Summary of Discussions

The question was considered as to whether and in which way the user requirements may be met by the planned data handling subsystem.

The users were not very interested in the technical aspects but were rather concerned of the consequences to their future work. They stressed that the new system should provide at least the speed of access and the userfriendliness of the present one, but improvements were requested.

The discussion demonstrated that a number of problems of the users could partly be solved by providing more online storage capacity. It was proposed that storage and retrieval of data in the new data handling system should take place in a standardized format in a databank system. The users requested the changes of data formats connected with the reorganization of the archives into a databank should not greatly affect the existing programmed software. This concerns especially the programs used for processing data in form of model files (for definition see chapter on user requirements).

Sequences of model files are produced containing often only slight changes of the model used. For the users it is important to know which model file belongs to which model. This information is stored in descriptors accompaning each model file. It was proposed to store this information in a databank. The same arguments hold for the storage and retrieval of fields. Though agreement was reached that the format of report archives should not be changed very much by introducing the databank system some users complained about the tortuous retrieval of time series of single observations. The users stressed also that in any case the generation of data substreams should be possible because specific data subsets of the archives are repeatedly used e.g. for the test of new models. In order to avoid long access times and/or unnecessary efforts, extracted data should be stored in separate data sets. It was however agreed that all redundant data of the archives should be eliminated as far as possible.

By means of the user facilities for storage and retrieval of data, i.e. PUTDATA and GETDATA, the newly available possibilities were illustrated. They should permit storage and retrieval of data in a databank system preceded or followed by data transformations. Examples of typical data manipulation processes were presented by the Operations and Research Departments. They provided an excellent opportunity to study the proposed PUTDATA and GETDATA facilities on behalf of their efficiency and userfriendliness. After checking it was agreed upon that unsettled questions should be considered again after a more detailed specification of PUTDATA and GETDATA.

Besides, the above mentioned input/output facilities, the user should be provided with the facilities GETFILE and PUTFILE in order to store and retrieve non-standardized data formats.

The user requirements for FORTRAN callable subroutines in connection with PUTDATA and GETDATA were intensively discussed, especially the runtime access via GETDATA. A solution to this problem was postponed.

SUMMARY REPORT

1. Conceptual Models of the Data Handling System

The conceptual model presented in figure 1 is suggested as a base for following discussion:

The presented graph is a combination of the open system interconnection model representation and a representation of a data handling model. The combination is based on the fact that with the data handling system working in a distributed environement,the data handling functions are inherently combined with data communications functions. The data communications part of the model illustrates the tasks of communicating between the user's computer and a separate data handling processor by means of a layered software. The part of the model which refers to the data handling processor only reflects the relation of the data communications functions to the data handling functions in the same computer.

The system as presented is interfaced by a user programme or job using a special user interface. The interface will enable to activate any of the data handling requesters that is GETDATA or PUTDATA or GET/PUT FILE which in turn communicating with corresponding service servers in the data handling processor will satisfy the request. All the layers represented by the model will be involved in some extent. The file transfer layer in the data communications part of the model is supposed to be a service for transferring bulk data with possibly establishing files in respective computers to be used by the application level processors, i.e. service requestors or service servers. All the other network part layers of the model have the standard meaning of the open system interconnection model.

On the data handling processor's side the service server modules are combined data communications data management processors. They may use multi-transformation access to data expressed here by three layers shown as following GETDATA and PUTDATA servers on the data handling side. The layer named EXTERNAL is responsible for matching the general data model implemented in the data handling processor to the user required views. The layer named CONCEPTUAL is responsible for transformations between the conceptual, that is a general data model and specific internal representations of the data. The INTERNAL layer is responsible for transformations from internal data structures to specific system structures. It should be noted here that all of the mentioned transformation layers can be moved across the sytem even to the part on the other side of the data highway. The picture as presented here corresponds to the situation where all of the transformations are performed in the data handling processor. The file handling layer represents a system to manage hierarchical storage while the device handling

