BOCHMANN, GREGOR V.

--Overview of protocols in distributed videotex systems.

Overview of Protocols in Distributed Videotex Systems

Gregor V. Bochmann

Département d'informatique et
de recherche opérationnelle
Université de Montréal

May 1981

This report is prepared under contract 20SU.36100-0-9506 for the Department of Communications, Canada.

1. Introduction

This study gives an overview of high-level protocols for distributed videotex systems. We assume that the reader is familiar with videotex systems and possible applications. A review of present videotex systems [Ball 80] shows that all existing systems started out as a centralized system on a single host computer. Here we take the view that the future videotex systems will be partly implemented as distributed systems (see for example [Ball 80]), and that interworking between different systems (called "subsystems" in this study) is desired.

Different scenarios of distribution are considered in section 2 of this report, and their impact on the required protocols is discussed in the subsequent sections. We put the discussion into the framework of the OSI Reference Model [ISO 81] which gives some guidelines for the development of computer communication protocols. We concentrate in this report on the higher-level protocols for videotex applications. At this point, it seems too early to propose specific protocols in this area. Therefore the discussion is kept on a level addressing the functions to be provided by the protocols.

2. Some scenarios of distribution

In this section we discuss some possible scenarios which characterize different forms in which functions of a videotex system may be distributed over different computers. This discussion gives the framework for the consideration of communication protocols in the following sections. We note that similar scenarios for distribution are discussed in [Tomp 81]. Some considerations of distribution in videotex systems may also be found in [Chit 79] and [Ball 80].

2.1. Videotex system functions and components

A videotex system contains a number of different functions which, in most present systems, are realized in a central videotex database computer. We consider in the following that these functions are distributed over different components of the videotex system. Each of these components, in turn, may be implemented in one, or multiple computer systems. These computers are connected through a local or long-distance communications network. For simplicity, we consider in this subsection each of these functions or components separately, as if each would be implemented in a separate computer.

2.1.1. Videotex access machine (VAM)

As shown in figure 1, the user terminals (UT) are connected to the VAM through which they access the different videotex services. The functions of the VAM (user identification, service selection, accounting, possibly keeping some user context during a videotex session andor between sessions, possibly adapting various terminal andor database communication standards, etc.) are described in more detail in [Ball 80].

2.1.2. Different kinds of databases

[Tomp 81] identifies three kinds of database structures that seem to be suitable for videotex information retrieval:

(a) Page-oriented databases which correspond to the present videotex databases, such as Prestel or Telidon.

(b) Record-oriented databases, the data structure of which corresponds to relational, hierarchical or network databases.

(c) Keyword-oriented databases, which allow retrieval of information guided by user provided keywords, similar to some existing bibliographic information databases.

We note that databases of different kind may be combined, and implemented on a single computer.

2.1.3. Videotex updata machine (VUM)

Similar to the VAM, this machine provides access to the databases for the update operations by the information providers (IP). The typical functions of a VUM include the functions of a VAM and some of the following:

(1) update facilities for selected parts of the database(s),

(2) automatic checking of consistancy constraints of the updated information, (such as presence of cross-referenced

pages, etc),.

(3) automatic checking of consistancy constraints involving several databases (if for example the keyword database contains pointers to pages in the page-oriented database, see also [Tomp 81]).

2.1.4. System management

Under this heading we consider functions necessary for the management of the videotex system, such as monitoring the system utilization, monitoring the state of the different system components, configuration control, etc.

Note: The 3-rd party database shown in figure 1 is not mentioned in the above list since it will in general encompass several of the above functions, in particular the functions of the databases, the VUM, and system management.

2.2. Videotex subsystems and distribution

The following subsections discuss different, partly distributed scenarios which we consider realistic for the not too distant future (in fact a scenario of 3-rd party databases is already reality now). The kind of distribution we consider involves different system components at different geographical locations, as well as "local distribution" where different system components are connected through a local network or some other communication facility.

We first consider a partitioning of the videotex system into several "subsystems". We define a "videotex subsystem" to be a self-contained part of the overall videotex system providing videotex service to a given group of users, possibly within a particular geographical area. For example, the Prestel computers providing videotex service in the different areas in Great Britain may be considered different subsystems of the Prestel system.

A typical structure of a videotex subsystem is shown in figure 1. This figure also indicates some possible interaction paths between different videotex subsystems. The situation will not always be as simple as the present Prestel system. As far as the distribution of the information is concerned, several cases may be distinguished:

(a) Multiple copies of the same information in different subsystems (not necessarily complete duplication, as in the present Prestel system).

(b) Partitioned information, i.e. some amount of information (of similar kind, or logically connected) is partitioned over different subsystems, usually according to the access pattern (for example Ottawa entertainment information is more often accessed from Ottawa than from Montreal).

(c) Local interest information only available in a given subsystem.

In addition to the geographical distribution leading to the concept of subsystem, we consider in the following different scenarios for the distribution within a given subsystem.

2.2.1 No distribution within a subsystem (see figure 2)

All functions of a subsystem are implemented in one computer system, which clearly may involve a number of front-ends for the communication with the UT's, and a number of back-ends for handling the databases.

2.2.2 Multiple VAM's (see figure 3)

Several VAM's handle different groups of users. All VAM's communicate with one database computer which contains the storage, update and management functions. The VAM may contain a partial database used as a "cache". For a more detailed discussion see [Ball 80].

2.2.3 Separate databases (see figure 4)

We may consider the following possibilities: (a) databases of different structure are implemented on separate computers (for example, a keyword database on a special-purpose high-performance computer). (b) a database of a given structure (for example page-oriented) is partitioned onto several computers. Such a possibility may be considered for organisational, or performance reasons.

If in these cases the VUM should enforce some consistancy constraints that involve several of the database computers, it could be reasonable to implement the VUM on yet another computer.

2.3 Distribution of authority

We have assumed so far that the videotex system, consisting possibly of several subsystems, was administered by one single autority, as is the case presently in Great Britain. However, it must be assumed that in general different videotex subsystems belong to different administrations. This is the case when international interworking between the videotex systems of different counties is considered, or within a country with competing communication and information providers, such as Canada. We make the assumption that interworking between videotex systems belonging to different autorities is desired.

Another important case of distribution of authority are 3-rd party databases. As indicated in figure 1, these are databases that are administered by an independent organisation and are accessible by the users from the videotex terminal. In some cases these 3-rd party databases are databases that exist primarily for some other purpose, and provide (as a secondary purpose) an interface for videotex users. In other cases, the database is primarily developed for providing information for videotex users. In these cases, there does not seem to be any fundamental difference between such databases and the databases that are part of a videotex subsystem (for example in the distribution scenario 2.2.3).

