Proceedings of the USPS 4th Advanced Technology Conference,
Automation Technology Workshop, Washington, D.C.,
November 5-7, 1990.

An Approach to a System Architecture for Post Office Automation

James S. Albus, Edward Barkmeyer, and Albert Jones
National Institute of Standards and Technology
Gaithersburg, MD 20899
USA

ABSTRACT

The National Institute of Standards and Technology (NIST) is involved in designing a systems architecture for a letter-sorting automation project for the United States Postal System. This paper describes the approach being used to develop that architecture. It gives NIST's view of a general systems architecture and outlines our effort to apply that general view to the specific problem of letter sorting.

1. INTRODUCTION

The United States Postal System (USPS) has embarked on a long-term automation program. The objective of this automation program is to increase efficiency and reduce the cost of mail handling and sorting operations. The first phase of this program focused on the automation of the various sorting operations at General Mail Facilities (GMF). This included the use of Optical Character Readers (OCR) to read addresses and bar code envelopes, Bar Code Sorters (BCS) to sort letter mail, a Flats Sorter, and a Small Bundles Sorter among others.

A second phase, which concentrates on letter processing only, has recently begun. It has two major goals: 1) to automate the movement of letter trays within the GMF, and 2) to automate the letter-sorting decision making process. The former requires the development of hardware and control software for a rapid tray transport system. This system must be capable of moving trays from one place in the facility to another. The latter requires the development of a system to plan, schedule, and control all letter sorting activities.

To meet both of these goals, it will be necessary to design an "intelligent" system which effectively integrates tray movement with the rest of the sorting decisions. NIST is designing an architecture which will make this integration possible. Our view is that mail processing and handling operations are very similar, at least conceptually, to part processing and handling operations in a manufacturing facility. Consequently, we intend to use the expertise gained in building the Automated Manufacturing Research Facility (AMRF) project [SIM82] to build that architecture.

2. BACKGROUND

For purposes of this paper, we define an intelligent system as one which acts appropriately in a dynamic, uncertain environment. Appropriate action is defined as that which maximizes some utility function. This utility function is defined by the user and represents his view of the tradeoffs among several cost/ benefit criteria. It is well known that, in the postal system, these criteria change as the state of the environment changes. An example of this is that the priority given to sorting a specific set of mail changes 1) as the scheduled departure time for that mail draws near, 2) as the volume of mail in different stages of processing fluctuates, and 3) as the availability of resources varies. This means that the choice for the most appropriate action will depend on the state of the environment. To deal with this time-dependence, an intelligent system must, at the very least, determine what is happening on the facility floor and respond to it as quickly as possible. In our opinion, it must do more than react, it must plan.

Our view of an intelligent system is depicted in Figure 1. As shown, it contains three basic elements: sensing, understanding, and acting. Sensing involves both measurement and processing of sensory data. Understanding involves the development and maintenance of a "world model" of the state of the environment. This includes updating the model from sensory data, and using the model to simulate plans and determine deviations from planned future events. Acting involves using the current world model to 1) predict future events, 2) control the activities of machines or human workers so that those events happen as predicted, and 3) react to deviations from predicted events.

In Figure 2, we indicate how one might apply these concepts to the letter-sorting environment. Sensors are required on most automated equipment. Some examples include address readers in the OCR, bar-code readers in the BCS, position sensors, and devices that indicate the position and identification of trays, conveyors, mail bags, trucks, planes etc. The world model contains a representation of the current "state" of the post office. This might include a best estimate of the status of all machines and people; the current task they are performing; and the position, content, and destination of trays, conveyors, and carriers. Action consists of the decisions now in effect and the future events to occur as a result of those decisions. Some examples include the current sort scheme for each OCR and BCS, scheduled completion times for each machine, tray movement schedule, and personnel assignments.

NIST has had a great deal of experience in applying these concepts to several different physical systems. As noted above, the most extensive application has been to manufacturing and robotics. We have generalized our results to many other systems including the NASA space station [ALB89a], autonomous vehicles [ALB88], and coal mining automation [ALB89b], among others. We have found it convenient to partition the entire intelligent system into three subsystems: resource management, data management, and communications management. Each of these subsystems is typically distributed over a large number of computing modules that may be widely separated in space.