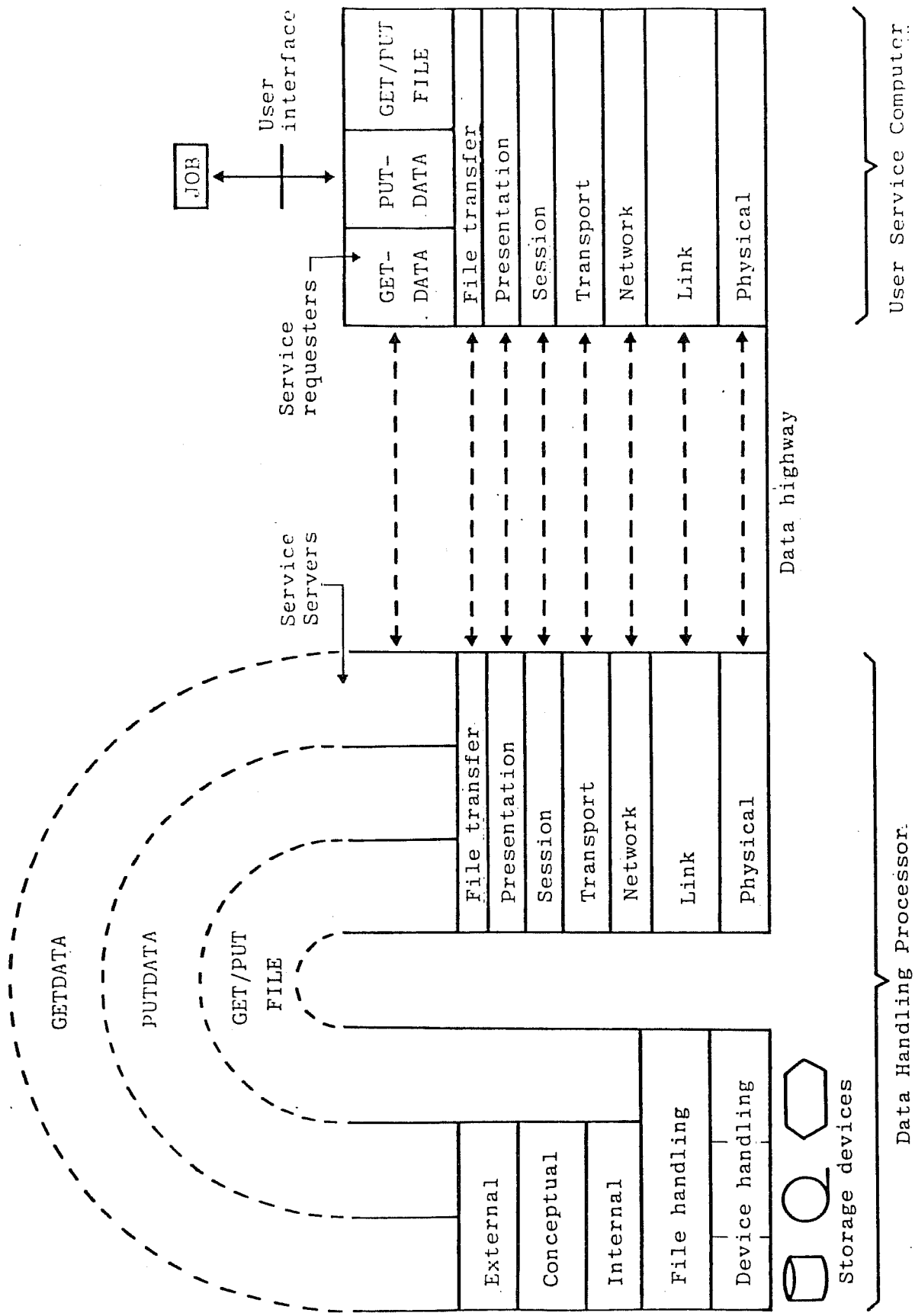

Fig. 1 Representation of combined data management/network model

layer contains the specific operating system interfaces to different storage devices. The following layers are considered as a result of the issued invitation to tender: On the data handing processor layers file handling and device handling and,within the scope of the network model,the layers file transfer, session, transport and underlying ones.

We do not believe that we are concerned with the discussion of the layers just mentioned .

2. GET/PUT FILE

GET/PUT FILE represents a very basic service to be implemented in the new system. It should allow users of the system to transfer files from a user service computer to the data handling processor and back. This file transfer may in the case of text files also include character code translations which would be situated in the model in the presentation layer of the network. In the case of transfer of binary files the transfer would be absolutely transparent to the extent that a file can only be reused on the computer from which it originated.

At present we do not have enough detailed knowledge to be able to judge how much of the network software, that will be provided by a potential vendor responding to the invitation to tender,can satisfy the need for implementation of GET/PUT FILE. The best we can hope for is that the software supplied by the vendor will implement what in the model is represented by a file transfer level. The GET/PUT FILE server on the data handling part of the model will have to interface with a level labelled FILE HANDLING in such a way as to provide the user with all the facilities that are offered by the file handling system supplied by the vendor. Implementors of the GET/PUT FILE facility should aim at hiding clumsy details of file transfer methods like submitting a job to a remote computer which then initiates file transfer from the users. However, we assume that requests for file transfers are always done explicitly by users.