3. The protocol architecture of Open Systems

A Reference Model for Open Systems Interworking is being developed by the ISO TC97 Subcommittee 16. This model defines a number of protocol layers for so-called Open Systems (see for example [ISO 81] or [Zimm 80]). The present work of the standardization committee concentrates on the development of service and protocol specifications for the higher layers of this architecture. The purpose of this work is to obtain protocol standards that make the interworking of heterogeneous computer systems easier. It seems therefore reasonable to try to use any results of this standardization efford for the communication of the different videotex subsystems and 3-rd party databases, and possibly also for the communication between the different components of a subsystem.

We cannot give here a complete explanation of the OSI reference model, its communication services and protocols. We only give a very short overview, and also refer to some other related work in progress on protocol development which seems to be relevant to the problems considered in this report.

3.1 Overview of OSI

The reference model distinguishes the following seven system layers:

(1) physical layer

(2) link layer

(3) network layer

(4) transport layer

(5) session layer

(6) presentation layer

(7) application layer.

For the lower three layers a number of protocols have been defined by ISO and CCITT (X.20, X.21, X.25, etc.) which could be adopted for videotex networks. A specification of a transport protocol is close to be finished. The work on the higher layers is less advanced. Many of the problems for videotex networks lie in these areas. Therefore we concentrate our discussion in the following on the higher-level protocols.

Different working groups of ISO TC97SC16 work presently on a file transfer protocol, a virtual terminal protocol, and the problems of system management (which are usually allocated to the application layer, in lack of any more specific place in the OSI model). A "database access protocol" would also be needed for certain scenarios of distributed videotex.

3.2 Other developments

We think that the following developments should be taken into account in the development of protocols for distributed videotex systems.

3.2.1 Teletex protocols: CCITT has developed protocols for the Teletex service. The following aspects of these protocols are particularly interesting for our purposes:

(1) The lower layer protocols provide the same transport service over circuit and packet-switched networks. They also can be used for the interworking of the different videotex system components connected to circuit-switched and packed-switched networks.

(2) The session and so-called "document" protocols provide a kind of transfer service for text files.

(3) The presentation conventions (coding of certain characters and page formatting functions) are not completely compatible with the corresponding conventions used in videotex systems.

3.2.2 OSI protocols over Cable TV networks: A recent study [Cave 81] discusses how the principles of OSI could be used over CTV networks for providing an "open" environment through which a variety of different information services could be accessible to the terminal users. The study concentrates on the lower layers of the OSI model. For these layers the CTV network provides a particular environment. For the higher layers, however, there does not seem to be a difference between the context of that study and our report.

3.2.3 Priority for 3-rd party databases: In the development of videotex in Germany, the provision of access to 3-rd party databases has a high priority. It is planned to use for the communication with a 3-rd party database the protocols developed for OSI, as far as they are applicable (for example, the Virtual Terminal Protocol). At present, however, interim protocols are used until ISO will have defined the corresponding protocols for OSI.

3.2.4 Interworking with bibliographic databases: The National Library of Canada has created a Task Group on ComputerCommunication Protocols for Bibliographic Data Interchange with the task of developing appropriate protocols within the next two years. The access to bibliographic databases from videotex terminals is one of the problems addressed by this group. Such databases may play the role of 3-rd party databases within the videotex network. Another problem addressed is a user interface for searching. Such an interface may also be useful for other videotex applications, as discussed in [Ball 81].

4. Protocols for different distribution scenarios

In this section we give an overview of protocols in distributed videotex systems. Using the framework of the OSI Reference Model, the protocols for the different scenarios mentioned in section 2 are considered. Some comments on the needs for standards are also given.

4.1 The OSI reference model applied to distributed videotex systems

In this first subsection we consider the application of the OSI Reference Model to the distributed videotex scenarios. We think that these considerations somewhat clarify the issues involved, but do not necessarily imply that standard protocols should be used within the videotex system. It is also to be noted that the higher-level protocols for the OSI Reference Model will probably not be ready before some years .

As shown in figure 5, and discussed in more detail in [Tomp 81] and [Ball 81], the interaction of a user or IP, respectively, may be envisaged as proceeding through a "user agent", or "update agent", respectively. In the case of the user interaction, the user agent implements a user-friendly interface which interacts with the user, and it translates this interface into a database-oriented interface which is provided by the database system.

The interface between the user agent and the UT is, in OSI terminology, a virtual terminal interface. For videotex, and in particular Telidon, it should support graphic facilities as provided by the videotex terminals. It is therefore through the use of this terminal protocol that the friendly user interface is realized by the user agent.

The interaction between the user agent and the database uses, in the OSI terminology, a file or database access protocol. While present work of SC16 is more focused on file transfer protocols, is seems that database access protocols would be a subsequent point of study. A file transfer protocol would be typically used for exchanging a file of videotex information, including a larger number of pages, or a keyword directory, etc., between several databases, possibly new updates. A database access protocol would be used for accessing individual information elements (such as a single display page), making a keywork search, or making an limited update of the database.

The database access protocol is typically transaction oriented, i.e. different requests (in particular retrieval requests) are handled by the database as individual, unrelated operations. (For a discussion of transaction-oriented database access for videotex, see [Ball 80]). Two parts of a database access protocol may be distinguished: (a) the retrieval protocol, and (b) the update protocol. It is clear that for the user agent only the retrieval protocol is required, except for applications in which the user updates the database. In many cases such applications will be implemented in a 3-rd party database. (We note that applications where the user updates the database, or where the user interaction with the system results in some observable action, such as the delivery of a pizza ordered through the videotex system, are sometimes called "transactional applications" [Ball 81]. This use of the term "transaction" is different from its use above, which corresponds to the usual terminology of database technology [Date 77]).

The interactions of the system management module shown in figure 1, with the other parts of the videotex system clearly represent some kind of management protocol. The work of SC16 in this area is only in its beginning so that not much can be said about its applicability to videotex systems.

4.2 Interfaces and protocols for distributed videotex systems

The distinction between "interface" and "communication protocol" is not always clearly made. An interface is usually a set of conventions that determine the possible interactions of two (or more) system modules when they are "close" together. When additional problems are involved due to implementation in different system components possibly geographically distributed, communication protocols are introduced in order to provide some equivalent primitives for the interaction between the two modules.

4.2.1 Lower-level protocols

The lower-level protocols to be used for the communication between the different system components shown in the figures 1 through 4 depend strongly on the distance between these parts and the data transmission medium used. Standard protocols, over dedicated circuits, or circuit- and packet-switched networks, such as X.20, X.21, X.25, X.75, and the lower layers of Teletex could be used for the communication between the system components to the left of the figure. The IP terminals (IPT) and UT's usually access the system through the telephone network using a special asynchronous modem, or through a CTV network (for a discussion of OSI oriented protocols over CTV networks see [Cave 81]). Additional access facilities to videotex systems through the usual access conventions for host computers could also be foreseen.

