A Logical Structure of the Computer System

TAKAYA ISHIDA*

1. *Introduction*

It is important to take the computer system as just one type of the information processing system. There are a great number of organizations for information processing in the world, composing various kinds of information processing systems. They usually work dependently or independently with each other making larger information processing systems, which will constitute further larger information processing systems. Thus it will be impossible to design the truly desirable computer system without regard to its surrounding information processing system. In spite of this fact the computer system has been so far designed without global view of information processing system, thus causing difficulties such as great discontinuity between a computer system and men or other computer systems. Some papers have been made about a logical structure of the computer system[1]~[3], but they were all mainly on this basis, namely never beyond the computer system.

We define here a set of information as a File and a container of a File as a Volume, thus regarding all the computing resources as Files or Volumes. And we intend to formalize the logical structure of the computing system on this new concept of File and Volume. Then such controversial specific systems as multi-processor system, multi-user system, man-machine system or multi-level storage system will appear as just some variations of the original computer system.

2. *Information Space*

The information space provides the basis of the information processing system. Hence, we should begin with defining the information space when discussing about the information processing system.

The previous definition of File, a set of information, implicitly means that a collection of Files is also a File. On the other hand, the information space may be considered to be composed of information. Thus in a sense we might regard the information space as a File.

However only by the notion of File we can't completely define the information space because it never implies where it exists or how it can be accessed.

This paper first appeared in Japanese in Joho-Shori (the Journal of the Information Processing Society of Japan), Vol. 11, No. 5 (1970), pp. 260-268.

* Kamakura Works, Mitsubishi Electric Corp.

Consequently, in order that a File may exist as acceptable one, a field of access should be given to it. We name the field of access of a File the Volume of the File. A Volume is conceptually a container of a File and in this sense a File provides the Status of its Volume. Especially, the Volume which has no File is called to be in the Empty Status. The definition of Volume in connection with that of File implies that a collection of Volumes also makes a Volume. Then we may regard the information space as one big Volume consisting of many smaller Volumes.

In summary we consider that all Files are accessible only through their Volumes.

2.1 Volume Definition

The definition of Volume will be further detailed in this section. A Volume can be uniquely located in the information space by being specified three attributes: Physical Characteristics, Physical Location and Logical Characteristics. In other words, a Volume should be represented by a subset (V) of the Volume Space being the product space $(X_c \times X_l \times Y_c)$ of the Physical Characteristics Space (X_c), the Physical Location Space (X_l) and the Logical Characteristics Space (Y_c). Moreover, when we denote the map from the $X_c \times X_l \times Y_c$ space to the $X_c \times X_l$ space by m_1,

$$m_1 : X_c \times X_l \times Y_c \longrightarrow X_c \times X_l ,$$

the physical domain covered by a Volume V can be represented by $m_1(V)$. An example of a Volume is illustrated in Fig. 1.

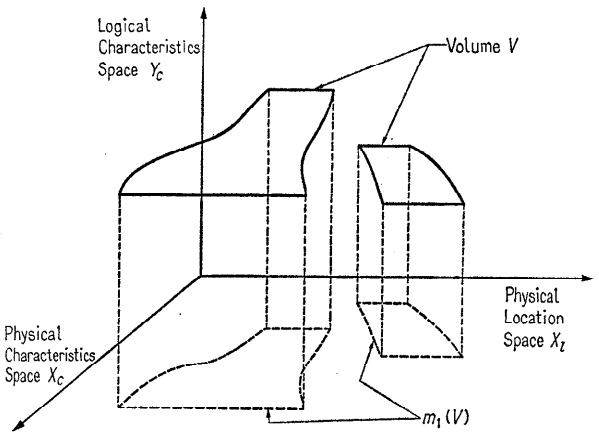


Fig. 1. Volume Space.

The two physical attributes, Physical Characteristics and Physical Location, indicate the whereabouts of a Volume in the Physical Media Space mentioned later.

The logical attribute, Logical Characteristics, specifies where the Volume

exists in the logical structure of Volumes as well as its capability concerning File accessing as shown below:

(1) Whether its File can be entered from other Volumes or transferred to other Volumes.

