from the viewpoint of the process per-se. The

discussion resulted in the creation of general statements - general with regard to the described earth science compilation systems as a whole. The next question to be answered is whether or not generalisation can extend further than merely as a statement. That is, can processes be truly generalised in real terms? In real terms the box on the flow chart becomes a process carried out by a computer software module. In this context, real generalisation means that the same box-name appearing in two or more different flow charts represents the same software module in two or more different compilation systems, or the same module applied to more than one type of data, a significantly higher order of generalisation than that of the general statement.

At this point we must therefore seek out and attempt to surmount the next series of obstacles. Namely obstacles to the realisation of generalisation.

An obvious and formidable obstacle that workers in this field encounter is the multiplicity of mutually incompatible data set contents and structures employed. Another obstacle is the diversity of programming languages and types of computer employed. Attempts to establish a generally accessible earth science data bank even within one country and within a narrow range of disciplines have failed because of the insurmountability of these obstacles. Each participant was pleased to contribute, provided that everyone else adopt their particular data structure, content, and means of manipulation'. One might expect however, that different types of data collected by a single organisation and processed by the same machine, would provide an ideal environment for generalisation. Shih (1973) describes the "shipboard computer system for processing and displaying Bathymetric, Gravimetric and magnetic data at sea" employed by the Bedford Institute of Oceanography. Three types of data collected at the same time, by the same organisation, to be processed on the same machine. Reading the description of the 36 programs listed in this work, however,

reveals only two which appear to be of a truly general nature. "Plot axes for plotting profiles" and "read x, y from paper tape and plot x, y on plotter". "Plot bathymetry profiles" "plot magnetic anomaly profiles" and "plot gravity anomaly profiles" appear in three separate programs. The obstacle to generalisation in this case was probably the very small memory capacity of the computer being used. Generalisation may have demanded too much core storage.

To summarize, obstacles to the realisation of generalised methods of earth science survey compilation are:-

i) Restrictions imposed by computer capacity.

ii) Diversity of machines and programming languages.

iii) Diversity of data structures.

The first obstacle has been, or is being, surmounted, simply by the rapid rate of development of E.D.P. hardware. Random-access and mass-storage memory capacities that were inconceivable ten years ago, are now commonplace. Hence, if generalisation does in fact make greater demands upon computer capacity these demands can be met at minimal extra costs.

The second obstacle, that of diversity of machines and programming languages, is also surmountable as a result of hardware developments and the system-software development that have accompanied them. Compilers for all the commonly used programming languages are today available even for most types of mini-computer. Modern microprocessor based computers are capable of "emulating" other types of machine. i.e. are capable of being made to mimic a variety of other types of machines thus facilitating the transfer of software. Hence, if generalised compilational software was written in the standard form of one of the more common computer languages, or designed to be so written, then this obstacle could be overcome.

2.3:1 The principal obstacle: diversity of data content and structure

The third obstacle, that of the diversity of data structure and contents is the most formidable one. It is possible to impose a rigid structure and content convention on a given type of data provided that consensus can be attained among the workers in the discipline. Dampney et. al. (1978, p. 216), for example, make recommendations for an industry-wide standard formats for the digital recording of seismic data. It would be very difficult to establish rigid content and structure conventions for all types of earth science survey data. Even if it were feasible it would only assist in generalisation within a discipline and not across several disciplines.

Such conventions could in fact, retard rather than advance the interests of generalisation as significant new developments in technique or instrumentation could require a revision of the standards.

This problem, the third obstacle, is not unique to earth science compilation systems. It is to be found in all realms of computer activity – science, commerce, industry, etc. – and has been solved with varying degrees of success for many specific and some general applications. It does not appear to have been solved for specifically geophysical applications. A search by this writer though of over 700 references containing the highly general keywords "computer" and "geophysics" failed to reveal a solution specific to the problem at hand.

An examination of work in other fields may, however, reveal a solution close enough to be adaptable to the current problem. At least, it could provide useful guidelines or hints as to a solution in the current field.

Accordingly, the next chapter will address itself to such an examination..

3.0 Existing solutions to the problem of diversity of data structure and content.

The problem, in essence, is as follows:

With all high level programming languages in common use, it is a simple matter to create a program which applies a specific process to a data set of specific form and content. This means one data form – one module. Hence, as has been described in the examples in Chapter one, essentially the same process may be expressed in many different modules, each one tailored to a specific form of data.

In the context of this work, "generalisation" requires programs which are independent of the data content and structure, and vice versa.

In their standard forms, the high level languages have some limited facilities for data independence. FORTRAN, for example, permits a program to read the data record format at run-time. Thus different data formats may be input to the same program without requiring internal changes. The facility does not, however, permit higher level structures to be defined. Thus if not just the record structure changed but also the juxtaposition of records of different structure then program changes would have to be made.

A further disadvantage arises from the fact that it is preferable to include the data structure definition with the data set rather than maintain it separately. If this was done using the "run-time format input" facility, then the data set would become "language dependent", i.e. a data set defined by a FORTRAN format at its head could not be interpreted by a COBOL program unless quite complex "translation routines" were specially written for the COBOL Program.

Hence, for comprehensive data independence, specialised software modules must be created (either with a high level language or in an assembly language) which may be called from a user program to effect data transactions which, from the user's viewpoint at least, are independent of data structure.

3.1 Data base management systems (DBMS)

The most widespread use of data-structure-independent software is within "data base management systems" (or "information retrieval systems"). At one time the bulk unit of data was regarded as the "file" in which all records were of the same form and content, hence data specific software was adequate. Modern usage now regards the bulk unit of data as the "data base" – a collection of several files which will have different contents and which may also have different structures. If the same software module is to be applicable to all data within the data base it must perforce be data-structure independent.

Martin (1975, p. 19) defines a data base as follows:

"A data base may be defined as a collection of interrelated data stored together without harmful or unnecessary redundancy to serve one or more applications in an optimal fashion: the data are stored so that they are independent of programs which use the data; a common and controlled approach is used in adding new data and in modifying and retrieving existing data within the data base. One system is said to contain a collection of data bases if they are entirely separate in structure".

Date (1976, p. 1) provides a much briefer definition:

"A data base is a collection of stored operational data used by the application system of some particular enterprise".

Olle (1976, p. 389) defines the data base in contrast to the above mentioned file as follows:

"The difference between a data base and a file, in terms used prior to the advent of data processing, is perhaps analogous to the difference between a thoroughly cross-referenced set of files in cabinets in a library or in an office and a single file in one cabinet which is not cross-referenced in any way."

The important difference is that the data base must be stored in the computer on direct-access storage (such as disks) in order for the computer's central processing unit to be able to utilize the cross-references within a reasonable time."

The relationship between the data base and the data base management system (DBMS) is described by Olle (1976, p. 390) as:

"A data base is a set of data stored in some special way in direct access computer storage. A DBMS is the software that handles the storage and retrieval of the records in this data base."

Describing in greater detail the role of a DBMS Martin (1975, p. 66) lists the following eleven events which take place when an application program reads a record by means of a DBMS (see Fig. 14).

"1. Application program A issues a call to the data-base management system to read a record. The program states the programmer's name for the data type and give the value of the key of the segment or record in question.

2. The data-base management system obtains the subschema (or program data description) that is used by application program A and looks up the description of the data in question.

3. The data-base management system obtains the schema (or global logical data description) and determines which logical data type or types are needed.

4. The data-base management system examines the physical data-base description and determines which physical record or records to read.

5. The data-base management system issues a command to the computer operating system, instructing it to read the requisite record(s).

6. The operating system interacts with the physical storage where the data is kept.

7. The required data are transferred between the storage and the system buffers.

8. Comparing the subschema and schema, the data-base management system derives from the data the logical record needed by the application program. Any data transformations between the data as declared in the subschema and the data as declared in the schema are made by the data-base management system.

9. The data-base management system transfers the data from the system buffers to the work area of application program A.

10. The data-base management system provides status information to the application program on the outcome of its call, including any error indications.

11. The application program can then operate with the data in its work area".

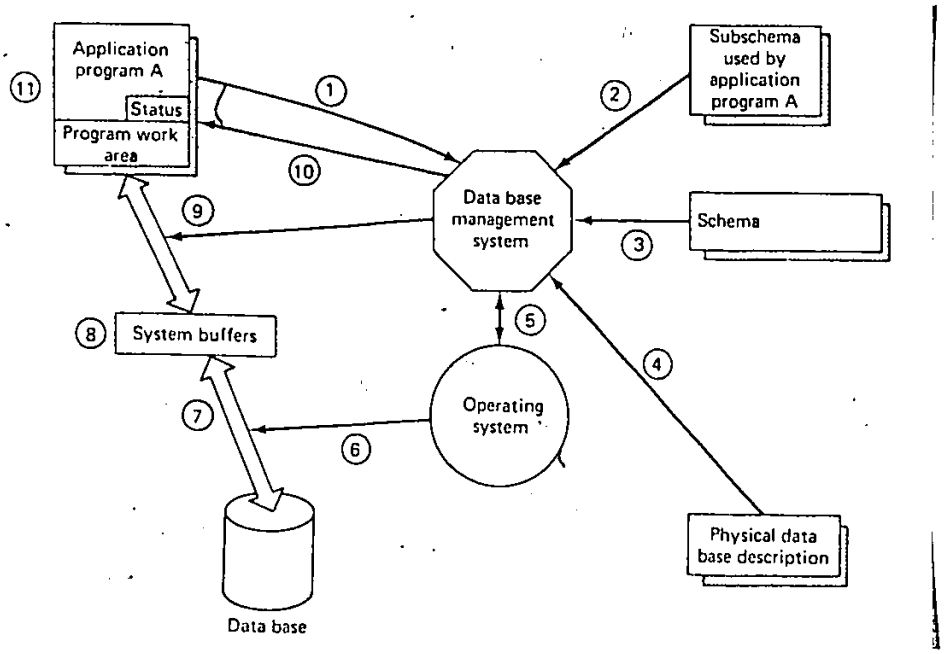


Fig. 14 The sequence of events when an application program needs a record, using a data base management system. (after Martin).

3.2 Data independence within the DBMS

All the authors cited above in this chapter rank "Data independence" as being of paramount importance to the DBMS.

Martin (1975, p. 27) extends the concept to two levels – logical and physical independence. Physical independence means that the content of the basic data aggregate (number, type and order of attributes) and locations of each aggregate may change without changes to the application programs. Logical independence means that the overall logical structure of the data set may change without necessitating changes to the applications programs.

The basic means by which the DBMS achieves data-independence as revealed in Martin's "eleven events" above – specifically by events 2, 3, 4 and 8, are:

i) There exists a "sub-schema" which describes the data from the viewpoint of the program (e.g. the program needs a string of consecutive pairs of X, Y coordinates to plot a graph)

ii) There exists a "schema" which describes the "entire logical data base" e.g. what, inclusive of X and Y, does the data base contain and what relationships exist between different components.

iii) There exists a physical data base description which relates the names and relationships in the schema to addresses and linkages in the physical storage medium.

iv) The program calls for its X, Y data set; the DBMS employs the physical data description to isolate X, Y from the other contents and retrieve them. If the structure of X, Y as defined in the schema differs from the program requirement defined in the subschema, the DBMS transforms the data to the form required by the program.

Hence, regardless of changes to the data content and structure of the data base, the programmer may call for data simply by name and retrieve it. Hence data-independence is achieved.

Date (1976) has a somewhat different viewpoint with almost entirely different terminology. In essence, Martin's "schema" is Date's "Data model" and martin's "sub-schema" is Date's "sub-model". Martin's "physical description" does not find a counterpart in Date, due to an essential difference in their concepts and method of presentation. Figures 15 and 16 illustrate Martin's and Date's viewpoints and terminologies. Note that in Martin's diagram the sub-schema, schema and physical description are independent. On the diagram (15) there appears "mapping" which floats in space outside of the sub-schema. It is indicated that this mapping is carried out "by the data base management system". However:

If i) the schema contained only a description of the logical data base,

ii) the physical description was purely of the physical data base,

iii) the mapping (relationship of one to the other) was carried out by the data base management system,

then iv) the DBMS would have to contain the mapping relationships between schema and physical description. In order to do so it would also have to contain an image of both the schema and physical description thus making their independent existence superfluous, i.e.

Schema	DBMS	Physical
(S)	(S=P)	(P)

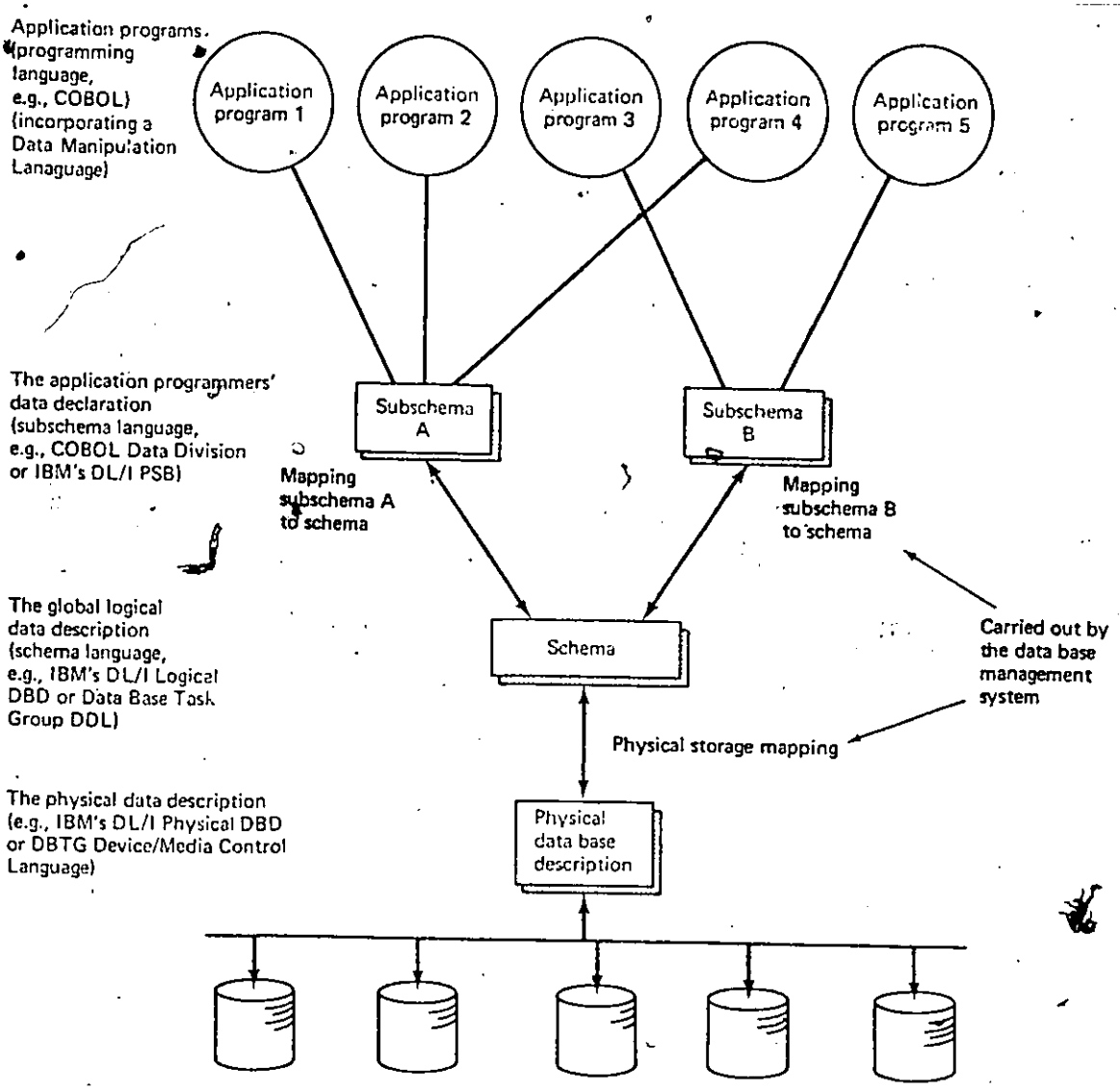


Fig. 15 Architecture for a data base system (Martin)

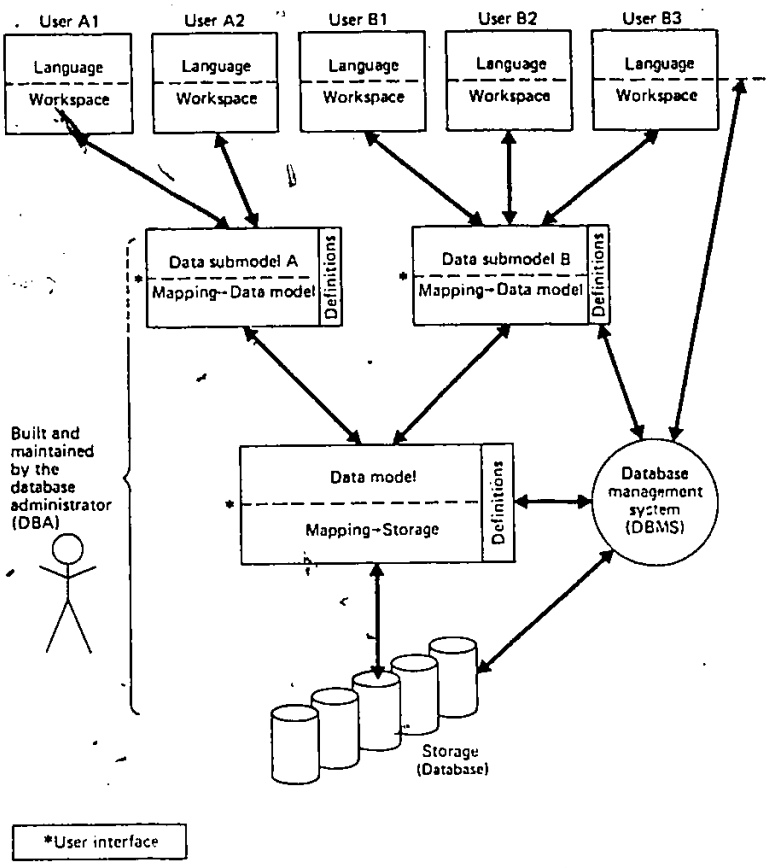


Fig. 16. Architecture for a data base system (Date)

Date's depiction is a much clearer and realistic description of how data independence is maintained. As can be seen in Fig 16, the data sub-model incorporates the mapping relationship between itself and the data model. The data model in turn incorporates the mapping relationship between itself and the physical storage.

If the physical storage changes, the data model need not change. All that is needed are corresponding modifications to the mapping; data model-storage (physical independence). If the data model changes, all that is required are changes to the mapping; – data sub model-data model (logical independence). To retrieve a user record, the DBMS merely follows the "map instructions" from the sub model to physical storage.

The DBMS concept therefore, clearly exhibits the sought after property of "data independence" i.e. the top-level user program may request data by name only and continue to have the request fulfilled regardless of differences in data content and structure between different files within the data base, and regardless of changes made to these the contents and structures over the course of time. Such is the capacity required of a generalized Earth Science compilation system. i.e. to plot a profile of radiometric data or aeromagnetic data with the same module requires that the systems, not the user, take care of the differences in structure and content between the two data sets.

3.3 Implementations of the DBMS Concept

Current implementations of the DBMS fall into three general categories on the basis of the underlying structural model. The categories are "Hierarchical", "Network" and "relational". The mathematical bases of such structures are described in detail by Berztiss (1975).

Hierarchical systems are based upon the "tree" structure, and network systems on the network or plex structure (see Section 4.1). Description and comparisons of several such systems may be found in Canning (1971) and a CODASYL report (1971, no. 1). Tsichritizis and Lochovsky (1976) present a survey of hierarchical systems. The commonly cited work on network systems is a further CODASYL report (1971, no. 2). Relational systems appear to be implemented only in experimental form (Date, 1976, p. 41).

Deutsch and Fong (1978) describe the characteristics which differentiate individual DBMS software packages and provide a list of fifty seven currently available systems.

3.4 Applicability of the DBMS to the generaliation of Earth Science Survey compilation

Sutterlin (1979) categorizes many DBMS's of different types on the basis of their capacities, i.e. general vs specialized; single file vs multifile; machine dependent vs machine independent.

Such a categorization allows the majority of systems to be rejected at the outset as being inapplicable to generalization of Earth Science survey compilation, i.e. machine independence and multifile capability are essential requirements. This leaves only one remaining degree of freedom of choice – specialized vs general. The literature search having failed to discover any systems specially designed for the compilation of earth science surveys, only generalized systems remain. The outlook is, however, bleak in this direction also. Hemperl and Ries (1979) state that an adequate system for generalized scientific data management does not exist today.

It must further be stated that the current work does not seek for generalization within the sciences as a whole, but merely for generalization of compilation systems applied to Earth Science survey data – a double specialization.

The general compilation processes for the five exemplified disciplines were described in Chapters 1 and 2. Chapter 4 will deal with details of data structures in these disciplines. For the moment, however, we may briefly examine the characteristics of earth science survey compilation particularly relevant to the applicability of an existing generalized system.

Two consideration are paramount; volume of data and types of retrieval.

i) Volume of data

Bristow (1979) describes a newly developed airborne gamma-ray spectrometry system. Such a system is capable of sampling up to 1024 energy channels at rates of up to 4 samples per second. Each channel/sample is a 16 bit word, hence the maximum data acquisition rate is 8 k bytes/second or approximately 30 M bytes/hr. During the field season as much as 1000 hours of on-survey flight may be executed resulting in the acquisition of 3×10^{10} bytes of digital data by a single acquisition system.

All of the cited general works on data base management systems state that the data base must be on some direct access storage device. The largest capacity of commonly available devices is approximately 3×10^{8} bytes on a disc system. Thus 100 high-density disc devices would be necessary to contain the maximum output from a single season's operation of a single multi-channel gamma-ray spectrometer.

This is an extreme case of course. The output of aeromagnetic data per season is considerably less. It is still large enough, however, to rule out the use of disc storage for more than temporary residence. Gravimetric data output is still less, but again large enough when accumulated over several years to make permanent on-line storage very costly. Geochemistry and drift sedimentology alone fall within the data volume limits that would make permanent on-line storage a reasonable proposition. Though reasonable, in terms of cost, such storage is, however, not necessary for the compilational processes. As will be explained below

i) types of retrieval

Most compilation process retrieval requirements are satisfied by linear-sequential input from, for example, a digital magnetic tape device. The compilation of aerogeophysical and gravimetric data, however, requires, for at least one stage in the process, "direct" or "random" access. The retrieval requirement for these stages, though, is well known in advance and is not random access at the record level but random access to (very) large blocks of sequential data (e.g. sorting the lines of traverse data into geographical order; combining and sorting flight path data segments, transposition of levelling or numerical surface matrices, etc). As such, the sophisticated unpredefinable, random access to individual records that the DBMS provides is not required of an earth science compilation system. This is not to say that they could not be

employed, but to do so would incur an enormous penalty of computational overhead. The "eleven events" that Martin describes as taking place when an application program gets a record from the data base via a DBMS would become about 11 billion events during the course of compilation of a large aerogeophysical survey.

The only advantageous application of a generalized or specialized data base management system would be to drift sedimentological or geochemical data alone, after compilation is complete. As will be demonstrated, however, the provisions for processing during compilation may themselves prove satisfactory for post-compilational requirements.

In conclusion, we may note that the concept of data-independence as demonstrated by the generalized DBMS remains as a vital requirement for a generalized earth science survey compilation system. The problem of achieving it, however, also remains, as the implementation of the concept within a DBMS is not applicable to the needs of earth science survey compilation.

What must be sought for and found is a concept of data structure pertinent to the needs of survey data and compilational overheads. It is the opinion of this writer that the best place to begin the search is within these very data and processes themselves.

4.0 Content and structural characteristics of Earth Science survey data sets

4.1 Fundamental concepts of data content and structure

4.1.1 Content

A great amount of work has been done and a great amount written on the subject of data content and structure. The language employed is necessarily complex, different workers use different words to describe the same thing (e.g. "record" = "segment" = "tuple") and sometimes use the same word for different things. Martin (1975, p. 44) in describing the nature of data invokes three "realms", namely "reality" "information" and "data". The first is where things exist, the second refers to how we conceive of the information pertinent to these things and the third is where the information is stored in digital form. In the realm of reality there exist "entities" which possess "properties". In the realm of information the property is represented by an "attribute" which has an "attribute value". In the data realm the attribute is represented by a "data item" which has a "data item value".

Sutterlin et. al. (1977) illustrate the variety of terminology by providing commonly used alternatives for many of the terms they employ. They define four "levels of structure" which have influence in the design of a data base namely "the natural data structure level" (Martin's "real realm"), the "seeming data structure level" (Martin's "information realm"), the "abstract data structure level (No parallel in Martin) and the "storage data structure level". (Martin's "data realm"). Sutterlin et. al. then go on to describe the "entity" as the "basic unit" of the data base. Where Martin had "data element name" Sutterlin et. al. have "field name" for which they provide the alternative terms "element," "property", or "characteristic". Martin's "data element value" is referred to as "Field Content (Item, Property, or Measure)".

Martin and Gordon (1977) state:–

"An entity is an object or event of interest, attributes have values which describe an entity."

These authors then go on to give an example referring to the analysis of a collection of geological field data. They state:–

"STATION is an obvious candidate to be considered an entity in the data model for this application, attributes of the entity STATION could include location, observer's name, and reference number."

A statement with which this writer cannot but agree. They continue in the next paragraph, however by stating: –

"A petrologist might insist that another entity be PETROLOGY with attributes name, color, grain size, mineralogy etc. – – – this model then would include a relationship – 'HAS' – between the entity STATION and the entity LITHOLOGY".

It is this writer's opinion that the latter paragraph creates unnecessary complexity by permitting one entity to "have" another entity as it necessitates that the set of entities be divided into sub-sets, according to their states of "possession" and "possessedness".

This complexity can be obviated by redefining PETROLOGY as a "property" rather than an entity, e.g. "STATION" possesses the property of LITHOLOGY.

Let us therefore establish the terminology to be used in this work.

The entity, the object of interest, we will retain strictly in the real world. Data sets do not contain entities. We can specify the nature of our entities more appropriately by referring to them as "samples", a geologically familiar term. That disciplines such as aerogeophysics do not collect actual physical samples is no obstacle to this usage as one can simply invoke the existence of an immaterial sample – i.e. the sampled point in space and time.

We will state that entities possess properties which can be measured or observed and the measurement or observation is an attribute of the sample. All things that can be recorded which are pertinent to the sample are attributes. To have some data items known as attributes and others as entities and yet others as "identities" is not only confusing, it is superfluous. Furthermore, qualitative differentiation between data items at such an early stage tends to channel the ways in which we think about them at later stages. This in turn tends to channel the methods we design to manipulate them which leads to artificial and unnecessary restrictions on their manipulation.

An attribute, therefore, is a specific value of an Earth science or related property. As such, an attribute must consist of two components, the attribute type (the property being measured or observed) and attribute value (the result of the measurement or observation.) When the attribute is digitally encoded, the attribute value becomes a "data element value" the attribute type a "data element type".

All the attributes related to an individual entity, when put together, constitute an "aggregate". A grouping of the attributes of more than one entity will be referred to as a "super aggregate" and a grouping of a subset of the attributes of a single entity will be referred to as a "sub-aggregate".

Surveying consists of sample (entity) collection. Data aquisition is observing or measuring and recording the attributes of the entities. The data set is the collection of all the aggregates of attributes recorded. To summarise:–

i) ENTITIES possess PROPERTIES

ii) SPECIFIC VALUES of properties are measured or observed.

iii) the specific value of a particular property is an ATTRIBUTE of the entity.

iv) An attribute has two components its TYPE (the particular property) and its VALUE (the observation or measurement as quantified).

v) All attributes of a particular entity constitute an AGGREGATE which defines and describes the entity.

vi) the digitally encoded form of an attribute is a DATA ELEMENT. Attribute type becomes DATA ELEMENT TYPE. Attribute value becomes DATA ELEMENT VALUE.

vii) the digitally encoded form of all attributes of all entities is the DATA SET.

4.1.2 Structure

"Structure" of the data set refers to the manner in which the above mentioned data contents are put together to form the data set, and the pathways that are provided to allow retrieval of desired data items. The lowest order of structure would be one with no deliberate structure at all. Retrieval would consist of selection and examination of items in turn until the desired item was found. The highest order of structure would consist of the items being stored in labelled locations to which existed a comprehensive cross reference catalogue allowing the location of any item possessing a given set of attributes to be found directly by reference to the catalogue. i.e. in any search, regardless of the criteria for selection of the item, only that or those items which satisfied the criteria would be handled.

The most appealing feature of the "no structure, no catalog" system is the ease of "updating" the file or data set. Any changes required are simply made – no other action is necessary – and the retrieval system functions just as well as before. With the highly structured, fully catalogued system such as the library, deleting or adding a single book requires that an entry be deleted from or added to all the various

catalogues. The most extreme example of the cost of updating a catalogue is provided by the case where the collection of items is also catalogued by some sequential position index. In which case a single addition to or deletion from the collection requires every catalog reference to items further down the sequence to be changed. (e.g. the index of contents of a book; If an extra chapter is added, the page numbers of every succeeding chapter will very likely have to be changed.)

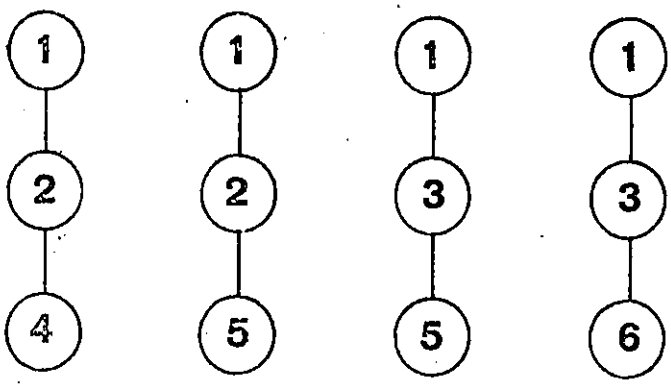
It is evident that the appropriateness of a structural system depends upon the contents of the data set, and on the manipulations to which it will be subjected.