We assume that in all cases the lower-level (up to, and including layer 4) protocols provide a uniform transport service which is used by the higher-level protocols discussed below.

4.2.2 Higher-level protocols

We consider now each of the interfaces A through M shown in figure 1, and discuss the higher-level communication protocols that could be used when this interface extends between distributed system components.

In the following discussion, the notion of a "user agent", and an "update agent" are important. As shown in figure 5, these agents translate the computer-oriented database access and update interfaces into the human-oriented interfaces seen by the user and IP, respectively. (See also section 4.1 and [Tomp 81]).

4.2.2.1 The UT-VAM interface (A)

This is a terminal interface. In the existing videotex systems, the protocols used here are sometimes quite dependent on the terminal technology and the underlying transmission medium (Prestel, alpha-mosaic graphics, etc.). The Telidon standard represents a more terminal and transmission independent approach, which is preferable since it simplifies the adaptation to future developments in terminal and transmission technology.

It is also through this interface that the user interface is presented. A discussion of this problem area may be found in [Ball 81]. Here we only assume that the adopted terminal protocol supports the functions needed for realizing an appropriate user interface. The importance of alpha-numeric input by the user for many applications is stressed in [Ball 81].

In the case that the terminal is not directly connected to the VAM (see also sections 4.2.2.2(a) and 4.3), or that a non-videotex terminal is used (interface A2 in figure 1) it may be useful to adopt a standard Virtual Terminal protocol for the communication between the VAM and the terminal. Even if the physical terminal used in this context does not support graphics (and therefore most graphic information would be lost to the user), this would allow the access to the videotex system from a terminal which is normally used for interworking with other (Open) systems.

4.2.2.2. The VAM - DB interface (B)

The protocols discussed here may be used within a given (distributed) videotex subsystem, for the interaction of a VAM module of one videotex subsystem with the database of another subsystem, or for the interaction with a 3-rd party database. We distinguish the following two cases:

(a) The user agent is implemented in the database: In this case the VAM is transparent during the interaction between the user and the database. We note, however, that the VAM may play a role during the sign-on of the user, during the selection of the database by the user, and may implement the user agent for another database. The protocols would be very similar to those mentioned under point 4.2.2.1.

(b) The user agent is implemented in the VAM:
As pointed out in section 4.1, in this case the protocol could be characterized as a database access protocol. the functions of this protocol are discussed in more detail in section 5.2.

(c) The user agent is implemented in an intelligent terminal:
In this case the VAM-DB protocol could be the same as in case (b), and even the UT - VAM protocol would be similar.

4.2.2.3. The IPT - VUM interface (C)

Like the UT - VAM interface, this is a terminal interface. The adopted protocol must provide the same functions as the UT - VAM protocol, and in addition provide suitable functions for data entry and updating (for more details, see section 5.1).

4.2.2.4 The VUM - DB interface (D)

The here adopted protocols must provide the functions of the database access protocols, mentioned under point 4.2.2.2, and in addition suitable functions for the updating of the database.

As in the case of the VAM - DB interface, we have to distinguish different cases depending how much processing is done in the VUM compared to the DB. We consider the following two cases.

(a) The update agent is implemented in the database: The situation is similar to case (a) of 4.2.2.2.

(b) The update agent is implemented in the VUM: The adopted protocols must provide the functions of the database access protocol, mentioned under point 4.2.2.2(b), and in addition suitable functions for the updating of the database(s). In the case of multiple databases within the same videotex subsystem, the update agent must include the necessary updating synchronization rules to keep the different parts of the information base consistent. (How much consistency is needed depends on the particular application, and the nature of the information).

4.2.2.5 VAM accessing data from a distant videotex subsystem (I)

The protocols used for this purpose would be similar to those mentioned under point 4.2.2.2. Different scenarios may be considered depending on the extent that the "user context" is handled by the originating VAM (see also [Ball 80]).

4.2.2.6 VUM obtaining information from a distant videotex subsystem (K)

It seems that the information used to update a database is often obtained in the form of a copy from another (possibly non-videotex) database. An extreme example are the present Prestel databases, which are all copies of a "master". Such updates often involve large amounts of data. Therefore a file transfer protocol would be appropriate for such applications.

In other cases, only selective updates may be required. If the information to be updated is determined by the local subsystem, the VUM may use a database access protocol (as mentioned under point 4.2.2.5) for obtaining the new information. If the information to be updated is determined by some "master" subsystem which sends these updates to the other subsystems involved, again a file transfer protocol could be used for sending the individual updates in a batch (as the present Telidon IP terminals do).

4.2.2.7 Interactions of the system management module (M)

Some typical functions to be provided by the protocols involving the management module are discussed in section 5.3.

4.2.2.8 Interactions between VAM's of different subsystems (F)

Communication between the VAM's of different videotex subsystems may be implemented for the following purposes:

(a) Allowing the user access to remote videotex subsystems, in the case that direct access as under point 4.2.2.5 is not implemented.

(b) Exchanging directory information, keywords, etc.

(c) Accounting.

4.2.2.9 Interaction between VUM's of different subsystems (G)

Communication between the VUM's of different subsystems may be implemented for update coordination of the subsystems.

4.3 Needs for protocol standardization

In this section we discuss the need for standards for the interworking of the different components of a distributed videotex system. In general, there seem to be two main reasons for standardization of communication protocols:

(a) to simplify the interworking of differnt systems, and

(b) to take advantage of standard communication hardware and software that is widely used, well understood and inexpensive.

For the higher-level protocols considered in this report, point (b) may be relevant in the future when OSI standards will have been established and are widely used. For now, we concentrate on point (a). This point is mainly relevant when the interworking systems, or system components, belong to different organizations and authorities. In the case that the different components belong to the same organization, it is easily conceivable that non-standard communication protocols between the components are specially designed for the particular requirements of the videotex system in question.

In the following subsections, we discuss the needs for standardization for some of the scenarios mentioned in section 2.

4.3.1 Interworking of independent videotex subsystems

We assume that interworking between several independent videotex subsystems is desired. These subsystems may be the "national" videotex systems of different countries, or regions within the same country, or may include private videotex systems (we may consider, for example, the cases that the information from a public database is accessible from a private videotex system of some company, or that the information of a private system is also accessible through the public system).

We conclude from the diagram of figure 1 that protocol standards are needed for the communication between the subsystems over the interfaces F, G, I, K, and M. We discuss them in turn.

4.3.1.1 Access to "foreign" databases: Access to "foreign" databases may be obtained through the interfaces I or F. Depending on whether the user agent, during the interworking, is provided by the "originating" or "source" subsystem, the communication protocol used for the interworking will be of type database access or virtual terminal (see also section 4.2.2.2).

