An architecture for a business and information system

by B. A. Devlin
P. T. Murphy

The transaction-processing environment in which companies maintain their operational databases was the original target for computerization and is now well understood. On the other hand, access to company information on a large scale by an end user for reporting and data analysis is relatively new. Within IBM, the computerization of informational systems is progressing, driven by business needs and by the availability of improved tools for accessing the company data. It is now apparent that an architecture is needed to draw together the various strands of informational system activity within the company. IBM Europe, Middle East, and Africa (E/ME/A) has adopted an architecture called the E/ME/A Business Information System (EBIS) architecture as the strategic direction for informational systems. EBIS proposes an integrated warehouse of company data based firmly in the relational database environment. End-user access to this warehouse is simplified by a consistent set of tools provided by an end-user interface and supported by a business data directory that describes the information available in user terms. This paper describes the background and components of the architecture of EBIS.

The environment for information retrieval and reporting within the IBM Europe, Middle East, and Africa (E/ME/A) countries is today characterized by functional systems that deliver information to specific groups. This is often achieved in a redundant and costly way because organizational or functional barriers prevent the free transfer of information. With effort it is possible for any independent and innovative user or group to seek out new data and obtain it. Therefore, a new direction and strategy must be one that promotes and encourages the use of data and makes the required information available to every person in a particular interest group who has need of it.

Today, each function, such as marketing, administration, or finance, tends to have its own method of handling general reporting services and end-user access to data. Most functions are using some form of APL-based system to carry out their reporting and information requirements, but these systems are quite different from one another.

This situation involving eclectic systems, which provided the impetus for the development of a new, more uniform, architecture for information retrieval and reporting, is not unique to IBM. Many companies have developed along similar lines, where discrete needs for particular sets of information have motivated independent and often conflicting procedures for obtaining the necessary data from the established operational systems.

A number of limitations of the current query systems have been identified. Most of us when we are new to an organization experience the fact that data retrieval is not easy for the novice user. Similarly, when it comes to structuring and reporting data, the new user often finds that standard reports are difficult to define and change, requiring detailed knowledge of the data source. Today's informational systems are not directed at the entire potential end-user community. Because of the dependency on detailed knowledge of the operational systems, many potential end users, especially at management level, have become *requestors* of information rather than *accessors* of information. Managers and other users require current information, and existing information systems frequently do not respond quickly enough to business changes or users' needs. The reason for this is that a specific action must be taken by the Information Systems (IS) organization for each change made. If the information exists but is in an unsuitable format for general querying, the Information Systems organization must devise a specific extraction procedure for each occurrence. This can lead to a substantial delay, because IS has to deal with many such *ad hoc* requests. Another possibility is that the information may exist in some form, but the user is not aware of it because there is no central catalogue of available information. We have found that reference information, such as commercial directories, is often not promoted or made available to potential users.

Directions for end-user data access

Users today may be characterized as belonging to one of two distinct groupings:

- *Dependent users* are those who require full support, assistance, and guidance through all options and facilities by a comprehensive menu system incorporating advanced help facilities. These intended users include managers, professionals, secretaries, and clerical personnel.
- *Independent users* are those who are experienced in computing systems and can work with a variety of different decision-support systems. This class of users includes such professionals as financial analysts, systems engineers, administrators, and business planning specialists.

Great progress has been made within E/ME/A in providing basic office support services, such as electronic mail and on-line diary, to the majority of users who depend on the system for prompting and full menu support. However, dependent users still do not have full and easy access to the business data they require. Most information retrieval and reporting is still carried out by professionals in the independent category of user on behalf of dependent users.

The solution to the requirements just discussed is the framework provided by an informational system, the architecture of which is discussed in this paper. This system, which is known as the E/ME/A Business Information System, or EBIS for short, delivers an integrated and consistent informational system to end users in any business function. It is run by a support organization that has the responsibility for promoting and supporting access to data by end users.

Users can now focus on the use of the information rather than on how to obtain it.

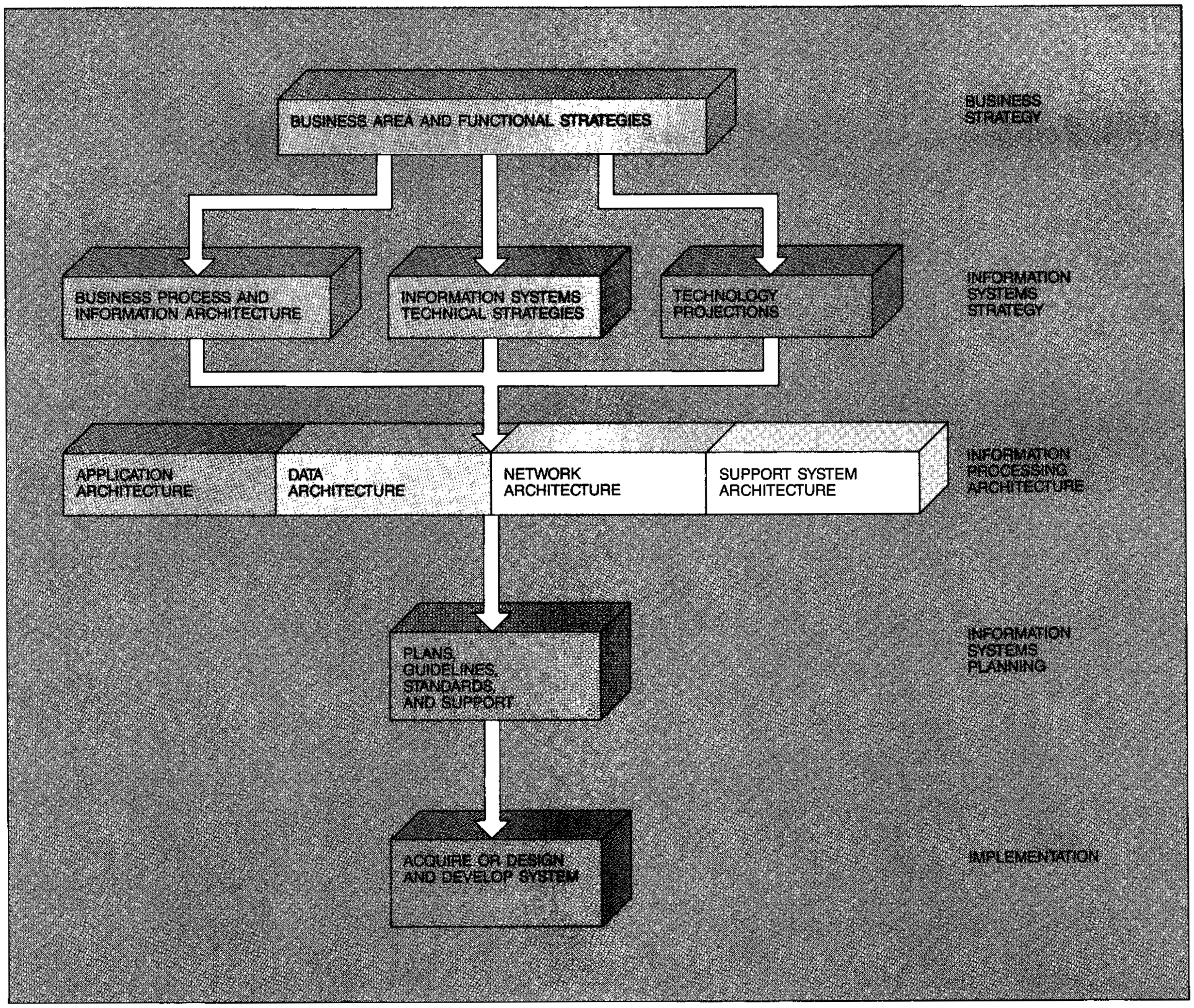
Because a single source of data is provided for end-user access, there is one logically consistent set of services to be learned; in the system EBIS is replacing, users must be educated and supported in a variety of different interfaces. In addition, there is a single support structure responsible for delivering business information to users, and a single contact point is responsible for its quality and accuracy. Users can now focus on the use of the information rather than on how to obtain it.

Architecture differs from design in that architecture provides a long-term goal and represents the overall design target and direction for a particular aspect of information processing.

Within information processing, the Information Systems organization must be guided by the business strategies of the company. This is essential to the continued success of the support systems provided for the company. The overall framework for information processing in IBM takes into account how the business and functional strategies must drive the Information Systems strategies, which in turn determine the various Information Systems component architectures. These architectures, illustrated in Figure 1, are the following:

- Application architecture
- Data architecture
- Network architecture
- Support system architecture

Figure 1 Overall information processing architecture



EBIS architecture

EBIS defines the architecture for the implementation of an information retrieval and reporting service. It defines the components and functions, and it defines their interrelationships. The architecture is based on the assumption that such a service runs against a repository of all required business information that is known as the *Business Data Warehouse* (BDW). The EBIS architecture is based on the four architecture classes previously listed. However, the primary dependence is on the data architecture and the application architecture. The data architecture is driven by the business process and the information architecture. The application architecture recognizes the need for each application to be responsible for providing complete information extracts to a downstream reporting service.

A necessary prerequisite for the physical implementation of a business data warehouse service is a business process and information architecture that defines (1) the reporting flow between functions and (2) the data required.