Three basic categories of structure can be described as graphic forms. Namely "flat files," trees" and "networks." (Martin, 1975). In a flat file, relationships between items are implicitly expressed by placing all related items together in the same "record" or aggregate. With trees and networks relationships between "nodes" (items) are explicitly expressed by linkages which join the related nodes. A tree structure is, in reality, a special case of the network with specific properties. It is hierarchical. The uppermost level of the tree has only one node (the root). Every other node has one and only one node related to it at the next higher level, (its "parent") and may have zero, one or more nodes related to it at the next lower level (its "children"). With a network any node may be related to any other node or nodes without restriction.

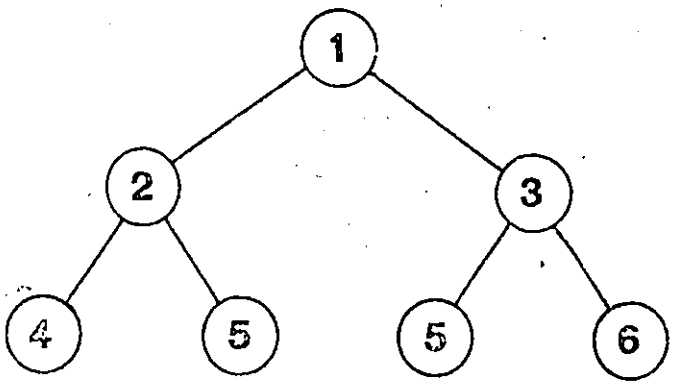
Figure 17 shows these three types of structure.

Certain physical implications are, present in these graphic forms. It can be seen, in the case depicted by figure 17, that a network may be decomposed to a tree or group of trees and a tree may be decomposed to a flat file or group of flat files. The penalty for this decomposition is the necessity to replicate nodes which

A) FLAT FILE



B) TREE



C) NETWORK

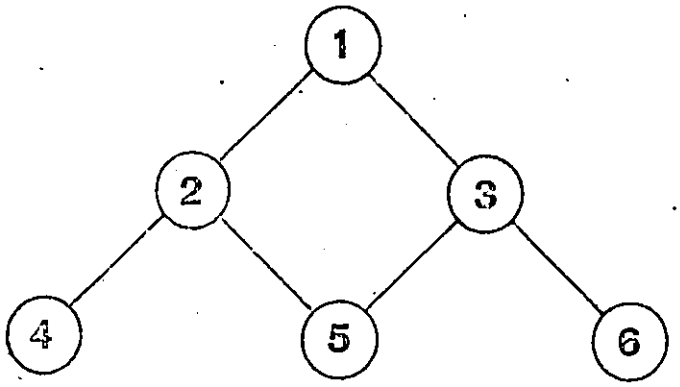


Fig. 17 The three basic types of structure. These three examples are equivalent. C may be decomposed to B, B may be decomposed to A.

previously occurred only once. The system shown in figure 17 requires 6 nodes to express it as a network, 7 nodes as a tree and 12 nodes as flat file. For more complex networks the redundancy would be much greater. The consequence of this in terms of the data set and its manipulation is that structural simplicity is a requirement which may conflict with economy of storage and efficiency of retrieval.

We can say in conclusion that with regard to data structure, a certain freedom of choice exists i.e. if we can impose a complex structure we could also impose a simple one. The actual choice should be the best compromise between structural simplicity and economy of storage/retrieval.

It is also evident that the degree of complexity of structure that can be advantageously achieved depends to some degree upon the existence of natural relationships in the data. i.e. A network may be decomposed all the way down to a flat file but if the nodes of a flat file contain no repeated values then we could not reduce the number of items stored by development of a network.

We will now examine the content and structural characteristics exhibited by real cases of Earth science survey data.

4.2 Earth Science Survey data Sets

We will define the "cardinal form" of the Earth science survey data set as follows.

i) All the entities described belong to the same class (e.g. aeromagnetic measurements not mixed up with drift sedimentology data)

ii) All the aggregates belong to the same class (i.e. each aggregate contains the same number and type of attributes, only the attribute values differ from aggregate to aggregate)

iii) There is a unique one-to-one relationship between the entity set and the aggregate set. Hence no aggregate can describe more than one entity and no entity can be described by more than one aggregate. Thus aggregates have an independent existence related formally only to their entity and not to each other. e.g.

aggregate No. 1) = N, X, Y, A, B, C (1)

aggregate No. 2) = N, X, Y, A, B, C (2)

... etc ...

Where "N" could be a sample number, X, and Y spatial coordinates and A, B, C, etc, other attributes of the particular sample.

We have, in fact, shown that most data sets should be reducible to this form. But we have also shown that even if the data set is reducible to this form it may not always be efficient to do so.

We may therefore expect valid Earth science survey data sets and subsets to possess some other form, either because they are still in the process of compilation or because it is more efficient.

The two important reasons for employing some alternative form are those of economy of storage and efficiency of handling, but often, the principal dictator of the actual form is the "dimensionality" of the data.

Although digital measurement and digital processing must deal solely with discrete quantities, one may conceive of data as being spatially continuous in one or more dimensions according to the type of data. The aggregate described above as the cardinal form, though possessing X–Y coordinates is in fact zero-dimensional, referring only to a single point in space. Other types of data may differ and we may speak of "point", "line", "plane" or "volume" data, each of which will possess its own,

most appropriate structure dictated by its dimensionality. Hruska (1976, p. 299) in discussing the problems of data management systems applied to economic geology notes that the "environmental" sciences, have perhaps the greatest need for improved data management systems due to this fact. A basic aggregate form may be employed however, which appears to be similar to the cardinal form i.e.:-

$$(SD)\ A,B,C, \qquad (3)$$

where (SD) is a definition of the spatial region, point line, plane or volume, to which the attributes A etc. apply. Examples of all four types of spatial region exist within digitised geological information. Other than the obvious point-sample data, the spatial definition (SD) could consist of the depths to two points on a core sample, the core between which possesses a single aggregate of geological attributes. A single aggregate of geological attributes could likewise be applied to a spatial region which was planar (an outcrop upon a geological map) or to a volume of space (a stratum whose upper and lower boundaries are defined over an area).

In all the above mentioned cases a spatial region is defined to which a succeeding aggregate of attributes applies. An attribute has been defined as a single specific value of a particular property. Then with this form of aggregate each of the specific attributes must apply to the spatial region as a whole, i.e. the attributes are constants within the spatial region – a 3 dimensional spatial definition of a stratum followed by the attribute "Limestone" means that all points within the space possess this attribute.

Cases exist however, where one or more of the attributes are not constant throughout the spatial region referred to, and for which a single specific attribute value is inadequate. Two alternative means may be taken to satisfy this requirement.

The simplest is to subdivide the region into sub-regions over which the attribute or attributes in question can be considered to be constant. This allows the original spatial region to be described by a set of several aggregates of the same form as previously employed, each of which describes a sub-region of the original.

The other alternative is to retain the definition of the complete spatial region as a separate entity but employ some means within the aggregate to define the variation of the attribute (s) over the region. This type of aggregate could take the form:–

$$(SD)\ F(A), F(B)..... \qquad (4)$$

where F(A) etc. are a set of mathematical functions of the attributes which describe their behaviour over a region of space rather than a constant value applicable throughout the region.

Within the realm of Earth science surveys the "Function" describing the behaviour of an attribute over a region of space is only rarely available as an analytical algebraic expression. Variations over a region are usually defined by a numerical expression i.e. a set of discrete numerical values distributed at points throughout the region. (to distinguish this case from that of the algebraic function, the notation "Ai, Bi" will be employed where "Ai" represents a string of several values of the attribute "A" etc.).

This fact makes the first alternative – subdivision of the spatial region – a commonly encountered case. Many cases can still be found however, which adopt the second alternative even though the attributes within the region are defined only as a set of spot values. The practicability of such a usage depends upon the spatial distribution of the spot values within the spatial region. If the distribution is

undefinable by some separate function, then each spot value must possess its own definition of location and no advantage is to be gained by separating the spatial definition from the attribute list.

If however a simple means exists to separately define the distribution of the spot values then a different aggregate structure can effectively be employed.

The simplest form of separately definable distribution, and the one most commonly encountered, is a regular distribution, i.e. the spot values are distributed at points separated by constant distance in each of the applicable dimensions – equispaced in one dimension at points along a line; equispaced in two or three dimensions at points falling on the nodes of a regular grid.

Definition of the distribution is a necessary component of the aggregate but it can be stated separately from the attributes and the spatial region to which they apply; the aggregate thus taking the form:–

$$(SD)\ (DD)\ Ai,\ Bi,..... \qquad (5)$$

where (DD) is the distribution definition of the attributes.

This being the most complex form of aggregate so far described, a pertinent, real example is appropriate, e.g. consider a data set containing the definition of magnetic total field over a 1:50,000 scale NTS Map sheet at points situated on the nodes of a regular UTM grid. The data set could be structured as a flat file cardinal form. For each data point the aggregate would contain the elements

$$X,\ Y,\ G \qquad (6)$$

where X and Y could be Lat-long, or UTM coordinates (or both) and G is the total field. The definition of the spatial region covered is implicit; that is – all coordinates specified lie within the particular NTS sheet. Such a data set would be simple to

construct but would require at least three elements for each data point. (storage inefficiency) Furthermore, with such a structure it would be difficult to predict the location of individual attributes in the data set according to their spatial location; (retrieval inefficiency). A much more economical structure, and one which facilitates data retrieval is possible as the data points fall upon a regular grid. The aggregate would be in the above mentioned "complex" form:–

$$(SD)\ (DD)\ Gi \qquad (7)$$

where (SD) would define the map area (NTS sheet number, Latitude-longitude boundaries); (DD) would define the data distribution (UTM of grid origin, north & east grid cell dimensions, number of rows and columns) and Gi would define the total field values over the grid simply as a listing of values to be encountered by passing along successive grid rows by the conventional route (left to right, lowest row first). By these means only the attribute whose variation cannot be predefined in a concise manner need be present for each data point. The locational attributes are fully described by a few elements preceding the list of specific values of the other attribute.

As the number of aeromagnetic data values within the map sheet increases, the space saving over the cardinal form approaches 66% and the position of a magnetic field value within the data file may be predicted from its spatial location on the map. (e.g. A total field value in column K on row L will be the N^{th} element in the data set, where; $N = K + (L-1)*NGR$, and NGR is the number of grid points per row)

By our original definition of terms, an aggregate is an assemblage of the attributes of an entity. We may name our entity the "sample" – in this case merely a sampled point in space and time, and the spatial coordinates of samples are just as much attributes of the entity as are geophysical or geological observations. The last

example, (the "complex" form of aggregate for gridded aeromagnetic data) however, refers to not one, but several "samples" and should be described as a "super-aggregate". i.e. a restructured collection of many individual aggregates in a form best suited to the needs of data retrieval and economy of storage. Something, though, has been lost by the re-structuring, and conversion from cardinal to complex form and back again involves more than mere re-arrangement of the data.

This introduces the concept of implicit "information" which must be held in mind when actual instances of data structures are being examined from theoretical viewpoints. In the above example for an aeromagnetic grid, the data set contains the parameters of a function which allows the spatial coordinates of the aeromagnetic values to be generated. It does not contain the spatial coordinates themselves. Hence aeromagnetic values laying in some sub-region of the map could not be retrieved directly. A computation module must be included in the retrieval system to create the cardinal form aggregates.

We will now examine two specific, and contrasting examples, in greater detail.

4.2.1 Aeromagnetic data

The compilation process for aeromagnetic data (Section 1.4) involves a progression through several data sets of different content. The three principal data sets are the two initial ones – In-flight data and Post-flight track data and the final combined mappable data set. In addition there exist several intermediate or derived types of data set such as the leveling data sub-set and the interpolated regular grid data for contouring. The in-flight data essentially consist of a set of in-flight aeromagnetic measurements each of which possesses an associated

"sample number" – a fiducial number. The post-flight-recovered track data essentially consists of a set of pairs of spatial coordinates each pair of which also possesses a fiducial number. The essential content of the final, mappable data is a set of corrected aeromagnetic measurements each of which possesses a pair of spatial coordinates. The purpose of the fiducials in the first two data sets is to allow correlation between the two so that they may be combined into the third data set.

As stated, the above components are the essentials. Such data would be sufficient to allow the compilation processes to be applied and a map produced. The simplicity of this data content, however, would require that the data be handled and processed record by record – i.e. one aeromagnetic value and fiducial at a time, one pair of track coordinates with associated fiducial at a time, etc. The retrieval requirements of the first compilation stages – physical verification – are simple, and retrieval of one record at a time would be adequate. For the later processes, however, logical verification, sorting and merging, leveling etc. – the retrieval requirements are more complex, and though it would be possible to fulfill the requirements from such basic data the processes would be protracted and costly and the detection of errors hindered. Accordingly further attributes have always been included in the data set, which have the effect of identifying a larger quantity of data that may, in logical terms, be handled and processed as a unit. Such elements include Line number, Flight number, Map sheet number, etc. This allows the retrieval, for example, of all data on a given map sheet as a logical unit.

The stage at which the in-flight and post-flight data sets are merged has as its objective the assignment of spatial coordinates to aeromagnetic measurements. The required spatial order of the resulting data set will have been defined before this stage and hence each of the two data sets can be sorted into this order before being

merged. This obviates retrieval of elements of one or both sets at random. By sorting, the elements are positioned within the data set in the same order that they will be called on by the merging program – a relatively simple retrieval requirement.

The next stage, leveling, has more complicated requirements. The normal survey practice is to fly a set of control lines perpendicular to the principal survey traverses. The first goal of this stage is to retrieve the data from some or all of the points where traverses and control lines intersect. The purpose is to form a set of data aggregates each of which contains the data from a traverse and from a control line at the point where the two intersect. The control line data could be broken into short segments, each one covering an intersection point, labeled with the traverse number at the point, then the segments sorted by traverse number and spatial coordinate into the same order as the traverse data itself. The elements of the respective data sets would then be arranged in the order that they would be called by the first stage of the leveling process. This would once again obviate the need for non-sequential retrieval of data elements. Non-sequential access to data will eventually be required at some stage of the work, and for the efficiency it gives, might as well be incorporated at this point. In this context, non-sequential retrieval means that neither the control nor traverse data sets need be re-ordered before submission to this stage. The traverse data can be taken in sequential order with no need to backtrack, but elements from the control line data will be required in non-sequential order. (i.e. The intersection points may be taken in order of occurrence along each traverse. One traverse at a time. But each traverse could cross <u>all</u> control lines, therefore segments of <u>all</u> control lines will be required for <u>each</u> traverse.)

Though such a retrieval requirement is not strictly random it is definitely nonsequential and may be regarded as random for all practical purposes. A further requirement for non-sequential retrieval may arise at the gridding stage, depending upon the type of interpolation function and gridding algorithm employed.

We may describe the data structures of the various stages of the aeromagnetic survey data set as follows:–

i) In-flight data set.

The common form is a super-aggregate:–

$$L, Gi, Ai \tag{8}$$

where; L contains information pertinent to a whole flight line of data, such as line Number, direction, date etc.; Gi is the set of aeromagnetic measurements and Ai is the set of auxiliary information such as fiducial number, altitude etc.

ii) Digitised track data.

The data is often output from the digitiser in a cardinal form as a result of design peculiarities and limitations of such devices, e.g.:–

$$M, L, F, X, Y \tag{9}$$

where;– M = Map Number; L = line number; F = fiducial number and X, Y are the cartesian coordinates of the digitised point.

Often, the first process applied to the data is super-aggregation by map or line number to the form:–

$$M, L, (X, Y, F)\, i \tag{10}$$

iii) The leveling data sub-set.

Such data provides a good example of the advantages of separation of a data distribution function from the data itself. The basic form of the aggregate is cardinal e.g.

$$T, C, GT, GC, X, Y \tag{11}$$

where: T and C are traverse and control line identifiers; GT and GC are the gamma values on the traverse and control line respectively, and X, Y are the spatial coordinates of the point at which the traverse and control line intersect.

The points of intersection of all traverse and control lines will not lie on a regular spatial grid therefore pre-definition and separation of this distribution function is not feasible. Each intersection however may be considered to possess another set of coordinates – the traverse and control lines concerned – which when ordered to suit the needs of the leveling process, can be made to possess a regular distribution. i.e. The intersections are ordered as the nodes of a grid where each row is a control line and each column a traverse. (The fact that not all traverses will necessarily intersect each control line – a fact which would destroy the regularity of the distribution – presents no obstacle. One merely creates null aggregates for the missing intersections.) The distribution function may then be separated as a list of the control line identifiers followed by a list of the traverse identifiers. After this definition the sub-aggregates defining each intersection are listed in the conventional matrix order with each sub-aggregate containing only the attributes GT, GC, X, Y. The super aggregate now has the form:–

$$Ti, Cj, (GT, GC, X, Y)i, j \tag{12}$$

where; "i" is an index ranging from 1 to NTR (No. of traverses)

and j is an index ranging from 1 to NCL (No. of Control lines)

It can be seen that the principal component now contains only 4 rather than the previous 6 elements, a valuable saving of space. Furthermore, the location within the data file of any particular intersection can be predicted from the indices of the particular traverse and control line involved.

iv) The final-mappable data set.

This data set contains all the survey information, in a verified and corrected state, required to produce a map. Auxiliary information will generally have been deleted by this stage. The necessary attributes of each data point are G, the total field and X, Y the coordinates of the point of observation. A usable data set would therefore be a set of aggregates each in the cardinal form G, X, Y. Examination of the major processes to which this data is submitted suggest that a more complicated structure would be more efficient. The production of stacked profile maps or the generation of an interpolated grid for production of a contour map require that the data be retrieved as groups each of which contain the data from a complete traverse, or that segment of a traverse which lies upon a particular map sheet. Furthermore, both types of map often require that the flight path, as recovered, be plotted as an underlay to the profiles or contours.

To facilitate these requirements a data structure of the following type can be employed.

$$LD, (P, Q, F)\, i\, (X, Y, G)\, j \qquad (13)$$

which represents a super-aggregate preceded by the line identification information, LD, followed by the recovered track data, which is then followed by the geophysical data where:–P, Q are the spatial coordinates of each traverse track point; F is the fiducial number of the point; X, Y, are the spatial coordinates of each aeromagnetic measurement point and G is the aeromagnetic measurement value.

v) Gridded data.

For automatic contouring and for other development processes, a prerequisite is a regular grid of interpolated data values for which a super-aggregate form, as previously described, is appropriate. i.e.:–

$$(SD)\ (DD)\ Gi \qquad (14)$$

where:–
(SD) defines the spatial element, in this case the map or survey boundaries containing the data. (DD) defines the distribution of the data within the space i.e. grid origin and spacing; G, is the set of grid values in conventional order.

The above data set structures would apply, with only minor modifications, to most other linear or two dimensional Earth science data sets (e.g. Airborne gamma spectrometry, Marine gravimetry.) Although the cardinal form aggregate of each of the data sets is quite simple, data manipulation and storage requirements result in a much more complex "super-aggregate" form being necessary.

The second example we shall consider has a decidedly non-simple cardinal form.

4.2.2. Drift sedimentology data

Shilts (pers. comm.) gives the following list of attributes which are present in the Geological Survey of Canada drift sedimentology data file.

The types of attribute fall into five major categories: Namely:–
identity and locational information, geochemical, mineralogical, geotechnical-physical, and storage locations.

Each of these categories may contain many sub-categories. i.e.:-

1) Identifying information:-

sample no.

Single, split or multiple sample

NTS sheet no.

UTM coordinates

Collector

Depth

Sample type

Stratigraphic position

Oxidation state

Underlying bedrock (1)

Underlying bedrock (2)

Laboratory sample type identification

Colour (field)

Colour (laboratory)

2) Geochemical data

Fraction	Analysis method	Results for:-
type 1 (–250 mesh)	A.A & fluorimetry	Cu, Pb, Zn, Ni Cr, Co, Ag, Fe, Mn, U, Cd.
	Spectrometric	33+ Elements
	Neutron activation	U
type 2 (Sand-size lights)	_	Ni, Cu, M, Al, Si, Na, Mg
type 3 (Meth. Iod. heavies)	_	Same as for type 1, but mostly Ca, Pb,Zn, Fe, Ma, Ag, U, Ni, Cr
type 4 ("Wet" clay)	_	Same as type 1
type 5 ("Dry" clay)	_	Same as type 1
type 7 (Sand size)	_	Cu, Pb, Zn, Co, Ni, Cr
Magnetic fraction		Ni, Cr
type 8 (Crushed rock)	_	Same as type 1

3) Mineralogical

Calcite % (Chittlik)

Dolomite % "

Total carbonate "

Total carbonate % (Solution)

Organic Carbon %

Magnetic minerals wt %

Magnetic susceptibility of sand

Heavy minerals wt %

Clay Minerals %

Chlorite

Illite

Kaolinite

Vermiculite

Smectite

Mixed layer

Serpentine

Exchange capacity

ph

4) Geotechnical-physical properties

Liquid limit %

Plastic limit %

Plasticity index %

Natural moisture content (actual) %

" " " (minimum) %

Gravel %

Sand %

Silt %

Clay %

Mean grain size Ø

Sorting Ø

Kurtosis Ø

Skewness

% in grade at Ø intervals (17 values)

Bulk density

Specific gravity

Porosity %

Permeability

5) Storage location

Address of Bulk original sample

" Sand & gravel washed from clay

" Clay remanent

" Heavy mineral slide

" Clay mineral slide (s)

" Heavy minerals

" " " crushed

" Light minerals

" 250 mesh separate

" Magnetic separate

The above list of attributes for a single sample would occupy approximately 300 or more data elements. Of the attributes noted, very few are likely to occur as common factors among many such records, with the exception of the NTS sheet number and "Collector". Hence little reduction in the bulk of data will be gained by super-aggregations.

In fact, this data file, as is mentioned in section 1.3, contains sub-aggregates. i.e. All attributes of each sample are not collected into a single aggregate. The file consists of the identity and locational sub-aggregate for all samples, followed by the geochemical sub-aggregates for all samples etc.

The compilation process per se for this data is short and simple, consisting essentially of verification of each sub-aggregate, then placement of it in the final storage file. Hence no prolonged and repetetive manipulation process occur which

would enforce some more developed structure – as is the case with aeromagnetic data. The subsequent retrieval of the data for analysis etc, is however, very complicated. So complicated, in fact, as to further inhibit creation of a more developed structure.

Typical retrieval requirements could range from a relatively simple "UTM coordinates and copper concentration for all samples in a specific NTS sheet"; to the more complex "Ni concentration and magnetic susceptibility in type 4 fractions for all Surface tills that lie on Basic Volcanic Rocks" or some such.

The data file could be structured in cardinal form as fixed format records, as is the cardinal form of aeromagnetic data. Thereby, each time a 300 element record was read in, the identity of an attribute would be given by its position in the record.

Such a structure has certain advantages, mainly that of simplicity, but in this case it has also two major disadvantages.

The contents of the aeromagnetic data set, are the same regardless of the agency carrying out the compilation, and have remained essentially the same since the inception of aeromagnetic surveying. The complexity of the drift sedimentological data however, makes it almost certain that the content will differ significantly between organisations and will change frequently in time.

The first disadvantage of the above mentioned structure therefore, is that programs have to be changed every time the data changes. The second disadvantage is that the set of attributes that a sample can possess, or those whose values are available for recording will vary from sample to sample and over the course of time. The fixed length, fixed format record type of structure, requires that a space be

reserved in each aggregate of the data set for all attributes that are potential inclusions. The data set would therefore contain a lot of empty space – an inefficient state of affairs. The above described problems could also be found in other forms of earth science survey data, though probably not to the same degree.

In summary, we can say that for drift sedimentology and similar types of data the situation is opposite to that for aeromagnetics and those types of data similar to it. In the later case, the simplicity of the basic aggregate permits, and the prolonged but repetetive retrieval processes demand, the development of several complex structures. In the former case, the complexity of the basic aggregate and the variability of the retrieval requirement positively inhibit such development.

In conclusion it would seem reasonable to state that Earth science survey data sets possesses distinct content and structural characteristics. But characteristics of great complexity and variability. The statement by Hruska (ibid) was not an overstatement.

Let us now examine some of structures described in terms of the "graphic" structures.

4.3 Earth science data sets as graphic structures

As concluded in section (4.1.2) above, the degree of structural complexity that a data set should exhibit, is that producing the best compromise between simplicity and the efficiency of storage and retrieval. The degree that it can exhibit depends upon the natural relationships present in the data.

Whatever the structure, it should be expressable as one of the three fundamental structural forms – Flat file, Tree or Network.

It should be noted at this point, that as Martin (1975, p. 79) points out, the "relationships" implied by the branches of the tree or the links of the network are solely "logical" relationships. The tree or network diagram does not, and can not, say anything about how these logical relationships are to be expressed in a physical implementation of the data set. The only physical implication is that one data item or group of data items exists for each of the nodes on the diagram. Hence the physical relevance of the increase of redundant storage as a network is decomposed, via a tree, to a flat file.

We will now examine graphic forms of some of the real data sets described in chapter 4.0 above.

Digitised flight path data

The aeromagnetic flight path data is an excellent example of reduction of redundancy by development of structure. The data as it emerges from the digitiser is in flat-file form. Each aggregate contains all the attributes pertinent to an individual sample. In this case the "sample" is a point on a map. The attributes are:- the map upon which the point lies (M), the flight line on which it lies (L), the X and Y coordinates of the point with respect to the digitiser frame and the fiducial number (F). Two well developed natural relationships exist. Each individual map number is related to many line numbers and to an even greater number of track points, and each line number is related to many track points.

The first task of the compilation system is to improve storage/retrieval efficiency by development of the flat file into a tree structure.

The physical implementation honours the logical representation of the tree in that any given value of "Map No." is stored once only and within any map, any given value of "Line No." is stored once only. On the tree diagram one may descend from "Map" directly to any one of the lines and from the line to any one of the data values.

This relationship however is not explicitly honoured in the physical implementation as a sequential file on magnetic tape. One cannot "jump" to any point on a tape. The file structure on the tape is therefore more accurately depicted by a symbolic representation as follows : i.e.–

$$\text{File} = (\text{Map}, (\text{line}, (X,Y,F)_i)_j)_k \qquad (15)$$

Where "i" indicates repetition from 1 to NP (No. of points on the line)

Where "j" indicates repetition from 1 to NL (No. of lines on the map)

Where "k" indicates repetition from 1 to NM (no. of maps)

During the compilation process of sorting and joining the line segments a random access mass storage device is used. The logical structure therein is closer to the tree representation in that one can "jump" directly to any map and from there to any line. The individual data values however are still inaccessable by random access – they must still be accessed sequentially. Hence the most accurate depiction of this structure would be a hybrid combination of the tree and the above symbolic notation.

Aeromagnetic grid data

This data, as shown in section 4.0 above, has a "super-aggregate" structure which was depicted symbolically as

$$(SD), (DD), Gi \qquad (16)$$

Where:– "SD" is the definition of the region of space involved (map sheet no. lat., long. of map boundaries etc.)

Where:– "DD" is the "distribution definition" (the grid origin, X and Y grid cell dimensions and the number of rows and columns in the grid) and "G_i" was the set of all grid values.

The symbolic notation depicts the data file exactly as it lies on the magnetic tape. The relationships are clearly stated in the explanation of the meaning of the symbols. To depict this structure as a tree or any other graphic structure is, however, somewhat difficult. i.e. the physical and logical relationships of the elements are evident when described as above, but the linkages are not direct. As noted in chapter 4 "implicit" relationships are present which require computation to develop them into explicit ones. The structure is eminently practical and well suited to the needs of Earth science compilation, but clearly unsuited to depiction as a graphic structure in any one of the three possible forms.

The aeromagnetic leveling data set

The symbolic depiction of this data set in chapter 4 above was:–

$$TR_i, CL_j, (X, Y, GC, GT)_{i,j} \qquad (17)$$

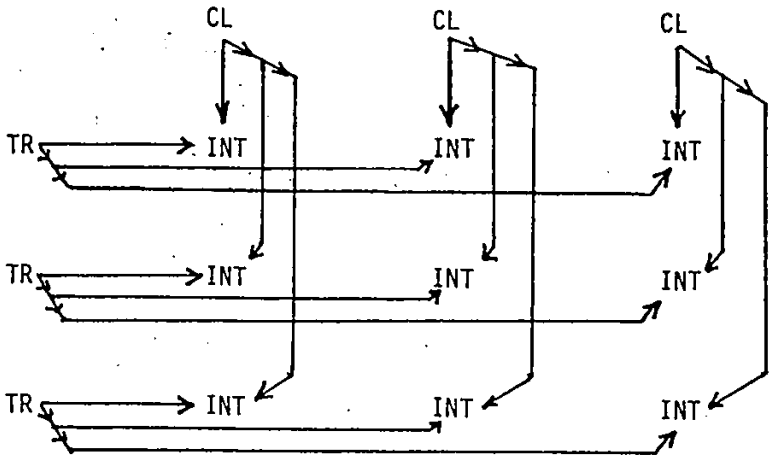
Where – TR_i is the set of all traverse names

CL_j is the set of all control line names

and $(X, Y, GC, CT)_{i,j}$ is the set of data from all control line – traverse intersections. This structure is well suited to formal depiction as a network. (Fig 18).

The logical relationships are exactly as shown by the network linkages. But once again, in the physical implementation these relationships are implicit and must be made explicit by computation. This does not present any disadvantage. On the contrary, the very nature of the data makes this approach advantageous.

Regardless of how an explicit link is made, it will require at least one "word" of storage to define the link. This means that for each traverse number a linkage item would have to be stored for each and every control line (ten or twenty in the average survey). Each control line would require a link for each and every traverse



TR - unique traverse number

CL - unique control line number

INT - data from one TR-CL intersection

Fig. 18 Aeromagnetic levelling data set network structure.