4.3.1.2 Updating a "foreign" database: This possibililty is probably of less practical importance. As above, we consider different cases depending where the update agent is implemented. If it is implemented in the "foreign" subsystem a virtual terminal protocol can be used. If it is implemented in the "source" subsystem a database access and update protocol could be used. In both cases, special attention should be given to the consistency between the data in the two videotex subsystems. In the simplest case, the normal update agent of one of the subystems may be used and the consistency of the two databases depends on the vigilance of the IP's.

4.3.1.3 System management: System management information may be exchanged over the interface M. Some management information may also be exchanged over the interfaces I, K, F, and G. For example, accounting information could be provided on a item-by-item, or call-by-call basis to the originating subsystem.

4.3.2 Third party databases

We assume that interworking is desired between a videotex subsystem and a 3-rd party database, as shown in figure 1. Protocol standards would be needed for the communication over the interfaces E and L.

The main function of the protocol over interface E would be data access. Again, depending where the user agent is implemented for the interaction of the user with the 3-rd party database, we may consider the following cases:

(a) User agent in "originating" subystem: The communication with the 3-rd party database would involve a database access protocol. It may be the same as the one used for interworking with "foreign" videotex subsystems.

(b) User agent in the 3-rd party database: The communication would involve a Virtual terminal protocol. It may be the same as the one used for the interworking with a "foreign" videotex subsystem.

Some management functions would also be useful over the interface E, such as 3-rd party billing (charging information on a item-by-item or call-by-call basis).

item-by-item, or call-by-call basis to the originating subsystem.

4.3.2 Third party databases

We assume that interworking is desired between a videotex subsystem and a 3-rd party database, as shown in figure 1. Protocol standards would be needed for the communication over the interfaces E and L.

The main function of the protocol over interface E would be data access. Again, depending where the user agent is implemented for the interaction of the user with the 3-rd party database, we may consider the following cases:

(a) User agent in "originating" subystem: The communication with the 3-rd party database would involve a database access protocol. It may be the same as the one used for interworking with "foreign" videotex subsystems.

(b) User agent in the 3-rd party database: The communication would involve a Virtual terminal protocol. It may be the same as the one used for the interworking with a "foreign" videotex subsystem.

Some management functions would also be useful over the interface E, such as 3-rd party billing (charging information on a item-by-item or call-by-call basis).

The management interface L could use a similar protocol as used for the interworking with "foreign" subsystems over interface M.

4.3.3 Conclusions on standards

The protocol standards needed for the interworking of different independent videotex subsystems and for the interworking with 3-rd party databases are very similar. Ignoring the problem of updating "foreign" databases, the following protocols are involved: -- Virtual terminal protocol (if the user agent is provided by the "source" subsystem or the 3-rd party database, -- database access protocol (if the user agent is provided by the "originating" subsystem), and -- some management protocol (the functions of which have not yet been identified).

We note that for the purpose of exchanging batches of updates, or larger fragments of the information in a database, a file transfer protocol would be useful. However, a file transfer protocol alone is not sufficient, since the internal structure of a transfered batch of updates will be determined by a database update protocol that must be adhered to by both partners.

5. Protocol function relevant for videotex

We discuss in this section the different protocols that have been identified in the above section for use in distributed videotex systems, pointing out some important functions that should be provided by these protocols.

5.1 Virtual terminal protocol

The purpose of a Virtual Terminal protocol is the definition of conventions for the display of information on a terminal and the reception of information from the terminal user in a form independent of the physical terminal structure, as much as this is possible. Most recent proposals for VT protocols [Gien 78, Schi 78, Naff 78, Magn 79] distinguish the following three classes of terminals (and protocols):

(1) scroll mode terminals

(2) page mode terminals, and

(3) data entry terminals.

The Telidon terminal protocol [Teli 79] uses the terminal screen usually in "page mode", i.e. different parts of the display may be displayed in different parts of the screen in any order, and the display may be modified through successive interactions with the user. We conclude that the basic mode for vidoetex applications is the page mode VT protocol.

The VT protocol for videotex should also include graphics and photographic extensions (such as realized for Telidon) since such extensions are very useful for many applications.

The discussion of VT protocols is related to the adaptation of existing systems to the videotex environment. One important consideration is the access to existing databases from videotex terminals. This reduces mainly to the problem of displaying on the videotex terminal (allowing for up to 40 characters per line) the output from a database that is originally formatted for an ASCII terminal with 80 or more characters per line. In the case of an application using a scroll mode terminal protocol, the problem may be solved by introducing additional "new line" commands, however, this approach is not feasable when the output consists of tables that extend over more than 40 character columns. At the end of the page of the videotex terminal, either a scroll mode advance of the text could be used, or "new page" commands could be automatically introduced.

A data entry VT protocol would be useful for a number of videotex applications that involve input from the user for transaction-oriented processing (for instance tele-shoping, banking, etc.). Data entry protocols are usually oriented towards form filling, where the screen (or part of it) contains the outline of a form containing certain fields to be filled in by the user and other fields already filled in by the application program (see also [COST 80]).

Another form of collecting information from the user is by prompts, which interactively invite the user to input one item of information at a time. This approach can be used with the scroll mode as well as page mode VT protocol. In this case the interaction with the user and the verification of the user input is handled by the application program. In order to off load part of this processing from the videotex database computer or the videotex access machine, one can envision the use of more intelligent terminals that perform the information collection dialog with the user. The processing by the terminal could include the analysis of the user's input with respect to the expected possible user answers (for instance in CAI applications), or simply allowing for a free format entry of certain data, such as for instance a date.

There seem to be the following two approaches to the development of terminal protocols that support such flexible user interactions.

(1) Study the typical requirements for interactive data collection, and extend the VT protocol to include an "interactive data collection mode".

(2) To add a "telesoftware function" to the VT protocol which allows the reception, by the terminal, of a program that will be executed by the terminal [Hedg 78]. This program would be selected be the application to include the necessary interactions with the user for collecting the desired information.

The telesoftware function of a terminal may be used not only for data collection, but also for any other processing that involves few interactions with the database and many interactions with the user, or much local processing.

5.2. Database and file access protocol

We consider in this section only protocols to access relatively small parts of one given database. We do not address the following two related points:

(a) How are large batches of updates, or copies of larger parts of a database transfered between two system components ? We note that a file transfer protocol may be used for this purpose, but the format of the transfered file should probably adhere to one of the access protocols discussed in this section.

(b) How does the VAM know in which database the information can be found ? This is a difficult problem in general. Particular schemes may be conceived for different parts of the problem. For instance, in the case that page-oriented information is partitioned over several databases, a cross-reference may point to a page that is not in the same database. How is the consistency between the two databases ensured ? It seems that some not too complex protocols can be designed for this purpose. What are these?