A data architecture should cover all aspects of the data required to support the business. However, most data architectures to date have concentrated on the requirements of operational systems within each function. *Operational systems* are those that have been developed to support the business operations, e.g., order entry, distribution, billing, accounting, and payroll. Informational systems requirements include all aspects of reporting and analyzing data, and hence, there is a need for an extension of the basic data architecture to support this environment.

Informational systems are normally provided as a set of end-user computing facilities that offer a query and reporting service for handling standard queries, general inquiries, *ad hoc* queries, formatted reports, and complex data analysis. Thus the end-user computing facility provides a complete range of decision-support facilities. Informational systems usually run against a subset of data extracted and copied from the operational systems.

Operational systems support the fundamental company business process, maintaining and updating the company databases. Operational systems implement the full security, legal, and audit standards required by the company in the processing of the company databases.

For several reasons, it is preferable to have the information required for general end-user access separate from the operational systems production data—the primary source of information. The three most important reasons are the following:

- Ensuring that the performance of the production systems is not disrupted by *ad hoc* queries or analyses
- Requiring that information needed by end users is not changing as they use it, i.e., point-in-time data
- Operational systems databases designed for the high-volume operational processes that are not, in general, suitable for answering the unpredictable queries of end users

The EBIS architecture assumes that informational systems will continue to be separate from the operational systems that supply the base company data for periodic update. The EBIS architecture forms one part of the overall E/ME/A End-User Computing Services architecture and complements such functions as office systems and direct access to operational systems.

Although such systems may be provided on different computers or operating systems, the end user is isolated from that complexity by a *gateway* that provides a single-system image of all services. This facility is based in the Intelligent Workstation (IWS). The overall positioning of EBIS within the End-User Computing Services is shown in Figure 2.

In the performance of duties within the company, an end user requires access to a well-defined subset of the total business information of the company. The boundaries of this subset are defined by the tasks performed by the user. Thus a salesman responsible for a particular set of customers may require access to a number of items of information about those customers, but about those customers only. It is conceptually convenient and it matches the user's perception to think of the business information and any user's subset of that information as a set of tables. An example set of tables in a Business Data Warehouse is shown in Figure 3.

Although data may reside in multiple locations, the appearance is of a single source.

Within the total set of tables that describe the business, it is unusual for any two end users to have access to exactly the same subset. Even where the subsets are similar in content, a user's perception of their structure may be quite different. Thus, of two users who have access to the same customer orders table, one user may be interested in how much each customer is spending, whereas the other may look only at which products are selling best.

The EBIS architecture provides such tables in a relational database environment. It also provides for an integrated set of user-friendly tools to access the information, to help the user understand its importance and interrelationships, and to process it as the user requires. To ease access to the data and to achieve a coherent framework for such access, it is vital that all the data reside in a single logical repository, the Business Data Warehouse (BDW). Although

Figure 2 EBIS as part of the End-User Computing Services

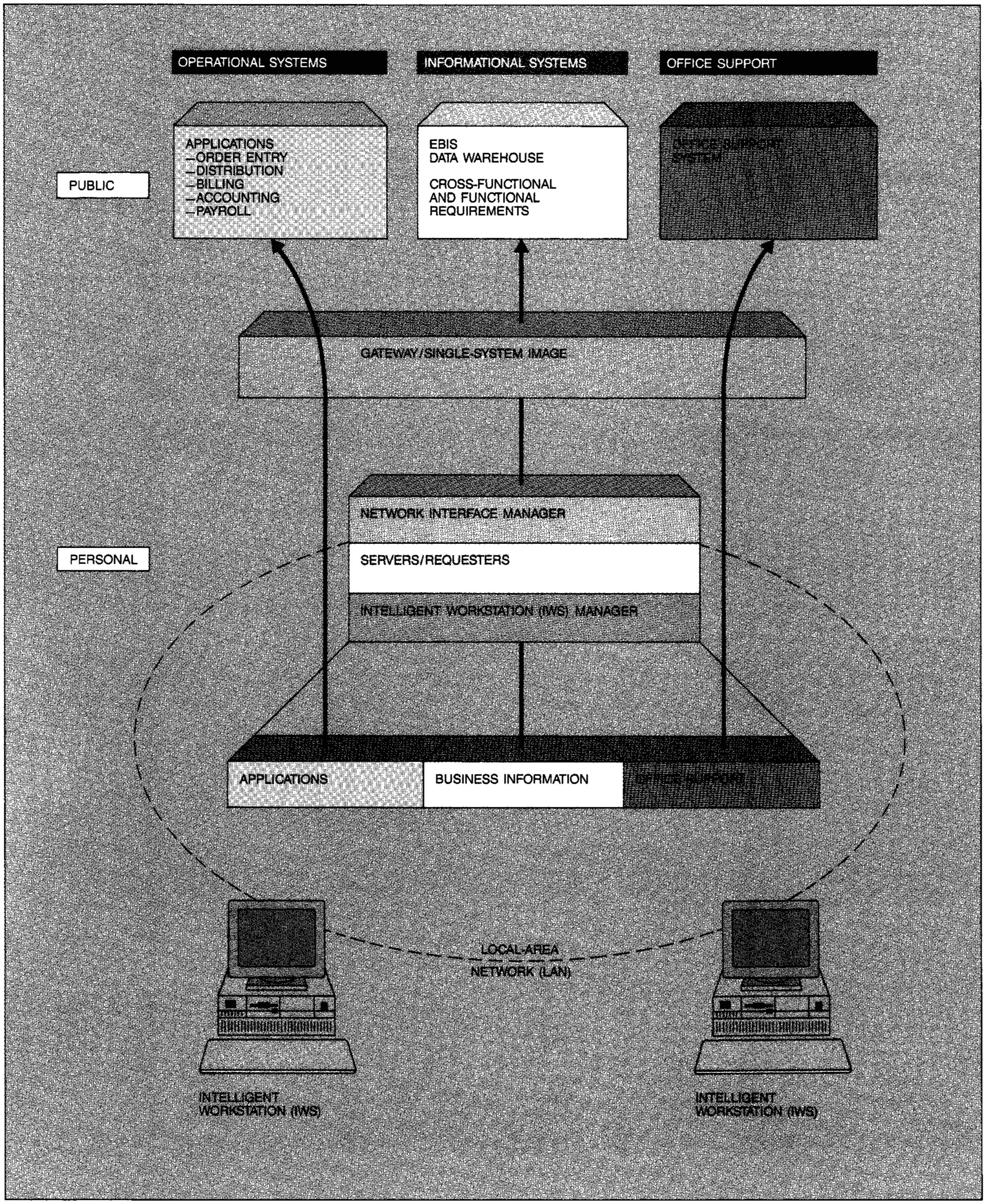
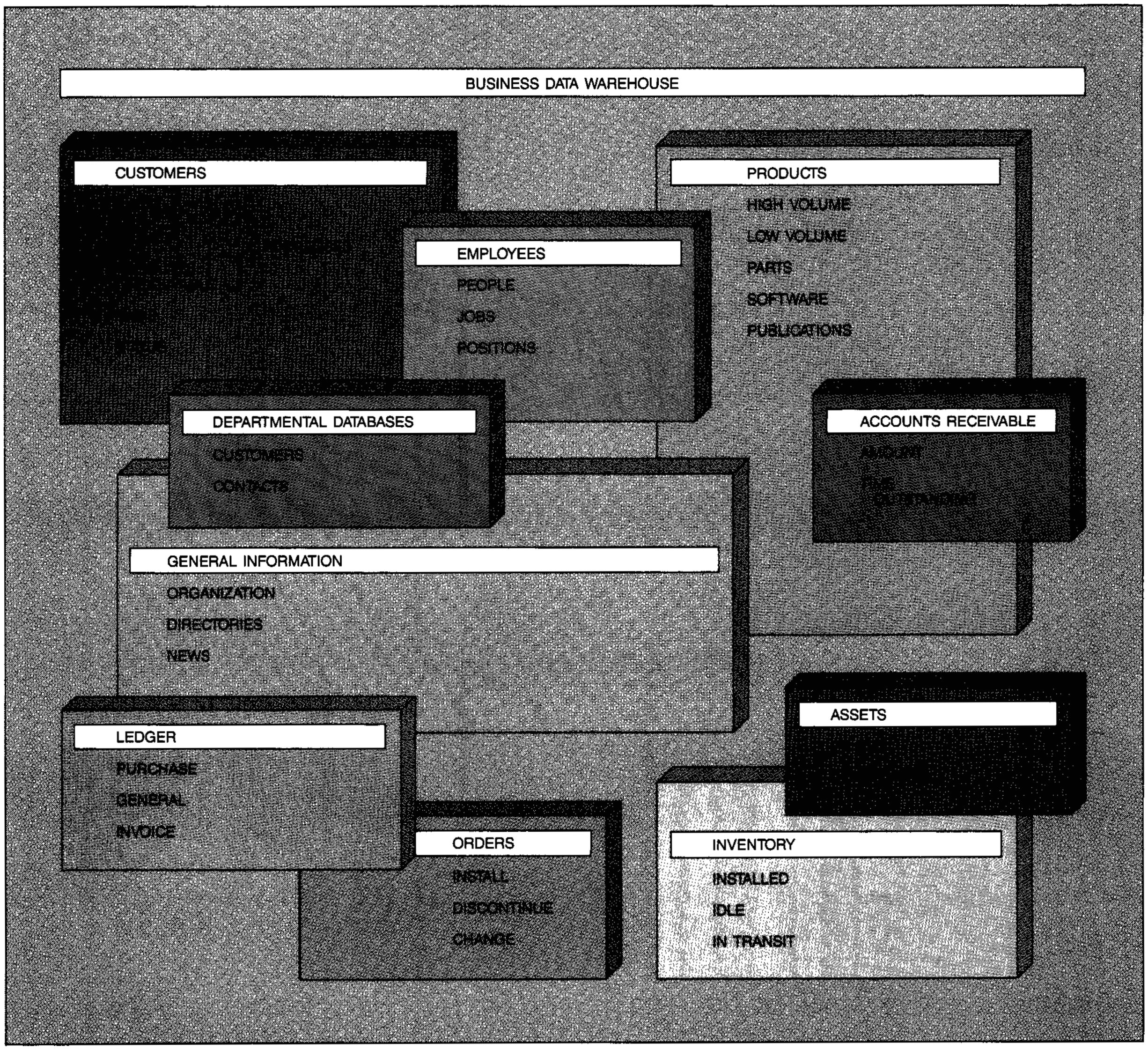