(several hundred in the average survey). Thus for N control lines and M traverses, N x M linkage items would have to be stored. Not prohibitive, but unnecessary. The intersection data is in the form of a rectangular matrix stored in the conventional manner and hence the links can be generated as required. i.e. The intersection data for the K^{th} control line and the L^{th} traverse is the I^{th} entry in the file where $I = K + (L-1)*M$

Drift sedimentology

The drift sedimentology data set is a flat file. In a flat file, relationships are expressed implicitly by placing all related items together. All the data elements in one sub-aggregate are attributes of a given sample and are related to each other by these means. Some of the elements are likely to be replicated in several sub-aggregates throughout the file. i.e. Specific values of "NTS map sheet No.", "collector" and others, are likely to re-occur many times. Hence a more complex structure could be developed. The first question to be asked though – as we have stated several times – is "does a more complex structure produce a better compromise between simplicity and storage/retrieval efficiency?" We have already noted that the retrieval criteria, in this case, are highly complex and variable. A fact which has inhibited the development of a more complex structure. If, however, there was a well developed natural relationship which was also likely to occur frequently in retrieval criteria this could be used as the root of a tree. For example, "NTS map sheet no." This is a well developed natural relationship. i.e. it is likely to have a fairly large degree of replications among the sub-aggregates.

Let the file be broken down into sub sections, each one preceded by a "map header" and containing all the sample descriptions which lie in the particular map. Storage efficiency is marginally improved – The element "map no." now occurs once at the head of the sub section rather than in every aggregate, producing a reduction of about 0.3% in storage space. Retrieval efficiency could, however, be substantially improved, if "NTS map sheet no." was one of the retrieval criteria. i.e. If samples were sought within a particular map sheet, then only the subsection headed by that map no. need be examined. All other sections could be bypassed entirely if their map number was not the one sought. In the flat file form, every aggregate would have to be examined to determine if its map sheet number was the one sought. In the minimal tree structure, only every subsection need be examined. If there were 100 samples per map sheet, then retrieval efficiency would have been improved by 99%!

A remarkable improvement, but a conditional one. Conditional not only upon the nature of the retrieval criteria but also on the existence of some efficient physical implementation of this logical relationship. Even so, the potential is worth considering. It can also be extended. The tree could be developed into several branches of relationship e.g. Second branch – "sample type"; third branch stratigraphic position etc. As the number of levels of branching increases, so however does the complexity, and the potential improvement in retrieval efficiency suffers diminishing returns. This is because the branch structure is created as a model of the most likely retrieval hierarchy. As the tree develops, the retrieval hierarchy it models becomes more and more specific and therefore, less and less likely.

As the retrieval criteria for these data are so variable, the best structure, in the logical sense, would be a network. With a tree, one must start at the top (root!)

and work down. If the root quantity is not one of the retrieval criteria then retrieval is so much less efficient. With a network one may start at any node and branch in any manner. Hence one could begin at the node representing the principle member of the retrieval criteria then branch to the node representing the next retrival criterion. Logically simple and efficient. Let us consider the physical pros and cons of a fully developed network. No attribute value would be replicated in the file. If specific values of many attributes in the flat file are likely to recur often, the network structure could possibly result in a 50% or greater, saving in data storage space.

The linkages must somehow, also have a physical expression. This could take the form of "pointers" i.e. Indices which give the storage address of a related attribute. Each unique attribute value would possess pointers to the storage addresses of those values of the other types of attribute with which it was associated and also pointers to all the members of its own type. e.g. Each unique map no. would possess pointers to each of the rock types with which it was associated. That is, which occurred within the map sheet. Each rock type would possess pointers to those map sheets in which it occurred, etc.

The logical model would thereby become a physical reality. e.g. If the requirement was "retrieve all samples in map number NTS 75, collected by Shilts, where the bedrock was granite". The system would skip directly from one unique map number to the next until "NTS 75" was found. It would then go directly to all "collectors" associated with this map until "collector = Shilts" was found. From this it could go directly to all bedrock types until the unique occurrence "granite" was found. What then? the system could go to all sample numbers related to granite.

But this set is not the one sought. The required set is "sample no. where map = NTS 75 and collector = Shilts and bedrock = granite) which is a very specific sub-set of all the sample numbers associated with the broader case "bedrock = granite". The problem is illustrated as a Venn diagram in figure 19. The entire data file is the set labelled "A". When the system finds the MAP node with value NTS 75 it has immediate access to all sample numbers associated with this node. (Sub-set B). The system then moves to the BEDROCK node with value GRANITE and has immediate access to all sample numbers associated with this node (subset C). Finally at the COLLECTOR node with value SHILTS, it has direct access to all sample numbers associated with this node (subset D). The region of interest is however, subset E, the intersection of subsets B, C and D. Subsets B, C and D are defined by "bundles" of pointers each one emanating from a single node. (Imagine figure 19 as a stage. Above the stage are many spot lights, each spotlight is a node in the network. The cone of light from each spotlight makes a circular patch of light on the stage. The bundle of rays in each cone is analogous to the bundle of pointers emanating from each network node. No node exists which has a bundle of pointers which defines all of, but no more than, subset D.)

The computer can "consider" only one thing at a time. The inescapable physical reality is therefore, that the system would have to:–

i) Find the node NTS 75. Store all associated sample numbers. (set B)

ii) Find the node GRANITE. Store all associated sample numbers (set C)

iii) Find the node SHILTS. Store all associated sample numbers (set D)

iv) Retrieve each member of set B in turn then test for its presence in both of sets C and D. If not present in both, discard it.

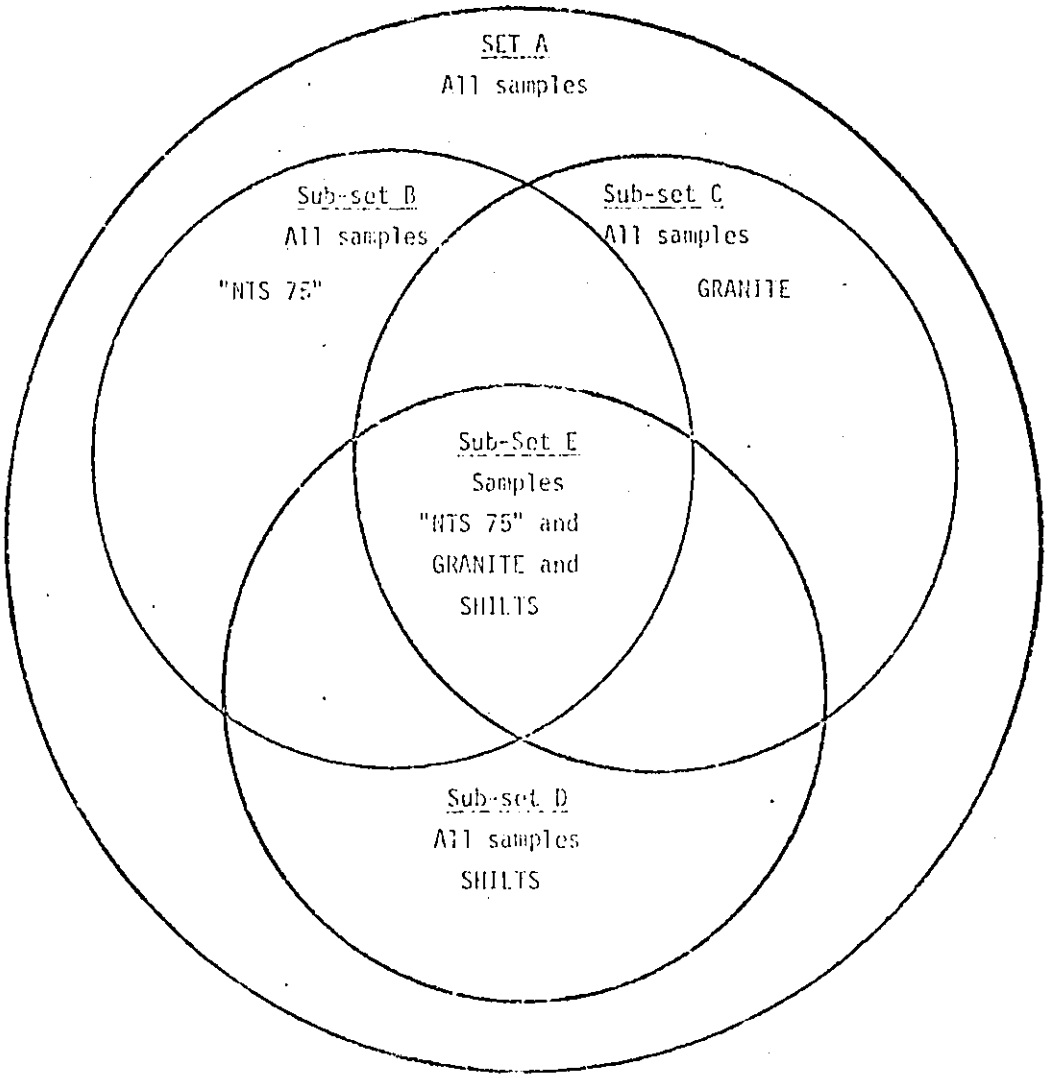


Fig. 19: Venn diagram of the drift sedimentology data retrieval problem.

The target set, E, would be those members of set B remaining at the end. Thus the "improvement" in retrieval efficiency promised by the logical properties of the structure becomes somewhat less of an improvement in physical terms. Still a potential improvement though, and one not necessarily beyond the bounds of physical realisation. We must still, however assess the effects of this system on storage economy. As stated, the number of actual attribute values to be stored would diminish by 50% or more. Given a moderately sized file – 300 attributes for each of 1000 samples, say – we would end up with a few as 100,000 unique attribute values. There still are 300 types of attribute though, and for a fully developed network, each type would require a pointer to all the specific values of every other type with which it was associated. e.g. each unique value of map number would require a pointer to all rock types that occurred within it, to all collectors who collected within it, to all bedrock types encountered within it, etc etc.

Little more need be said. The increased complexity of the network system does create some "logical" improvement in retrieval efficiency in this case. It also could create more than 50% data storage reduction – of attribute value storage. If fully developed however, it would require the storage of several thousand pointers for each unique attribute value thus increasing the size of the file more than one thousandfold and consequently, (even if possible to implement such a monster) causing a catastrophic decline in retrieval efficiency which would eclipse any "logical" improvement.

The inapplicability of a fully developed network structure to the drift sedimentological data file seem to have been proven.

It was, however, a proof by reductio ad absurdum, and it is reasonable to protest that the application need not be carried to absurd limits. One could stop at a "reasonable" depth of networking perhaps.

A reasonable depth however, implies once again that preferred retrieval paths can be pre-determined. If so a simple tree structure offers clear advantages not burdened with the exponentially increasing complexities of a network.

It has been shown that a network can be broken down into a tree or several trees. One possible compromise would be the provision of means to take the cardinal form, flat file, and generate trees from it as required. Thus if a large number of retrievals was foreseen for which a preferred structural hierarchy could be defined, the appropriate tree structure could be generated ad hoc. The hierarchies found to be in greatest demand could be retained as alternative versions of the master file.

It is, however, evident that the apparently cumbersome flat file has distinct practical advantages over its more logically "streamlined" cousins.

4.4 Earth science data sets compared to other types of data

As noted, the examples cited by Martin concern business data sets whose inherent characteristics differ from those of Earth science data. Almost every data base management system in existence was created to serve business needs and hence was taylored to the characteristics of the structures inherent in business data. If these structures are similar to those found in Earth science data then one could employ the systems for Earth science data. The characteristics of Earth science data however, differ from those of business data in several crucial aspects. e.g.;–

Business data sets have natural, externally imposed limits to their size, and naturally imposed limits on their rate of growth. Consider a payroll system for example. The number of employees in any given organisation will fluctuate but will remain fairly constant. Barring economic catastrophy its rate of change will also be fairly constant. Likewise an inventory system, the number of items in stock, the suppliers and the customers will fluctuate within fairly narrow limits over a long period.

Furthermore, business data sets are usually self limiting. Employee records are removed from the file soon after the employee leaves the company. Hence no extra space is required for the records of the replacement.

The Earth science data sets on the other hand have naturally, externally imposed drives to continually grow. Technological and scientific advances are constantly increasing the rate of data aquisition and the number of different parameters that are observed or measured. Furthermore Earth science data sets tend to be cumulative. New observations rarely replace old ones.

Hence the designer of the business system can expect, with reasonable confidence, that the system will be adequate to future needs.

The designer of the Earth science system, however, is faced with the clear possibility, that by the time a system is implemented, its capacity and capabilities will have been outstripped by the growth in size and complexity of the data set.

In the matter of commonality of attributes which lends itself to advantageous development of physical structure, business data is usually more tractable. Unique values of the business data attribute are generally members of a small set.

(5 departments, 12 subdivisions, 25 salary codes, etc). Many Earth science attribute values however are members of very large sets. (UTM Northing in metres from the equator. Elemental compositions from percent to p.p.b., aeromagnetic field measurements with a dynamic range of 1 in 10^6, several thousand known mineral types etc.) Those attributes which may be members of a reasonably small set within any given survey, such as NTS map sheet or geologist's Name, tend to be largely independent variables resulting in a many-to-many relationship. This, plus the added problems of multi-dimensional data inhibits the development of complex data structures for Earth science data sets as compared to business systems.

Before it is thought that this writer considers that the problems of business systems are trivial in comparison with those of the Earth sciences, we will redress the balance with the third comparison. That of volatility.

As noted above, entries in an Earth science data bank are very rarely changed or removed. The overwhelming majority of transactions consist of additions to the file and retrieval of selected parts without changing the file content. Furthermore, although the Earth scientist requires efficient (cheap) and effective retrieval, he or she is rarely in a great hurry. Retrievals are usually made for a purpose known well in advance. Hence the Earth science data system could be created so as to permit one or two authorized persons to update the file periodically and a larger (but still relatively small) number of people to "queue" their retrieval requests and still receive satisfactory service. If the occasional error or delay occurred it would be bearable.

Compare the above requirements with those of, for example, the international airline reservations system, thousands of data elements are deleted, changed or added each minute, day and night. The change must be made virtually instantly lest the same seat be sold to several people (as often happens, one must admit).

In summary we can say that fundamental differences exist between the characteristics of Earth science and business data. Consequently, the structural precepts and systems developed to suit the characteristics of business data and its manipulation requirements may be difficult to apply to Earth science compilation.

4.5 Conclusions

It is stated that all data structures can be represented as one of the fundamental forms – Flat file, Tree or Network. We have seen, however, that some structures – real ones of demonstrable practical value – are difficult to express as any one of these forms due to the presence of implicit information, the explicit form of which must be computed.

It is also stated that, given the presence of natural relationships, the flat file can be developed into a tree or network structure. This produces an improvement in logical access to data items and a reduction in the number of data items stored. We have seen however that in the physical implementation the improvement in retrieval efficiency can be significantly less than promised and may, in some cases, be negative. i.e. a lowering of efficiency. We have also noted that the linkages, so simply drawn on the graphs of trees and network, must also have some physical expression if the data set is to honour the properties of the model. Physical expression means stored numbers, and little is to be gained in storage economy if more linkage items are created than replicated data items eradicated. On the contrary, storage requirements can increase substantially in the real case, unless a fortuitous simplicity of data structure exists.

One problem is that the structural concepts employed, and their potential advantages as real systems, were conceived of in the light of the characteristics and manipulation needs of business data. We have demonstrated that fundamental differences – pertinent to structural development – exist between business and Earth science data. Hence one might expect to encounter some difficulty in realizing these advantages with Earth science data.

This writer contends however, that the root problem lies in the great gap between the remarkable ability of the human being to comprehend complex patterns and the pedantic simplicity of the computer "brain."

Diagrams of networks, though the human sees them as a simplification of the problem, are two dimensional. Even more complex structural relationships are depicted graphically as perspective drawings of three dimensional systems. A perfect example of this is to be found in a work by Bouillé (1976), presenting a model of a scientific data bank. Among many other complex illustrations is one depicting a hierarchical structure. The illustration closely resembles a perspective drawing of the wings of a triplane with all the struts and bracing wires in place.

The computer is however, one dimensional. All multi-dimensional concepts and structures must be mapped into one dimension by the analyst who creates the algorithm to implement the model. The computer also demands everything in explicit numerical form. The conceptual simplification of the data set attained by a network or other type of diagram is deceptive.

The eventual implementation of a model will have more in common with the original unsophisticated flat file data set, than either of them have with the intermediate, multidimensional, conceptual model. Hence in certain circumstances, advantage may be gained by keeping the conceptual stage closer to the line between the simple data set and the simple mind of the computer.

It was previously seen that a simple symbolic notation could often describe a data set more accurately than a graphic diagram.

The next section will address itself to the creation of a one dimensional, symbolic notation, data structure model which both accurately represents Earth science survey data, and is appropriate to the limiting realities of the computer.

5.0 A linear model for Earth Science data structure.

We have defined the data set as a digitally encoded description of a set of entities. All of the entities possess the same set of attributes. To use the terminology of the relational data base, each attribute constitutes a "domain". Each individual entity is described by its attributes. For example, if the entities were people, we could describe each individual person by the attributes of height, weight, sex, colour of hair, age, etc.

We have defined an attribute as possessing two components - a "type" and a "value". The "attribute type" is a real-world property of the entity. "Mass" and "colour" are two examples of real-world properties of physical objects. The "attribute value" is an observed or measured value of the property. e.g. "50 kilograms" is a possible value for the attribute type "mass"; "blue" is a possible value for the attribute type "colour".

We have stated that the attribute in the information world is equivalent to the "data element" in the data world. Data elements possess type and value components corresponding to the attribute type and value components.

The group of attributes which describe an individual entity was called an "aggregate" (equivalent to "tuple" in relational terminology.) The data set is composed of many such aggregates, one for each of the entities that the data set describes.

5.1 Notation for symbolic representation of the data set.

On the basis of these definitions we will establish a notational convention for represention of the data set in a concise symbolic form. The sole purpose of the notational convention is to permit investigation of the abstract properties of the data set in isolation from real-world physical connotations.

Definition of the rules of the notational convention follows:

i) Properties of entities are represented by a string of one or more upper case alphabetic characters. Different character strings serve only to distinguish between different properties. e.g "M" could represent the property of mass, "COL" the property of colour, etc. The character strings may be mnemonic but need not be so.

Whether the properties are real or abstract, quantitative or qualitative is irrelevant. The sole consideration is that the property be measurable, observable or definable in some manner for the purpose of describing or identifying the entities concerned.

ii) A specific value of any property is represented by an integer number. Different numbers serve only to distinguish between different values of the same property. e.g. For the property of colour, "3" could represent "red" and "17" could represent "blue". The numbers do not represent a quantification of the

property, they merely identify a single specific value. What the actual value is, is irrelevant .

iii) As noted above, an attribute consists of a type/value pair. The type refers to a specific property of the entity, the value is a specific value of the property. Hence, by the two preceeding rules, an attribute is represented as an alphabetic string (type) and an integer number (value.) e.g. "A7" could represent "mass: 7 grams" and "B3" represent "colour: blue". If a value exists for an attribute but the actual value is not important, a lower case alphabetic subscript can be used in place of the number e.g. "Bi" would represent the general case of a value "i" of property "B".

iv) An aggregate of attributes describing a single entity is represented by a string of attribute symbols separated by commas. e.g. "A7,B3,C5" could represent the height, weight and age of a specific person. It could equally well represent the uranium, thorium and potassium content of a specific rock sample.

We will state, by definition, that the relationship between attributes of the same entity is equivalent to the multiplication operator. i.e. in the notation, the comma separating the attributes of an individual entity stands for "multiplied by".

v) The data set containing the descriptive aggregates of many entities is represented by a sequence of aggregates separated by plus signs. e.g. "A7,B3,C5 + A2,B3,C1 + A8,B5,C2" could represent measurements of the U, Th and K concentrations for each of three separate rock samples.

We will state, by definition, that the relationship between the aggregates of different entities of the same class is equivalent to the addition operator. Hence the plus sign in the notation has its usual arithmetical significance.

5.1.1 Summary of notation.

To summarise the notational convention defined up to this point:-

1) An attribute consisting of two components, type and value, is represented by an upper case alphabetic string for the type (a particular property) and an integer for the specific value of the property. e.g. -

A7

- could represent "mass: 10 grams" or "colour: blue". An alphabetic string followed by a lower case alphabetic subscript represents the general case. e.g., if "A" represented the property of colour, then -

Ai

- would represent "some or any particular colour".

2) An aggregate of the attributes of one entity is represented by a string of attribute symbols separated by commas. e.g -

A7,B5,C2

- could represent a description of a particular entity by the attributes "colour: blue", "mass: 10 grams", "volume: 220 Ml". The comma represents the multiplication operator.

3) A data set containing the aggregates describing many entities is represented by a sequence of symbolic aggregates separated by plus signs. e.g -

A7,B5,C6 + A2,B1,C1 + A1,B1,C9

- could represent a data set describing three entities of the same class. The plus sign represents the addition operator.

5.2 Symbolic representation of an aeromagnetic data set.

We will now employ the above defined notation to symbolically represent an actual data set.

5.2.1 Representation of the cardinal form of the data set.

The "cardinal form" of a data set has been defined as follows: All of the entities described by the data set belong to the same class (i.e. all are described by the same set of attribute types), and there is a unique one-to-one correspondence between entities and aggregates.

The symbolic representation of the cardinal form of the aeromagnetic flight-path data described in section 4 above, is as follows:-

M1,L1,X1,Y1,F1 + M1,L1,X2,Y2,F2 + ... M1,L2,Xi,Yi,Fi +

... M2,L1,Xj,Yj,Fj + ... Mk,Lk,Xk,Yk,Lk ... etc. (1)

The entities are points along flight lines marked on maps, the attributes of each entity are as follows:-

M	map number
L	line number
X	X positional coordinate,
Y	Y positional coordinate,
F	fiducial (serial) number.

The first aggregate (M1,L1,X1,Y1,F1) describes the first entity - the first point on the flight path - by the map and line upon which it lies, by its X and Y coordinates with respect to some reference frame, and by its fiducial number. The second aggregate (M1,L1,X2,Y2,F2) describes the second flight path point, etc, etc. Note that, according to the rules of the notational convention, the second point is on the same map and line as the first, but has different coordinates and fiducial number. Successive aggregates describe points on different lines, then on different maps.

The elipsis (...) in the expression indicates that a sequence of aggregates, unspecified in number, has been omitted. As defined above, the lower case alphabetic subscripts ("i", "j" and "k") as attribute values indicate "some/any specific value". i.e. these indicate the general-case of the aggregate.

All of the information available for description of the first entity resides in the first aggregate and this aggregate contains no information descriptive of any other entity. The same is true for all other aggregates, hence by definition this is a representation of the cardinal form of the data set.

5.2.2 Representation of the super-aggregate form of the data set. The above representation of the cardinal form of the data set was derived simply by observation. i.e. The original, non-symbolic, description of the data set was examined and its symbolic equivalent written out, attribute by attribute, aggregate by aggregate, according to the rules of the notational convention.

The representation of the super-aggregate form could also be derived in the same way. Another means exists, however. Once having created a symbolic representation of the data set, no further reference to actual data is necessary to derive the representation of alternative forms. These can be derived by formal processes applied to the symbolic representation.

With the comma between attributes of the same entity representing multiplication and the plus sign between aggregates representing addition, expression (1) is a conventional algebraic expression. As such, it may be subjected to the formal processes of algebraic manipulation.

As previously noted, and as can be seen in expression (1) above, A specific value of the Map Number attribute is common to many lines and data points. Furthermore, A specific value of the Line Number attribute is common to many data points. Removing these common factors of expression (1) to a higher order of placement ("factorisation") results in the following expression:-

$$M1\ (\ L1\ (\ X1,Y1,F1 + X2,Y2,F2 + \ldots) + L2\ (\ Xi,Yi,Fi\ .+$$

$$\ldots\ M2\ (\ L1\ (\ Xi,Yi,Fi + \ldots) + L2\ (\ Xi,Yi,Fi\ \ldots \qquad (2)$$

Expression (2) exactly describes a super-aggregate form of the data set as is developed from the cardinal form by the first aeromagnetic data compilation process (section 4.2.1).

5.3 The fundamentals of the model.

The representation of the data set by an algebraic expression and the derivation of alternative structural forms of the data by formal algebraic manipulation as demonstrated above, are the bases of the model.

The data set contents and structure are modeled by the content and structure of the algebraic expression. The processes of data manipulation are modelled by conventional algebraic manipulation of the expression.

The model has been defined in terms of the "information world" of attributes. By the definitions in this work, however, there is a one-to-one correspondence between components in the information world and components of the data world. Hence the model also describes the data world.

5.3.1 Symbolic representation of the model.

Expression (1) is an expanded form of the concise expression for the data set. The general-case aggregate in expression (1) was given as -

$$Mk,Lk,Xk,Yk,Fk \qquad (3)$$

As aggregates are joined by the addition operator, the entire data set in its cardinal form can therefore be defined as -

$$\sum_{k=1}^{N} Mk, Lk, Xk, Yk, Fk \qquad (4)$$

- where N is the number of aggregates within the data set.

Or with alternative notation;

$$\text{SUM}\ (\ Mk, Lk, Xk, Yk, Fk\ )\ k = 1, N \qquad (5)$$

The factorised form of the data set is defined as -

$$\sum_{i=1}^{NM} Mi \sum_{j=1}^{NLi} Lj \sum_{k=1}^{NPij} Xk, Yk, Fk \qquad (6)$$

- where NM is the total number of Maps

NLi is the total number of lines on Map i

NPij is the total number of flight-path points on Line j of Map i.

Expressions (4) and (6) above, employing the SIGMA to indicate summation, though formally correct are also cumbersome. We can, however, extend the notational convention to produce a less cumbersome form whilst retaining the features neccessary for symbolic representation of the model and investigation of its properties.

The SIGMA serves to indictate that the specific members of the general-case aggregates are joined by the addition operator. This has been established by definition and actual application of the summation operator will never take place. i.e. the addition-operator relationship between aggregates is purely algebraic and does not represent a potential computational operation. Hence we can dispense with the SIGMA.

We could retain the lower case subscript indicating the presence of attribute value, however:-

a) this is superflous to most of the requirements to come, and

b) the range of the summation is irrelevant if the summation operator will never be invoked.

Hence for general representation of the model, the subscript can also be omitted. It can, however, be employed wherever neccessary to illustrate a specific case.

Enclosure in parenthesese will be employed, as in expression (5), to indicate that the contents of the parentheses are replicated. On these bases, the concise notation for expression (4), the cardinal form, is -

$$(M,L,X,Y,F) \qquad (7)$$

Levels of parentheses will be employed in their normal algebraic sense, as in expression (2), to indictate the order of factorisation. Hence, the concise notation for expression (6), the factorised form is -

$$(M (L (X,Y,F))) \qquad (8)$$

5.4 Further properties of the model and other Earth Science data sets. The capability to derive the hierarchical (factorised) form of the aeromagnetic flight-path data set from the flat-file original (cardinal form) solely by algebraic manipulation of the model has been demonstrated in section 5.2.2 above.

Other types of data, however, are described in chapter 4.0 whose structure can not properly be described or formally developed solely by this category of manipulation. Further properties of the model must be explored to accomplish this.

5.4.1 The dot-product operator and the Drift Sedimentology data set. The cardinal form of the data set requires that each entity be described by a single aggregate. In the drift sedimentology data set, there exists a set of sub-aggregates which describe the identity and location properties of all entities, followed by a set of sub-aggregates which describe their geochemical properties, followed by a set which describe their mineralogical properties etc, etc (see section 4.2.2.) Each entity is described by five separate and dispersed sub-aggregates.

The data set therefore, is not in cardinal form. Nor is it in the factorised form of expression (8) as the separate sub-aggregates do not represent common factors removed to a higher order of placement.

To illustrate, let us assume a simplified form of the drift sedimentology data set with a single attribute in each sub-aggregate. The cardinal form would be -

(A,B,C,D,E) (9)

- with all attributes in a single aggregate, the aggregates repeated to describe each entity once only (by the notational convention, enclosure in parentheses denotes repetition.) In contrast, the actual form appears to be -

(A) (B) (C) (D) (E) (10)

e.g. the first attribute repeated for each and every entity followed by the second attribute for each and every entity, etc. There is, however, a serial, one-to-one correspondence between the repeated sub-aggregates in (A) and (B) and (C) etc. Corresponding members are attributes of the same entity. The relationship between such attributes has been defined as that of the multiplication operator. The multiplicative combination of corresponding members of two related groups is achieved by taking the dot product of the groups. Hence, this data structure can be defined correctly and completely by employing the dot product operator. The correct model expression for this data set, therefore, is -

(A).(B).(C).(D).(E) (11)

Expanding this expression according to the rules of dot product multiplication and expressing the result in model notation, produces -

(A,B,C,D,E) (12)

- which is the cardinal form.

As was demonstrated for the factorised form of the aeromagnetic data set, the drift sedimentology data structure can also be derived from the cardinal form and vice-versa, solely by formal algebraic manipulation, yet the derived expressions maintain an exact description of the actual, real data structures described.

5.4.2 The cross-product operator and the aeromagnetic levelling data set.

The cardinal form of the aeromagnetic levelling data set (see section 4.2.1, expression (11) and section 4.3.) can be summarised as -

$$(C,T,I) \tag{13}$$

- where C is a control line number, T is a traverse number and I describes the intersection of the traverse and control line. All possible traverse-control line pairs are covered by the data set. Hence, each individual specific value of C will be repeated for every value of T. This means that each value of C is a common factor in many aggregates, hence the data set can be restructured in a factorised form as -

$$(C (T,I)) \tag{14}$$

i.e. each unique value of C occurs once only in the data set, followed by its intersections with all traverses. Then the next unique control line number followed by all its intersections, etc.

The same logic applies to traverses also, however, and an equally valid factorisation would be -

$$(T (C,I)) \tag{15}$$

The form that the actual data set takes, however, is apparently -

(C) (T) (I) (16)

i.e. All unique values of C stated once only, followed by all unique values of T, followed by a value of I for all possible combinations of a values of C with a value of T (all possible intersections.) This is not a cardinal form, nor is it a factorisation in the form of expressions (14) and (15), nor is it the same structure as that of expression (10) as there is no necessary one-to-one correspondance between (C) and (T) or between (T) and (I).

The numbers of aggregates in (T) and (C) (the numbers of traverses and control lines respectively) are independent. The number of aggregates in (I), however, is not an independent variable. It is the product of the number of traverses and the number of control lines (e.g. 10 traverses and 4 control lines makes 40 possible intersections.)