While present videotex databases are page-oriented, there seem to be two other database structures that seem very useful for videotex applications, as explained in more detail in [Tomp 81]. Each of them needs a particular form of access protocol, as discussed in the following subsections.

5.2.1. Page-oriented database access

While present videotex systems exhibit only the user interface for access to page-oriented databases, we consider here the interactions between the user agent (see section 4.1) and the database (for a critical views of present page-oriented database interfaces, see [Tomp 81b]). The user agent contains such information as the number of the presently displayed page and possible cross-references from this point.

A request for a new page of information would usually contain the following information:

-- number of page requested.

-- logical database, "magasine" or similar, of which the page is part.

-- options for accounting.

A response from the database would usually contain the following information:

-- number of page,

-- logical database, ...

-- the content of the page,

-- cross-references valid from this page,

-- possibly some information about keywords that are valid responses by the user after the display of the page,

-- possibly "title" of page, or list of keywords assigned to it for retrieval purposes,

-- information about access rights, such as closed user groups, etc.

-- accounting information.

5.2.2. Access to record-oriented databases

As pointed out in [Tomp 81], most videotex applications of record-oriented databases involve queries on only a single relation (or entity) at a time. Therefore the difference between relational, hierarchical, and network database models is of minor importance in our context.

Some proposals for access protocols to record-oriented databases can be found in the literature [Bane 78, Page 80].. We note that for many applications one or several fields of a record may contain text or graphical information for display on the terminal. In contrast to traditional record-oriented data, such fields are usually of variable length. Instead of storing this information directly in the record, it is conceivable to store the information in a separate page-oriented database, and to include in the record only a pointer (or page number).

5.2.3. Access to keyword-oriented databases

Different structures for keyword-oriented databases exist. Corresponding access protocols could be designed. Some simple keyword related information was already mentioned in section 5.2.1 for page-oriented databases. More complex structures are realized in library information systems. Access protocols at the user interface level are discussed in [Ball 81]. A standardization of the user interface also implies a standardization of the retrieval functions provided by the system. We do not go into any more detail in this report, and close with the following remarks:

Keyword searches generally require in addition to the function of retrieval of information associated with a given keyword (or keyword expression), also the function of spelling corrections and looking for related keywords (using a thesaurus, or word stemming). It may also involve string matching of the entire information in respect to the searched word. For more detail, see for example [Ball 81].

It is possible to consider the page-oriented and keyword-oriented database access protocols as a special instance of the record-oriented protocol. For example, the cross-references indicated in section 5.2.1 could be considered particular fields in the "page record" which has a particular format. Similarly, keyword-oriented information associated with some information (page) could be handled as particular record fields. It may be possible to develop some standard record format for videotex applications, involving cross-reference fields, keyword fields, etc., and also variable-length fields that contain the "basic" information, i.e. text and graphic information to be displayed.

5.3. System management protocols

The overview of the protocols in section 4 showed that the protocols used by a videotex subsystem for the interaction with another subsystem, and the interaction with a 3-rd party database can be quite similar. This holds also for the management aspects of the interactions.

This section gives a list of functions that seem to be important for the interworking of different videotex subsystems with one another, and with 3-rd party databases.

5.3.1. Status exchange

The following kind of status information may be exchanged:

(a) system up ? present load, response time, etc.

(b) kind of service provided

5.3.2. Gateway management

Gateway management includes:

(a) establishment and closing of gateways (either for time-shared use for many different user requests, or a single gateway per user)

(b) information exchange over a gateway, possibly involving some Presentation translation)

(c) gateway related accounting

5.3.3. Selection of database, directory information

We consider under this heading such functions as:

(a) exchange of information about services provided in view of selecting a database (3-rd party, or other subsystem) for a particular user application

(b) exchange of directory information for the establishment of a global, "high-level" directory of services available to the user

5.3.4. Accounting

The form of the accounting procedures will depend on such administrative questions as "where is the user account kept ?", "is 3-rd party billing used ?" [Sola 76], "how much detail will the user see on the bill for services ?", etc.

In [Ball 80], it is suggested that the accounting for the user activities could be well handled by the VAM obtaining accounting information from the databases and other service computers for the different transaction made on behalf of a user.

If copies of information from a given source are kept in different locations, it may be necessary to distinguish the rates and accounting procedures for "wholesale" and "end-user sale".

5.3.5. Access rights

Large parts of the videotex information will be publicly available. The access to other parts may be regulated by protocols involving closed user groups. Other approaches may also be suitable.

6. Conclusions

The following paragraphs summarize the main conclusions from the foregoing discussions.

A number of different scenarios for distribution must be foreseen for future vidoetex systems. They range from the distribution of the different functions of a videotex system over several computer (over local or long distance communication networks, see for instance [Ball 80]), to 3-rd party databases and the interworking of several videotex systems with all the possible problems of distributed authority, and data partitioning and replication.

For the interaction of a videotex system with a 3-rd party database and for the interworking with other systems (possibly in other countries, or operated as a competing enterprise) very similar protocols could be used.

The concepts developed in the OSI Reference Model are applicable to videotex applications. In particular, the OSI protocols mentioned below may be used for distributed videotex systems. (We note that this study does not go to the level of detail as to consider particular protocols. On the other hand, higher-level protocol standards are not yet developed).

The higher-level protocols for distributed videotex systems may be classified into the following calsses. Standard OSI protocols developed for these purposes should be applicable.

(a) Virtual Terminal protocol
(b) database access protocol (access and possibly update of selected records)
(c) file transfer protocol (for sequential files)
(d) management protocol

The following questions are probably not sufficiently addressed by the present OSI work:

(a) database access in general. (The ISO TC97SC16 work concentrates on file transfer).
(b) File access to "page-oriented" databases (see section 5.2).
(c) Virtual Terminal protocol including graphics and image facilities (full Telidon standard).

7. Points for further study

We give in the following a list of problem areas which merit particular attention, and in which further research is needed for solving the problems of distribution and interworking of different videotex systems and applications.

7.1. Naive user interfaces: For more detail see [Ball 81].

7.2. Procedures for interactive form filling (see section 5.1).

7.3. Standard for down-line loaded executable code (see section 5.1).

7.4. File access protocol for page-oriented databases (see section 5.2).

7.5. Interworking of page, record and keyword-oriented databases (see section 5.2).

7.6. Protocols for retaining consistency between pointers and information in page-oriented distributed databases.

7.7. Protocols for managing information that is partitioned over several databases, for example finding out where a particular fragment of the information resides.

References

[Ball 80] A.J.S. Ball, G.V. Bochmann and J. Gecsei, "Videotex networks", IEEE Computer 13, 12 (Dec. 1980), pp. 8 - .