Figure 3 Example of contents of a Business Data Warehouse

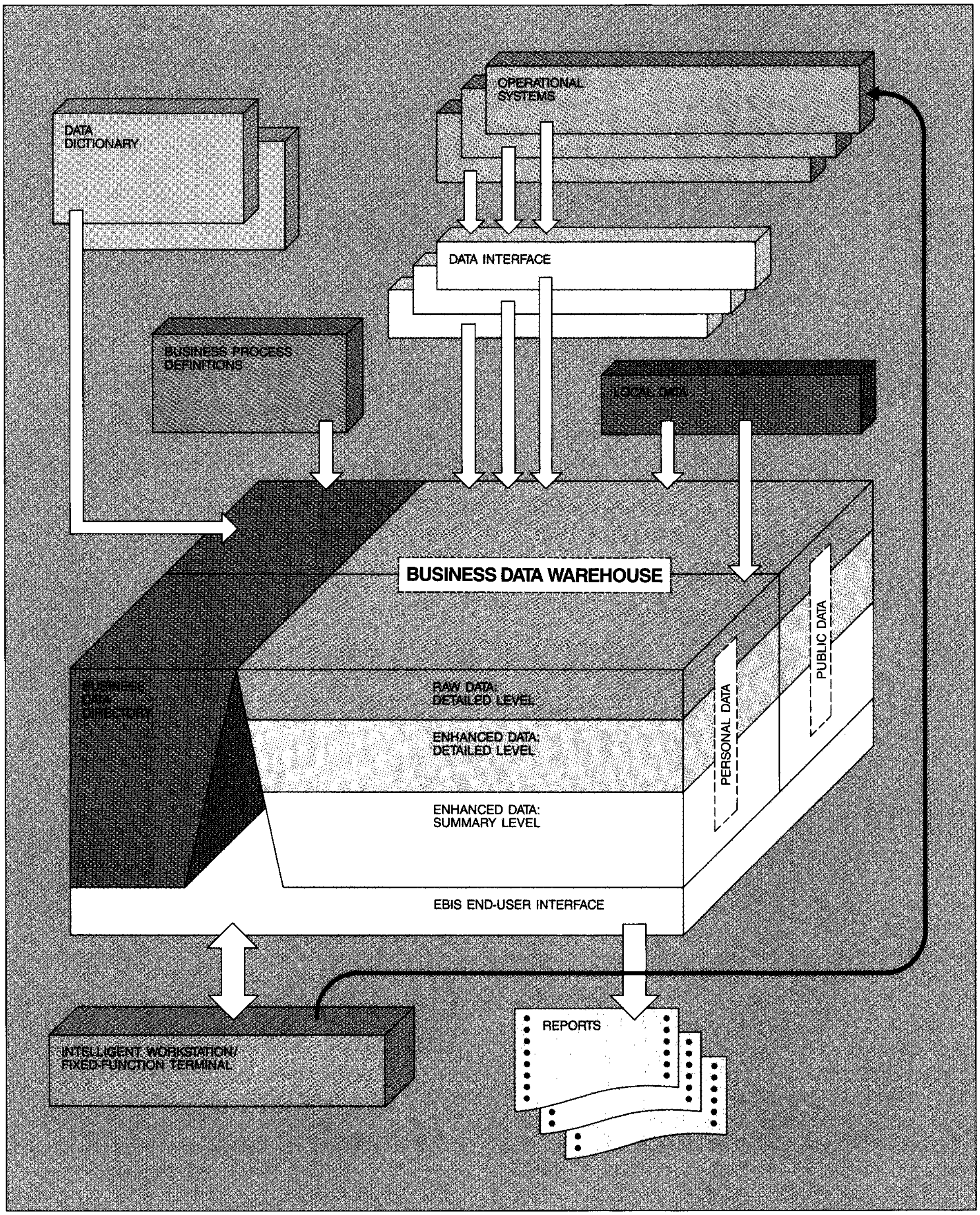


the data may physically reside in multiple locations, the end user sees only a single source that satisfies all informational needs.

As shown in Figure 4, data are received from the operational systems in an agreed-upon format and stored in the BDW. Such data are usually at a detailed level. A second important source of data for the BDW is local departmental files or databases. Descriptions of the data and the business rules are also stored in a Business Data Directory (BDD), which allows the user to productively use the stored business information. Within the BDW, the raw data are enhanced to obtain new information that is of general interest. The new data can be either at a detailed level or at a summary level more appropriate to the management of the business. Finally, the information is presented to the end user for viewing on the terminal, for further manipulation on the Intelligent Workstation, or for use in repetitive reports.

The EBIS architecture is confined neither to any one operating system nor to any one data access method. However, recognizing the current environment of

Figure 4 Overview of the EMEA Business Information System



most end users and their changing expectations, together with the technology available today and its

Each user sees information from different company tables combined in a way that makes the data most meaningful.

future directions, the EBIS architecture has the following characteristics:

- Access to data may be from a cross-functional or functional viewpoint.
- The strategy is based on an open architecture. The sources of the data stored in the BDW are limited only by the user's needs. In obtaining the information from EBIS, the user is not restricted to a predefined set of tools.
- The data manager is based on the relational model (i.e., on DB2 or SQL/DS).
- The design is modular, with standardized and well-defined interfaces between the different modules. This allows changes in technology or business practices to be more easily accommodated.
- The operational systems developers are responsible for providing data to the BDW through a standard data interface agreed on by the operational system and EBIS developers.
- Data can be stored in several locations: central relational database, distributed relational databases, Intelligent Workstations, or file servers.
- Use is made of local processing power where available. Functionality and data are distributed.
- EBIS does not permit changes to public data stored in the BDW. The end user who wishes to make changes to data for which he is responsible must do so through the operational system.

The EBIS architecture consists of seven distinct components, each with a specific task or responsibility. These components are now discussed.

Business Data Warehouse (BDW). The BDW is the single logical storehouse of all the information used to report on the business. The BDW is based on the relational concept. In relational terms, the end user is presented with a view or number of views that contain the accessed data. The structure of these views is determined solely by the requirements of that user. The user thus perceives a set of tables containing only the needed columns, although these columns may have been obtained from a number of different tables of company data. Figure 5 shows how the end user of a relational database can be isolated from the complexity of the total company data by being provided with only the relevant subsets of the company tables. In addition, each user sees information from different company tables combined in a way that makes the data most meaningful.

The *data administrator* is responsible for the logical structure of the BDW, as defined in the company data model. As shown in Figure 5, the data administrator has access to the total data of the company, and his perception matches the logical data layout of the company data model. These tables are cross-functional, and must be previously defined in a company-wide business data analysis. This analysis also defines the functional subsets of the total data, and is used to create functional views of the data or even functional warehouses if necessary.