In this case, the entity is the "intersection". The traverse and control line numbers of a given intersection are attributes of the same entity and hence are related by the multiplication operator. The cardinal form contains all possible traverse / control line pairs. The multiplicative combination of all possible pairs of members of two independent groups is achieved by taking the star product (vector outer product) of the groups. Hence the construct -

(C) * (T) (17)

- when the star product is taken, would produce a sub-aggregate set -

(C,T) (18)

- containing all possible pairs of the individual members of (C) and (T)

This sub-aggregate set is now in serial, one-to-one correspondence with the members of (I) in (16) and is therefore related to it by the dot product operator. Hence a model expression can be written which correctly and completely describes the aeromagnetic intersection data set structure. i.e:-

$$(C) * (T) . (I) \qquad (19)$$

Expanding this expression according to the rules of star and dot product multiplication and expressing the result in model notation, produces -

$$(C,T,I) \qquad (20)$$

- which is the cardinal form. Hence, the aeromagnetic levelling data set structure can be derived from the cardinal form solely by formal algebraic manipulation, and vice-versa.

5.4.3 The null operator and the aeromagnetic archival data set.

The aeromagnetic archival data set structure (see section 4.2.1) can be summarised by the model expression -

$$(M (L (F) (G))) \qquad (21)$$

It is a factorised form similar to expression (8), grouped hierarchically first by Map (M), then by flight line (L), then with the lowest level sub-aggregate groups (F) and (G). The first of these, (F), describes the set of navigational fixes along the flight line. The second group (G) describes the in-flight aeromagnetic measurements at points in space along the flight line. The coordinates of the measurement points were derived from the navigational fixes but the measurement points differ in number and location from the navigation points.

Hence, there is no one-to-one correspondence between (F) and (G). Furthermore, taking all possible pairs of navigational and measurement points is meaningless. Therefore, neither the dot nor the star product operator can be used to express formally the relationship between these two groups. What, therefore, is the relationship ?

To determine this, we observe that M and L are higher order factors of both (F) and (G), and therefore we can expand expression (21) by multiplying out the higher order factors to produce -

$$(M,L,F)\ (M,L,G) \qquad (22)$$

This result is exactly what we should expect. It consists of two separate, independent, cardinal forms. Refering to the description of the aeromagnetic compilation process (section 1.1.4), we see that the process does indeed begin with two independently acquired data sets - the track recovery data and the in-flight data. These two data sets describe two different entity sets - navigational fixes on a map, and in-flight measurements. Hence, two separate cardinal forms are unavoidable.

Although we have two separate cardinal forms, there are common factors (M and L) between the two forms. The legitimacy of grouping common factors from different cardinal forms to produce expression (21), is demonstrated by the fact that the independent components retain their independence in (21) and that the original independent cardinal forms are recovered on expansion.

The legitimacy can be formalised by stating that two sub-aggregate groups at the same level, which can not be related by any other operator are related by the "null" operator. The operations which change the levels of sub-aggregate groups (factorisation and expansion) can be applied across the null operator but it will not, itself, be affected by these operations. The null operator is indicated simply by the absence of any other operator between two adjacent lowest level aggregate groups.

Hence, the sought for relationship between the lowest level sub-aggregate groups of expression (21) is that of the null operator - an indication that the sub-aggregates were originally derived from separate, independent cardinal forms.

5.5 Data manipulation operations.

Changing the structure of a data set from its basic cardinal form to one of the more advanced structures described above is a data manipulation operation. i.e. attributes are moved - manipulated - relative to each other. Sub-aggregate contents change but the overall attribute content of the data set remains unchanged. We can extend the symbolic notation for the model to permit the description of such operations by employing an arrow to denote development of one structure from another.

e.g. the expression -

$$(A,B,C) \dashrightarrow (A (B (C))) \qquad (23)$$

describes the development of the factorised form. Other operations are the development of dot product groups:-

$$(A,B,C) \dashrightarrow (A).(B).(C) \qquad (24)$$

and the development of star product groups by compound factorisation:-

(A,B) —> (A)*(B) (25)

The development of the aeromagnetic leveling data set is a combination of the operations shown in expressions (24) and (25). The combination of two distinct cardinal forms under a common factor as was the case with the aeromagnetic archival data set, is expressed as:-

(A,B) (A,C) —> (A (B) (C)) (26)

Note the preservation of the null operator.

All of these operations can, of course be carried out in reverse to reconstruct the cardinal form from a developed form.

5.6 The physical structure of data.

The model, as established to this point, is a "logical" model of the data set. i.e. the logical relationships between attributes and aggregates are expressed by the non-alphabetic symbols in the model expression. The logical content of the data set is expressed by the alphabetic symbols - the attribute descriptors.

Figure 17 (section 4.1.2) illustrates the logical structures of other types of model. Tree and network diagrams can be drawn giving the identity of the data items in the nodes. e.g. a tree diagram could be drawn for the aeromagnetic data set in expression (8) with M as the "parent" branching down to many L as first generation "children"; then each L in turn branching down to many X,Y,F children. Figure 18 (section 4.3) actually depicts the aeromagnetic levelling data set of expression (19) as a network and identifies the attribute content of the nodes.

The tree and network diagrams, whilst depicting logical structure, provide no information whatsoever about the physical storage of the data and hence how individual data items might be accessed. This is presumably all that one should expect given that we are dealing with a "logical" model.

The algebraic model, however, can exceed this expectation to a significant degree and provide valuable information on the physical storage of the data.

5.6.1 Sequential access structure.

The scientific data sets described possess two crucial characteristics:-
A) they are large and hence demand efficient access methods; but -
B) sequential access satifies a majority of the processing needs.
Accordingly, the optimum storage medium is digital magnetic tape. Magnetic tape is a linear storage medium. Regardless of the logical relationships between data items, the physical manifestation of data item values is as groups of bytes stored in linear sequence along the strip of magnetic tape. There is no way to access the Nth item without first passing physically over the preceeding N - 1 items in sequence. Hence the term "sequential access".

The algebraic model, as is clearly demonstrated by the non-summarised form shown in expression (1), is also strictly linear - a linear sequence of characters representing attributes and their relationships to each other.

The relationship between attributes of the same entity is multiplicative. Hence it is commutative. i.e. in terms of content and relationships;

$$(A,B) = (B,A) \qquad (27)$$

Therefore, we can make the order of attributes within an aggregate the same as the physical order of the corresponding data items in a data record without violating any of the rules of the model. Hence the model can serve as a physical description of the data as well as a logical model. (An alternative employment of the property of commutivity is given in section 5.7 below)

□

The legitimacy of this concept can be demonstrated by consideration of expression (5) in which the summation of the data set, implied by the existence of the addition operator between the aggregates, is explicitly stated.

If we replace the arithmetical operation "SUM" in expression (5) with the computer operation "READ", we create a formally correct instruction for retrieving the data set. i.e. the instruction:-

$$\text{READ (} Mk,Lk,Xk,Yk,Fk \text{) } k = 1,N \qquad (28)$$

- is the equivalent of the FORTRAN statement:-

```
READ ( LUN ) (M(K), L(K), X(K), Y(K), F(K), K = 1, N)        (29)
```

Which would serve to input the data set from external sequential storage to internal random access storage in arrays in the computer memory. Replacement of "READ" by "WRITE" produces the corrollary statement required to output the data arrays from memory to external sequential storage.

The same applies to all of the other structural forms defined as model expressions. Every occurrence of a right hand parenthesis implies physical access via an iteration of the form "I = 1,N", where N is the number of members in the group delimited by the right parenthesis, and I is the identity index to each individual member.

Hence the model does provide information about the physical structure of the data. Sufficient, in fact, to generate formal instructions for physical data input and ouput.

It is stated in section 5.3.1 that "... actual application of the summation operation will never take place.", for which reason, the formal summation symbols were deleted. It is interesting to note, however, that the summation symbols were not entirely redundant. They do, in fact, represent an operation which will, in fact, take place. Actual physical data input/output is represented by the summation of a model expression.

5.6.2 Direct access structure.

It is stated in section 5.6.1 above, that "sequential access satisfies a majority of the processing needs [of the scientific data sets described]". The remainder of the processing needs could also be met by sequential access but only at the cost of a prohibitively large amount of redundant data input/output. To meet these remaining needs efficiently, direct access to data is required.

The principal direct access data storage medium is the magnetic disk. On a disk, data is still stored in a linear sequence, hence the linear model is still applicable. On the disk, however, chosen points in the data can be accessed directly without sequential passage over the preceeding data items. After accessing the chosen point, data retrieval proceeds sequentially until the requisite amount of data has been recovered.

Various means exist to implement direct data access, but all have essentially the same structure - an index exists to the points which may be accessed directly. The index can be structured hierarchically if required.

A data set to which direct access may be made is therefore in a significantly different state to one which must be accessed sequentially and means must be found to indicate this in the model expressions describing data sets.

As noted above, once a direct access jump is made to a point in the data set, retrieval is sequential from that point onwards. Processing requirements dictate, therefore, that the direct access points coincide with natural structural boundaries. i.e. if direct access was made to the middle of a group of sub-aggregates, only the latter half of the group would be retrievable by the subsequent sequential access.

The grouping of sub-aggregates in the model expressions represent structural boundaries. The parenthesese indicate repetition of the group they enclose. e.g. in the data set -

$$(M (L (X,Y,Z))) \qquad (30)$$

- the outer parenthesese indicate repetition of the M (map) group and its two subordinate groups. The next lower level of parentheses indicate that the L (line) group and its single subordinate group is repeated for each individual map. The lowest level parenthesese indicate that the X,Y,Z group is repeated for each individual line.

If this data set was stored on a direct access device, we could create an index to the start of each individidual map group and then be able to retrieve all the data for any one map directly without sequential passage over the preceeding map groups. We can extend the symbolic notation for the model to indicate the existence of such an index. If direct access is possible to individual members of a group, a bar will be placed over the first attribute within the group. e.g. for the case described above, the expression would be;

$$(\overline{M} (L (X,Y,Z))) \qquad (30)$$

There is no bar over the L group in expression (31), hence individual members of this group for any given map can only be accessed sequentialy after direct access to their parent map group. However, if the expression was;

$$(\overline{M} (\overline{L} (X,Y,Z))) \qquad (32)$$

Then after direct access to any individual map group, a second level of direct access could be mades to any individual line group that the map contained.

The process of converting a data set from sequential access to direct access is described an expression of the form;

$$(A (B)) \longrightarrow (\overline{A} (B)) \qquad (33)$$

5.7 Considerations of the physical content of data.

The logical content of the aggregates and sub-aggregates is defined by the names of the attributes they contain. Physical content refers to the actual values assigned to individual attributes.

5.7.1 Physical order.

In actual data sets, groups of related data items (i.e. individual aggregates) can be, and often are, ordered according to the value(s) of one or more "key" data items in the group. For example, aeromagnetic in-flight data records are in ascending order of the fiducial number data item.

If our concern is to describe processes which are sensitive to, or which alter the physical order of, aggregates according to the values of key attributes, we can exploit the property of commutivity an a different way.

We can define that the individual members of a parenthesised group are arranged in order according to the physical content (values of the sub-aggregates in the group.) The order in which the sub-aggregate descriptors are written, defines the precedence, e.g.:-

(P,Q,R) (34)

Indicates that the individual aggregates are physically ordered according to the values of P. If two or more consecutive members have the same P values, then these will be ordered by Q, etc. The expression -

(P,Q,R) --> (Q,P,R) (35)

- describes the process of sorting from order by P to order by Q.

This rule applies to factorised groups in the same sense. e.g. for:-

(M (L (P,Q,R))) (36)

- individual members of the major group are ordered by M. Within each of these, the members of the next lower sub-group are ordered by L, and within each of these, the residue group is ordered by P, Q and R in precedence as written.

The word "within" is important. Once factorised, an ordered sequence cannot exceed the bounds of a sub-group.

i.e. in the above example, the ordered sequence of M continues throughout the whole data set - but the first level sub-group could not

be ordered by L continuously throughout the whole data set exept by a fortuitous combination of aggregate values. The sequence for L can only be continuous within an individual M group. In general, the sequence will be broken between M groups. To illustrate, consider a cardinal form with specific values as follows:-

M1,L2 + M1,L3 + M2,L1 + M2,L4 + M2,L6... (37)

When this data set is factorised, the result is -

M1(L2 + L3) + M2(L1 + L4 + L6 ... (38)

Note that the L values are in order only within a group. The sequence is broken between groups.

A quite common process is the deletion of data items that are superfluous to the needs of the next and subsequent processes. If the M Sub-aggregate were deleted, the resulting data set could be expressed as

(L) (39)

This expression is misleading, however, as it implies that the data is primarily ordered by L, which would not be so immediately after the deletion of M. If we delete the M factor from expression (38), the result is -

L2 + L3 + L1 + L4 + L6 ... (40)

Local "pockets" of order exist, but the data set must be sorted before it can be described by expression (39) which implies order by L value.

This condition can be indicated by an addition to the notational convention. If deletion of a sub-aggregate would result in a disordered data set, then the deleted sub aggregate can be replaced by a minus sign to indicate the fact. i.e. After deletion of M but before re-ordering by L, the correct expression is -

$$(- (L)) \qquad (41)$$

In general, the collating sequence for ordering the sub-aggregates will be ascending order of value. If an alternative sequence is used, a numeric superscript can be employed to indicate this. e.g:-

$$(P,Q) \dashrightarrow {}^{1}(P,Q) \qquad (42)$$

- could serve to indicate a process of sorting from ascending order of P value to some other (e.g. descending) order. The zero superscript can be reserved to describe an un-ordered data set.

5.7.2 Entity sub-classes.

Cases can be found, within the given examples of compilation systems, of the development of a logical sub-class of the basic entity. Members of the different sub-classes and the aggregates which describe them are physically identical but possess

certain logical characteristics which distinguish one sub-class from another. For example. In the Geochemical compilation system the entity is (say) a water sample and the aggregate which describes each sample is of the form:–

$$SN, LOC, AR \qquad (43)$$

– where:– SN is a Sample Number, LOC a locational sub-aggregate, and AR a sub-aggregate containing the results of the various analyses.

This is the case during the initial stages of the compilation system, but at one point the entity suddenly divides into several sub-classes. i.e. The simple water sample suddenly becomes either a routine sample, a field-duplicate, a lab duplicate a control sample or a blank, and it is necessary to correctly identify the sub-class to which each aggregate belongs.

A similar case exists in the aeromagnetic compilation system. During the first part of the compilation (verification, standardisation, coordinate transform) all lines are treated equally.

At the start of the second part of the compilation (sorting, combining and leveling adjustment calculation) two sub-classes of line develop – traverses and control lines. During the third and final part (application of adjustment, gridding etc.) control lines disappear and we are once more left with just "lines."

One could cover these cases by setting up different entity classes at the outset and maintaining them throughout. This however is unacceptable for three reasons.

i) It would make the symbolic expression too long.

ii) It is artificial. The system does see only "sample" or "lines" before recognition of sub-classes. And all aggregates are treated equally.

iii) Members of the sub-classes before recognition are intermixed at random. The fact that they are all equal in the eyes of the processes applied, allows description of an ordered data set. If the sub-classes were to be identified explicitly by the notation. Their presence as a random admixture would produce a disordered data set no longer describable as one of the cardinal or factorised logical forms.

Hence recognition of separate entity sub-classes should not be made until the data set undergoes the first process which required this recognition. By these means the symbolic description honours the reality of the processes and the data sets remain describable as logical structures. A simple notational convention will be employed to indicate such cases. ie. if class A contains sub-classes B, C, D etc. this will be stated by a simple equation:-

$$A = B + C + D \text{ etc.} \qquad (44)$$

This equation will be placed beneath the symbol representing the process which first recognizes the existence of subclasses and separates them. e.g. In the case of the aeromagnetic data set, when the class of lines is separated into the two sub-classes of control lines and traverses the notation will be as follows:-

$$(M(L(X, Y))) \xrightarrow[L = T + C]{} (M(T(X, Y)))\ (M(C(X, Y))) \qquad (45)$$

5.8 Non-conformable data structures

Although the notation so far created is capable of accurately describing all the data sets developed within the various compilation systems described, we cannot assume that it will describe every data set that the Earth scientist could contrive. A counter example can in fact be found from the aeromagnetic compilation system. It is the data set as created by the airborne data aquisition system and presented to the start of the compilation system. It contains an unfortunate interaction between logical and physical structure. The cardinal form aggregate would be:–

$$(F, L, A) \qquad (46)$$

where:– F is a fiducial number, L is a line number and A is a sub-aggregate containing measured magnetic field value, altitude, Doppler etc. The real physical structure however is neither cardinal or factorised. The data is in physical blocks of a constant length. Each block begins with a header record stating the line number and other information. The rest of the block is filled out with required F, A aggregates. It takes several blocks to contain all the data from a given line. Hence the same line number will occur at the start of several consecutive blocks.

When a block is encountered which contains a line number different from that of the previous block this indicates that we are on a new line. The data is certainly not in cardinal form as the line attribute is not present in every aggregate. Neither is the data completely factorised as more than one occurrence of each unique line number is present.

The outstanding feature of this data set however, the one which merits the name we shall give to all such structures, is that a change in line number indicates that the line changed somewhere in the previous block. Just where, is impossible to

define. The situation is saved from disaster, however, by the fact that all lines begin with a run-in of superfluous data. Hence taking the start of the new-line block as the actual start of the line, works.

We will call this and all such structures "illogical". i.e. structures which do not conform to the rules of the logical model and which therefore cannot be described by model terms. We will represent this structure by a list of its sub-aggregate descriptors enclosed in triple parentheses to indicate that some, albeit illogical, grouping exists, e.g.:-

$$(((F, L, A))) \qquad (47)$$

Every such occurrence requires special ad-hoc software to decypher and untangle. Such structures are often unavoidable, and therefore forgiveable, as products of an automated data aquisition system with its built in restrictions. They are avoidable, and therefore unforgiveable, as computer output.

5.9 Summary of Notation

To summarise these additions to the notation:-

i) The order in which sub-aggregate descriptors are written indicates the precedence of these sub-aggregates in the ordering of the members of the groups or sub-groups, e.g.:- the members of the set:- (P, Q, R) - are ordered by P, Q, and R in that precedence.

If it is necessary to indicate that the order is not common ascending order but some special collating sequence. A numeric superscript may be appended to indicate the fact and to provide reference to the description of the sequence. e.g.:-

$$({}^{2}P,Q) \rightarrow ({}^{3}P,Q) \qquad (48)$$

denotes sorting from special order No. 2 to special order No. 3.

ii) Deletion of a sub-aggregate, if such would leave a disordered group, is indicated by replacement of the descriptor by a minus sign until such time as the group is re-ordered by the remaining sub-aggregates.

e.g.:– deletion of M from:–

$$(M(L(P, Q))) \qquad (49)$$

– results in:–

$$(-(L(P, Q))) \qquad (50)$$

– until sorted by L to produce:–

$$(L(P, Q)) \qquad (51)$$

iii) Random access may be made directly to individual members of a group via an index, this is indicated by the presence of a bar over the sub-aggregate descriptor(s) to which the index refers. e.g.:–

$$(\bar{M}(L(P, Q)) \qquad (52)$$

Indicates that individual members of the highest level group may be accessed directly on the basis of the value of the sub-aggregate M. After which retrieval of L and P, Q members must progress sequentially.

iv) Recognition and separation of sub-classes of aggregate is indicated by a sub-class composition equation beneath the symbol for the process which effects this. e.g.:–

$$(L(X, Y)) \xrightarrow[L = T + C]{} (T(X, Y))\ (C(X, Y)) \qquad (53)$$

v) Any data set so structured as to be incapable of description by the model notation will be termed an "illogical" structure and indicated as such by enclosure of its sub-aggregate descriptors in triple parenthesis e.g.:–

(((L,F,A))) (54)

The process which converts this data to logical structure will be an ad-hoc software module.

5.10 Software/data independance

As noted in the quotation from Martin in chapter 3, data elements are meaningless unless data element type can somehow be associated with data element value.

In the model, the sub-aggregate descriptor is an abbreviated representation of one or more data element type-value pairs e.g.- "P" could stand for "A1, B5, C3".

Hence the association is a physical one within the data set. Such is rarely the case in practice. In most program applications the data element name has no physical existence in digital form. What do exist are the names of variables within the program to which data element values are assigned by input.

The common method is to specify, within the program, an input lists of variable names whose order exactly matches that of the data elements on their recording medium and whose content changes to match the gross structure of the data set. This is the nature of the FORTRAN "READ" statement. The method is simple, efficient, and adequate in most cases. One disadvantage is that if the structure of a particular data set does not match that preset within the program then either the data or the program must be changed to suit. In most cases this is an easy task and an infrequent requirement and hence provides no serious hinderance to the working of a system.

If however, a system is intended to be capable of processing many different and widely varying data types which can not be forced into the same structural form,

then this disadvantage becomes crucial and it is necessary to eliminate it. Jeffery and Gill (1976) state that "Independence of data from the processing software is essential for any synthesis of scientific data because it is impossible to define all possible combinations of data format and structure at the time of writing the software."

Data could be allowed to assume a variety of forms and be independent of software if the data possessed a structure exactly the same as that of the model (i.e. element name-value pairs).

Implementation of such a form in real terms would be by preceding each digital data element with the digitally encoded version of its name. Hence a set of analysis results for example would be recorded digitally as:-

Cr, 103, Cu, 22, Ni, 36, Pb, 125.

This would work in practice but would double the size of the data set as compared with a recording such as:-

103, 22, 36, 125.

- which is the commonly used form. In such a form the connection between attribute type and value is maintained externally to the digital data set, ie. the user has the connection documented. Provided that the documentation is not lost, and that the person preparing the data does not change the order of the values, then the user can write into a program:-

READ Cr, Cu, Ni, Pb

which will cause the attribute values to become correctly re-attached to their type names.

Though economical of storage space, this method in practice has certain disadvantages. External maintenance of the data significance, correct statement of it within a program, and correct modification of the program whenever the data order or content changes, all require human effort and are all subject to human error.

The preferable compromise would be to have the attribute type separated from its value for the sake of economy but the connection somehow retained within the data for the sake of efficiency.

This can be done within the model by further exploitation of its algebraic properties.

5.10.1 Factorisation by attribute type.

The way that presents itself is to ascribe a multiplicative relationship between the type and value of an attribute. A relationship identical to that already ascribed between different attributes within the same aggregate, ie. "A1, B6, C5" may be considered as "A, 1, B, 6, C, 5" thus attribute types may be separated from their values and removed as common factors themselves. e.g.:-

A1, B6 + A2, B1, + A3, B2 (55)

- can be factorised as:-

A, B (1, 6 + 2, 1 + 3,2) (56)

Thus, the attribute type factors have been extracted and sent to the head of the data set, where they occur only once rather than being repeated in every aggregate.

Although the problem would appear to have been solved, interpretation of the solution in terms of data elements and real implementation has certain unsatisfactory aspects.. ie:-

i) For maximum condensation of the data set, both whole attributes (type and value) and attribute types alone would have to be extracted as common factors thus the data set:-

A1, B5, C2 + A1, B1, C7 + A1, B3, C6 (57)

- could be factorised as:-

A1, B, C (5, 2 + 1, 7 + 3, 6) (58)

- The factor group extracted contains a mixture of whole and part attributes. Thus some means would have to be incorporated to distinguish between these two categories of information.

ii) The original form of aggregate representation in which attribute type and value were inseparable, permitted algebraic manipulation to any degree while maintaining the integrity of the data. The new form allows manipulations which could destroy the integrity of the data. In algebraic terms, variables are commutative. eg:-

A1, B5 = B5, A2 (59)

Likewise, in terms of integrity of information:- "colour = blue, weight = 3.0" is equivalent to "weight = 3.0, colour = blue". Thus in the original form the property of commutivity could be applied freely to the model without unacceptable consequences in terms of the information being modeled. The same property applied to the new form would, in algebraic terms, allow that:-

A, 2, B, 5 = A, B, 5, 2 = 2, B, 5, A etc (60)

which is tantamount to saying that "colour = 3.5, weight = blue", or worse still "weight = colour, 3.5 = blue". i.e. the integrity of the information is lost. Hence if advantage is to be gained by factorisation of attribute types in isolation, then formal

means must be taken to protect the integrity of the data. Attribute type and value must be regarded as a special kind of "ordered pair" not subject to all ordinary algebraic rules.

Fortunately the practical realities of Earth Science data and compilation processes themselves enforce a structure which promotes factorisation of attribute types..

Let us, consider an aeromagnetic data set which contains the along track total field measurements. The cardinal aggregate form of the data set could be:-

map no., line no., X coord, Y coord, gamma value.

It is a natural property of the data set, that only a few values of "map no." will exist and each will be common to a great number of aggregates; that many values of "line no." will exist but each will still be common to many aggregates; and finally, that very many values of "X-coord", Y-coord" and "gamma" will exist with very few values (if any) common to more than one aggregate.

It is a property of the processes that are to be applied to the data, that factorisation by the first two attributes will order the data in the way it is most likely to be required.

Hence factorising the aeromagnetic data file by whole-attributes would produce a result of the form:-

M1 (L1 (X1, Y1, G1 + X2, Y2, G2...) + L2 (Xn, Yn, Gn + ...) + L3...) + M2(L1 (etc). (61)

Further factorisation after this by types in isolation eventually produces the results:-

M, L, X, Y, G (1 (1 (1 (1, 1, 1 + 2, 2, 2...) + 2(1, 1, 1...)...)+2) (62)

i.e. the cardinal form "M, L, X, Y, G" has "floated" to the head of the entire data set thus being separated from, and hence no longer to be confused with, attribute values.

The order of attribute values must however be rigidly conserved within the sub-aggregate if nonsensical results are to be avoided when the data set is expanded to a less factorised form. This however is the natural state of affairs within a data set. If no action to the contrary is taken the order of elements will not change.

A further convention can also be adopted to permit not only a description of the cardinal form content, but also description of the factorisation state of the data set. This can be accomplished by giving to the attribute type group the same structural levels as possessed by the data e.g., the attribute type group: "M, L, X, Y, G", can be constructed as M(L (X, Y, G)) with the group of attribute types being nested in the same way that the actual data groups of element values are nested.

Such a group, containing attribute type definitions only, will be given the name of a "data definition group" as opposed to data groups per-se.

Thus the desired compromise has been achieved. The data element names have been separated from their values thus producing an almost 50% reduction in storage volume. The association between the names and values however remains in digitally encoded form within the data set. If the structure and/or content of the data set change, the structure and content of the factorised group of element type names will also change accordingly. Thus the data set becomes internally "self defined" and does not rely upon external, non-digital, definitions.

To implement this structure would require software that did not "expect" any particular structure (i.e. did not have the required data structure programmed into its input statements). The program would merely expect the presence of data elements

pertinent to its process. At program run time the first information read from the data file would be the factored-out data element names. Their juxtaposition would inform the program of the physical locations of the corresponding data element values. Hence the system would be "data independent". It should be noted that the term "data independent" realy means data structure independent, as no program which is not a null process can be entirely independent of data content. e.g. a program to calculate the free air correction for gravity data must be supplied with gravity values and altitudes. It could not "make do" with drift sedimentology data.

5.10.2 Unfactorisable or inhomogeneous data sets

The hierarchical group structure represented by the model provides a means of reducing the bulk of a data set by factorisation of both common attribute values and common attribute types.

Factorisation by attribute value presupposes that at least one attribute within a data set possesses only a small number of unique values, (smaller than the number of aggregates). Then by sorting and factorisation, this attribute can be removed from many individual aggregates and recorded once only at the head of the group to which it is common. Factorisation by attribute type presupposes that the given attribute type structure is common to many aggregates. Real cases exist however for which one or both of these presupposes are invalid.

If the first supposition (that attributes exist which are common to several aggregates), is invalid, then the data set simply cannot be condensed by attribute value factorisation. It may however, still be possible to abstract the attribute type definition. For example, consider the following symbolic data set:-

A4, B3, C2 + A2, B1, C5 + A1, B2, C1 (63)

This set cannot be factorised by removing common attributes to a higher order of placement as no common attributes exist. The same attribute type structure, however, is common to all three aggregates and this can be removed and placed at the head of the set as a data definition group. e.g.:-

(A,B,C,) (4, 3, 2, + 2, 1, 5 + 1, 2, 1) (64)

Hence a data set which is un-factorisable by attribute value may still, be amenable to factorisation by attribute type. All data sets which are not composed of identical aggregates will in fact contain an unfactorisable residue at the lowest level. Hence invalidity of the first supposition still permits appreciable condensation by attribute type factorisation.

An example of such a case is found in geochemical compilation. One data subset has the form:-

(SN, RES) (65)

-where SN is sample number and RES the set of analysis result for several elements. SN has all unique values by definition and any analysis results which were common to more than one aggregate would be purely fortuitous. e.g. a copper concentration of 50 p.p.m. in two separate samples is coincidence. Unlike common values of "Map number" or "collector" which are systematic: Hence even if common values exist, factorisation would be inappropriate.

Such a data set could however be advantageously factorised by attribute type e.g.:

(Sn, Cu, Ni, Pb, Zn)(1, 1, 1, 1, 1, + 2, 2, 2, etc... (66)

which obviates the appendage of a type name to each and every value.

Invalidity of the second supposition is shown by the following symbolic data set.

A1, B5 + C3 + D1, E5, F4 (67)

This represents the highest attainable degree of inhomogeneity of a data set. Its aggregates are not only all different in their logical content (attribute type structure) but also in the number of attributes in each aggregate.

Such a case could arise if the entity set were inhomogeneous. i.e. if each entity belonged to a different class and therefore possessed a different set of properties to be observed and recorded.

Earth science data sets, however, refer always to a homogeneous entity set. e.g. All geochemical samples or all gravimetric measurements. A data set resembling the above symbolic form can still arise though as a development of a "sparse" data set. One in which a large proportion of the data elements possess null values (i.e. their value is not available for recording.) Such can be the case with drift sedimentological data. Of the 300 potential attribute values, perhaps 50% or more may not be available at a particular time. Furthermore, the absent group will not be the same for each sample, e.g. some samples may have been analysed for copper and not uranium and vice versa. The data set could still be structured with a constant attribute content for each aggregate by employing some convention to denote that the recorded value was a null, e.g. a negative number for an analysis result could serve to indicate that the result was unobtainable. This would then permit, at least, factorisation by attribute type with the resulting condensation of the data set. The redundant storage of null values would still, however, remain.