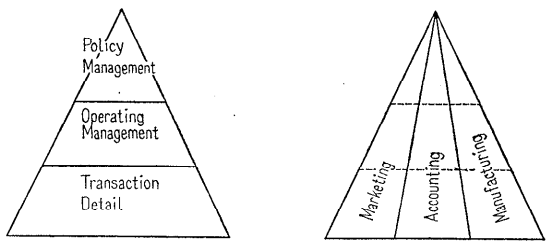
(2) Whether its File can be executed by other Volumes as a procedure or the Volume can be an executor of Files in other Volumes.

(3) Information filtering, that is, picking up only valid input or output information for the Volume.

(4) File organizing, that is, giving inter-relationships among input or output information for the Volume.

(5) Information representing, that is, specifying the form in which each input or output information for the Volume will be represented.

In order to illustrate the concept of Volume, let us look at the utilization of Files in an enterprise. The set of all information necessary for management of a certain enterprise may be considered to constitute the File of the enterprise. But all the information is not always necessary for all the members in the enterprise. Some members may require a part of the File and others may need other parts of the File, sharing some information each other. Hence, the same File should be organized quite differently for each class of members (for example, horizontally or vertically as shown in Fig. 2[4]. In this case one unique Volume will be given to each class of members, satisfying all their needs though sharing the same File.



(a) Horizontally Organizing (b) Vertically Organizing

Fig. 2. Examples of File Organizing in MIS.

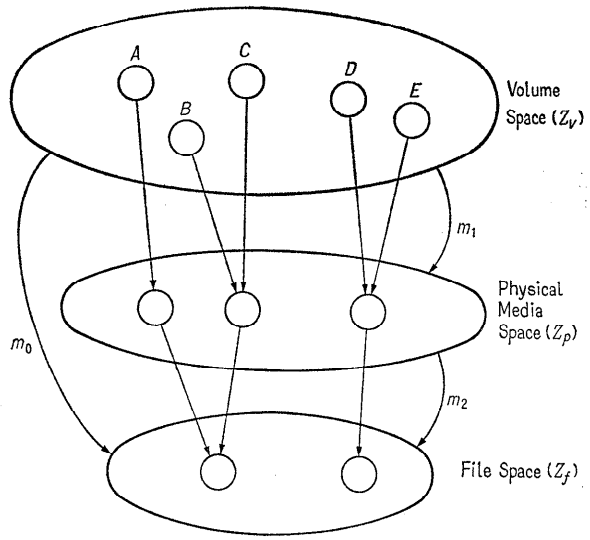
Now let us name all the set of information the File Space (Z_f), the $X_c \times X_l \times Y_c$ space the Volume Space (Z_v) and the $X_c \times X_l$ space the Physical Media Space (Z_p). In addition, we define maps, m_0, m_1 and m_2 among these spaces as follows (Fig. 3):

$$m_0 : Z_v \to Z_f, \quad m_0(Z_v) = Z_f$$

$$m_0 = m_1 \cdot m_2, \quad m_1 : Z_v \to Z_p, \quad m_1(Z_v) = Z_p$$

$$m_2 : Z_p \to Z_f, \quad m_2(Z_p) = Z_f$$

If $m_0(V_n) = m_0(V_m)$ $(n \neq m)$, then Volumes V_n and V_m share a File. Further-



Where, Volume A and Volume B, and Volume A and Volume C are both in the relation of Logical File Sharling, while Volume B and Volume C, and Volume D and Volume E are both in the relation of Physical File Sharing.

Fig. 3. Mapping from Volume Space to File Space.

more, if $m_1(V_n)=m_1(V_m)$ they are called to be Physical File Sharing, and if $m_1(V_n)\neq m_1(V_m)$ they are called to be Logical File Sharing. In the former case, changing the Status of one Volume results in changing the Status of the other Volume.

2.2 Logical Structure of Volumes

The Volume which provides the entire information space of a given information processing system is called the highest rank Volume or the rank 0 Volume of the system. The rank 0 Volume will be usually composed of one or more exclusive rank 1 Volumes. Generally speaking, a rank i Volume except terminal Volumes is composed of one or more exclusive rank $i+1$ Volumes ($i=0, 1, 2, \cdots\cdots$). Terminal Volumes are those which have no composing Volume.