3. DATABANK

The databank that is accessible by the user interfaces which are provided by GETDATA and PUTDATA comprises reports, operational archives and research archives. Conversions between the user format of data and the storage format of data will be performed as necessary. The basic entities in the databank are observations, fields and model files. Additionally, it is suggested that the field descriptors will be maintained as a database of itself,which means that the descriptors will be as well separate entities of the databank. So far the problems related to archiving of fields have been studied. For model files we suggest to take an experiment name as an attribute . Further details have to be worked out.

As concerns observations the problems of the report database have not been studied in depth. The report database should remain basically as it is at present with some changes to structure to improve flexibility and to facilitate storage and retrieval on the data handling processor. Some enhancements should be added later to allow users to define the format of the data delivered to them, the user format being independent of storage format.

We now consider the storage of fields. Information on the properties of fields are contained in descriptors. At present each field has a descriptor of 16 Cyber words, a set of fields has another three descriptor records plus one file descriptor. It is suggested to use the information available in these descriptors to construct descriptors of a new format such that they are intelligible by the data handling processor. The need for safe and flexible handling of these descriptors is best satisfied by the establishment of a database to all descriptors. Putting the descriptors in a database allows for flexibility in redefinition and access. Available products should be evaluated to find out if they are suitable for this purpose. We want to emphasise that all the data stored in the databank should be in a storage standardized format. Fields and descriptors should be stored in such a way that they are intelligible by the data handling processor. On the other hand model files are treated as atomic items that is,their contents need not be intelligible for the data handling processor.

4. PUTDATA

In what follows we deal only with fields. Similar facilities for model files and observations will have to be studied. PUTDATA converts fields and descriptors in user format to databank format for storage. The data format for the user input file of PUTDATA can be real numbers,packed integers or unpacked integers. The storage strategy is determined by descriptor contents and by PUTDATA parameters. The PUTDATA parameters contain the following information:

- Class e.g. operational, research, subclass etc,
- Local file - containing the fields and user format descriptors,
- Function - which may be ADD, REPLACE, DELETE,and
- the precision of the representation.

PUTDATA does not provide the possibility of converting user file input data from grid to spectral representation and vice versa.

5. GETDATA

GETDATA provides the facility for the retrieval of single fields or sets of fields depending on the attributes given as argument to GETDATA. GETDATA also provides some facilities for the conversion of the data representation. The

attributes that can be given to GETDATA to identify fields to be retrieved are listed in table 1. We have introduced a notion of a combined parameter which combines a level with a variable which should be especially suitable for time series retrieval. Some attributes allow values only as single items, others allow them as single items, lists, ranges of values and all possible values that can be retrieved satisfying all the other search arguments.

The sequence of parameters given determines the sequence of a result in such a way that a parameter which is given later in the sequence varies more rapidly. It is left to the implementor how he matches an access strategy with the required sequence of results. Transformations of the data representation foreseen are: Change of spectral resolution, conversion to grid format, extraction of grid areas, adaptation of mash spacing and conversion of data from packed to unpacked format and eventually also conversion from integer to real. Manipulation facilities for retrieved data by means of arithmetic operations for calculating averages should be included in GETDATA. It might be necessary to include a reformatting algorithm to produce dissemination format. Facilities similar to GETDATA will have to be provided for observations but not necessarily for model files.

6. Concluding Remarks

Due to the lack of detailed technical information, we have not been able to be very specific in many respects. On the other hand, perhaps we avoided in this way to put unnecessary restrictions with regard to software that might be available from manufacturers. The proposed concept of the data handling processor seems to be able to fulfill even future requirements of users in a very good way because it is flexible and the databank concept will certainly facilitate the data access for users. It may even be able to open up new ways and new possibilities for research.

One general note is that from the point of view of the overall computer complex modularity and future growth, it is recommended that as much of the data handling functions should be located as close to the data handling processor as possible in order to provide as much of computer independence in the system as possible.

Attribute	Attribute Values	Remarks
class	single	operational/research/ subclass/...
field form	single	spectral/grid
type	list	forecast/analysis/....
coordinate	single	sigma/pressure level
start time	single/list/range	
timestep	single/range/list/all	
variable	single/list/all	T/...
level	surface/single/list/all	500mb/...
combined parameter	single/list	200mb T

Table 1 List of Attributes