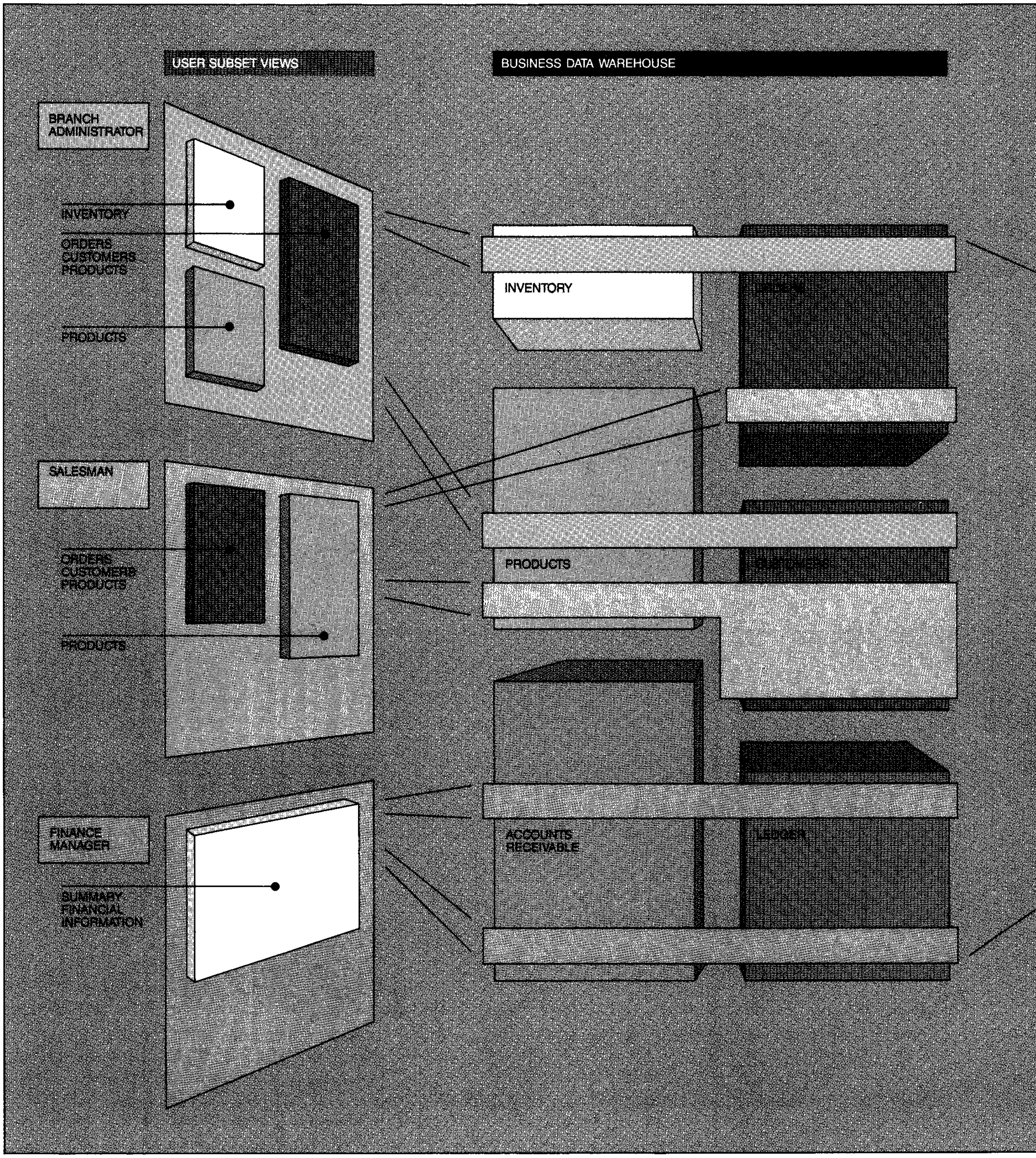
Database design and administration. The design of a company data model is a prerequisite for a successful implementation of a company-wide cross-functional BDW. The company data model expresses the relationships among the variety of data elements used in different parts of the business and the basic business processes. It also defines the interrelationships among the various data elements.

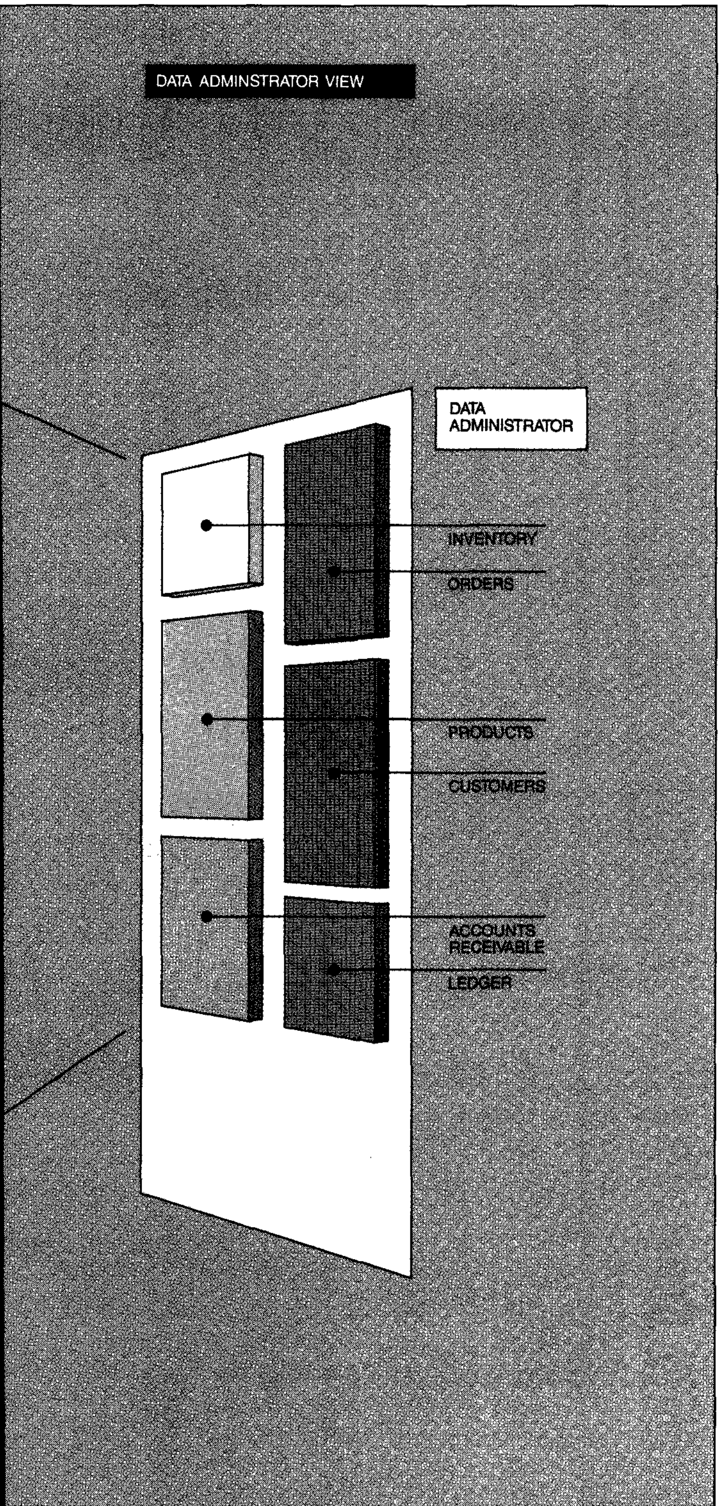
Database design, that is, the design of the table layout and relationships, depends on the prior data analysis and data modeling phase that produces the company data model. Rigorous data analysis is necessary before the inclusion of information from each operational system into the BDW.

The EBIS database designer has an advantage over his counterpart planning for a hierarchical database, because of the flexibility of the relational concept. There is a greater degree of independence between the logical layout of the tables and the physical layout of the data. This insulates the end user from some of the technical compromises that have to be made for the sake of performance or storage constraints.

The BDW assumes full use of the table-and-view structure described previously. The data are con-

Figure 5 Different views of the Business Data Warehouse





tained in tables designed to satisfy two conflicting requirements. There is the need for logically structured, fully normalized tables that are easily understood and maintained. At the same time, there are system performance constraints.

It is important to realize that, whereas the concept of an integrated BDW and the process of normalization eliminate data redundancy, the introduction of performance constraints reintroduces some redundancy at the base-table level. Table design for a relational database thus takes place at two levels. On the first level, the data analysis and normalization procedures produce a set of logically coherent tables that correspond to the views of the data administrator. These tables may also be used as the base tables. On the second level of design, which may take place only after a pilot, performance constraints are identified, and the base tables (but not the data administrator views) are redesigned to improve performance.