[Ball 81] A.J.S. Ball and J. Gecsei, "User interfaces for future videotex systems", Technical report, prepared for the Department of Communications, Canada.

[Bane 78] J. Banerjee, D.K. Hsiao and F.K. Ng, "Data network - a computer network of general purpose fron-end computers and special-purpose back-end database machines", Proc. Computer Networks Protocols Symposium, University of Liege, 1978, pp. D6-1 to D6-12.

[Cave 81] J. Cavers, "Application of the OSI reference model to CATV - based data services networks", to be presented at NTC 1981; see also "A data services protocol for CATV - based information systems", MacDonald, Dettwiler & Ass. Ltd., Richmond, B.C., Canada, Sept. 1980.

[Chit 79] A.M. Chitnis and J.M. Costa, "Videotex services: network and terminal alternatives", IEEE Trans. Consumer Electr. CE-25, 3 (July 1979), pp. 269-278.

[Cost 80] COSTPRO (Canadian Organization for the Simplification of Trade Procedures), Canadian Trade Information System, 1980.

[Date 77] C.J. Date, An Introduction to database systems, Addison-Wesley Publ. 1977.

[Gien 78] M. Gien, "A file transfer protocol", Computer Networks

2 (1978), pp. 312-319.

[Hedg 78] J. Hedger, "Telesoftware", Wireless World, Nov. 1978.

[ISO 81] "OSI - Basic reference model", Draft Proposal 7498, see also ISO TC97SC16 N537 (1980).

[Magn 79] F. Magnee, A. Endrizzi, and J. Day, "A survey of terminal protocols", Computer Networks 3 (1979), pp. 299-314.

[Naff 78] N. Naffah, "High level protocol for alphanumeric data-entry terminals", Computer Networks 2 (1978), pp. 84-94.

[Pope 80] R. Popescu-Zeletin et al., "The virtual file system", Technical report HMI-B 333, Hahn-Meitner-Institut, Berlin, 1980.

[Schi 78] P. Schicker and A. Duenki, "The virtual terminal definition", Computer Networks 2 (1978), pp. 429-441.

[Sola 76] I. de Sola Pool, "International aspects of computer communications", Telecommunications Policy 1, 1 (Dec. 1976), pp. 33-51.

[Teli 79] "Picture description instructions for geometric and photographic image codings in videotex systems", CCITT document CAN/COM VIII/2, COM VII - No. 105 (FEb. 1979).

[Tomp 81] F.W. Tompa, G.V. Bochmann, and J. Gecsei, "The application of current database technology to videotex", Technical report, prepared for the Department of Communications, Canada.

[Tomp 81b] F.W. Tompa, J. Gecsei and G.V. Bochmann, "Data structuring facilities for videotex systems", to be published in IEEE Computer.

[Zim 80] H. Zimmermann, "OSI reference model - the OSI model of architecture for Open Systems Interconnection", IEEE Trans. Com-28, 4 (April 1980), pp. 425-432.