If the proportion of null values was sufficiently high one could optimise storage efficiency by taking the opposite approach. Namely sacrifice the storage reduction afforded by attribute type factorisation so as to avoid storage of null values. By these means, an inhomogeneous data set of the form shown symbolically would arise. Each real value would be recorded in direct association with a digitally encoded type name. Attributes for which no value exists would simply be left out. Thus the data integrity is maintained. In this data set the self-definition is not "globally implicit" (i.e. stated once at the start and applied implicity to all aggregates) but "locally explicit". (Each aggregate contains its own definition).

Hence exploitation of this property of the model would allow even the most intractable data set to be structured in a self defined, software-independent form.

5.11 Formal definition of the model syntax

Figure 20 defines the model syntax in Backus-Normal form.

Further constraints are as follows:

i) The "bars" indicating a direct access data set must begin over the leftmost and extend to the right over as many subaggregates as are indexed for direct access.

ii) The subaggregate composition (number and type) of the data set description to the left of a module must be identical to the composition of the data set to the right unless

<PROCESS>::=<INPUT><MODULE><OUTPUT>

<INPUT>::=<DATA SET>|<DATA SET><INPUT>

<OUTPUT>::=<INPUT>

<MODULE>::=⟶|⟶<SUB-CLASS EQUATION>|{<NAME>}

<NAME>::=<ALPH>|<ALPH><NAME>

<SUB-CLASS EQUATION>::=<ATTRIBUTE>=<SUB-CLASSES>
|<SUB-CLASSES>=<ATTRIBUTE>
|<SUB-CLASS EQUATION><SUB-CLASS EQUATION>

<SUB-CLASSES>::=<ATTRIBUTE>+<ATTRIBUTE>|<ATTRIBUTE>+<SUBCLASSES>

<DATA SET>::=<RESIDUAL GROUP>|(<AGGREGATE><RESIDUAL GROUP>)

<RESIDUAL GROUP>::=(<AGGREGATE>)|<RESIDUAL GROUP><OPERATOR>
<RESIDUAL GROUP>

<OPERATOR>::= |.|*

<AGGREGATE>::=<ATTRIBUTE>|<ATTRIBUTE>,<AGGREGATE>
|<SUPERSCRIPT><AGGREGATE>|<BAR><AGGREGATE>

<ATTRIBUTE>::=<ALPH>|<ALPH><ATTRIBUTE>|-

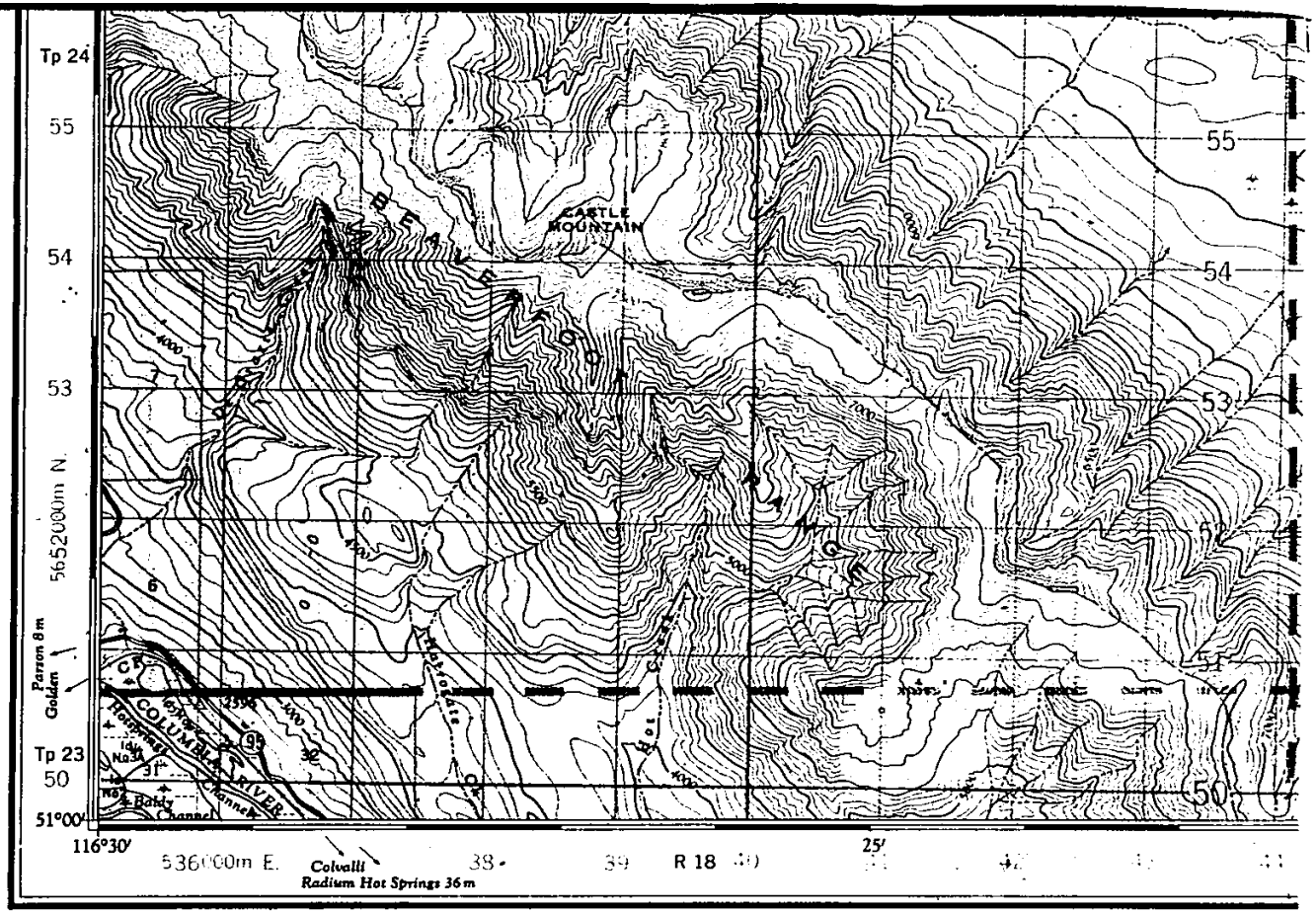
<ALPH>::=A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z

<SUPERSCRIPT>::=<INT>|<INT><SUPERSCRIPT>

<INT>::=$^{0}|^{1}|^{2}|^{3}|^{4}|^{5}|^{6}|^{7}|^{8}|^{9}$

<BAR>::=¯

Fig. 20 Formal definition of the syntax of the model notation.



Produced and printed by the SURVEYS AND MAPPING BRANCH, DEPARTMENT OF MINES AND TECHNICAL SURVEYS, 1961, from air photographs taken in 1951.

Copies may be obtained from the Map Distribution Office, Department of Mines and Technical Surveys, Ottawa.

MOUNT G

KOOTENAY D

BRITISH COl

WEST OF FIFTH MERIDIAN - OUES

SCALE 1:50,000

Miles 1 0 1

Metres 1000 500 0 1000

Yards 1000 500 0 1000 200

Roads	Routes
hard surface, all weather	pavée, toute saison
hard surface, all weather	pavée, toute saison
loose surface, all weather	de gravier, toute saison
loose surface, dry weather	de gravier, période sèche
cart track	de terre
trail or portage	sentier ou portage
Railway, normal gauge, single track	Chemin de fer, voie unique (écartement normal)
Horizontal control point, with elevation	Point géodésique avec cote ..... 454 △
Bench mark, with elevation	Repère de nivellement avec cote ..... BM 157

CONTOUR INTERVAL 100 FEET
Elevations in Feet above Mean Sea Level
North American Datum 1927
Transverse Mercator Projection

ÉQUI
Élévations er
Réseau

MAGNETIC DECLINATION 22°31' EAST
AT CENTRE OF MAP 1961
Annual change (decreasing) 4'

DÉC
DI

A) the composition of the data set to the right is a sub-set of that to the left (deletion),

or B) the difference in composition is exactly resolved by the sub-class equation(s),

or C) the module is a development process of the form {name}

iii) The subaggregate composition of the data set(s) to the right of a development process must contain at least one subaggregate description not present in the subaggregate composition of the data set(s) to the left of the module.

6.0 The compilation model

We now possess a means to accurately describe the structure and logical content of the data set regardless of the level of structural development.

Changes of form of the data set are brought about by submission of the data to one or other of the four fundamental types of compilation processes described in chapter 2. Manipulation processes are accurately modeled by algebraic manipulation of the data model. We can therefore represent a manipulation process simply by an arrow connecting two different data forms. e.g.:–

$$(P, Q) \longrightarrow (P(Q)) \tag{1}$$

The particular manipulation process which took place is evident from the difference between the two data set descriptors.

Hence, the various categories of manipulation appear as follows

i) Sort.

$$(P, Q) \xrightarrow{o} (Q, P) \tag{2}$$

or

$$({}^{2}P, Q) \longrightarrow ({}^{3}P, Q) \tag{3}$$

ii) Merge physically separated subaggregate sets.

$$(P) \cdot (Q) \longrightarrow (P,Q) \tag{4}$$

iii) Separate into subaggregate sets.

$$(P, Q) \longrightarrow (P) \cdot (Q) \tag{5}$$

iv) Create a matrix

$$(P) * (Q) \longrightarrow (P, Q) \tag{6}$$

v) Delete sub-aggregates

$$(P, Q) \rightarrow (P) \tag{7}$$

vi) Change from sequential to random access by index.

$$(P(Q)) \rightarrow (\bar{P}, (Q)) \tag{8}$$

vii) Recognition and extraction of sub-classes of aggregate

$$(P, Q) \xrightarrow{Q = R + S} (P, R)\ (P, S) \tag{9}$$

Data manipulation processes do not change the overall sub-aggregate contents of the data set(s) involved. Only the data set structure changes. In a verification or display process neither the sub-aggregate content nor the structure changes. In the former case the data emerges from the verification process with the same structure and logical content that it had on entry. Only the physical content, which is not represented in the model, may have changed. i.e. Erroneous values of certain data elements may have been corrected or deleted but the order and nominal content of the data is unchanged.

In the case of display processes, a data set as we have described it, does not emerge from the process at all. Hence the data set can not be said to have changed. What emerges is a man-readable form of the data.

We will represent verification and display processes by a V and a D respectively in the following manner:–

$$(P(Q)) \rightarrow V \tag{10}$$

$$(P(Q)) \rightarrow D \tag{11}$$

The remaining type of process, development, differs from the previous three types in that the logical content of the data set changes. New types of sub-aggregate are created. In many cases, whole new data sets with their own particular structure

are created. As the nature of development processes varies considerably throughout a given system and the content of the new data set depends upon this nature, we will represent this type of process by a process name in-brackets. e.g.:–

$$(X, Y, Z) \{\text{Interpolate}\} (P, Q, R) \qquad (12)$$

Unlike verification and display and most types of manipulation, two or more separate data sets may be required as input to a development process and two or more output data sets may result. It has already been stated that the model is not concerned with the physical location of the data, hence, two or more data sets would be represented simply as a concatenation. The rules of the model notation ensure that the concatenation is correctly interpreted. e.g.:–

$$(P, Q)\,(X,Y)\ \{\text{Process}\}\ (R, S)\,(A, B) \qquad (13)$$

The outer set of parentheses always encloses the entire data set. A right, followed immediately by a left parenthesis indicates two groups at the same level. The absence of an operator between the groups indicates their independence. Hence the formulation shown, exactly describes the case of two independent data sets being input to the same process, and two independent data sets being output.

People may argue that it also appears to imply that they are input or output one after the other, which may not be the case. If not so, and if it matters that the case be distinguished, the data set descriptors can be stacked to indicate parallel input or output. e.g.:–

$$(P,Q)\ \{\text{Process}\}\ (R, S) \qquad (14)$$

$$(X, Y) \qquad (A, B) \qquad (15)$$

Note 1

Two data sets between which exists a dot product relationship have a one to-one correspondence between their individual members. Hence if the same sub-aggregate descriptor appears in both data sets this denotes a duplicate sub-aggregate set. e.g. In the case of geochemical field data and station coordinate data, the sub-aggregates describing the field information in the former will differ from the sub-aggregates describing the locational information in the latter. But both will possess the same sub-aggregate set of sample numbers. Hence the notation for the field data would be:-

$$(S, F) \tag{16}$$

-where S is the sample number sub-aggregate and F the field data sub-aggregate. For locational data, the notation would be:-

$$(S, L) \tag{17}$$

- where S is the same sample number set as before and L is the locational data.

When the two sets are merged it is evident that there is no need to retain the sample number data element from both. Hence the process would be described as:-

$$(S, F)\cdot(S, L)\rightarrow(S, F, L) \tag{18}$$

If two sets are to be merged in the above manner or if two sets are to be input in parallel to the same development process; one which requires corresponding sub-aggregates from each set; then it is evident that both sets must be in the same order or at least one set be in random access form. Hence the following operations are immediately recognizable as invalid:-

$$(P, Q)\cdot(S, P)\rightarrow(P, Q, S) \tag{19}$$

$$({}^{1}P,Q)\cdot({}^{2}P,S)\rightarrow(P,Q,S) \tag{20}$$

valid operations are:–

$$({}^{1}P,Q)\cdot({}^{1}P,S)\rightarrow({}^{1}P,Q,S) \tag{21}$$

$$(P,Q)\cdot(\overline{P},S)\rightarrow(P,Q,S) \tag{22}$$

Note 2

Two groups between which a star product relationship exists need not be in one-to-one correspondence. Hence the order within each group is irrelevant to the execution of the star product operation. The order in which the two groups are written however, denotes the order in which the operation will be carried out and hence the order within the resultant group. e.g.:–

$$(P)*(Q)\rightarrow(P,Q) \tag{23}$$

– the operation that took place was that the first member of P was grouped with every member of Q in turn, then the second member of P with every member of Q etc. Consequently a formulation such as:–

$$(P)*(Q)\cdot(R) \tag{24}$$

Implies that R is ordered as it would be in the case:–

$$(P(Q(R)) \tag{25}$$

Note 3

Single, upper case, alphabetic characters are not mandatory as sub-aggregate descriptors. Combinations of letters may be used as a mnemonic to assist in the interpretation of complex expressions. The comma or parentheses delimit the individual sub-aggregate descriptors.

Note 4

If a particular data set is not in common order but in some special order but the order does not change throughout a long series of processes, then the special order superscript can be omitted and the special order described by a foot-note if necessary.

We will now employ the model notation to describe actual compilation systems.

6.1 Examples of autocompilation systems described by the model notation

6.1.1 Aeromagnetic Auto Compilation

i) In-flight data

$$(((L, FI, GR, AUX)) \rightarrow (^{\circ}L\ (FI, GR)) \rightarrow V \rightarrow D \tag{26}$$

ii) Flight path data

$$(M, L, XD, FF) \rightarrow (M(L(XD, FF))) \rightarrow V \rightarrow D \tag{27}$$

$$\rightarrow \{\text{Coord. Trans.}\}(M(L(XU,FF))) \rightarrow (L(XU, FF))) \rightarrow \tag{28}$$

$$\rightarrow \{\text{Join line segment}\}^{?}(L(XU, FF)) \tag{29}$$

iii) Combine in-flight and flight path data.

$$(^{\circ}L(FI, GR)) \rightarrow (\bar{L}(FI, GR)) \tag{30}$$

$$(\bar{L}(FI, GR))\ \ (L(XU, FF) \rightarrow (L(XU, FF)\ (FI, GR)) \rightarrow \tag{31}$$

$$\{\text{Interpolate}\}(L(XU, FF)(XP, GR)) \tag{32}$$

iv) Calculate intersections and extract intersection data set.

$$(L(XU, FF)\ (XP, GR)) \xrightarrow[L = T + C]{} (T(XU, FF)\ (XP, GR))\ (C(XU, FF)\ (XP, GR)) \tag{33}$$

$$\{\text{Calculate/extract}\}(C, T, XI, GI) \rightarrow (C)^{*}(T)^{\cdot}(XI, GI) \tag{34}$$

v) Calculate level adjustment along control lines and apply to traverses.

$$(C)*(T)\cdot(XI, GI)\{\text{CL adj}\}(C)*(T)\cdot(XI, GI, AC) \quad (35)$$

$$(C)*(T)\cdot(XI, GI, AC)\rightarrow(T)*(C)\cdot(XI, GI, AC) \quad (36)$$

$$(T)*(C)\cdot(XI, GI, AC)\ (T(XU, FF)(XP, GR))\{\text{Interpolate/Apply}\}\rightarrow \quad (37)$$

$$\longrightarrow (T(XU, FF)(XP, GR, AG)) \quad (38)$$

vi) Interpolate regular grid and contour

$$(T(XU, FF)(XP, GR, AG))\{\text{pick grid intersections}\}(XG, IG, GC) \quad (39)$$

$$\text{Interpolate } (GC) \quad (40)$$

Sub-aggregate descriptors

L	-Flight line number
FI	-In-flight fiducial
GR	-Aeromagnetic measurements
AUX	-Auxiliary (altitude, Doppler etc.)
M	-Map Number
XD	-Track point coordinates w.r.t. digitiser frame.
FF	-Track point fiducial
XU	-Track point coordinates w.r.t. UTM frame.
XP	-In-flight data point coordinates. (UTM)
T	- Traverse number
C	-Control line number
XI	-Coordinates of C-T intersection
GI	-Aeromagnetic values on C and T at intersection
AC	-Leveling adjustment to control line
AG	-Leveling adjustment to each data point.

XG -X coordinate of traverse/grid line intersection

IG -Grid line index number.

GC -Aeromagnetic value at traverse/grid line intersection

GG -Interpolated aeromagnetic value at nodes of grid.

Special ordering sequences

L - When not otherwise noted, in E-W order across survey.

T - as for L

C - N-S order across survey.

GG - Conventional matrix order. L-R along rows. Rows bottom to top.

6.1.2 Geochemical Compilation

i) Field and positional data.

$$(S, F) \rightarrow V \tag{41}$$

$$(S, L) \rightarrow V \rightarrow \{\text{Convert to UTM}\} (S, U) \tag{42}$$

$$(S, F) \cdot (S, U) \rightarrow (S, F, U) \tag{43}$$

ii) Wet chemistry data

$$(S(EW, RW)) \rightarrow V \tag{44}$$

iii) Automated AA analysis data

$$(EA(S,RA)) \rightarrow (EA,S,RA) \rightarrow (S,EA,RA) \rightarrow (S(EA,RA)) \tag{45}$$

$$\rightarrow \{\text{Stats}\} \rightarrow D \tag{46}$$

iv) Merge to extract duplicates and control

$$(S,F) \cdot (S(EA,RA) \cdot (S(EW,RW)) \xrightarrow[\substack{EA + EW = E \\ RA + RW = R}]{} (S,F(E, R)) \tag{47}$$

$$(S, F(E, R)) \xrightarrow[S = O + D + C]{} (D(E,R)) \cdot (C(E, R)) \tag{48}$$

v) Final merge etc.

$$(S, F)\cdot(S, L)\cdot(S(E,R))\rightarrow(S, F, L(E, R))\rightarrow(S, F, L, E, R)\rightarrow(E, S, F, L, R)$$
$$\rightarrow(E(S, F, L, R))\rightarrow D \qquad (49)$$

Sub-aggregate-descriptors

S - Sample No.

F - Field data

L - Digitised station coordinates

U - Station Coordinates in U.T.M.

EA - Element analysed for by atomic absorption

RA - Result of this analysis

EW - Element analysed by wet chemistry

RW - Result of this analysis

O - Ordinary sample

D - Duplicate sample

C - Control sample

E - Element

6.1.3 Drift sedimentology autocompilation

i) System as described.

$$(A)\rightarrow V \qquad (50)$$

$$(G)\rightarrow V \qquad (51)$$

$$(M)\rightarrow V \qquad (52)$$

$$(T)\rightarrow V \qquad (53)$$

$$(S)\rightarrow V \qquad (54)$$

$$(A)\ (G)\ (M)\ (T)\ (S)\rightarrow(A)\cdot(G)\cdot(M)\cdot(T)\cdot(S) \qquad (55)$$

ii) Possible extension of system. Produce cardinal form and factorise as simple tree according to preferred retrieval criteria.

$$(A)\cdot(G)\cdot(M)\cdot(T)\cdot(S)\rightarrow(A,G,M,T,S)\xrightarrow[A+G+M+T+S=D]{}(D) \quad (56)$$

$$\xrightarrow[D=RA+RB+RC]{}(RA,RB,RC)\rightarrow(RA(RB(RC))) \quad (57)$$

Sub-aggregate descriptors

A - Auxiliary

G - Geochemical

M - Mineralogical

T - Tech/phys

S - Storage

D - All attributes

RA,RB,RC-etc-selected retrieval criteria.

6.1.4 Airborne gamma spectometry autocompilation

i) In-flight data

$$(((L,F,A,C)))\rightarrow(L(F(A,C))) \quad (58)$$

ii) Extract background measurements. Compute and apply background and other corrections.

$$(L(F(A,C)))\xrightarrow[C=CR+CB]{}(L(H(A,CR)))\ (L(G(CB))) \quad (59)$$

$$F=H+G$$

$$(L(H(A,CR)))\ (L(G(CB)))\{\text{Interpolate Background}\}(L(H(BC))) \quad (60)$$

$$(L(H(A,CR)))\ (L(H(BC)))\{\text{subtract background}\}(L(H(A,CC))) \quad (61)$$

$$(L(H(A,CC)))\{\text{Compute and apply other corrections}\}(L(H(A,CF))) \quad (62)$$

$$(L(H(A,CF)))\{\text{Convert to equivalences}\}(L(H(A,EE,TC))) \quad (63)$$

iii) Merge with flight path and separate elements, plot profiles and contour

$$(L(H(A,EE,TC)))\cdot(L(H,X))\rightarrow(L(H,X(A,EE,TC))) \quad (64)$$

$$\{\text{Interpolate data point coordinates}\}(L(Y,A,EE,TC)) \quad (65)$$

$$(L(Y,A,EE,TC))\xrightarrow[EE=EU+EK+ET]{}(L(Y,A,EU,EK,ET,TC)) \quad (66)$$

$$\rightarrow(L(Y,A,EU))\cdot(L(Y,A,EK))\cdot(L(Y,A,ET))\cdot(L(Y,A,TC))\rightarrow D \quad (67)$$

Sub-aggregate descriptors

L - Line number

F - Fiducial

A - Auxiliary

C - Count rates for four windows

CR - Count rates for routine measurements

CB - Background count-rates

H - Fiducial for routine measurements

G - Background measurement fiducials

BC - Background corrections interpolated at routine measurement points.

CC - Background corrected count rates.

CF - Final corrected count rates.

EE - Elemental equivalences

X - Coordinates of fiducial point

Y - Coordinates of data point

TC - Total count

EU - Equivalent uranium

ET - Equivalent thorium

EK - equivalent potassium

6.1.5 Land gravimetry auto-compilation

i) Field data; calculate and apply drift correction, convert to absolute gravity units.

$$(T,A,R,BS,TR) \rightarrow (BS(TR(T, A, R))) \quad (68)$$

$$(BS(TR(T, A, R))) \xrightarrow[T = TS + TB]{} (BS(TR(TS, AS, RS))) (BS(TR(TB, AB, RB))), \quad (69)$$

$$A = AS + AB$$

$$R = RS + RB$$

$$\{\text{Interpolate}\}(BS(TR(TS, AS, RS)\cdot(DC))) \rightarrow (BS(TR(TS, AS, RS, DC))) \quad (70)$$

$$\{\text{Apply}\}(BS(TR(TS, AS, RC))) \quad (71)$$

$$(BS(TR(TS, AS, RC))) \ (BV)\{\text{adjust}\}(TS, AS, GA) \quad (72)$$

ii) Calculate and apply corrections.

$$(ET(TS,AS,GA)\{\text{Earth tide}\}(TS,AS,GE) \quad (73)$$

$$(TS,AS,GE)\{\text{Free Air}\}(TS,AS,GF) \quad (74)$$

$$(D)(TS,AS,GE)\{\text{Bouguer}\}(TS,AS,GB) \quad (75)$$

iii) Merge with positional coordinates.

$$(TS,AS,GB)\cdot(X,Y) \rightarrow (TS,AS,GB,X,Y) \rightarrow (X,Y,TS,AS,GB) \quad (76)$$

iv) Terrain Correction

$$(C,P,Q)*(X,Y,TS,AS,GB)\{\text{Calculate}\}(X,Y,TS,AS,GG) \quad (77)$$

Sub-aggregate descriptors

BS - Base station number

TR - Traverse number

T - Time of reading

A - Altitude of station

R - Gravimeter reading

TS,TB
AS,AB } times, altitudes and readings at survey stations
RS,RB

and base station respectively

DC - Drift correction

RC - Drift corrected reading

BV - absolute gravity value at base station

GA - Absolute gravity value at survey station

ET - Earth tide corrections

GE - Gravity value corrected for earth tide

GF - Free air anomaly

GB - Bouguer anomaly

D - Density of material beneath each station

X,Y - Station positional coordinates

C - Gravity contribution from vertical prism

P,Q - Positional coordinates of vertical prism

GG - final gravity value on geoid.

6.2 Common processes

The generality of different processes and types of process as described verbally in section 2 can now be seen in the formal notation of the model.

All verification routines are essentially the same and as such the single symbol V honours this generality.

Display routines will vary in their generality. i.e. Highly specialised displays such as fence diagrams are very limited in the breadth of data types to which they can apply. We have already noted, however, that we are concerned with a basic system and within such, many generalised display routines could exist, e.g. General profile plotting or a contour package.

The generality of manipulation routines is evident from the model. e.g.:–

$$(P,Q) \rightarrow (P(Q)) \tag{78}$$

is a factorisation process totally independent of the significance of P and Q.

Development routines also exhibit generality when their components which are strictly manipulatory are isolated. e.g.:–

$$(P, X) \xrightarrow[P = Q+R]{} (Q,Y)\ (R,W)\{\text{interpolate}\}(Q, Y)\cdot(S) \rightarrow (Q,Y,S)\{\text{apply}\}(T,Y) \tag{79}$$

$$X = Y+Z$$

describes all of the following specific compilational processes.

i) Extract the base station subset in gravimetric compilation. Interpolate the drift value for all readings between base stations, and subtract it from the station reading.

ii) Extract the control reading subset from a sequence of automated AA analyses and interpolate and subtract the drift from all routine analyses between.

iii) Extract the background measurement subset from an airborne gamma spectrometry data set and interpolate and apply the background correction for intermediate values.

iv) Aeromagnetic diurnal removal (not shown in the compilation system example) is essentially the same with the exception of the first stage. The diurnal data set is already separate.

All of the above examples employ a linear interpolation function. Other functions would not hamper generalisation if the interpolation module was kept separate from the other processes.

Aeromagnetic leveling, ostensibly highly specialised, contains only one process not found in many other places in the various systems. Namely "Calculate C-L adjustment" the remaining processes are common manipulation processes and "interpolate and apply" identical to the above example.

We can, therefore, in abstract but formal terms, describe a general linear system whose components could be re-arranged to suit most of the specific compilational needs of five very different Earth Science disciplines leaving only a few specialised sub-processes to be accounted for by ad-hoc software. By dint of such generalisation it could be expected that the system would be applicable to compilational requirements beyond the five examples given.

The next, and paramount, question to be answered is "how can such a system be implemented in real terms". i.e. working software.

Our search for an answer is simplified by the choice that was made to create a linear model. One that had more in common with the nature of the data and the linear mind of the machine.

The next part of this work will devote itself to finding that answer. i.e. to determination and specification of the means to create a software implementation of the model.

7.0 Realisation of the model data structure

The model exists as a symbolic expression of algebraic form. Its properties as a model have been demonstrated. It lies, however, in Martin's "information world". The next task is to create its analog in the data world. That is, to design the means to express the symbolic model in a digital form which retains its valuable logical properties.

The practical objectives of the structural design are to achieve in reality the potential improvement in efficiency promised by the theory. These are two-fold; data storage economy and retrieval efficiency. It has been shown that the model as so far defined can afford greater storage efficiency that a flat file form. The attainable storage efficiency will depend upon the extent to which the data set can be condensed by factorisation. This will in turn depend upon the logical and physical content of the individual data set. Hence little more need be said about this aspect of efficiency until specific, real data sets are considered. We will now examine the other aspect of efficiency.

7.1 Levels of selectivity in data retrieval

Retrieval efficiency is the dominant aspect of overall efficiency of a data model design. Retrievals may be categorized in three different ways, i.e.

i) Bulk retrieval

All data within the data set is to be input to the process. (Though not all data so input will necessarily be actively involved in the process computations). A verification routine would very likely employ this type of retrieval.

ii) Type selective retrieval

Certain types of data within the data set are required by the process. Other types within the same data set are to be bypassed by the input routine. Such retrievals characteristically occur with data sets containing groups between which the null relationship exists, e.g.

(M(L(FP)(IF))) (1)

An aeromagnetic data set is factorised by M (Map), then by L (line) within the map group. At the lowest level however, there exist two independent residues, FP (flight path data) and IF (In-flight measurements).

To plot the base map would require the M, L and FP groups to be retrieved, but not the IF groups. When the input routine encountered an IF group, it would bypass it in search of the next relevant group.

iii) Content selective retrieval

This category is the most complex one in terms of the mechanisms required. It could also involve pre-selection by type (category ii above).

In this category of retrievals, data groups whose physical contents satisfy certain criteria are passed on to the process. Those groups which do not satisfy the criteria are bypassed.

The drift sedimentology retrieval problem previously discussed is a case of content selective retrieval

In a real system, the routines to carry out each of these three categories of retrieval would be present as a chain of increasing selectivity. Categories ii and iii both require initial input from a category i routine. Category iii may require input via both categories i and ii or via category i alone.

Category ii requires control input specifying the group type(s) required. Category iii requires control input specifying the content criteria for retrieval. In a truly general system, category i would also be to some extent selective as it would require control input specifying the data set required. Figure 21 shows this selectivity hierarchy.