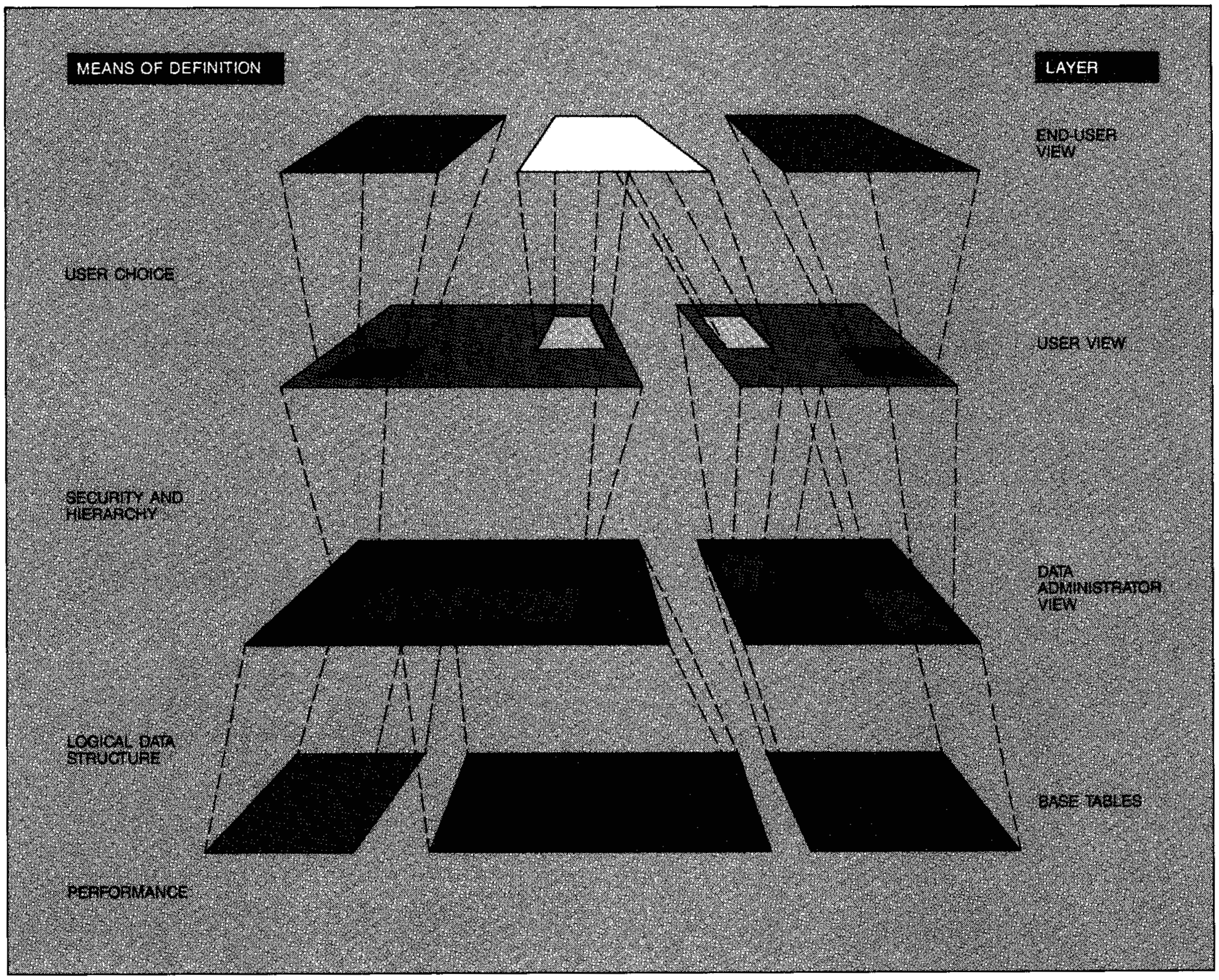
The user has only indirect access to the data administrator views because another layer of views is interposed. This layer is based on the security level and access needs of the user. Figure 6 illustrates the view structure through which the end user accesses the base tables. This layered structure, although apparently cumbersome at first sight, is highly beneficial to both the data administrator and the end user. The purpose and definition of each layer are now given.

The *base tables* are designed to enhance the performance of the system. Their layout is known only to the data administrator. Base tables may be changed frequently, and they may or may not reflect the logical structure of the information.

The *data administrator views* reflect the logical data structure and are seldom changed except by the addition of extra fields. Because the end user has access (indirectly) to the data administrator views only, that user knows the data only through their logical layout.

The *user views* are mapped in a one-to-one relationship with the data administrator views. Thus, the user views are logically structured views of the data, with certain rows or columns omitted. The restrictions on the user views are defined by the security and hierarchical details contained in a user profile that is described later.

Figure 6 General database structure for the Business Data Warehouse



The *end-user views* present the information to the user as he wishes to see it. They may correspond to the user views, or they may be defined by the user himself or implicitly by a task to be carried out. This allows the user to focus only on those rows or columns of the data that are currently of interest.

Data in the Business Data Warehouse. Public data within the BDW constitute information that is generally available (subject to security restrictions) and may not be altered by the end user. Although all public data are treated identically by the BDW, we find it conceptually useful to further divide public data into two categories based on the scope of the data. The first category is organizational data—information that is applicable and of interest to the organization as a whole. In business terms, such an organization might be a country that functions essentially as an independent unit within the business. The second category of public data is departmental data. This information is used solely by one group of users within the organization. Such a group might be defined by site, department, business function, or any other useful criterion.

Personal data are those owned by a user; they may be changed and used at the user's discretion. The EBIS architecture allows personal data to be stored in the relational database, and such data are inaccessible to other users of the BDW (unless the owner chooses to share that information). As far as the owner is concerned, personal data are indistinguishable from system-supplied information, with the exception that the user may modify the personal data as he sees fit. EBIS cannot, of course, ensure the integrity or consistency of such data.

End-User Interface (EUI). The EBIS architecture must provide its users with a consistent window on the stored data, independent of the sources and its

The productivity of the end user is enhanced by the usability of the interface through which the data are accessed.

users' needs. Beyond providing tools to enquire into the information and generate reports in a fully integrated manner, EBIS must be a vehicle that supplies the data to particular tools favored by individual users, functions, or countries. The increasing sophistication of end users, including their preferences for particular methods of processing data, is becoming an important area for increasing productivity.

The productivity of the end user is enhanced by the usability of the interface through which the data are accessed.[1] The EUI must be adaptable, providing fast access to data for experienced users while giving maximum guidance to novices. Thus, a full-menu system, customized to the user profile, aids the novice, while a fast path and abbreviated command entry must be available for the experienced user. The most commonly used facilities must be integrated such that their appearance and usage conventions are as similar as possible. Flexibility is required to give the user access to any special tools used for a specific task. The data shared among information system, office systems, and operational system environments must be portable. A comprehensive help facility must be provided, including a full on-line tutorial. The installing locations must be able to modify and extend the menus and help functions to provide national-language versions of them.

The End-User Interface (EUI) provides the user with access to every module of EBIS. However, in terms of providing an information retrieval and reporting service to end users, the EUI is most closely coupled with the Business Data Warehouse (BDW), Business Data Directory (BDD), and user profile. In particular, the BDD and the user profile are essential in generating the menus and options that are presented to the user. The internal architecture of the EUI and its relationship to the three EBIS components just given are shown in Figure 7.

The EUI provides services to the end user in five general categories. Such services are provided in varying degrees of completeness by such program products as QMF, AS, and IC/1.

Data services provide the user with the language or languages used to access the information in the BDW or BDD. These languages may be conceptually divided into the following three categories:

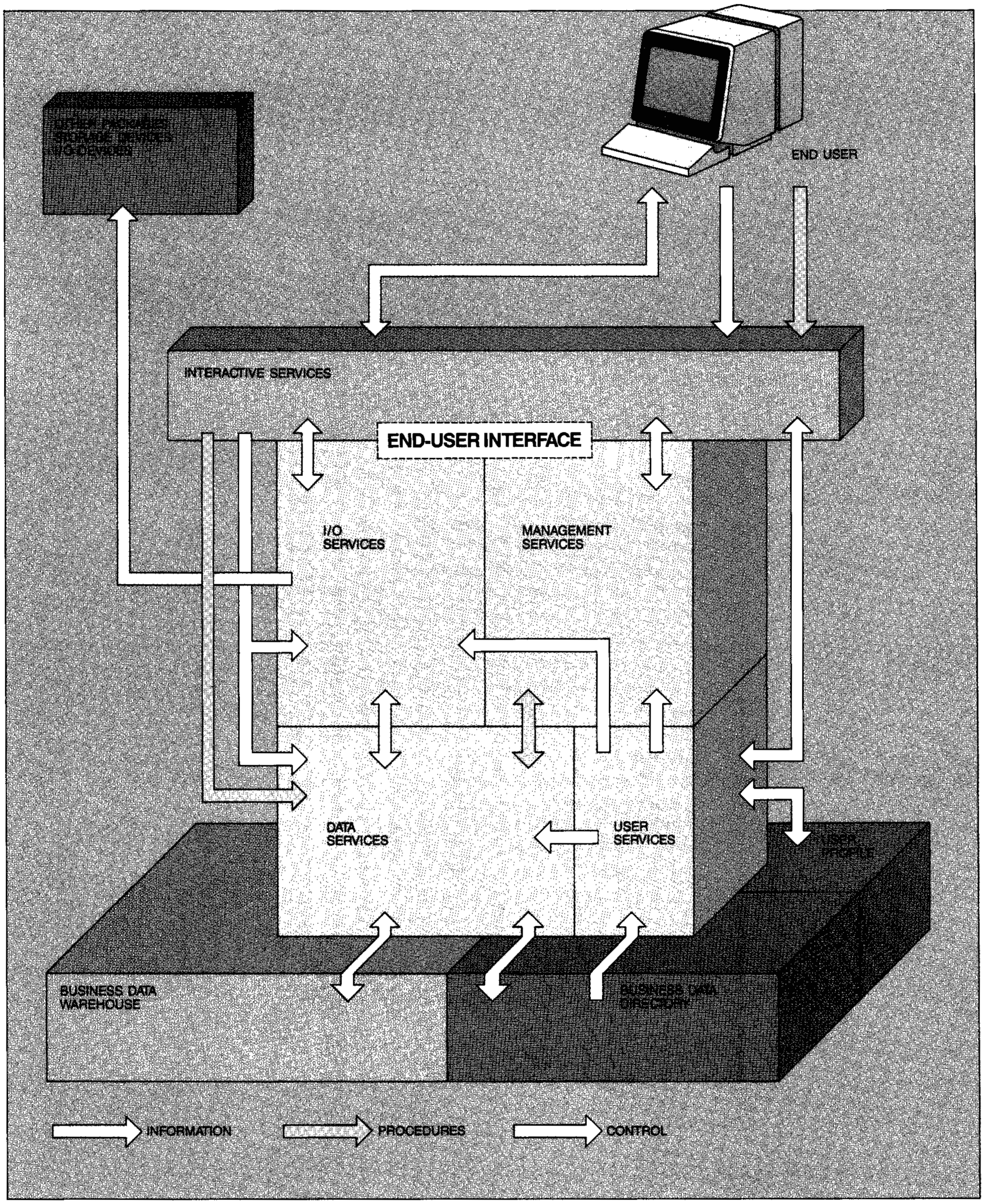
- *Data-retrieval languages* communicate directly with the BDW and BDD and allow selection (or insertion) of data elements according to logical conditions. The process of selection of data is a query.
- *Data-manipulation languages* act on the data provided by a data-retrieval language and perform arithmetic and statistical functions on the retrieved data. Examples include addition of data elements, averages, summations, etc.
- *Data-formatting languages* act on the data resulting from either of the previous two steps and allow the result to be formatted as required for printing, for inclusion in documents, or for input to other programs.