Now we can classify all these Volumes according to their ranks. Let us denote the set of rank i Volumes by S_i and the map from S_i to S_{i-1} by μ_i,

$$\mu_i : S_i \rightarrow S_{i-1} \quad (i=1, 2, \cdots\cdots) ,$$

then the logical structure of Volumes will be expressed by $S_0, S_1, S_2, \cdots\cdots S_N$ and $\mu_1, \mu_2, \cdots\cdots \mu_N$, where N is the maximum rank number in a given information processing system. An example of this hierarchical structure of Volumes is shown in Fig. 4.

So far, without the concept of Volume, the file structure has been represented by using the single map often called file directories which might be considered the forced merger of two essentially different maps, m_0, m_1, m_2 and $\mu_1, \mu_2, \cdots\cdots$

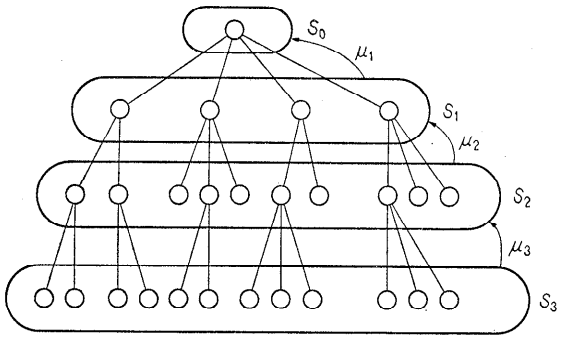


Fig. 4. Logical Structure of Volumes (in case of $N=3$).

μ_N in our sense. Consequently, in such a system file management is a very troublesome work. Still more, file sharing should be represented by introducing special means such as Link [5], which, in case of erroneous deletion of some entry in a file directory, might cause an endless looping for a file search process [5].

With our concepts, when a File is shared by two users two Volumes will be prepared, each of which is for each user, while the maps, m_0, m_1, m_2 designate the relationship of their File sharing.

2.3 Physical Media Space

A Volume becomes actually existent only when specific domains in the Physical Media Space are allocated for it.

The Physical Media Space consists of subsets of physically continuous points. Let us name each subset a Physical Volume (P), which could be freely specified by the designers of the system. For instance, for a computer system one memory module will be usually taken as one Physical Volume but for a particular computer system this may be too large to build up a more elaborate system, where each memory bank which composes a memory module may be chosen as a Physical Volume.

Suppose an information processing system is constructed upon M Physical Volumes (P_i, $i=1, 2, \cdots\cdots, M$), then the following relationship exists among a certain Volume V and Physical Volumes in the system:

$$m_1(V)=\bigcup_{i=1}^{M} p_i, \quad p_i \subset P_i, \quad P_n \cap P_m=\phi \quad (n \neq m)$$

where ϕ is the empty set.

Among these Physical Volumes there will be many routes to physically connect two given Physical Volumes. This connecting relationship can be expressed by using the Physical Interface Volume which is hypothetically defined between two Physical Volumes. That is, the physical structure of a system is represented by one central Physical Volume, other peripheral Physical Volumes and their Physical Interface Volumes, where the central Physical Volume is arbitra-

rily chosen from the system (Fig. 5). Here, information is first transferred from one Physical Volume to its appropriate Physical Interface Volume and after that the information is transferred to the destined Physical Volume (Fig. 6).

One example of a Physical Interface Volume is a so-called Logical Channel [6]. Another one is a set of magnetic tape units which provide many routes to connect a specific magnetic tape reel to the CPU (central processing unit).

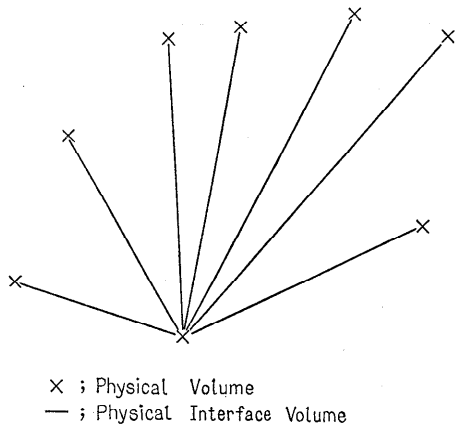


Fig. 5. Connecting Structure of Physical Volumes.