7.2 Data identity

The lowest level of data identity is that of the individual data element. This will be taken care of by the data definition system. The next level concerns the identity of the individual data group, i.e. the means whereby one can distinguish between two individual groups of the same type. This level of identification is required for content selective retrieval and is determined by the actual values of data elements within the group, e.g. the data set:–

(M (L (FP)(IF))) (2)

– contains many M (map) groups. To retrieve a particular map would require that actual values within the M sub-aggregate, which define individual maps, be examined.

Thus the individual is identified. Individuals, however, belong to a "family" i.e. the group type. All FP groups for example, belong to the same family. For type selective retrieval the type must be identified.

Within a data set, there exist data definition groups as well as data groups per se. Other kinds of group different from both of these could be conceived of, such as documentary text groups or groups which contain an index to the contents of the data set. This introduces a still further higher level of identity - a "species" of group in fact. It is evident that the species must also be identified so that the system can take appropriate action in each case.

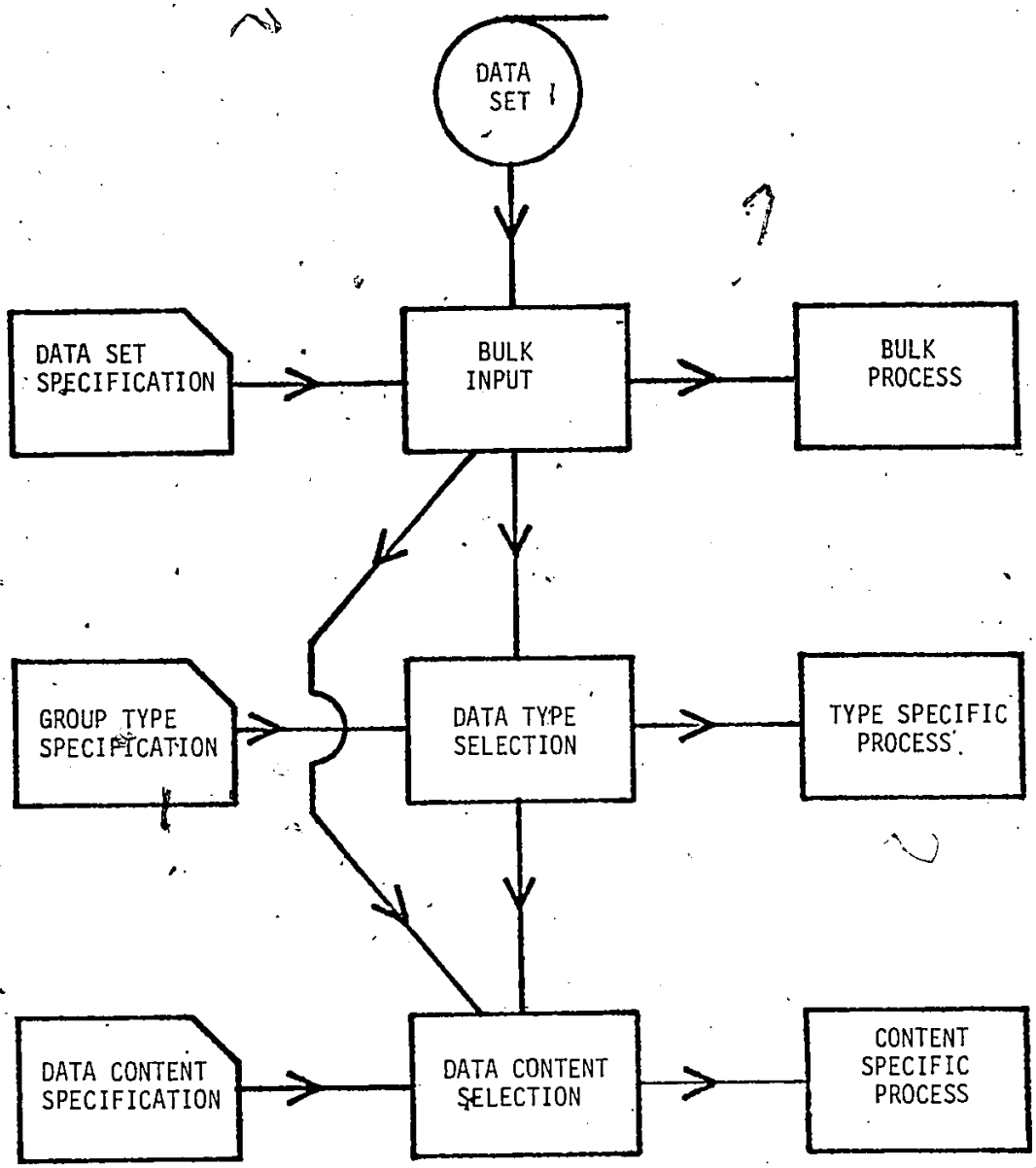


Fig. 21 Levels of selectivity in data input.

7.3 Location of data

In a real system, not only must data be identifiable, it is also necessary to delimit the various groups in some physical way so that they can be found when sought. It is also necessary that the hierarchy of groups and their interrelationship (dot, star or null) be indicated. One means to achieve this would be to store the element values concatenated into sub-aggregates as shown in the model, and delimit these sub-aggregates with special "control" elements representing the parentheses and the plus signs. So recorded, the string of digital data elements would be an exact analog of the model notation.

For bulk input, each successive group would be retrieved as those data elements enclosed by a pair of control elements. Crossing a left parenthesis would signify a descent of one level in the hierarchy. Such a system would work for bulk retrieval and similar methods are commonly employed to delimit data "records" or blocks, i.e. a special value or character combination in one or more elements is used to indicate "end of record" or "end of data" etc.

The interrelationship could be indicated by a special character between the groups, or better still stated once and for all in the data definition group. The physical location of every group boundary could not be predefined unless every group of each type was of the same length-an unacceptable restriction.

For type or content selective retrieval, the system would compare the group type or content with the specified selection criteria. When an irrelevant group was encountered, scanning would continue, with no transference of data, until the next relevant group was found. At which point transference of data would recommence.

Hence the data model could be expressed in digital form in a manner closely analogous to the symbolic notation. A form which would permit all three categories of retrieval. Unfortunately, the differences between the mechanisms of the computer and those of the human being make the closeness of the analogy disadvantageous.

For the human, to scan a string of characters in search of a particular combination is far easier than attempting to scan along a specified number of characters. For the computer the opposite is true. It takes less computational effort for the central processor to "jump ahead" a large number of "characters" than it does to "scan ahead" a single character and determine if the character represents data or a control element.

The improvement in retrieval efficiency afforded by the model structure is due to the fact that the data is "parcelled" into large groups. If some element (a group name or a high level factor) characteristic of the group as a whole, fails a retrieval criterion, then the whole "parcel" can be rejected.

The parcel could be as long as an entire map group in an aeromagnetic data set. A group which could contain tens of thousands of aggregates. To have to examine every character of every element of every aggregate of an unwanted group in search of the next control element would be highly inefficient.

A further problem arises due to fact that not all data sets are composed of binary encoded characters. When dealing with numerical data it is far more efficient to store and transfer data in pure binary form. In a pure binary number any configuration of bits is permissible and likely. Hence no configuration can be reserved for exclusive use in representing control elements. Fatal ambiguity would

therefore arise. The system would have no way to determine whether a bit configuration represented a group boundary parenthesis or whether it was merely a fortuitous combination of bits within an actual data element. The above two considerations, low efficiency and potential ambiguity, rule out data location by means exactly analogous to the symbolic model.

7.4 Necessary structural elements

The preceding sections have established that structural elements must be employed within the data per-se to identify the "family" and "species" of group and to delimit group boundaries. The definition of the relationship between adjacent groups at the same level, and the identification of individual data elements, can be taken care of in data definition groups preceding the data proper.

7.4.1 User and system group type names

The "species" of group refers to whether a group contains actual data or is a data definition group or other special type of group. Identification of "species" and corresponding appropriate action would be a matter for the main system software to take care of. The user need and should not have to be concerned with such matters. An element will be needed to contain this parameter and will be referred to as the "system type" of the group.

The "family" of group refers to the general type of data that a data group contains, e.g. aeromagnetic flight path data or geochemical analysis results etc.. This parameter is also a matter for system concern as it will provide the means for relating data definition groups to its associated data group. The parameter is,

however, very much a user concern as its content will be created by the user to label the data and be employed by the user as criteria for retrieval. It will therefore be referred to as the "user-type" of the group.

Both user-type and system-type names would have to be physically placed at the start of the group and would hence represent the group's left parenthesis of the symbolic notation.

7.4.2 Group external pointer

It has been agreed that to examine every element in a large group solely in search of a terminating control element, would be highly inefficient. The stated practical alternative is to jump directly to the group termination. This can be achieved by addition of a "pointer" to the control elements.

This pointer would indicate the location, in terms of number of elements ahead, of the group's terminating right parenthesis. No actual physical expression of this parenthesis need be present.

Skipping the requisite number of elements would move out of the current group to the start of the next group. This pointer will therefore be referred to as the "external" pointer.

7.4.3 Group internal pointer

If a group type, or its highest level factor(s) does not meet the retrieval criteria, the external pointer allows immediate transfer to the next group in sequence. If the group was not factorised this pointer would permit all necessary navigation through the data set. We must, however, allow for factorised groups. If a

factorised group was entered at its highest level and, at this level, satisfied the retrieval criteria, the need to descend to the next lower level would arise. e.g. in the given example of the data set:

(M (L (FP) (IF))) (3)

– the external pointer permits successive M (map) groups to be skipped until a required M content is found. Once found, however, the next step would be to descend to the lower L (line) level. Thus a second pointer is needed in addition to the "external" pointer. The external pointer defines the location of the imaginary right parenthesis closing a group and hence defines the start of the next group at the same level. The internal pointer defines the location of the next embedded left parenthesis, i.e. the start of the next lower level sub-group.

7.5 The control aggregate(group label)

The necessary control elements described above when placed in physical juxtaposition within the data set form the control aggregate. This in effect, stands for the group's opening left parenthesis and will be referred to as the group lable.

The group labels act as "stepping stones" for navigation through the data set.

Just as a left parenthesis begins the symbolic expression of the data set:

(M (L (FP)(IF))) (4)

– a group label would being the real data set.

The system, on encountering the first label, would firstly peruse the system type name of the group, then branch to the appropriate process. e.g. a data definition system type would elicit a response to establish the data aggregate content for the forthcoming data set.

If the system type identified the group as a data group, the system would examine the user type name. In the case of type-specific or content-specific retrieval, an unwanted group would have to be bypassed. This would be accomplished by forward spacing the data set according to the dictate of the external pointer.

In the case of a wanted group, the highest order factor would be input and attention would be transferred, via the internal pointer, to the next lower order factor.

7.5.1 Defining the hierarchy

Observing the symbolic expression of the data set, it can be seen that left parentheses always occur singly, separated from each other either by sub-aggregate descriptors or by descriptors and right parentheses. Each left parenthesis is also explicity represented by the group label. Hence when navigation consists of descending to successively lower order factors (i.e. movement through successive left parentheses) the hierarchy remains well defined. That is to say, each time movement is made following the internal pointer, the system knows that one level has been descended in the hierarchy. By keeping count of the levels descended, the current level is always known. When the lowest level is reached, this can be indicated simply by the convention of a zero-value internal pointer, i.e. "no lower level".

The case of navigation through right parentheses is not as simple. Firstly, it can be observed in the symbolic notation that several right parentheses can occur contiguously. Secondly, it has been stated that the right parenthesis has no explicit

representation in the data set. Where the right parenthesis would be, is simply the boundary between two elements of the data set. The left hand side of the boundary is the last element of a data aggregate, the right hand side is the first element of the next control aggregate, i.e. the next group label.

When one crosses this boundary in the symbolic notation, the number of levels that one rises in the hierarchy is indicated by the number of right parentheses traversed. In the data set these parentheses are absent so, therefore, are the means to maintain knowledge of the current hierarchical level. This is an untenable situation. It could be resolved by the addition of a control element which stated the number of levels risen at this point. A simpler method exists however.

So as to permit the navigator to "escape" from within a group when required, the system will remember the values of the external pointers of the current hierarchy. e.g. for content selective retrieval, the user may be finished with the IF data before the end of this group is reached. In order to skip to the next line (L), the external pointer of the current superior L group is required. The nature of the transaction could also be that the user is finished with the entire map (M) whilst still in the IF group. The next map can be found via the external pointer of the current M superior group.

The ambiguity resulting from the absence of explicit right parentheses essentially consists of not knowing if the end of the current group is also the end of the higher order group(s).

If the external pointer of a group is set to zero whenever the end of that group is also the end of the next higher group, the problem is resolved, i.e. when wishing to escape from a group its external pointer must be employed. If when recalled, the

value of this pointer is zero, the system then knows to test the value of the pointer of the next highest level group. This continues up the hierarchy until a non-zero pointer is reached which is then employed to escape to the end of the group. The number of zero pointers encountered is the number of levels risen in the hierarchy on traversing the end-of-group boundary. The zero value pointer will be referred to as an implicit pointer.

Although the model is basically a tree structure, usage of this implicit pointer permits a complex network of navigation paths through the data set. Fig. 22 depicts this network.

7.5.2 Retrieval paths through the hierarchy.

Navigation through the data set shown in figure 22 could follow any path that it is possible to trace along successive arrows in the figure. This is a great number of possible paths. More complex data sets will provide even more alternative routes. Many of the possible routes will, however, be unlikely to be followed in practice. Certain preferred types of route will predominate, i.e.:

i) Bulk retrieval

In the case where all groups within the data set are to be retrieved, the retrieval path is simply defined. The PI pointers are followed until a zero value PI is encountered. Then the PX pointers are followed until a zero value of this pointer is also encountered. The route then rises in the hierarchy via the implicit pointers and repeats the process from where the first non-zero PX pointer is encountered.

ii) Type or content specific retrieval

Type of content specific retrieval would progress by the previous route if the type or content was specified for residue groups only. If, however, the

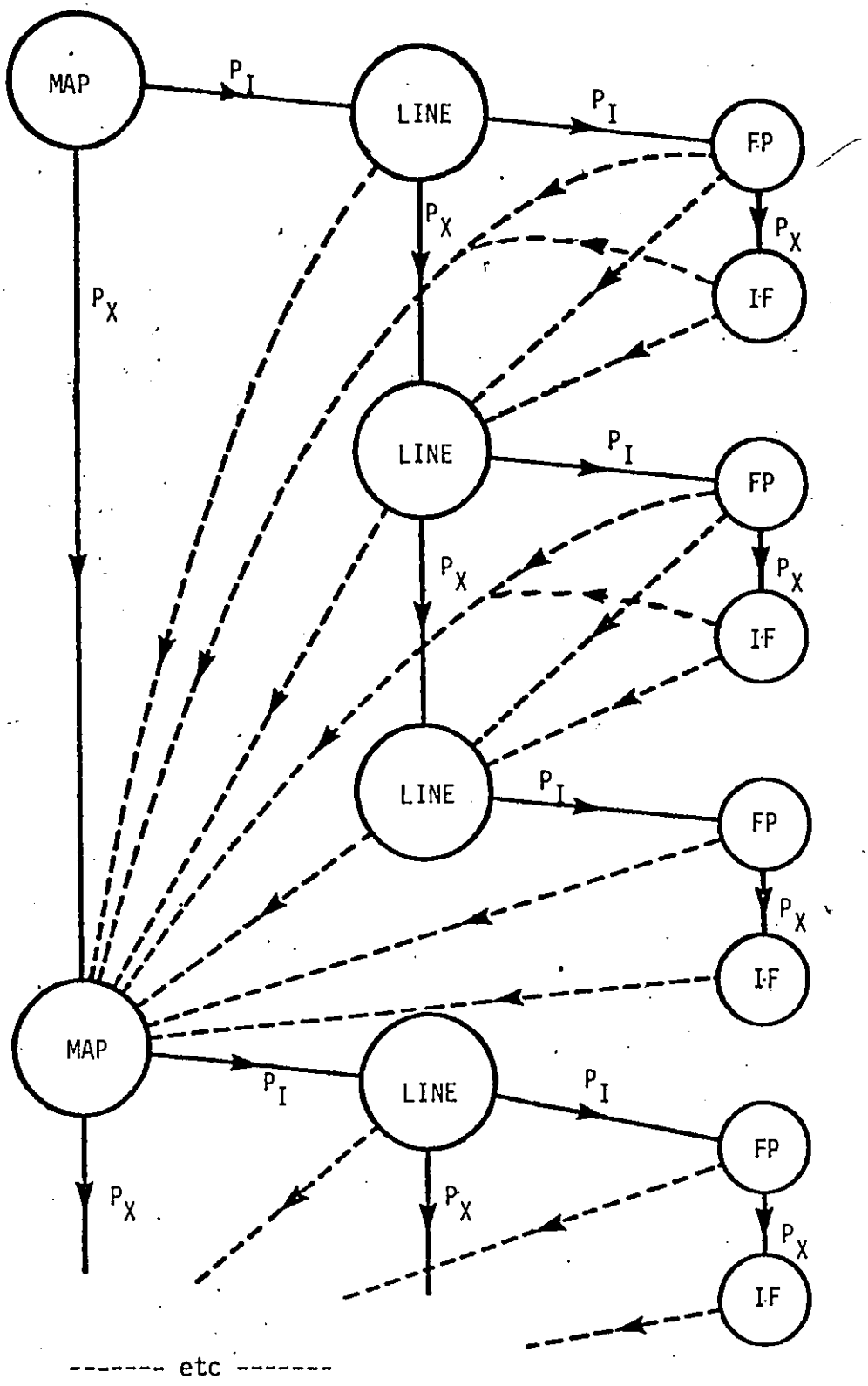


Fig. 22 Data navigation network provided by the pointers

P_X = external pointer

P_I = internal pointer

---◄--- = implicit pointer

MAP — group label

type or content of higher order factors was also specified more efficient routes exist. Using the example given in figure 22, if a particular MAP, LINE and FP combination was specified. Navigation would begin along the MAP level using the PX pointers, until the required MAP group was found, all subordinate groups would be bypassed directly. Once the MAP group sought was found, navigation would descend to the LINE level. Again progressing by PX only and bypassing all lower groups until the requisite LINE group was found.

Hence descent to the residue level would not be made until the residue groups so qualified by virtue of their higher factors. As a single bypassed map group could contain thousands of residue groups, this route is enormously more efficient than the previous route described.

iii) Retrieval by index

An index is compiled to permit random access to some level of the hierarchy (by means to be described in section 8.2.5). The nature of the pointers in the control aggregate permits creation of a different type of index, one that will permit movement within a sequential data set directly to a specified group.

The numerical value of either a PX or PI pointer defines the number of actual elements (data and control) between the start of a group and the start of either the next lower order group or next group at the same level. Hence the addition of successive pointers can create a "super pointer" capable of allowing many successive groups to be bypassed in one jump.

A single pass through the data could copy out all control aggregates, and if appropriate, a summary of group content sufficient to permit identification of individual groups. The index thus compiled could be placed at the head of the data set and employed to determine the exact location of any particular group within the data set, e.e. given the three level hierarchial data set:-

$$(A(B(C))) \tag{5}$$

-and the need to find the N^{th} C group in the M^{th} B group in the L^{th} A group, then its position PE (in terms of number of elements from the start of the data set) is given by

$$PE = \sum_{i=1}^{L-1} PX(A)_i + \sum_{j-1}^{M-1} PX(B)_j + \sum_{k=1}^{N-1} PX(C)_k \tag{6}$$

where $PX(A)_i$ are the values of PX pointers of successive A groups etc.

By these means, a single "jump ahead" is sufficient to retrieve any specified group at any level.

7.6 Definition of residual sub-aggregates

Structural elements have so far been defined which carry out the functions of the parentheses in the symbolic model.

In the most concise notational form, the only "control elements" are parentheses. In the expanded form which is predecessor to the concise form, however, there exists another structural element - the plus sign e.g.

$$(M_1 (L_1 (FP_1 + FP_2 \ldots)(IF_1 + IF_2 \ldots) + L_2 (FP \ldots . \text{etc} \tag{7}$$

This plus sign separates the repeated occurrences of each type of group at each level, and also separates the repeated sub-aggregates in the lowest level, unfactorisable residue, groups. It is evident that in the implementation of the model we must separate the elements of one sub-aggregate from the next within this residue; and that physical inclusion of a coded plus sign is not allowable.

The plus sign which appears before the repeated occurrences of whole groups, e.g.:

$$+ L_2 (FP \ldots \text{etc}) \qquad (8)$$

- presents no problem as it occurs at a group boundary and is therefore represented by the group label. Residual sub-aggregates, however, do not possess labels, not at this stage of development that is. (It will be shown later that certain types of data lend themselves to treatment of these residuals as groups in their own right as a practical measure.) At this stage, however, the residual sub-aggregates are of constant composition, i.e. each residual within a group of a particular type, contains the same number and type of attributes.

No formal means need therefore be taken to represent the plus signs. The data definition group will inform the system of the composition of the residuals. The system can then separate and retrieve individual residuals on the basis of this information, e.g. if the data definition group informs the system that the residual contains three elements, then successive extraction of each three consecutive elements in the lowest level group will meet retrieval needs without explicit delimitation of the residuals.

7.7 Comparison of the current system with DBMS concepts

Certain aspects ... may be described at this point to illustrate the similarities to, and differences between the current approach and that of the DBMS.

As noted (Section 3.2) the DBMS, in order to support several (possibly simultaneous) users, maintains several levels of definition and mapping relationships independent of the user program. Thus when the user program calls for a record by name the DBMS must:

i) Obtain the data sub-model employed by the user;

ii) Relate the record name to its description in the sub-model and determine the mapping relationship between the sub-model and data model.

iii) Obtain the data model and isolate the required segment of the data model via the determined mapping relationship;

iv) Determine the secondary mapping relationship between this desired segment and the physical storage.

v) Follow this second mapping relationship to the physical storage, retrieve the data requried then transpose it back through the chain of mapping relationships to the user's work space in the form required by the user.

After this the DBMS is free to carry out the same process via different mapping relationships for another user or for a different request by the same user.

A compilation system is, however, in a quite different situation. It will not be bombarded by a host of unpredictable requests from a variety of users with widely differing needs. At any one time only one "user" will exist – a module applying one or

another of the described compilation processes to one or another of the described data sets. The system will make thousands, even millions, of requests during the application of the process, but in the overwhelming majority of cases the form of the request is clearly established before commencement of the process, e.g. "Determine the apparent change in aircraft ground speed over every flight path fixed point in the survey".

The above request could, however, be put to several different data sets (one after the other – not simultaneously) of different content and structure. Thus the logical model seen by the program must be independent of the physical data set and mapping relationships must be established to ensure that program employs spatial coordinates to compute apparent speed – not aeromagnetic measurements or some such.

The number of requests makes a simpler relationship mandatory. Conceptually the same requirements exist as in the DBMS, i.e.

i) The program possesses a "data sub-model" – e.g. the records $(X, Y, T)_1$ and $(X, Y, T)_2$; where (X, Y, T) 1 and 2 are the spatial coordinates and time of passage over a consecutive pair of flight path fixed points respectively.

ii) There exists a "data model" which defines the content and logical structure of the data set aggregates, e.g. the aggregates are factored by line and contain 5 other attributes in addition to X, Y, and T.

iii) There exists a physical description which defines the storage address of the aggregates on the sequential storage medium, e.g. the X, Y, T data elements are the 3rd, 4th and 7th elements within the group "IFREC".

iv) Mapping relationships are necessary to cause transfer of a data element value to a program variable name.

The predictability of the requests, however, permits most of Martin's "eleven events" (Section 3.1) to occur once, and once only before the process proper begins thus greatly reducing the number of events which take place subsequently for each individual request.

The mechanism, in summary, is as follows:

i) Within the program there exists a correspondence table which defines the name and working space address of each program variable requiring input. ("Data sub-model" equivalent).

ii) The first event in the program run is that the user supplies a "correspondence list" which specifies the name of the data elements which correspond to each of the program variable names and the names of the groups in which these elements reside. This information is entered in the correspondence table. (Mapping; Data submodel – data model).

iii) At the head of the data set is a data definition group which defines the logical structure (factorisation state) and nominal content (data element names) of the data set (the data model). The physical structure of the data set corresponds exactly with the logical/physical structure of the data defintion group. The data element values occur in the data groups in the same order that their names occur in the definition group (mapping; data model-physical storage).

v) The process proper can now begin. The program calls to "find" the start of the first data group to be input. If other groups precede this, the FIND routine employs the pointers to bypass them. Once the group has been found the program calls to "BRING" the data into the work space.

vi) In the given case (as in the majority of cases) the data at this point is in a lowest level residual (repeating) group. To transpose and transport each successive "data record" to its equivalent "work space record" ranks, in Martin's terms, as only a single event, i.e. "Advance the base value of the input buffer pointer by the length of the residual group then transfer the contents of buffer address ID to working array address IV for all program variable-data element pairs." (See Fig. 23).

The actual process may be slightly more complicated in the case of mismatch between program variable type and data element type
but is still comparatively simple. Thus the equivalent of ten of Martin's eleven events have been consigned to occurrence once only at the start of the program run. An extra minor event (get the group start) occurs once only for each group. As the group in the majority of cases is a repeating group containing a large number of residual aggregates, this is a negligeable overhead.

In conclusion, we may note that the concept of data independence via variable mapping relationships between constant and independent logical and physical definitions is employed by both the current system and the DBMS but that the widely differing real conditions encountered by the two, necessitate a considerable difference in the mode of implementation.

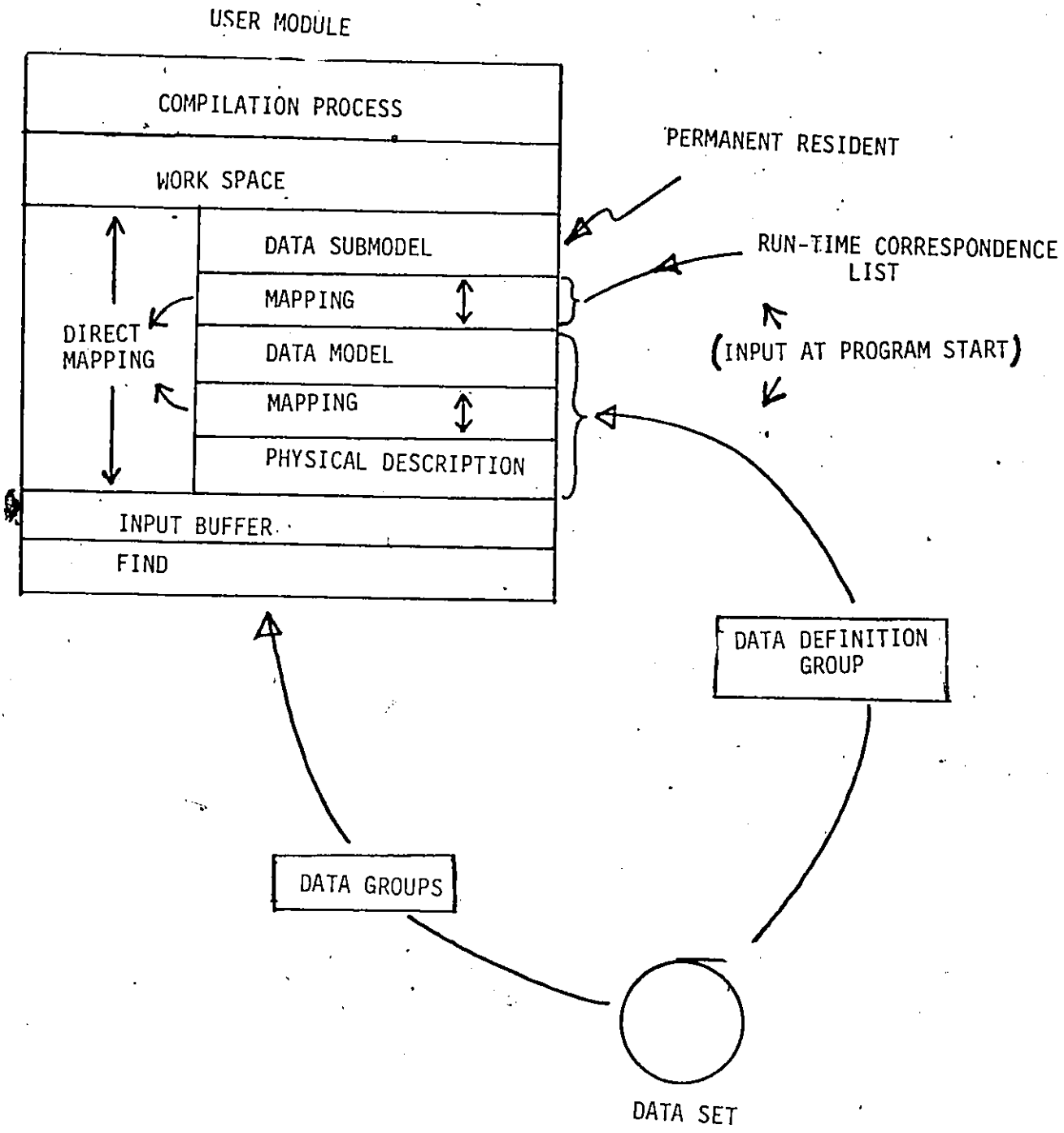


Fig. 23: Data-independence architecture within the generalized compilation system.

8.0 REALISATION OF THE GENERALISED COMPILATION SYSTEM

The data structure system so far designed is applicable to all the types of data described, in all of their respective levels of development. The data structure/content definition being included in the data set permits its exclusion from the processing software. This permits the creation of structure-independent software modules. Hence if a particular process is pertinent to more than one type of data, it should be capable of being applied to the different data types without modification of the process software, i.e. a generalised module.

The next requirement is to determine the software modules necessary for a generalised earth science compilation system. A module is an independently compilable set of program statements (i.e. a sub-program, several sub-programs, a main program with or without sub-programs or several main programs each with or without sub-programs). A set of program statements that is not independently compilable may not be described as a module. Modules are characterized solely by having a definable purpose and by being interchangeable according to requirements.

Modules may be made up of combinations of other modules, in a hierarchical grouping. As one descends in such a hierarchy the purpose of the modules becomes more and more limited and specific.

A system may therefore be described first as a few very high level modules. Then each of these modules may be described in terms of its own component modules in every-increasing detail. This process continuing until the "atomic" level of indivisible sets of program statements is reached.