For example, SQL and QMF together address all three categories. A *procedure* is a combination of steps to produce information in a form that is useful to the user.

Input/output services provide interfaces between the data services component and the user via an interactive services component and directly with physical I/O devices, storage devices, or such packages as graphics or office systems.

Management services are responsible for stored procedures in the EUI. These services include saving new procedures or queries at the request of end users, managing the availability of procedures on a company-wide, group, and personal basis, and scheduling the execution of procedures.

Figure 7 Components of the End-User Interface



User services provide miscellaneous facilities required by the user. The user may access or update his user profile using this interface. Default values stored in the user profile are passed to the other EUI components through a user profile service. Descriptive information stored in the BDD is made available to the other EUI components via the BDD service. This is particularly important to interactive services, where this information is used in constructing menus, and to data services, where the information is used in enhancing reports. Help screens are presented to the user via the help service. System news is made available to the user through the news service. The problem-reporting service provides users with a means of reporting problems and following up on any action taken.

Interactive services provide the interface between the user and the other subcomponents of the EUI. They are also used by components of EBIS that transmit information to or from the user. These services are the following: general menu handling, access to procedure-writing languages provided in data services, menu-driven procedure writing, fast-path inquiry, command interpretation and parsing, and parameter input prompting for procedures.

Access to data and procedures. All data in the BDW, regardless of their source, should be available through identical methods. Public data and personal data should be presented in a similar manner to facilitate comparisons by the user. Users need access to data in many different ways. From the end-user viewpoint, this may mean that data may be required in such formats as a simple table, a complex formatted report, a graph or chart, output of a local or central printer, part of a note or document, or formatted input for a transaction that is to be applied to an operational system. All of these views of the output information are produced by a procedure that runs a query against the BDW and manipulates and formats the result.

From a user viewpoint, procedures may be divided into two categories. The first are standard procedures that are run regularly with either identical or changed parameters. These may often be scheduled in advance. A small number of users require the ability to construct their own queries. The need to make *ad hoc* enquiries is a growing one and is beginning to extend to users who are not data processing professionals.

Interface to operational systems. The BDW is updated regularly with the changes to the operational systems. The manner and frequency of updates are determined by business needs. If frequent updates are required, only the changes in the operational systems databases are applied to the BDW by a table update procedure in order to reduce the volume of data to be transferred and the processing time necessary to do the update. On a longer-term basis, a reconciliation ensures that the BDW is still in agreement with the operational systems. The relevant parts of BDW are then overwritten with a new copy by a table load procedure from the operational system. If update is infrequent, or the table update method is not suitable for the data being transferred, the table load method is the only one used.

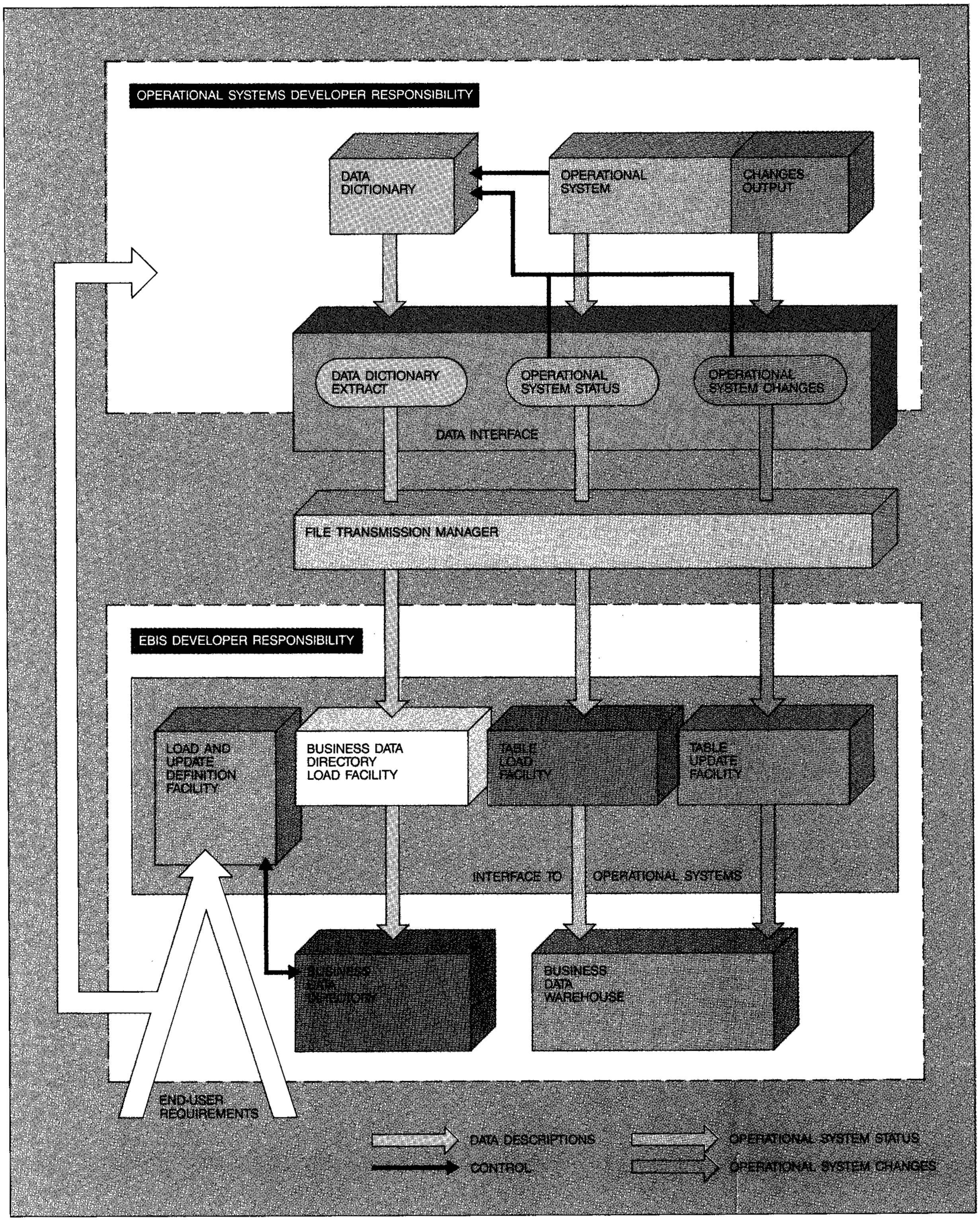
The transfer of information from the operational systems to the management information system requires the close cooperation of three groups of people. The developers of both the operational systems and the management information system must be aware of the requirements of both systems and must cooperate in the development of the interface between them. The third group in the process—the end users—must define the information that is required in the management information system. The EBIS strategy for interfacing with operational systems divides the responsibility for the interface in a logical manner among these three interested groups. Their differing responsibilities are expressed in the definition, provision, and reception of the contents of a *data interface.*

The data interface is the boundary between operational system and informational system responsibilities in the EBIS architecture. Any operational system that provides data to EBIS does so through the data interface.

Figure 8 shows that the data interface consists of the following three parts:

- *Operational system status* is an internally consistent, point-in-time image of the databases of the operational system. Physically, it is a sequential file extracted from the operational systems databases using a method provided by the operational systems developers.
- *Operational system changes* are a complete record of all updates that have been made in the operational systems databases since the previous status or changes extract. The operational systems changes are also a sequential file constructed by using a method provided by the operational systems developer. Ideally, the records in the changes

Figure 8 Business Data Warehouse update strategy



should be aligned with well-defined business transactions.

- *Data Dictionary extract* is a point-in-time image of the information in the Data Dictionary that describes the operational systems and the operational systems status and changes in the data interface. The Data Dictionary extract is a sequential file constructed by using a method provided by the operational systems developer.

Data reception in EBIS. Data presented to EBIS are entered into the BDW through one of two subcomponents, either the *table load facility* or the *table update facility.* Information from the Data Dictionary extract file is entered into the Business Data Directory (BDD) through the BDD load facility. The control information that drives these components has previously been constructed through a *load and update definition facility.*

The table load facility receives operational systems status information and loads it into the BDW, overwriting any existing copy of the data. The mapping of fields in the status file to tables and columns in the BDW is obtained from the control information previously defined through the load and update definition facility. A reconciliation procedure reports the existence of incompatibilities between the previous and current versions of the information.