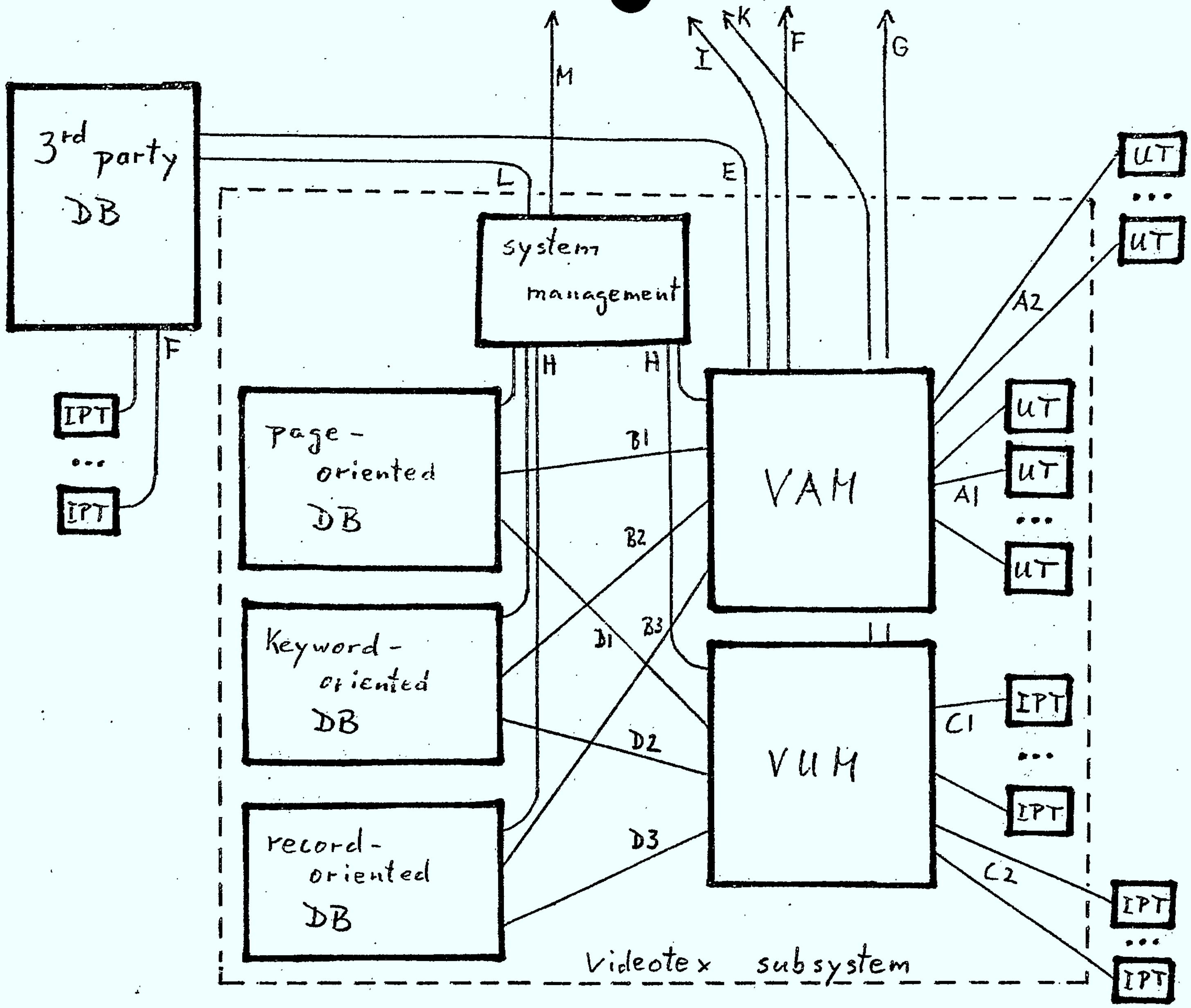

Figure 1

M: to management module of other videotex subsystems

I, K: to databases of other videotex subsystems

F: to VAM of other videotex subsystems

G: to VUM of other videotex subsystems

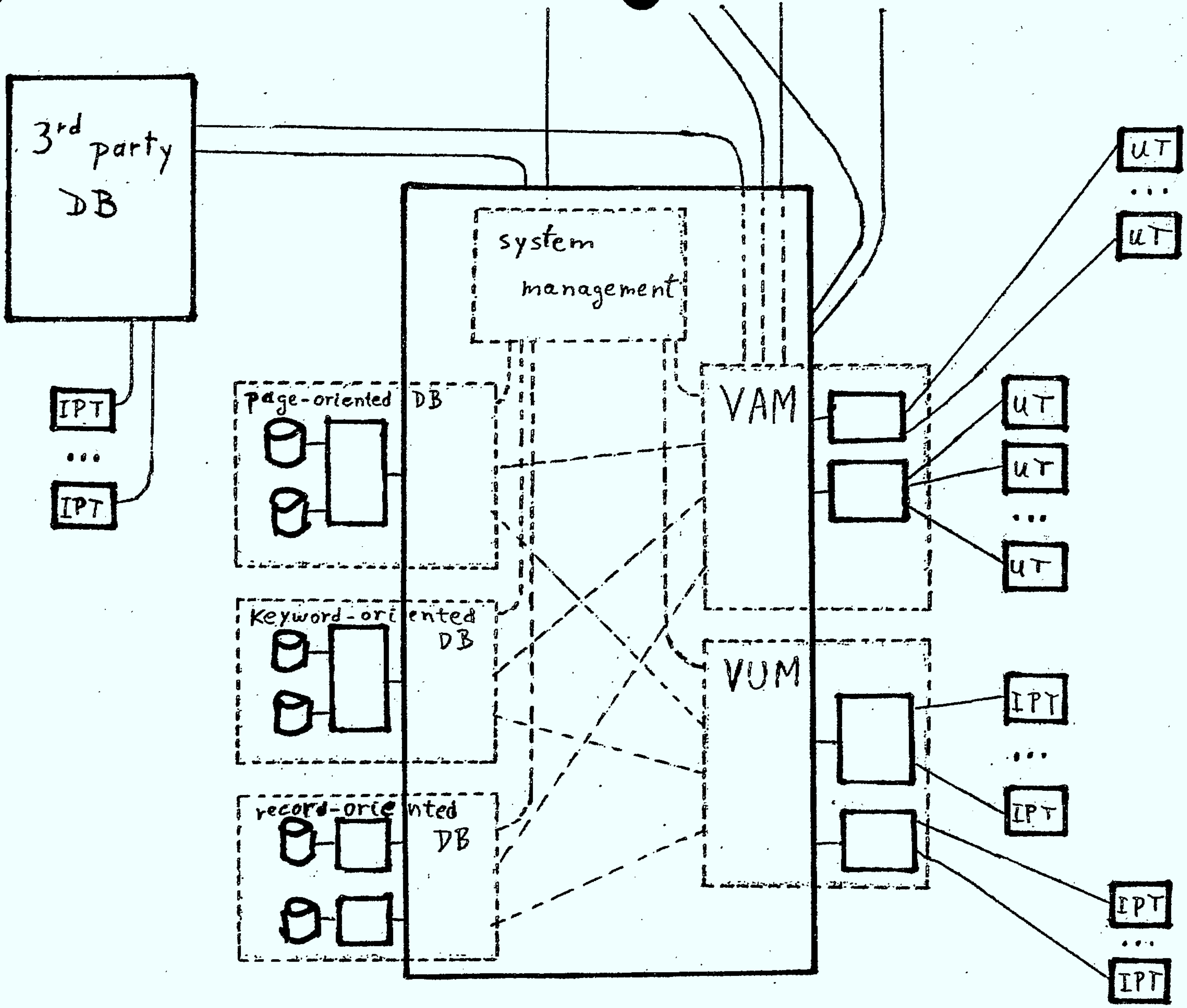
3rd party DB
IPT
IPT
system management
page-oriented DB
Keyword-oriented DB
record-oriented DB
VAM
VUM
UT
UT
UT
UT
UT
IPT
IPT
IPT
IPT

Figure 2

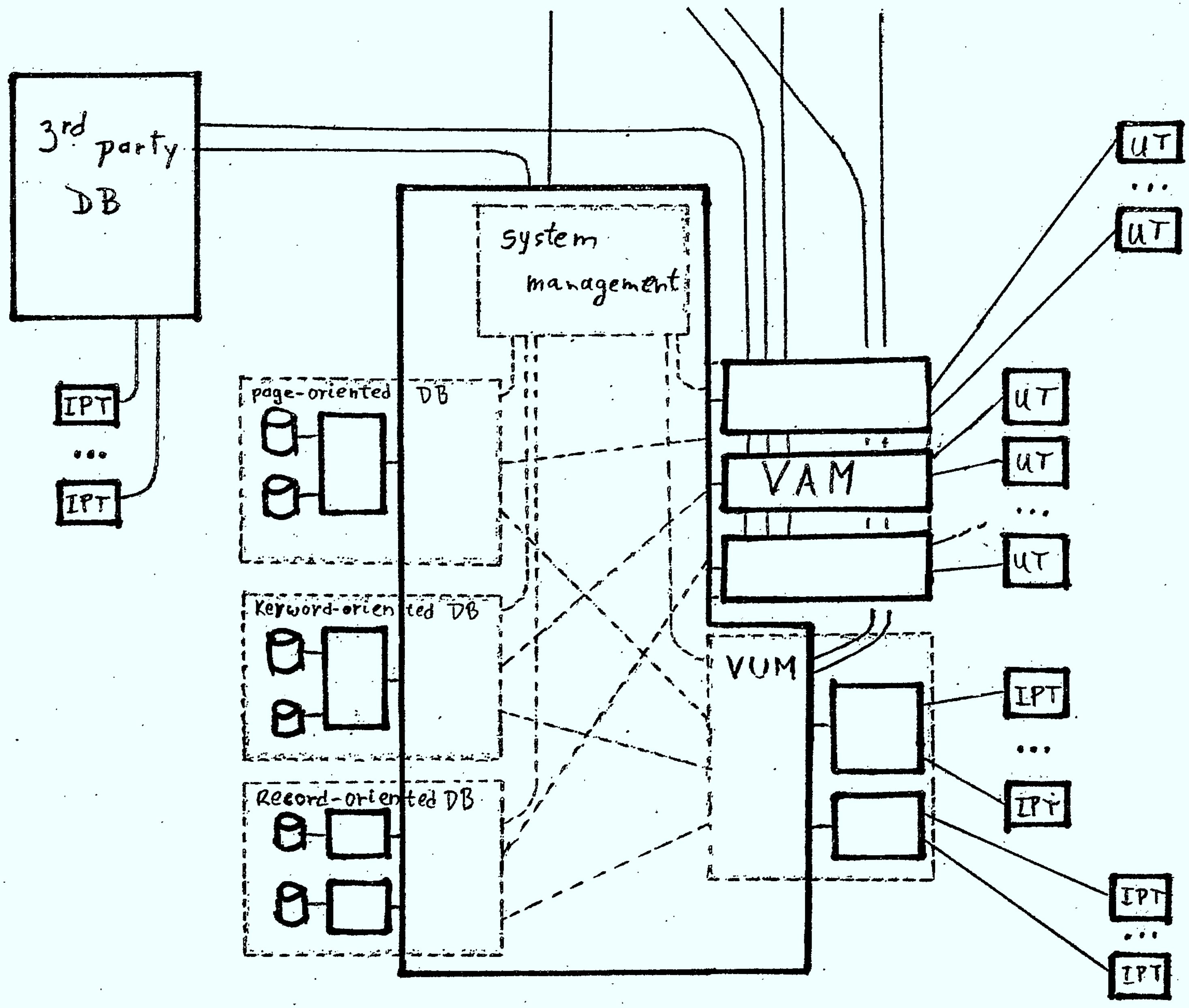
3rd party DB
IPT
...
IPT
system management
page-oriented DB
Keyword-oriented DB
Record-oriented DB
VAM
VUM
UT
...
UT
UT
UT
...
UT
IPT
...
IPT
IPT
...
IPT