8.1 Degrees of generality of software modules

The generality of modules will differ, and certain categories of generality can be defined:

i) Utilities

These will be found at all stages of all systems. They are not concerned with higher level compilation objectives but with execution of the most fundamental processes of the system such as basic input/output etc.

ii) General applications

These are concerned with a particular compilational task but are basic enough to be independent of data type. Hence they will be found at specific stages of all systems, e.g. a verification module. In some cases they may be made up, in part or in whole, of utilities.

iii) Special Applications

These are modules which perform a particular compilation task applicable to one, or only a small number of, data types. In many cases they may simply be a specialised combination of more general modules. Some, however, will inevitably be indivisible and data type specific, e.g. "one-off" routines.

The most fundamental utility routines such as input/output are too distant from our concerns with the structural model and earth science compilational objectives and need not be described in detail at this point. They are described in Appendix I. We will restrict ourselves to concern with the higher level modules more relevant to the subject of earth science compilation. Therefore, unless explicitly stated, it will be implicit that fundamental utilities are provided.

We will determine the necessary modules for each of the four basic types of compilational process in turn.

8.2 Manipulation modules

The compilation models in chapter 6 provide the basis for determination of the necessary manipulation modules.

8.2.1 Development of structure by factorisation

$$(A,B,C) \longrightarrow (A(B(C))) \quad (1)$$

Beginning with a cardinal form data set the first requirement that presents itself is to develop structure by factorisation. At the implementation level, this involves two stages, i.e.

i) Sorting

Having decided upon a factorisation hierarchy, the cardinal form aggregates must first be arranged in sequence according to this hierarchy. e.g. if the data set is to be factorised by M(map) then T(traverse) aggregates must be ordered in that sequence; by map in order then within each map in traverse order.

ii) Reduction to factorised form

Sorting in the above described manner would bring all aggregates with the same map number into contiguity. Likewise for all aggregates with the same line number within each map. The next process would be to scan the sequence of aggregates, implant the group labels and delete the redundant recurrences of common factors.

This process would require that each individual aggregate be examined in turn. The residue sub-aggregates not involved in the factorisation process would be automatically transferred to the output file. Factor attributes would be retained in memory. Whenever a boundary was encountered across which the value of a factor attribute changed, the appropriate group label and the new value of the factor would be implanted in the output file.

Sorting a sequence of aggregates and creation of a hierarchical data set by implanting labels are both likely to be needed elsewhere. Thus this task is best divided into three modules - a main "management" module, an "aggregate sort" and "labelled group creation".

8.2.2 Re-creation of cardinal forms.

$$A(B(C))) \longrightarrow (A,B,C) \qquad (2)$$

This is the reverse of the previous process. A factorised data set is to be expanded to cardinal form. The process required is:

i) Input factorised form group by group

ii) Retain higher factors in memory

iii) Add higher factors to each residual and output.

8.2.3 Refactorisation.

$$(A(B(C))) \longrightarrow (C(B(A))) \qquad (3)$$

This module is simply a combination of the two previous modules i.e. the data set would first be expanded to cardinal form then sorted and re-factorised to the new form.

8.2.4 Merging and separating of dot-product groups

$$(A)\cdot(B) \longrightarrow (A,B) \qquad (4)$$

For this process, the sub sets need not necessarily be in cardinal form. The process is essentially the same whether they are or not. It involves parallel input of two or more sub-sets, combination of the residuals, then output of the combined form, or vice versa. A single module would suffice in each case.

The "separate" module also takes care of the deletion process:

$$(A,B) \longrightarrow (A) \qquad (5)$$

which is merely separation into two streams, one of which is a cul-de-sac.

8.2.5 Creation of an indexed random access data set.

$$(A(B(C))) \longrightarrow (\bar{A}(\bar{B}(C))) \qquad (6)$$

This process is as follows:

i) Input the data group

ii) Store in memory the information chosen to identify the group

iii) Output the group to a random access storage location defined by a numerical key

iv) Append this same key to the identifying information in the index table to permit retrieval.

If the hierarchical level of the indexed group is high, the amount of information it contains may be too much to permit transfer in a single input-output operation. Hence the data may have to be segmented. This could necessitate a sub-index to each segment.

If the index is to extend to the lowest levels of the hierarchy, the amount of data in each indexed unit will probably be transferable by a single input-output operation. In this case, however, the index will have to be hierarchical itself, and could contain too many items to remain resident in memory. Although this would depend upon the size and complexity of the data set, it would be best to allow for it in the design of the module.

This would be done by placing the lower levels of the index itself on the mass storage device, e.g. given the process:

$$(A(B(C(D)))) \longrightarrow (\bar{A}(\bar{B}(\bar{C}(D)))) \qquad (7)$$

The index would itself be compiled as a hierarchical group structure, in the standard form, i.e. the index would have a structure identical to the factorised data set. The difference would be that each group would contain the identifying information and mass storage address of its associated data, rather than the data itself.

If, at any time, the available memory space became used up, the data throughput and indexing process would come to a temporary halt. The system would then re-process the index so far compiled, all lowest level (C) entries would themselves be despatched to mass storage leaving only a storage address in their place. The index could then be repacked into a much smaller space. When data throughput and indexing recommenced, indices to the C groups would only be temporarily compiled in memory. Each one when complete would be despatched to mass storage. Only the A and B levels of index would be retained in memory.

If his summarized index itself became too large for memory, the whole recapitulation process could be repeated to produce an index at the A level alone, both B and C indices thereafter being compiled temporarily then despatched to mass storage.

An index, once compiled, could be used in an immediately following process for random data retrieval. It could also be output to a sequential storage device then followed by the entire data set retrieved from random access storage. In this form it could be used to improve sequential retrieval efficiency as described in section 7.5.2. It would also permit the data to be placed directly on a random access device without having to recompile the index.

8.2.6 Group sort

$$({}^{1}A(B)) \longrightarrow ({}^{2}A(B)) \qquad (8)$$

When sorting aggregates, the largest aggregate encountered in the compilation systems described (Drift sedimentology≈300 attributes) is still small enough to be handled by the simple "bubble sort" type of routine.

Many requirements exist however in which the unit of data involved in the rearrangement is a whole group which can be far too large to handle by a simple sorting process. The best way to carry out this task is to pass the data through the mass storage, forming an index of the sort keys. Then the index itself is rearranged into the required order and hence the groups as retrieved also fall into this order.

This module is simply the previous random access indexing routine with the addition of a simple sort routine. If the index was so large as to require placement on random access storage itself, the module would apply itself twice, first to group-sort the index then to group-sort the groups per se.

8.2.7 Sub class recognition and separation

$$(A,B) \xrightarrow[A = P + Q]{} (P,B)\ (Q,B) \qquad (9)$$

This process is similar to the previously described separation of parallel sub-aggregate sets. The difference is that with parallel sub-aggregate sets, each individual aggregate is broken into two or more parts to form the output stream, whereas with sub-class separation whole aggregates are diverted into one or other of the output streams according to content.

This module can therefore be formed from the one for separation of parallel sub-aggregate sets, with the addition of a routine to recognize the sub-class of a particular aggregate.

8.3 Display Modules

Display, as noted, concerns the conversion of digital data into a form intelligible to the human being. The two basic types of display are print and graphics.

8.3.1 Print

Much of the necessary printed output will be taken care of within individual modules, e.g. such things as summary reports of process progression and diagnostic messages.

One thing to avoid within a module, however, is automatic lengthy listing of the contents of the data set. Such practices are time consuming and expensive and tend rather to bewilder the users than help them.

From time to time though, it will become necessary to examine the contents of the data set in some degree of detail. This case often arises as a result of peculiar problems with the data which are not amenable to solution by an automatic verification routine. In the kinds of system described in Chapter 1, this is usually taken care of by a host of ad-hoc "dump" routines, each one tailored to the particular structure of one type of data at one stage of its development. The existence of data whose content and structure is self defined presents a golden opportunity to discard this untidy collection of narrow-minded routines and replace the lot by a single, universal, data set content description routine (acronym -DSCD).

When a separate print routine is necessary for every stage of every data set, no one is willing to lavish great care and attention upon them to provide a choice of several forms of well ordered, easy to read output.

When only one routine is required, such becomes worth the effort. Hence a data set content description module (a sizeable program with many subroutines) is to be included in the display routines.

Such a program would, according to user needs, describe the contents of data set in a variety of ways from the briefest of summaries (overall group structure) to detailed listings of some or all data element values. It would also produce its output in a well structured, easily readable manner. (Such programs are known in data base management system circles as "Report writers", a somewhat exaggerated term).

This module would, in large part, be composed of other lower level modules such as type and content specific retrieval processes. It would also need a reasonably sophisticated "cosmetics" routine to paginate, tabulate, annotate etc., the output data.

Such is the advantage of the self-defined data sets that this single module is the only specialised print display routine necessary within the whole generalised compilation system.

8.3.2 Graphics

As noted in Chapter 2, many graphics packages exist which have within limits, general applicability. Some provide the user with a large variety of graphic facilities for a limited variety of data types. Others take the converse approach and provide the user with a limited variety of facilities for a broad range of data type. (Practical considerations prohibit, at the current state of the art, a graphics system which can provide all facilities for all data types.)

Any one of such packages, however, requires its input data in a specific form. Hence, to allow advantage to be taken of existing graphics packages, a utility module will be necessary which will convert our standard data form into that required for input to the external graphics package.

Such a module, however, is closely similar in purpose to the previously described DSCD. This module converts data to the special form needed for well-organized printer output. The basic components of this module, with a different supervisor module should largely suffice to produce output in the form required for external graphics packages. It, however, would require a greater degree of generality, as the variety of forms required for input to graphics packages greatly exceeds the variety needed for output to a printer. This module, therefore, having

achieved the level of generality capable of coping with the demands of a variety of graphics packages, can be elevated to the status of an Interface to External Systems (ITES). A high level utility for conversion of internal standard data to the specific form necessary for an external processing system - graphics or other.

One cannot expect, however, that all graphics display needs will be served by existing packages. Furthermore, it would be highly inefficient to have to exit from the system for every simple graphics requirement. It would be preferable if, for example, simple graphs and profiles etc. could be created within the system. The system, however, is not designed to serve any one particular graphics output device. Hence to maintain generality, graphics definition within the system should be in a device independent form. Special, quite simple, modules can be created to convert this general form to that required for each specific device.

Graphics definition will be an important part of the system and deserves a detailed analysis before deciding on the forms to be implemented. Such an analysis however, being only indirectly concerned with earth science survey compilation per se, will not be made at this point.

8.4 Verification

As is the case with print displays, many modules will contain their own verification routines. This is not a deviation from the principle of modularity but a necessary consequence of the nature of errors. Certain types of error, "syntactic" errors, are immediatley detectable at any point; alphabetic input when numeric was expected for example. Other types of error exist, however - "semantic errors" -

which are less immediately evident. Such errors are context dependent, i.e. no errors can be detected when the data is viewed in isolation. They only become apparent in the context of the process to which the data is submitted.

Accordingly, the test for correctness needs to be placed within the appropriate context, i.e. close to, or within the process module concerned. As a matter of good programming style, all process modules should contain some degree of self diagnostics.

The need still exists, however, for external independent verification processes. When a serious data error is detected within the process module, it is often too late to do anything about it. The process will often have to be aborted and processing up to this point will have been wasted unless some means exist to salvage the results.

It is obviously preferable to preview the data so that all detectable errors can be removed in advance, thus improving the chances of a clean run when the data is submitted to the process proper.

Again, as with print display, the self defined data permits a single set of general purpose modules to execute this task for all forms of data. Its two basic components will be inspection and correction, e.g.:

i) Test each relevant element against a set of acceptability criteria;

ii) Take action as specified in the cases of test failure.

8.4.1 Inspection

It is interesting to note that once again hidden instances of common processes have been brought to light. i.e. the first component above is identical to that required for content selective retrieval, and also to the previously described aggregate type recognition module required for sub-class recognition and separation. In all three cases a condition is stated against which the values of data elements within an aggregate are compared. For content selective retrieval the response when an aggregate meets the condition, was simply to transfer it to the recipient process. If the recipient process was output to one stream rather than another, this covers the needs of sub-class separation. In the current case, the recipient process is the correction routine.

Simple cases of these needs could all be met by a single module. In the interests of a more powerful and flexible system, however, at least two modules would be preferable. Both would be capable of application to all three of the above tasks, but at different levels of complexity (and hence cost of operation). The modules will be named ACT (for aggregate content tester). The lowest level of complexity (ACT1) would test each specified element against a set of acceptance/rejection ranges of value. Actions would be taken either on the basis of each individual test of an element or after completion of testing of all elements within an aggregate. In the latter case action would result only if all tests were positive, i.e. an automatic AND relationship would apply to the test results, e.g. "If

test 1 is positive AND test 2 is positive AND test 3 is positive etc. then act; otherwise continue with the next aggregate". The second level of complexity (ACT2) would permit submission of element values to a more complex logical function (allowing conditions containing AND, OR and EOR) or to an arithmetical function. This would allow:

i) recognition and separation of more subtly distinct sub-classes,

ii) Complex conditional retrieval as described for drift sedimentology;

iii) Logical verification functions to be applied.

8.4.2 Correction

This process also has broader applicability than merely within the verification module. It falls in fact, in the general category of data editing, and it too would best be provided for by several routines of increasing level of complexity.

The simplest level of complexity allows for replacement of an erroneous element or aggregate. The element can either be nullified or corrected. The former involves replacement of the element with a special value which alerts subsequent processes to bypass it. The latter involves replacement of the element with a value which will be acceptable to subsequent processes. This replacement value could either be predefined or could be calculated by a simple function, e.g. linear interpolation between the adjacent two elements.

In some cases it may not be appropriate or feasible to automatically replace an element i.e. the decision as to corrective action may have to be made by the user according to the particular situation. In these cases the program would report the

condition and flag the erroneous values. After inspection by the user and decision as to corrective action, a second program run would be made to make the replacements as decided upon. All of the above actions would be carried out by very simple modules attached to the ACT module.

The second level of complexity allows for physical deletion of erroneous data and/or insertion of erroneously absent data.

Unless the number and type of aggregates removed is equal to the number and type of insertions within any particular group, then changes to the control aggregates also become necessary. This requirement can, however, be satisfied by a two stage process.

The first stage will indicate the nature and location of errors. This can be done by the previously described modules of the first level of complexity. The second stage involves a generally applicable editing utility.

8.5 Development Modules

As noted in Section 6.2 above, quite specific development needs can often be met with a specialised combination of general purpose modules. In the examples given, the majority of the processes are manipulatory. The remainder consist of two specific development processes - "Interpolate" and "apply".

The "apply" routine involves the retrieval of aggregates one at a time then submission of selected elements to an arithmetical function, by which the value of existing elements will be changed or new elements created. This type of routine occurs in many places throughout several of the systems discussed. It is of the lowest level of complexity described in section 2.2.5.

A simple "unit aggregate development function" routine with provision for attachment of user specific functions would fully satisfy all the variety of such needs encountered within the systems described.

The "interpolate" routine is an example of the second level of complexity described in 2.2.5. It involves interaction between several different aggregates within each part of the data set. It is still amenable to generalisation however. Several different types of interpolation function will be required to meet the various needs. These, however, can reside in a subroutine library to be attached and executed by the user as required.

Development requirements will therefore be satisfied by two complimentary means:

i) Creation of special purpose modules by specialised combination of general purpose modules;

ii) Creation of a sub-program library to meet those requirements not satisfied by i) above.

Existing development programs may be used via the ITES external interface. Such software as statistical packages etc. can be accessed in this manner.

Description of necessary specific development routines will be made in chapter 9 when the needs of whole systems are considered.

8.6 Summary of Software Modules Thus far identified

i)	ACT	Aggregate content tester
ii)	SPSA	Separate parallel sub-aggregate sets
iii)	MPSA	Merge parallel sub-aggregate sets
iv)	AGSO	Sort contiguous aggregates or sub-aggregates
v)	FACF	Factorise cardinal forms (includes iv)
vi)	EXFF	Expand factorised form
vii)	REFA	Refactorise (includes v and vi)
viii)	INST	Create an indexed mass storage data set
ix)	GRSO	Group sort (includes iv and viii)
x)	SCRS	Sub-class recognition and separate (includes i)
xi)	DSCD	Data set content description
xii)	ITES	Interface to external system
xiii)	REEC	Replace erroneous elements by corrections
xiv)	DICI	Data inspection and correction version 1 (includes i and xiii)
xv)	UADF	Unit aggregate development functions.

An example of actual working software (Basic Input/Output and Manipulation) is presented in Appendix I.

9.0 MODULAR CONSTRUCTION OF SPECIALISED COMPILATION PROCESS

In the interests of brevity we will not attempt to reconstruct all the compilation systems described in Chapter 1 as individual, complete assemblages of modules. In fact, to do so would be to oppose the advances of generality made thus far. The intention of this work is to define a generalised base from which a great variety of more specialised systems can be constructed as needed. The same manipulation, general verification and display modules will re-occur in all systems. Individual systems will be characterised only by specialised combinations of general modules and by a small number of particularly specialised routines.

Specialised routines will in most cases reside in a subroutine library and be attached to a main program according to needs. This main program will do no development itself. It will exist solely as a management module to direct data to the pertinent processes. In consequence, the general design of this program can be established.

9.1 General Management Module Structure

All management modules in the compilation systems will have essentially the same design, i.e.

i) Initialise the process.

ii) Input control information.

iii) Find the first group of required type in the data.

iv) Descend the group hierarchy submitting the data at each level to the appropriate process along with control or background information.

v) Direct the results of the process to output and/or storage as background.

vi) Repeat from iii) until the data set is exhausted or some other terminating condition occurs.

vii) Finalise the process.

In theory, at least, a single general master program could be written to which control information would be input which specified the groups to be retrieved, the processes to which they were to be submitted, and the destination of the results.

This possibility should be kept in mind but in most if not all real cases to be encountered such a level of generalisation would be too much.

The ease of creation and maintenance, and the realistic needs of generalisation are adequately served by management modules individually tailored to the particular purpose.

The concept of the master program should however be the design basis for each management module, i.e. a consistent structure should be employed throughout. This is a matter of good programming style which will greatly help in maintenance of the software.

We will now describe the modular content and structure of three major types of application module.

9.2 Coordinate transformation of digitised track data

In aeromagnetics and airborne gamma spectrometry, as described in Chapter I, track data is plotted on base maps and then digitised. Initially each base map has its own reference frame and its units are arbitrary. The objective of this process is to relate all track points to a single reference frame, with true cartographic coordinates. Figure 24 depicts the structure of this module.

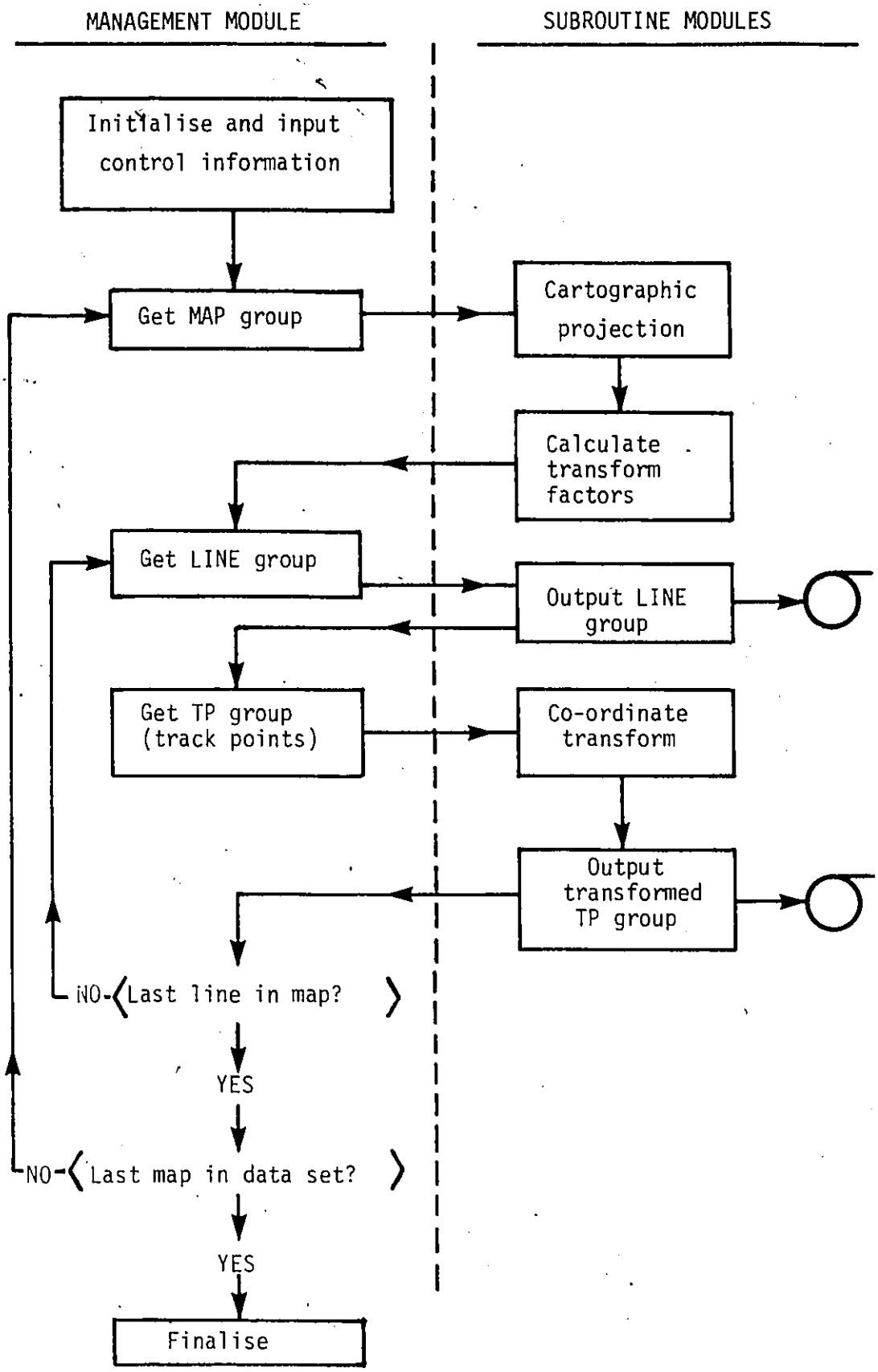


Fig. 24 The track data co-ordinate transform module.

The management module design is as exactly described previously. After initialisation and control input, the highest level groups are sought (MAP). From each one, the latitude-longitude values of the map corners are submitted to a general cartographic projection sub-routine which returns the cartographic coordinates of the corner points. These data and the digitiser coordinates of the map corners are then submitted to a second, general-coordinate-frame-relation, sub-routine. This routine, given the coordinate of a set of points in both of two different reference frames, calculates the rotation, translation and scale factors of the second frame with respect to the first.

The process now descends in the hierarchy of the MAP group to seek the LINE group. The process associated with this data is simply to output the line header.

One further descent retrieves the TP group, a lowest level residue of repeated values of track point coordinates.

These values, plus the transform factors, are supplied to the third subroutine, a general coordinate transform routine, which transforms all points to the cartographic reference frame.

These processes are repeated for all lines in all maps.

Included with each subroutine would be a context-dependent verification routine, e.g. in the context of the standard map sheet, the latitude-longitude coordinates of the corners should describe a geographic "rectangle" whose frame is orthogonal to the geographic frame. A test to see if this was in fact so would detect most miskeyed latitude-longitude values. Likewise, only two points are needed to determine all the transform factors between two frames. Four points are supplied.

This gives sufficient excess information to determine the isotropy of the digitiser frame, which assists in the detection of misdigitised points or distorted base maps.

While derived from the existing aeromagnetic compilation system, this module is applicable to any form of digitised track data, and the process sub-routines are of an even more general nature.

9.3 Extraction of intersection point data from a track network

In many survey disciplines one of the compilational processes involves finding an intersection point between ordinary traverses and control traverses. Once found, certain data items pertinent to the intersection points are derived or extracted and used to form the "intersection data set".

Each traverse and control line are described by a set of points along the track. Each point possesses at least a pair of spatial coordinates and usually an identifying parameter (a "fiducial"). Thus the traverse as a continuous line is made up of the notional straight lines (segments) which join each consecutive pair of track points.

Although ostensibly a quite complex and specialised two dimensional process, it can be reduced to a sequence of largely general linear processes. The first stage, carried out by completely general manipulation processes is to form separate data sub-sets of control lines and traverses respectively. A good practice with such data sets is to determine the limiting X and Y coordinates of each line and place this information in the line header data group. This having been done the control module for the intersection data set extraction once again follows the standard design, i.e.

i) Initialise

ii) Get a control line LINE group

iii) Get a traverse LINE group

iv) Submit the limit data from both to determine if any intersection is possible. If not, the relevant process is "Null", i.e. simply get the next traverse LINE group and repeat the process.

v) If an intersection is possible, descend to the FP track point group of both control line and traverse.

vi) Take each control line segment in turn.

vii) Take each traverse segment in turn.

viii) Submit both segments to general coordinate geometry routines to determine if and where the segments intersect. Output the intersection coordinates.

ix) Repeat from vii) for all traverse segments.

x) Repeat from vi) for all control line segments.

xi) Repeat from iii) for all traverses.

xii) Repeat from ii) for all control lines.

Figure 25 depicts the structure of this module. As can be seen the management module is involved solely with navigation up and down the hierarchy of the data set, and transfer of groups to necessary processes.

The navigation in this case is somewhat more complex than the previous case due to the two-dimensionality of the task. It is still, however, basically simple and adheres to the general structure described.

In the aeromagnetic compilation system, the intersection data set is submitted, with the main data set, to a levelling system. This system through in objective is

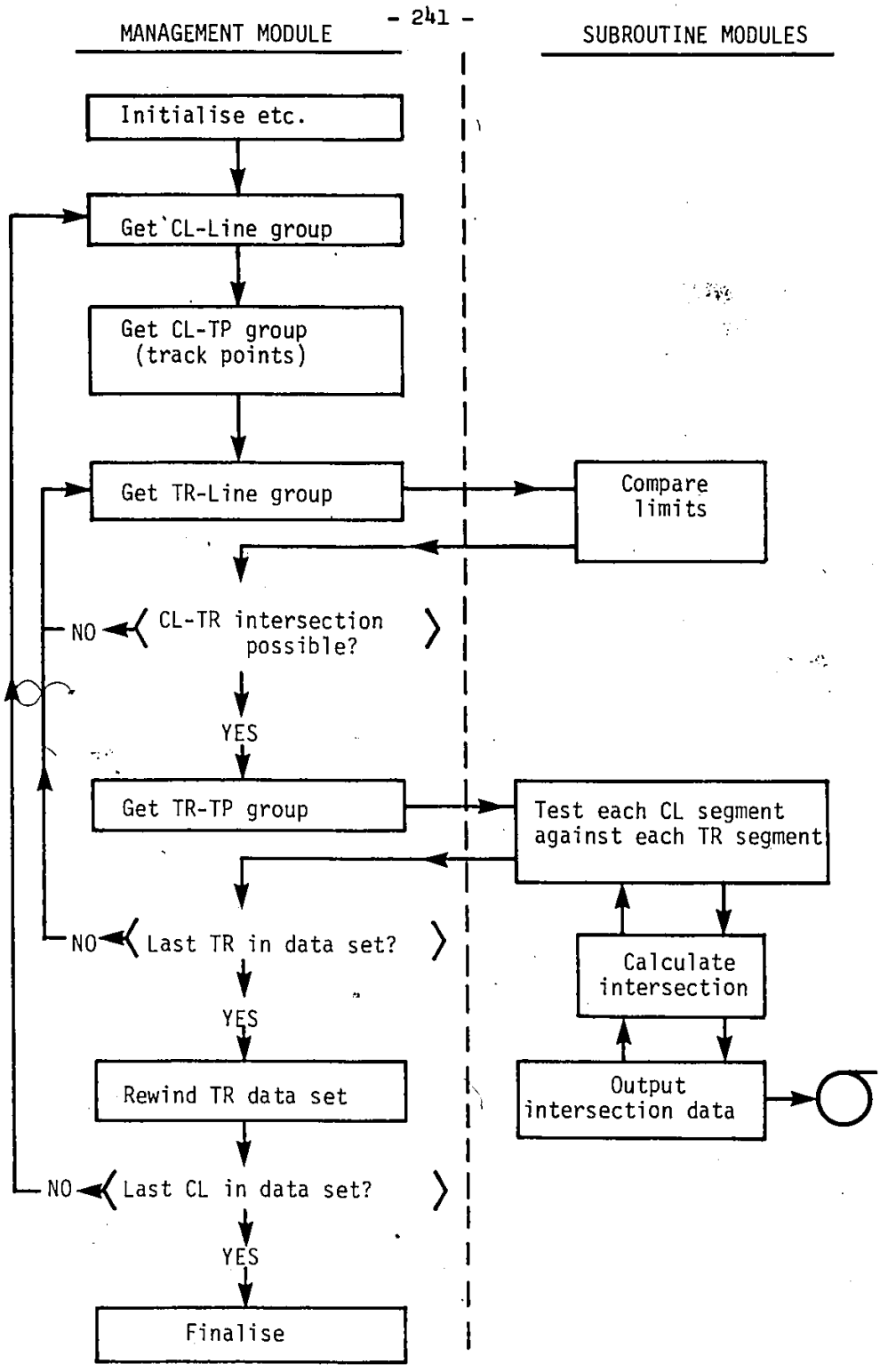


Fig. 25 - Intersection data set module.

highly specialised, is in fact largely made up of a sequence of standard manipulation routines between which occur small development routines, as is demonstrated in section 6.1.1 above.

Extraction of an intersection data set as described is a common requirement. The actual levelling process to which it is submitted is much more specialised and as such will not be described at this point.

9.4 Grid interpolation from profile data

The pre-cartographic stage of all earth science autocompilation systems which require an isoline map, includes a grid interpolation system. This produces a regularly defined numerical surface accross which isolines can be traced. The gridding system described for aeromagnetics is not restricted solely to that discipline. It is specialised only insofar as it is restricted to densely sampled data, i.e. sampling frequency in excess of the maximum spatial frequency within the quantity being sampled. It is therefore general enough to warrant description at this point. As with the levelling system, it too consists of a sequence of separate modules. The processes are as follows:

Phase I

i) Initialise etc.

ii) Input profile group (a chain of spatial coordinate pairs each with a value of the variable to be gridded)

iii) Interpolate and evaluate along the profile at the specified grid interval.

iv) Output the interpolated profile

v) Repeat from ii) for all profiles.