The table update facility receives operational systems changes and updates the tables in the BDW by inserting new rows, deleting existing rows, and updating existing rows as appropriate. The correspondence between fields in the changes file and the tables and columns in the BDW is established from the control information previously defined through the load-and-update definition facility. Both the load and the update facilities are automatically triggered by the arrival of status or changes files in EBIS.

The BDD load facility receives DD extract information and reloads tables in the BDD whenever new information is available. The mapping of fields in the DD extract file to tables and columns in the BDD is obtained from the control information previously defined by EBIS.

The load and update definition facility is a tool used by functional or cross-functional support specialists to define the tables and columns in the BDW and their relationship to the status and changes files that feed them. The load and update definition facility obtains descriptions of the operational systems status and changes files from the BDD. Descriptions of the tables and columns in the BDW are also obtained from the BDD. This information is presented to the support specialist, who can choose which fields to extract and the tables and columns into which they should be loaded. The support specialist is thus able to define a subset of the operational data that are required by users of EBIS and to change that subset as necessary. Provided that all the required data are available in the data interface, this can be achieved without raising requirements on operational systems or EBIS developers.

For the first time, the end user is given the benefit of the information stored in the Data Dictionary.

The original source of the information that describes the operational systems fields is the operational systems Data Dictionary. Because the support specialist uses this information in the load and update definition facility, it is vital that the operational systems field descriptions are maintained and managed using the full power of a Data Dictionary product. In addition, the descriptive text provided must be easily understood by the support specialist. These points suggest that the current role of the Data Dictionary will be substantially expanded by the requirements of the interface to operational systems.

Business Data Directory. The Business Data Directory (BDD) stores and manages the descriptions of the data stored in the BDW, the information concerning stored procedures, and the descriptions of business processes.

The BDD is a key component in the EBIS architecture, because it communicates with the end-user interface, the interface to operational systems, the Data Distribution Manager, and the Data Enhancement Manager.

The BDD serves two distinct functions for the end user.

Help function. The BDD provides a method of locating information, understanding its meaning, and finding the query needed to analyze it. In addition, the BDD aids in understanding the result of the query by providing descriptive information on the fields that are returned, on coded values in those fields, and on the currency or ownership of the data received. Such information might also be incorporated into a report. Thus, for the first time, the user is given the benefit of the information stored in the Data Dictionary that until now has been available only to the developers of large systems.

Menu function. The BDD plays a key role in the menu system of the end-user interface. The names, descriptions, owners, etc. of predefined procedures and queries are stored in the BDD. Menus that allow the end user to select an inquiry have their source in the BDD and are controlled by the information in the BDD. This restricts the queries to those owned by or capable of being accessed by the end user.

In the process of creating user-defined queries, the End-User Interface (EUI) presents the list of fields in a table for use by the menu system. This list is obtained from the BDD. Other information in the BDD restricts the range of values that may be entered for certain fields or ensures that key fields in a relationship are not omitted by the end user. In this sense, the BDD is similar to a knowledge base for an expert system, guiding the user in building his queries.[2] A similar use of such information is part of the "SMARTY" project.[3]

The following information appears in the BDD:

- Descriptions of tables, columns, procedures, and parameters for stored procedures
- Possible values for coded fields
- Date of last update of information
- Source of information
- Ownership of information
- Method of derivation of information

The primary source of the data in the BDD is the Data Dictionary, which should accompany the operational system. Because the Data Dictionary changes infrequently, the BDD is not updated regularly but is reloaded only when new information becomes available.

The operational systems Data Dictionary provides the descriptions, data types, and field lengths of the data items, which bear a rough correspondence to the columns or fields in the database. That dictionary also defines the relationships of these items to one another. It is the responsibility of the operational systems developer to provide this information.

Some information in the BDD is generated locally in EBIS, and this is updated as required. Descriptions of tables and views are maintained only in the BDD. This is done by the data administrator.

There are two occasions when data in the BDD must be maintained by a user. When the user creates his own views, tables, and tasks, he is requested to document them in the same manner as the corresponding public items. Thus the BDD is partitioned into two areas, public and personal. The personal area contains the personal descriptions. As a result, the user may develop applications that are easily transferred into the public domain if necessary.

Within any user group, there is information that is local to and shared among members of that group. Such information does not reside in the operational Data Dictionary. It is maintained by a data specialist within that group through this facility of the BDD.

Security and profile manager. Because EBIS funnels the business information to the professional and managerial end users for making decisions, it is vital for EBIS to closely control and manage the data it provides. The security of the BDW must be at least as good as that of the operational systems from which it derives its data. This is the case because the level of aggregation of the information is higher. Also, because the information is stored in such a way as to be easily understood by the intended user, it can be easily understood by others.

Security. Thus, the access of any particular user to the system is carefully controlled and restricted at a number of levels. Access to the entire EBIS system is controlled as in any other application and is managed by the overall computer security mechanism. Access may be restricted to particular parts of the EBIS system at the EUI on the basis of access tables maintained by a security and profile manager. Access to stored procedures may be controlled by the EUI as well. Access to data in the BDW may be restricted and controlled by the database manager through the use of views. The creation and maintenance of these views is achieved through the security and profile manager.

User profile. The purpose of the user profile is to define the characteristics of the user to the EBIS system. This profile consists of two parts. One part contains the security and access details for the user and cannot be changed by the user. The other part allows the user to define the preferred appearance of the system.

Data Enhancement Manager. The Data Enhancement Manager (DEM) manages the enhancement of public data in the BDW. The DEM is responsible for ensuring that predefined and scheduled procedures are run at the correct time, when all necessary data are available. The DEM provides authorized users with a facility whereby they can identify a procedure that is to be executed. The DEM also defines when a procedure should run, whether it should run regularly and if so, how often, and defines the prerequisites for program executions. The DEM procedures perform three types of functions. *Data consolidation* involves the generation of summary data according to defined business rules. *Data enrichment* is the combining of various data items extracted from the operational systems to create new information that is of general interest. *System data updating* in the BDD reflects changes that have taken place.

In relational terms, data consolidation involves the creation of new tables. The rows in these tables contain information that is aggregated from a number of rows from the raw data tables. For example, from the orders data, the total number and value for each machine type and model may be calculated from the raw data and stored in a summary table, with one row for each machine type and model.

For data enrichment, the relational process is that of adding extra columns to the original tables. Each value in the new columns is calculated from a number of items in the corresponding rows of the original tables.

Data Distribution Manager (DDM). In a group of companies or a single company with widely scattered locations or divisions, where computing power is likewise widely distributed geographically, the source of a piece of information is, in many cases, remote from the location that needs it. In the long term, distributed databases may address this problem by allowing queries executed in one location to access data in any database, regardless of location.[4]

However, even if data locations are transparent to queries, it is still essential to bring frequently accessed data as close as possible to the users. This is necessary to ensure good performance for data retrieval. Thus, the second key consideration for a distributed database is a copy-management mechanism.

A strategic, long-term direction of the EBIS architecture is to support a distributed database environment. Three components of the architecture are involved. The EUI supports queries that are independent of the location of the data, and the BDD supports the concept of multiple copies of the business information within EBIS. The DDM is responsible for the copy management of distributed data.