Figure 3

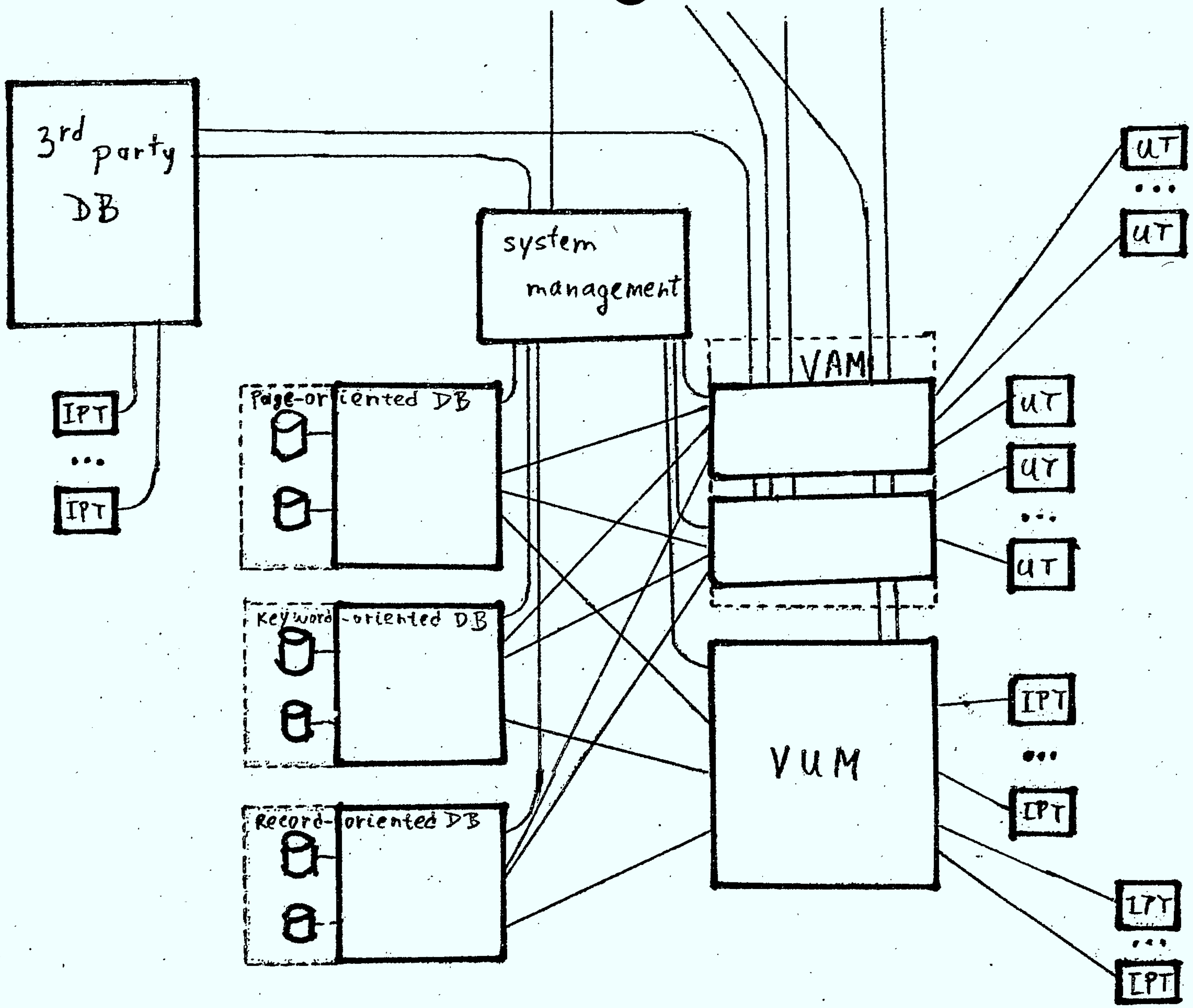
3rd party DB
system management
VAM
VUM
Page-oriented DB
Keyword-oriented DB
Record-oriented DB
IPT
IPT
UT
UT
UT
UT
UT
IPT
IPT
IPT
IPT

Figure 4

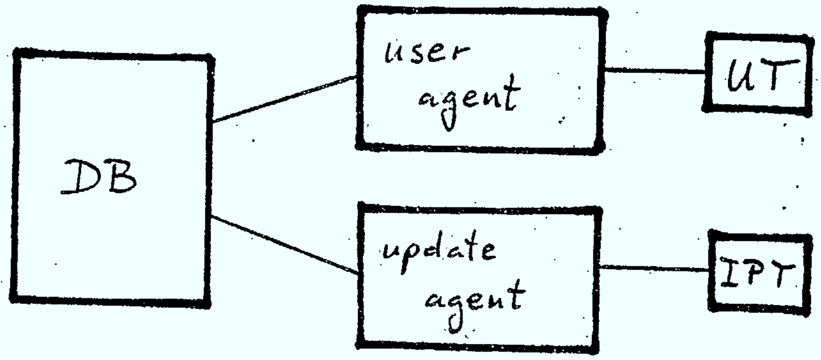

Figure 5