Phase II

i) Refactorise interpolated profile data set, delete profile sub-aggregate, sort residue by X. This produces a set of X-direction sections across all profiles, i.e.

$$(P(Y,X,IP)) \longrightarrow (Y(X,IP)) \qquad (1)$$

where P is the profile identifier,

Y is the Y coordinate of an interpolated point,

X is the X coordinate,

IP is the value interpolated at this point on the profile.

Phase III

i) Initialise etc.

ii) Input profile section group

iii) Interpolate and evaluate along the section at the specified grid interval to produce a grid row.

iv) Output the grid row

v) Repeat from ii) for all sections.

This is often followed by a second re-factorisation phase to segment the grid for presentation to the mapping package. In this system, it is interesting to note that the management modules for Phases I and III are identical. The only likely difference between the two phases is in the choice of interpolation function employed. Interpolation functions belong to a general class of subroutines, specific members of which can be attached as required. Hence in general terms only one basic module is required. (See figure 26).

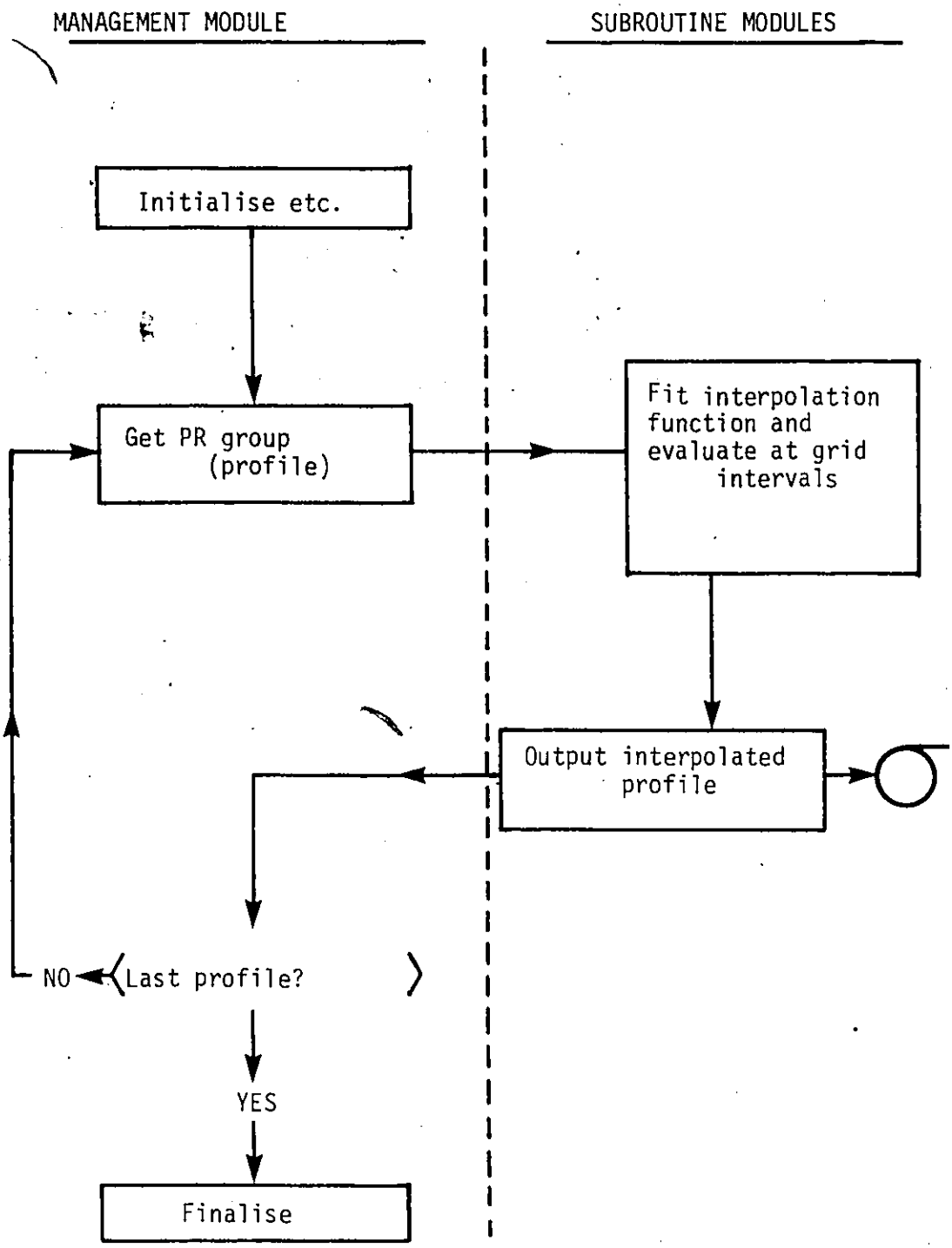


Fig. 26 Basic module for grid interpolation of well sampled profile data.

This chapter could continue at length, describing module after module for all possible compilation tasks. A brief look-ahead, however, satisfied this writer that they would all come out essentially the same - at least within the five disciplines studied. (airborne, gamma spectrometry, drift sedimentology, regional geochemistry, aeromagnetics and gravimetry.) i.e. each would consist of a management module of the same general form, calling upon members of a developmental sub-routine library. As these five disciplines represent a broad and representative cross-section of the earth sciences to which survey compilation is relevant, it is reasonable to predict that modules would be essentially the same for other disciplines also.

This prompted the writer to decide that the originally stated objectives of this work has been achieved.

10. CONCLUSIONS

We will briefly recapitulate the stages of this work and compare the results with the originally stated objectives.

10.1 Recapitulation

Recognition and definition of the four basic types of compilational process allowed us to escape from the narrow viewpoint of the individual discipline.

The principle obstacle to generalization of the software system was recognized to be the diversity of data content and structure. To surmount this obstacle, a high degree of data independence from software would be necessary.

Currently existing methods of achieving data independence within Data Base Management Systems (DBMS) were examined. Though their essential concepts were pertinent their practical applicability was severely limited due to the peculiarities of the data and processing requirements of compilation systems. This fact instigated a search for an effective data structure model peculiar to the needs of earth science survey compilation.

Examination of the highly effective structures encountered within the various discipline specific compilation systems, and recognition of the advantage of linearity in a model, led to the development of a discipline-independent data structure model which effectively described all previously encountered real cases of data structure

Formal algebraic manipulation of the model expressions were shown to be exactly analogous to the manipulation processes of compilation systems. Thus proving that at least this type of process could be made independent of discipline.

Maintenance of data description within the data set allowed for data-structure independent software. This in turn allowed generalisation of verification and display processes.

The last type of process, development, was admitted to be the least amenable to generalisation. However, when the developmental processes shown on the system flow charts were analysed in detail and expressed in the model notation it was demonstrated that many such processes were made up in greater part of a specialised combination of more general routines.

When the most specialised parts of these processes were analysed and process modules synthesised, it was found that the same, quite simple, general form of management module was applicable to all cases. The management modules would call upon a library of subroutines of varying degrees of generality.

10.2 Meeting the objectives

The goals of this work were to establish the viability of, and a design for, a generalised automated compilation system for earth science survey data. The objective being to eradicate the problems inherent in a multiplicity of independent, discipline specialised, systems. These problems were duplication of effort in creation, maintenance and education, and incompatibility of data structure.

The data structure system designed employs the same set of input/output modules regardless of data type or degree of structural development. This, in essence, is the meaning of compatibility of data structures. All data structures regardless of the degree of factorisation or the physical ordering of the aggregates, can be reduced to cardinal form, sorted to any specified order and refactorised in any way permitted by the common factors that exist. Thus two organisations dealing

with the same type of data, can structure the data differently if their individual needs differ, but if so, then either organisation can restructure their data to be compatible with the other simply by specifying the factorisation hierarchy to the REFA module.

If it makes scientific sense to combine two or more types of data in a multiparametric process (e.g. correlation of geochemical and airborne gamma spectrometry data), then if they are not already in compatible order they can be made so without creation or modification of software. Even the need to refactorise will not be a common event. Major processes such as verification and many display applications are independent of the factorisation state of the data.

Regardless of the factorisation state, all attributes are accessible and every cardinal form aggregate can be temporarily reconstructed in full by retaining higher order factors in memory as the lowest order residue passes through.

Any definable process, manipulation, display, verification or development can be expressed in a single software module. So expressed,it can be applied to any type of data to which it is pertinent. Hence the number of different modules needed will represent the number of fundamentally different processes to be applied. Previously, different modules were necessary to apply the same process to different data types. In many cases the situation was worse, different modules were needed to apply essentially the same process at different stages within a single compilation system. e.g. Several different input/output, verification and print display routines were needed within a given system due to changes in content and structure as the data passed through the system.

This writer's estimate is that the number of program statements required to fulfill the needs of the five described disciplines via the generalised system as proposed, would be greatly reduced. As there would be not only a reduction of redundancy between systems, but also within individual systems. The system as established meets the specified objectives.

10.3 Limitations of the system

10.3.1 Computational efficiency

The principal limitation of the system is one that affects any generalised system and any modular system, namely computational efficiency. The proposed system being both generalised and modular, it is doubly prone to this ill.

Generalisation necessarily means that things are done in the most general manner. What is generally applicable to many cases may not be the most efficient way in the individual case. e.g. any data structure in the system can be converted to any other by the refactorising process carried out by a single module. Many real cases will be encountered, however in which the desired structural change could be effected much more cheaply by an ad-hoc process which moves aggregates directly from one place to another without explicit expansion and refactorisation.

The cost of modularity is due to "overheads" incurred by the transfer of data and control. Computation via a set of subroutines can be more expensive than by "in-line code", i.e. a single program unit which contains all the processes necessary without having to transfer data to and from an external subroutine. The proposed

system will be highly compartmentalised being made up almost entirely of subroutines. Even though the choice of subroutine categories and their design will be made with strict attention to machine efficiency, the system could still be significantly more costly to run than a monolithic system.

This limitation is real. The questions to be answered are "How critical is it?" and "How can it be avoided or reduced?"

To answer the last question first; A well designed general purpose module or a subroutine based system, can be more computationally efficient than a poorly designed special purpose module or monolithic system. As far fewer general purpose modules are needed than special purpose modules, much more time can be dedicated to good design of the former than of the latter. Nothing, however, prevents the users from creating their own special purpose, more efficient modules to suit any need that arises.

With regard to the problem of modularity, if subroutine overheads are unbearable, the subroutine processes are there, ready made, for the user to combine into in-line code in a larger module. This is an enormously simpler task than having to break down a monolithic system into subroutines if the overhead of complexity becomes too great.

Hence means do exist to avoid or reduce these two forms of inefficiency should they become critical.

In answer to the first question "How critical is it?" it must be remembered that the objectives are concerned with improvements in human rather than machine efficiency (reduction in manpower resources expended on creation, maintenance and learning to use a variety of systems; reduction in both man and machine costs in writing and applying interface software to convert from one form of data to another.)

Hence a potential increase in processing time can be balanced against a certain decrease in manpower time. As the costs of computation have decreased 100 fold per decade, each of the past three decades, and the cost of manpower has doubled each decade over the same period, it is evident that improvement in manpower efficiency at the expense of machine efficiency is the wisest investment.

10.3.2 Constraints upon data content

In principle there are no constraints upon data content. In practice, ease of application will diminish with increase of data complexity, setting a practical upper limit on the data complexity.

A vitally important requirement is that users correctly interpret their data contents in terms of the concepts of entity sets, attribute sets and group relationships etc., described in Chapter 4 and 5. Otherwise manipulation processes could produce false results, e.g. if the wrong relationship (dot, star or null) is ascribed to two or more contiguous residue groups then automatic expansion will produce nonsensical results.

As data complexity grows, so will the likelihood of mis-interpretation. The practical upper limit to complexity will, however, be limited by the interpreters skill in this case, and can be expected to rise with experience.

As the inhomogeneity and variability of aggregate content increases, a different type of limit will impose itself. At some point, when the data set is for all practical purposes unfactorisable, and so inhomogeneous that a separate data description is required for almost every aggregate, then the practical advantages of the system break down, i.e. the data structure system is not applicable to such data.

One need not worry too much about this limitation, however.

If the applicability of the data structure system breaks down due to this case, so also does the applicability of compilation itself! The system was designed expressly for the needs of earth science survey data compilation. Whatever applicability it might find elsewhere is fortuitous.

Earth science surveying may be characterised as repeated measurement of one or more earth science parameters at systematically distributed locations over a defined region of space.

"Repeated measurements" and a "systematic distribution" of the location of these measurements, guarantee a data set suitable for treatment by the system. Extreme inhomogeneity and variability of aggregate content is not characteristic of earth science surveys (though, as noted, extreme variability of attribute value is distinctly characteristic of earth science data).

Certain cases will be encountered where a degree of inhomogeneity exists -e.g. a "sparse" data set whose ultimate aggregate content will be quite homogeneous but whose data is acquired slowly and creates inhomogeneity in the early forms of the data set. Means exist within the system to deal with this however.

10.4 Direction of future work

The general direction of future work is clear, develop algorithms and write specifications for system modules, then encode, test, document and release for use.

The overall priorities are also quite clear. Modules must be implemented in decreasing order of generality as more specialised modules contain and depend upon less specialised ones, i.e. the general system utilities must be created first to permit creation and retrieval of data sets; followed by the general print display system to

permit examination of data sets so created. The next set of modules to be created will be the manipulation routines then the general verification, graphics display and system interface. Finally, more specialised development modules can be constructed from the modules so far created with the addition, where necessary, of development subroutines specially created for individual needs.

Hence, in keeping with the basic philosophy of this work, compilation systems for individual disciplines will not be created separately one after the other. The relative complexity of the various systems and the necessary progression from general to specific, however, dictates that the means to create simpler systems will become available before the means to create more complicated ones.

Utilities, basic verification, basic manipulation to allow appropriate structures to be developed, and a selective retrieval routine would be sufficient for the needs of a drift sedimentology compilation system of the type described in Chapter 1. More advanced manipulation routines added to this would form the core of a geochemical compilation system. Creation of an aeromagnetic, airborne gamma spectrometer, or gravimetric compilation system would depend upon which type of development modules were created first. The similarities between certain phases of aeromagnetic and airborne gamma spectrometry compilation, however, makes it preferable to approach these two systems first, i.e. more general routines are involved.

10.5 Extension of the work to fields beyond compilation systems

Although the system is specifically intended as a means to improve the technique of automatic compilation, the compilation methods and the form of the results in any particular case should take into consideration the eventual end use of

the data, i.e. interpretation of data in terms of the geological contents and structures of the lithosphere. In the cases where interpretation (at least the computer automated aspects of it) concerns only one type of data, the essential needs are provided for with the system as described, i.e. the retrieval and manipulation processes permit the data to be presented to the interpretation "development" modules with whatever form and content is necessary.

Cases have been mentioned, however, which employ two or more types of data for interpretation purposes. As mineral resources become scarcer and the search for them becomes increasingly more costly, more and more sophisticated methods will have to be employed to improve the probability of success in any given case.

Principal among such methods are those which attempt to reduce the areas of uncertainty by integration of several forms of data. This can be done deterministically, for example, by combined magnetic-gravimetric interpretation. Both gravity and magnetics, as noted, have inherent uncertainty in their respective interpretations. A causative body may be determined which could give rise to the observed field, but no absolute means exist to determine whether the body is in fact that which lies beneath the surface. Boundary conditions must therefore be prescribed to keep the interpretation within "reasonable" limits. The assumption that the body responsible for a magnetic anomaly corresponds to some degree with the body responsible for a gravity anomaly restrains the interpretation of either to a form compatible with the other. Better still, a direct theoretical method, based upon potential field theory, could be used to create an interpretation at once compatible with both data forms.

The system as defined would appear to satisfy the needs of either of these cases, as the interpretation per se would become just another development module to be supported and supplied via the general modules.

The cases, however, which are amenable to such a deterministic approach, as in the cited case, are mostly concerned with specifically identified targets. The major problems occur before this stage. They are concerned with the process of deciding where to look for targets. They involve not just two or three but many more forms of information, and by the very nature of the information involved are probabilistic not deterministic. The probabilistic nature of geology is noted by Agterberg (1979, p. 201) who also states that "It will become increasingly necessary to integrate the results obtained by the various subdisciplines... The relatively new field of resource analysis provides one good reason for the integration of geoscience data of different types." Fox (1979, p. 141) mentions two specific examples -correlation between airborne gamma spectrometry data and corresponding geological maps to determine background and anomalous values for various geological units; and correlation of aeromagnetics with topography to determine which anomalies have strong terrain correlation and are hence suspect as indicators of sub-surface features.

Integration of different types of data requires physical and logical compatibility between the contents and structures of the different types of data. To attempt to establish a comprehensive but rigid standard covering all possible types of geoscience data that might conceivably be put in an integrated data bank would be a monumental task. Even if a standard could be established, it would be likely to meet with opposition whenever its implementation would require a structure to be imposed on some form of data which was radically different from the structures that the workers in the field were used to.

Thus, while comprehensive in theory, it is highly unlikely that such a "table of standard formats for geoscience data" would ever become comprehensive in practice. The success of multivariate analysis for prediction of mineral occurrences, however, depends upon obtaining the greatest possible quantity and variety of relevant data.

On the basis of the well known laws of perversity it is probable that the one type of data not available for the analysis would be the key ingredient that would result in effective predictions.

The proposed algebraic group data structure and generalised software system could actively assist in the development of a truly comprehensive integrated data bank by acting as an "interlingua".

It can be seen in examples in Chapter 4 and in the compilational models in Chapter 6, that sensibly structured data sets need little or no change to their content and order to convert them to an algebraic-group structure. All that is required is the addition of group labels. Conversely put, the algebraic group structure is flexible enough to adapt to any well organized data set, regardless of its content, with minimal change to the original form.

Hence one could expect resistance to conversion to this type of standard to be less than for conversion to a rigid standard. Once data is converted to the algebraic group structure it is essentially compatible with any other form of data with algebraic group structure; expansion, re-ordering, re-factorisation, separation and merging of data sets all being routine matters.

As was the case with the drift sedimentological data set, creation of highly structured compatible forms within the data bank may not be necessary. Provided

that all data was in algebraic group form, the data sets could be left in their original form and each user could convert all selected types of data to a more developed form most appropriate to that user's needs only when the occasion arose.

A corollary to the above is that if data is not logically structured then it would be more difficult to convert to the proposed standard. This, however, is in itself beneficial. The logically sound data structures described in Chapter 4 were not created on theoretical principals but as a reasoned but ad-hoc approach to particular individual cases. The example given of the illogical data structure was also created in an ad-hoc manner but as necessitated by the limitations of an in-flight data acquisition system and not by the needs of compilation. In the absense of abstract guidelines and definitions of logicality for data structures, little exists to restrain the emergence of illogical structures.

Even if the only outcome of this work is to provide a theoretical framework by which the "correctness" of a proposed data structure can be judged and thereby assist in the avoidance of illogical structure, then the effort of carrying out the work will have been well spent.

REFERENCES

Agterberg, F.P.

"Statistics applied to facts and concepts in Geoscience" Geology en Mijnbouw, Vol. 58 (2) p. 201-208, 1979.

Anon

Projeto Geofisico Brasil-Canada. "Historico E Atividades ate 30/09/77" Ministerio das Minas e Energia, Brasilia, Brasil, 1977.

Anon

"Mapping Software and Cartographic Data Bases" Harvard Library of computer graphics 1979 mapping collection, vol. 2, 1979.

1. CODASYL Systems Committee

"A survey of generalized data base management systems" Technical report, May 1971.

2. ——— Data base task group of CODASYL programming language committee, Report, April 1971.

Berztiss, A.T.

"Data structures: theory and practice" 585 pp. Pub. Academic Press, New York, San Francisco, Lond, 1975.

Bhattacharyya, B.K.

"Bicubic spline interpolation as a method for treatment of potential field data" Geophysics, Vol. 34, pp. 402-423, 1969

Brlener, S.

"Applications Manual for Portable Magnetometers": Pub. Geometrics, Sunnyvale, Calif. U.S.A. 1973.

Breiner, S., Lindow, J.T., Kaldenbach, R.J.

"Gamma-ray Measurement and Data Reduction Considerations for Airborne Radiometric Surveys": Proceedings of a Symposium on Exploration of Uranium Ore Deposits: International Atomic Energy Agency, Vienna, 1976.

Bristow, Q.

"A Computer-Controlled Geochemical Analysis System for use with Existing Perkin Elmer Double-Beam Atomic Absorption Spectrophotometers": J. Geochem. Exploration, Vol. 4, 1975.

Bristow, Q.

"A gamma-ray spectrometry system for airborne geological research" in Current Research Part C. Geol. Surv. Can., Paper 79-1C, p. 55-61, 1979.

Bouille, T.

"A model of a scientific data bank and its applications to geological data". Computers and Geosciences, Vol. 2, pp. 279-291, 1976.

Cagan, C.

"Data management systems". Pub. Melville Publishing Company, Los Angeles, California, 1973.

Cameron, E.M., Hornbrook, E.H.W.

"Current Approaches to Geochemical Reconnaissance for Uranium in the Canadian Shield": Presented at IAEA/NEA International Symposium on the Exploration of Uranium Ore Deposits, Vienna, Austria, March 29-April 2, 1976.

Canning, R.G.

"Trends in data management" Part 1, EDP analyser 9. No. 5 (May 1971): Part 2 EDP analyser 9 No. 6, June 1971.

Crain, I.R.

"Computer interpolation and contouring of two dimensional data: A Review" Geoexploration, No. 8, pp. 71-86, 1970.

"Review of Gravity and Magnetic Data Processing Systems". CSEG Journal, Vol. 8, No. 1, 1972.

Dampney, C.N.G., Funkhouser, D., Alexander, M.

"Structure of the SEG point data exchange and field formats (A proposed SEG digital recording standard) - Specification" Geophysics, Vol. 43, pp. 216-227, 1978.

Date, C.J.

"An introduction to data base systems" pp 1-2: Pub. Addison-Wesley Publishing Company, 1976.

Deutsch, D. and Fong, E.

"Characteristics of generalised data base management systems" in: Generalised data management systems and scientific information. Report of a specialist study" pp 27-48. Pub. OECD Nuclear Energy Agency, Paris 1978.

Dobrin, M.B.

"Introduction to Geophysical Prospecting": Pub: McGraw-Hill Book Company Inc., New York, Toronto, London, 1960.

Dods, S.D.

"Aeromagnetic Speed Check Program 'Speedy'" Resource Geophysics and Geochemistry Division, Geological Survey of Canada. 1974 (Unpublished).

Dreimanis, A.

"Tills: their Origins and Properties": in: - Glacial Till. Special Publication No. 12 of the Royal Society of Canada, pp. 11-49, 1976.

Fox, R.C.

"Detailed surface appraisal using geophysical methods" In: Computer Methods for the 80's in the Mineral Industry. pp. 141-154. pub. Society of Mining Engineers of the American Institute of Mining Metallurgical and Petroleum Engineers, Inc. New York, 1979.

Garrett, R.G.

"Field Data Acquisition Methods for Applied Geochemical Surveys at the Geological Survey of Canada": Geol. Surv. Can. Paper 74-52, 1974.

Granath, G.

"Heavy Minerals from Placer Deposits in the County of Nonbotten, Northern Sweden and their Provenance": Bulletin of the Geological Institutions of the University of Uppsala, N.S. Vol. 7, pp. 111-175, 1978.

Grasty, R.L.

"Airborne Gamma-Ray Spectrometry Data Processing Manual": Geol. Surv. Can., Open File Release No. 109, 1974.

Grasty, R.L.

"Uranium Measurement by Airborne Gamma-Ray Spectrometry": Geophysics, v. 40, No. 3, pp. 503-519, 1975.

_____(1)

"Applications of Gamma Radiation in remote sensing". pp. 257-276: Pub. Springer-Verlag, Berlin, Heidelberg, New York, 1976.

_____(2)

"A Calibration procedure for an Airborne Gamma-Ray Specrometer": Geol. Surv. Can., Paper 76-16, 1976.

Heiskanen, W.A., Vening Meinesz, F.A.

"The Earth and its Gravity Field": Pub. McGraw-Hill Book Company Inc., New York, Toronto, London, 1958.

Hemperl, V.E., and Ries, D.R.

"Requirements for the design of a scientific data base management system" in: Generalised data management systems and scientific information. Report of a specialist study" pp. 111, 128. Pub. OECD Nuclear Energy Agency, Paris 1978.

Holroyd, M.T.

"The Aeromagnetic Data Automatic Mapping (ADAM) System" Geol. Surv. Can. Paper 74-60, 1974.

Holroyd, M.T.

"System Guide to GECO - Automated Compilation System for Atomic Absorption Spectrometry Data": (unpublished) Geol. Surv. Can. 1975.

Hood, P.J.

"Geophysical Applications of High Resolution Magnetometers" in "Handbuch Der Physik", Pub. Springer-Verlag Berlin-Heidelberg, New York, 1971.

"The G.S.C. Aeromagnetic Gradiometer - a New Mapping Tool for Mineral Exploration": Northern Miner, Annual Review No. Nov. 1975.

Hood, P.J., Sawatzky, P., Kornik, L.J., McGrath, P.H.

"Aeromagnetic gradiometer survey, White Lake, Ontario (NTS 31F/7SE)". Geol. Surv. Can. Open File 339, 1976.

Hood, P.J., Holroyd, M.T., McGrath, P.H.

"Magnetic Methods Applied to Base Metal Exploration" Proceedings of the Symposium "Exploration 77" Ottawa, 1977.

Hruska, J.

"Current data management systems: problems of application in economic geology" Computers and Geoscience, Vol. 2, pp. 299-304, 1976.

Jeffery, K.G., Gill, E.M.

"The Geological Computer" Computers and Geosciences, Vol. 2, pp. 347-349, 1976.

Kogan, R.M., Nazarov, I.M., Fridman, Sh.D.

"Gamma Spectrometry of Natural Environments and Formations" pp. 1-47, 256-288. Pub. Israel Program for Scientific Translations Ltd: Jerusalem, 1971.

Lund, G.

"Regional Lake Sediment and Water Geochemical Reconnaissance Data, Ontario, 1977": Geol. Surv. Can. Open File No. 506, 1977.

Martin, J.

"Computer data base organisation" pp. 44-47: Pub. Prentice Hall, Inc., Englewood Cliffs, New Jersey, 1975.

Martin, G., Gordon, T.

"Data-base management systems - data models and query languages" Computers and Geosciences, vol. 3, pp. 387-393, 1977.

McConnell, R.K.

"Information Systems in Geophysics - the Management of the Canadian National Gravity Data Base"; in, "Geophysics in the Americas" Publications of the Earth Physics Branch, Energy, Mines and Resources, Canada, Vol. 46, No. 3, 1977.

McGrath, P.H., and Hood, P.J.

"An automated least-squares multi-model method for magnetic interpretation" Geophysics, vol. 38, no. 2, pp. 349-358, April 1974.

Nagy, D.

"Bouguer Anomaly Map of Canada" The Canadian Geographer, Vol. 14, No. 1, pp. 59-66, 1977.

Newton, A.R.

"Geological Interpretation of an Airborne Gamma-Ray Spectrometer Survey of the Hearne Lake Area, Northwest Territories": Geological Survey of Canada, Paper 77-32, 1977.

Nie, N.H., Hull, C.H., Jenkins, J.G., Steinbrenner, K., Bent, D.H.

"SPSS: Statistical package for the social sciences": Pub. McGraw Hill. 2nd ed. 1975, 675 p.

Olle, T.N.

"Data base and data base management" in: Encyclopedia of Computer Science pp. 389-395. Pub. Petrocelli/Chanter; New York, 1976.

Richardson, K.A., Darnley, A.G., Charbonneau, B.W.

"Airborne gamma-ray Spectrometric Measurements over the Canadian Shield: the Natural Radiation Environment" Proceedings of the Second International Symposium on the Natural Radiation Environment, Houston, Texas, U.S.A. August 1972.

Shih, K.G.

"Shipboard Computer Systems for Processing and Displaying Bathymetric, Gravimetric and Magnetic Data at Sea": Bedford Institute of Oceanography, Dartmouth, Nova Scotia, Canada. Report Series /BI-R-73-13/September 1973.

Shilts, W.W.

"Glacial till and Mineral Exploration": in: Glacial Till. Special publication No. 12 of the Royal Society of Canada, pp. 205-224, 1976.

Shilts, W.W.

"Detailed Sedimentological Study of Till Sheets in a Stratigraphic Section, Samson River, Quebec": Geol. Surv. Can. Bulletin 285, 1978.

Sutterlin, P.G., Gwynn, H.

"Computer Applications in the Analysis of Heavy Mineral Data from Tills": in: Research Methods in Pleistocene Morphology; Proceedings, 2nd Guelph Symposium on Geomorphology, pp. 109-133, 1971.

Sutterlin, P.G., Jeffery, K.G., Gill, E.M.

"Filematch: a format for the interchange of computer based files of structured data". Computers and Geosciences, Vol. 3, pp. 429-441, 1977.

Sutterlin, P.G.

"The future of information systems in the geological sciences". Presented at: Geochautauqua, University of Syracuse, Oct. 1979 (In print).

Tanner, J.G., Buck, R.J.

"A Computer-Oriented System for the Reduction of Gravity Data"; Publications of the Dominion Observatory, Ottawa, Volume XXXI, No. 3, 1964.

Tsichritizis, D.C., and Lochovsky, F.H.

"Hierarchical data base management: a survey" Computing Surveys Vol. 8, no. 1, 1976.

Wegner, P.

"Data structures" in:- Encyclopedia of Computer Science pp. 443-437. Pub. Petrocelli/Chanter; New York, 1976.

Worzel, T.L.

"Continuous gravity measurements on a Surface Ship with the Graf Sea Gravimeter". J. Geophys. Res., Vol. 71, no. 2, 1966.

Wyckoff, R.D.

"The Gulf Airborne Magnetic Meter"; Geophysics, Vol. 13, No. 2, 1948.