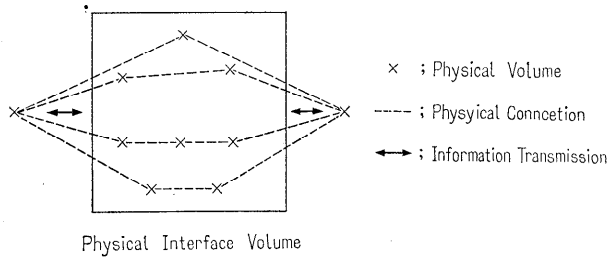


Fig. 6. Information Transmission between two Physical Volumes.

3. *Information Processing*

Information processing would be uniquely specified by its Procedure and its object Files. However in reality, however, the Procedure itself should exist as a File and there must be something which executes the Procedure. Thus information processing itself should be represented by a Volume which is particularly called an Information Processing Volume (for short, IP Volume).

The whole of an information processing system corresponds to one IP Volume. Similarly for the sub-works of one IP Volume other IP Volumes may be allocated. In this way the information processing system will be constructed by a recursive structure of IP Volumes. Using the concept of IP Volume we may say that information processing is the work to assign it an IP Volume and

change its Status to the desired Status.

The concept of IP Volume will be more general than that of so-called Logical Processor and Address Space in the point that the structure of IP Volumes does not essentially distinguish processors and data files, regarding each of these as just one component of the homogeneous recursive Volume structure.

3.1 Structure of IP Volume

Let us call the set of information which actually represents a Procedure a Procedure File and the File upon which the Procedure acts an Object File.

An IP Volume may be considered to be composed of 4 Volumes each of which will usually further comprise many lower rank Volumes (Fig. 7). The parts which these 4 Volumes play in the work of information processing are as follows:

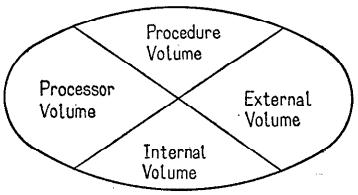


Fig. 7. Internal Structure of IP Volume.

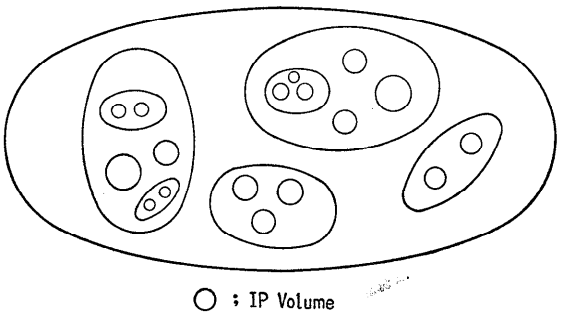


Fig. 8. Logical Structure of Information Processing.

(1) Processor Volume: executes the Procedure.
(2) Procedure Volume: keeps the Procedure File.
(3) Internal Volume: plays the role of a chopping board for processing the Object File.
(4) External Volume: keeps the Object File.

As we mentioned earlier, an IP Volume may dispatch smaller IP Volumes during its work, entrusting parts of its information processing to them. Then those IP Volumes become the components of the External Volume of the dispatching IP Volume.

For example, a computer system makes up an IP Volume. Suppose the

computer system is shared by many users (Principals [1]), this IP Volume will be divided into smaller IP Volumes each of which corresponds to a virtual computer system for each user. Furthermore the IP Volume for each user will be divided into IP Volumes for Jobs, the IP Volume for each Job into IP Volumes for Tasks, and the IP Volume for each Task into IP Volumes for lower level Tasks (Sub-Tasks).

3.2 Operation of IP Volumes

As mentioned above, information processing is accomplished by the cooperation of multi-rank IP Volumes. Thus next we have to look into the operation of IP Volumes.

There are two means for information communication between two IP Volumes: File Sharing and System Interrupt.

Using the means of File Sharing, one IP Volume places communication information into its sending Volume and after that the other IP Volume accepts the information from its receiving Volume, where the sending Volume and the receiving Volume are in the relation of Physical File Sharing.

On the other hand, one IP Volume can send an appropriate signal to other IP Volumes by issuing a System Interrupt which is any kind of interrupt from one IP Volume to other IP Volumes.

Next let us discuss about the States of IP Volumes. There will be two States: Running and Blocked. If the Procedure of an IP Volume is under execution, the IP Volume is called to be in the Running State. Otherwise, the IP Volume is called to be in the Blocked State. The Blocked State would be further divided into following two types:

(1) Time Blocked
(2) Space Blocked

The Time Blocked State is caused when the IP Volume has to wait for any kind of synchronization with other independently running IP Volumes, e.g., when awaiting the completion of an input/output operation, the completion of a Sub-Task, certain time elapsing or the release of a Physical File Sharing Volume.