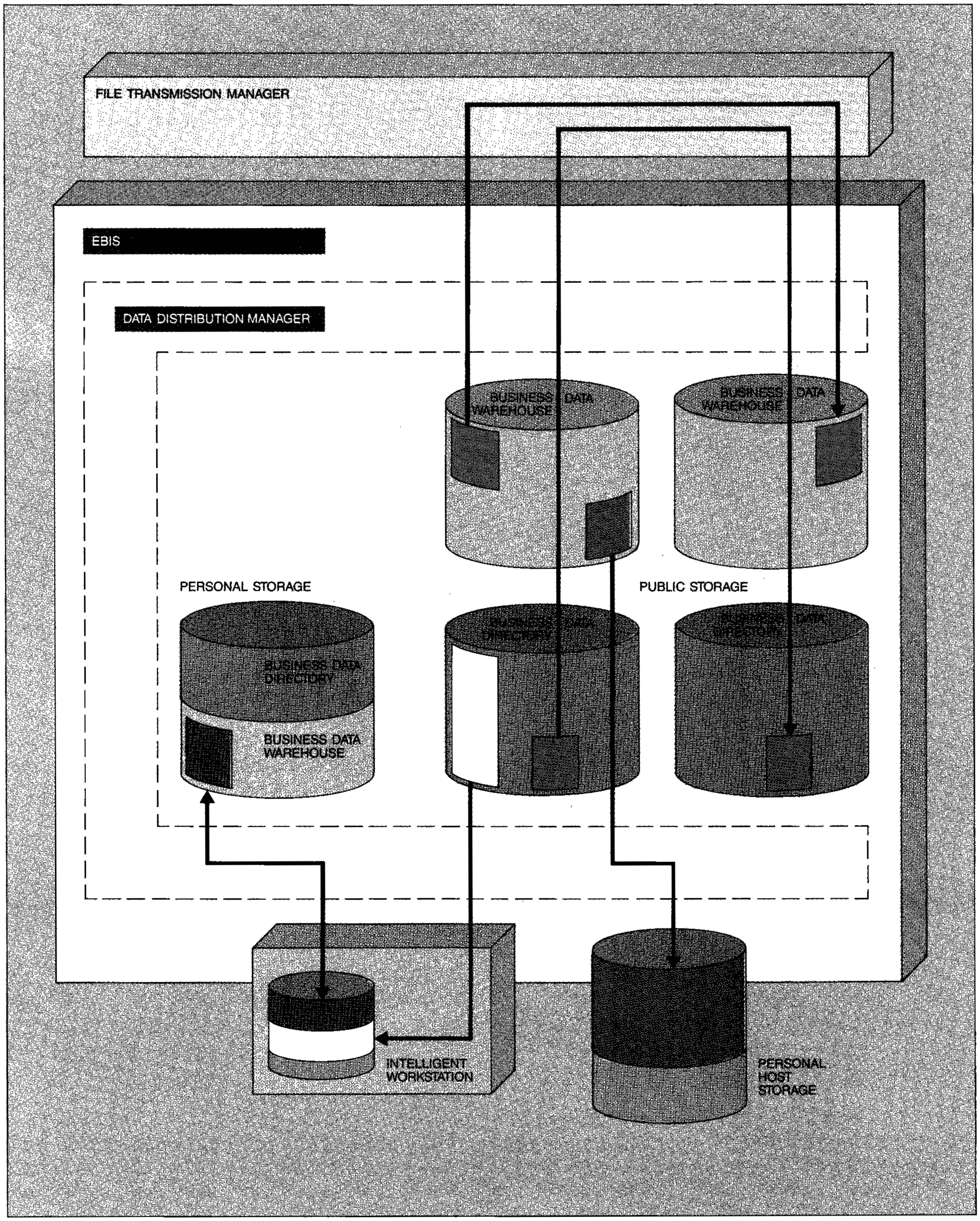
Figure 9 shows that public data and personal data in the BDW may be distributed, as may the information contained in the BDD.

End users may extract data from the master BDW and BDD tables and store the information locally, on an Intelligent Workstation (IWS), a file server, or personal storage on the host. The user informs the DDM if it is necessary to ensure the currency of the copy.

At the simplest level, ensuring currency of information involves merely reminding the user when the copy is out of date and giving an opportunity to reload the data. Automatic update is also possible, especially if the copy resides on personal host storage or on a file server, both of which are continuously available to the host. EBIS supports the concept of local or functional host processors serving a small group of end users with all the information they require. The master database receives information from the operational systems and then distributes parts of the data to other slave databases. Because public data cannot be altered by the end user, it is not necessary to transfer from slave to master. Thus, a transfer is a one-way event for a particular set of information. Copies of information in the BDD are also distributed through this mechanism. Copy management of public data is performed automatically, and its frequency determined by business needs.

It is useful to consider the particular benefits of intelligent workstation (IWS) usage in EBIS in terms of the physical IWS environment. Such consideration provides a basis for determining whether a particular function can be enhanced by the IWS. For example, the IWS provides personal and dedicated processing power to the end user. Also, local disk storage provides fast access to data. Session management facilities on the IWS allow concurrent access to multiple host systems. The IWS provides fast display refresh

Figure 9 Distributed Data Storage



and scrolling coupled with active keyboard scanning when processing locally. These facilities give users a fast, versatile interface to such functions as editing and graphics in a manner different from that provided through host/fixed-function terminal attachment.

The foregoing points indicate three functions that are most suitable for distribution to the IWS: (1) BDW support for personal data storage; (2) EUI support for menus, report manipulation, help information, and graphics; and (3) BDD support for local storage of regularly used descriptive information.

The trend toward solutions based on cooperative processing is exemplified by the Enhanced Connectivity Facility.

In a cooperative processing environment, the host and the IWS work together to provide data to satisfy a request. Thus, a request on the IWS for data that do not reside there is routed to the host computer. It is not necessary for the user to be aware of whether data reside on the IWS or on the host. In the interim, support for the IWS in the EBIS architecture is expected to develop in line with the current directions in cooperative processing.[5]

This trend toward solutions based on cooperative processing is exemplified by the Enhanced Connectivity Facility (ECF) products, which allow the effective sharing of resources and the transfer of data between the IWS and host environments.[6,7]

All the business information is available in the BDW. Any subset of that information can reside on any storage device, provided that device can be properly managed and controlled. Because the hard disk on the IWS is owned by and primarily managed by the user, it is apparent that only personal data or personal copies of public data can be stored there. When a user makes an enquiry on the BDW, a result table is returned that may be saved in the user's personal storage on the IWS. The management of such information is solely the responsibility of the user.

There is also a requirement, at least potentially, for a copy of a subset of public data to be supplied on a regular basis to a user's fixed disk. In that case, the copy management of such information would be the responsibility of EBIS; that is, a new version would be sent to the IWS only when the master copy on the host changes.

The primary advantage of such data distribution is the utilization of processing power so that the end user gains performance benefits when accessing data. The partitioning of the data into small subsets, where appropriate, speeds data access and thus improves response time for the end user. A second advantage is in making the data available to IWS-specific tools. However, these advantages must be balanced against the load that data distribution places on the network.

The interconnection of Intelligent Workstations via a Local-Area Network (LAN) allows the development of small departmental groups of users. Such users require frequent access to a common pool of information. The availability of a file server on the LAN allows such a set of data to be provided locally. The file server is thus an important staging post in the distribution of data to IWS-based users. It significantly reduces network traffic compared to the equivalent situation where each IWS user receives a similar subset of the company data.

Concluding remarks

EBIS is a comprehensive architecture aimed at providing a cross-functional business information system that is easy to use and that has the flexibility to change as the business environment develops, even at a rapid rate. The flexibility and crossfunctional support are a result of the relational database technology on which the EBIS system is based.

Relational database technology (DB2 and SQL/DS) is the strategic direction for data management in informational systems. As relational database products evolve functionally, the EBIS architecture will allow for a gradual extension so as to include full support for the Intelligent Workstation, multiple file servers, and multiple database machines.

Cited references

1. L. M. Branscomb and J. C. Thomas, "Ease of use: A system design challenge," *IBM Systems Journal* **23,** No. 3, 224–235 (1984).
2. See *IBM Systems Journal* **25,** No. 2 (1986), a special issue on knowledge-based systems.
3. P. Stecher and P. Hellemaa, "An 'intelligent' extraction and aggregation tool for company data bases," *Decision Support Systems* **2,** No. 2, 145–158 (1986).
4. R. A. Yost, "R* Overview," *Share 64.5*, Toronto, Ontario, Canada (1985); see also M. W. Blasgen et al., "System R: An architectural overview," *IBM Systems Journal* **20,** No. 1, 41–62 (1981).
5. B. C. Goldstein, A. R. Heller, F. H. Moss, and L. Wladawsky-Berger, "Directions in cooperative processing between workstations and hosts," *IBM Systems Journal* **23,** No. 3, 236–244 (1984).
6. P. N. Parr, J. S. Auerbach, and B. C. Goldstein, "Distributed processing involving personal computers and mainframe hosts," *IEEE Journal on Selected Areas of Communications* **SAC-3,** No. 3, 479–489 (1985).
7. B. C. Goldstein and J. M. Jaffe, "Data communications: The implications of communication systems for protocol design," *IBM Systems Journal* **26,** No. 1, 122–137 (1987).

General reference

The Management of End-User Computing, GBOF-1689, IBM Corporation; available through IBM branch offices.

Barry A. Devlin *IBM Ireland Information Services, Ltd., 2 Burlington Road, Dublin 4, Ireland.* Dr. Devlin joined IBM in 1985 at the Dublin International Development Centre, where he became technical architect for the EBIS architecture. He is currently involved in developing an architecture for supporting electronic communication between IBM and customers. Before joining IBM, Dr. Devlin worked for four years in the University College, Dublin Computing Centre, in the support, development, and installation of end-user computing facilities in a variety of environments, including MS-DOS, DEC-20, MUSIC, and VM/CMS. Dr. Devlin earned his B.Sc. and Ph.D. degrees in solid-state chemistry at the University College, Dublin. He has published a number of papers in the field of ionic conductivity in silver halides.

Paul T. Murphy *IBM Ireland Information Services, Ltd., 2 Burlington Road, Dublin 4, Ireland.* Mr. Murphy is manager of the Strategy and Architecture group at the Dublin International Development Centre. That group has prime responsibility for the definition and support of data architecture and data management within IBM Europe. Mr. Murphy joined IBM Ireland in 1976 as a systems engineer. In 1980, he was on assignment to the IBM UK Field Support Centre as the IBM UK DOS/VSE specialist. Mr. Murphy has also worked on relational database performance at the IBM World Trade Systems Centre in Boeblingen, Germany. In 1983, he was appointed manager of the European Office Support Group at the Dublin International Development Centre. Mr. Murphy has a B.E. degree in electrical engineering and an M.E. degree in engineering science for research into digital filters, both from the University College, Dublin.

Reprint Order No. G321-5311.