The Space Blocked State is caused when the IP Volume cannot get large enough demains in the Physical Media Space to keep it in the Running State, which will never be caused in the system having ideally large Physical Media Space, while the Time Blocked State is due to inevitable causes.

In addition we may define the degree of the Blocked State as follows: the more causes are responsible for the Blocked State, the stronger the Blocked State is.

The System Interrupt can be classified into two types: the Block Interrupt and Wake-up Interrupt [2], using the above concept of State. The Block Interrupt changes a Running State IP Volume into the Blocked State and a Blocked State

IP Volume into the stronger Blocked State. On the other hand the Wake-up Interrupt has no effect for a Running IP Volume, but for a Blocked IP Volume it changes the State into Running or decreases the degree of the Blocked State.

3.4 Volume Creation and Deletion

A Volume is created or deleted by its higher rank IP Volume. We name the procedure of Volume creation Volume Open and the procedure of Volume deletion Volume Close. Volume Open would be divided into following three types:

(1) Create a Volume in the relation of Physical File Sharing with an existent Volume, in the current IP Volume.
(2) Create a Volume in the relation of Logical File Sharing with an existent Volume, in the current IP Volume.
(3) Create an entirely new Volume in the current IP Volume.

Similarly, Volume Close may be classified as follows:

(1) Delete the specified Volume from the current IP Volume as well as requesting its higher rank IP Volume to create a Volume in it which is Physical or Logical File Sharing with the deleted Volume.
(2) Delete the specified Volume from the current IP Volume as well as requesting its higher rank IP Volume to delete Volumes from it which are Physical or Logical File Sharing with the deleted Volume.
(3) Simply delete the specified Volume from the current IP Volume.

In particular, the generally used term Cataloging [6] corresponds to the case where we request the IP Volume of a computer operating system to create a Volume in it using type (1) Volume Close.

3.5 Function of Processor Volume

Finally, let us consider the functions of the Processor Volume. The Processor Volume plays the central role of information Processing of its IP Volume. The essential functions of the Processor Volume may be:

(1) Control the flow of execution of its Procedure.
(2) Change the Status of its IP Volume according to its Procedure.
(3) Issue System Interrupts to other IP Volumes.
(4) Accept System Interrupts from other IP Volumes.
(5) Open or Close the Volumes (including its lower rank IP Volumes) in its IP Volume.
(6) Control the processing of its lower rank IP Volumes.

Note the Processor Volume is a kind of IP Volume. Hence, suppose we are given a certain Processor Volume, then using it we will be able to build up an IP Volume which also has the power of the Processor Volume but is more powerful than the original one. Repeating this process we may construct more

and more competent Processor Volumes, thus finally obtaining the desired IP Volume.

4. *Conclusion*

We are of the opinion that the computer system should be information system free in order to be a consistent component of a large information processing system, getting over so far emphasis such as device independent or configuration free on the limited notion of the computer system.

We believe the discussion in this paper would supply the competent tool for designing our aiming ideal computer system.

Acknowledgements. The author is particularly indebted to Kazuya Shimamura, Kazuma Ban and Shoichi Iikawa of Mitsubishi Electric Corp. for their encouragement and support.

References

[1] Dennis, J. B. and E. C. Van Horn, Programming Semantics for Multiprogrammed Computations, *Comm ACM 9*, 3 (Mar. 1966).

[2] Saltzer, J. H., Traffic Control in a Multiplexed Computer System, *MAC-TR-30* (Thesis), M. I. T., Cambridge, Mass. (July 1966).

[3] Lampson, B. W., A Scheduling Philosophy for Multiprocessing Systems, *Comm ACM 11*, 5 (May 1968).

[4] Head, R. V., Management Information Systems, A Critical Appraisal, *Datamation 13*, 5 (May 1967).

[5] Daley, R. C. and P. G. Neumann, A General Purpose File System for Secondary Storage, *Proc. FJCC* (1965).

[6] Clark, W. A., Functional Structure of OS/360 Part III, Data Management, *IBM Systems Journal 5*, 1 (1966).