2.1 Resource Management

It is important to determine the resources that must be managed for a given application. Usually, these resources can be classified into three groups: machines to carry out the processing, objects to be processed, and devices to move objects from one process to another. The definition of process, object, and transporter is system dependent. In the postal system, the objects are letters; the processes are facing, canceling, bar coding, and sorting; and the transportation devices can be conveyors, monorail carriers, or humans. Management includes those functions needed to plan, schedule, and control these resources to meet some user-defined objectives. NIST has long advocated the use of a hierarchical structure (see Figure 3) to carry out these functions. These structures have several interesting properties:

- Levels are introduced to reduce complexity and limit responsibility and authority,
- Each level has several modules which make the similar decisions and perform similar functions
- Each level uses information that decreases in granularity as one goes up the hierarchy
- Each level has a distinct planning horizon which decreases as you go down the hierarchy

Each module in this structure is simultaneously a supervisor to several subordinates and

subordinate to a single supervisor. Each module decomposes input commands from its supervisor into procedures to be executed at that level and subcommands to be issued to one or more subordinate modules. This command (or task) decomposition process is repeated until a sequence of tasks is generated which actuates processing equipment [ALB81]. The status feedback that is provided to supervisors by their subordinates is used to update the world model and to close the control loop by supporting adaptive, real-time, decision making. We note that it is only at the lowest level that feedback comes directly from sensors mounted on the equipment.

It is important to realize that human operators are, and will remain, an integral and important part of even the most advanced and automated facilities. Operator displays must be provided so that human operators can easily understand what is going on at all times and at all levels in the hierarchy. Operator input devices must be provided so that human operators can easily work with and, if necessary, override the system at a variety of levels. This can be done by entering data, commands, new priorities or schedules, and generating new plans.

From the preceding discussion, it is clear that the development of a hierarchical structure for a particular application requires a simultaneous decomposition of time, space, and decisions and an aggregation (or resolution) of information. To do this, one must answer the following questions for both the entire system and for each of its levels:

1. What is the temporal span and resolution?

   This affects the communication bandwidth and latency, control loop bandwidth, control cycle periodicity, data sampling intervals, and sensory processing integration periods.

2. What is the spatial span and resolution?

   This applies to scale and size of maps representing geometric details about floor layout, cart and conveyor routing, and transportation routing.

3. What is the control unit span and resolution?

   This applies to how many subsystems are under control of each supervisor, what decisions they can make, and to what level of detail subsystem commands are decomposed.

As a rule of thumb, as one goes up the hierarchy span increases, and resolution decreases about an order of magnitude per level. At any level, thetemporal and spatial span is typically between 100 and 1000 resolution elements.

2.2 Data Management

The main purpose of the data management system is to support functions in the resource management hierarchy with timely access to all essential data. In our view, it is essential to design and implement a data management architecture which is separate from but integrated with the resource management architecture. Such an architecture must address three major concerns: data modeling, database design, and data services. We give a brief description here; more details can be found in [JON89].

The first step is to develop a "conceptual model" of all the information required to execute

the resource management functions. Experts on individual functions in that architecture will perform the analysis and develop a conceptual information model for each functional area. Then the resulting "component" models must be integrated into a single "enterprise model". The enterprise model is the conceptual representation of the global information base.

Having arrived at a global enterprise model, we are then faced with the problem of mapping this model onto live databases. This process is called database design. It must result in databases which are consistent with the model and tuned to the timing and access requirements of the resource management functions that use them. Because of the evolutionary nature of large, complex systems, it is not always possible to start this process from scratch. There may be databases and data systems required by or embedded in component systems, which are difficult to alter or augment. There may also be large reservoirs of previously developed information which already have an imposed organization.

The data services portion of the data management architecture provides access to shared information. It is unreasonable to expect a common data service to provide the kinds and speed of service required at all levels of the resource management hierarchy. In most systems multiple data access and management systems will exist. What is important is that their interrelation is understood and coordinated. There are several important aspects of data services:

- "query processing" which is concerned with the interface between resource managers and information

- "transaction management" which is concerned with identification of information sources, scheduling, resolving conflicts among competing uses of related information

- "data manipulation" which is concerned with the actual execution of the operations on the stored information.

There are three architectures for data services which have been used with varying degrees of success in various business applications: centralized data and control, distributed data and control, and hybrid systems.

The totally centralized approach is the traditional design. There are now available high-speed, internally redundant, fault-tolerant, integrated centralized database management systems. But, even if such systems can keep pace with the growing demands and time constraints of automated systems, the centralized architecture is not workable from the point-of-view of subsystem autonomy. Consequently, the nominally centralized architecture will in fact consist of a collection of autonomous systems with their own, sometimes primitive, data services copying information to and from a single centralized facility, according to some externally-specified plan.

The canonical architecture for the totally distributed approach (Figure 4) consists of local data services which process locally originated and locally satisfiable requests, and negotiate with each other to process all other requests. The problems with this architecture lie in the transaction management area, and are related to the coordination of the separate systems.

The hybrid architecture depicted in Figure 5 attempts to combine the best features of both centralized and distributed architectures. Subsystem autonomy and high throughput are achieved by allowing local data services to process locally originated operations on local data. Information which transcends the scope of a local system is integrated by a

centralized "global query processor" which supervises the distribution to the appropriate sites. The global query processor acts as a central arbiter for resolving the characteristic problems of distributed control of information and for handling configuration changes in the available data services.

2.3 Communications Management

To understand communication, it is necessary to distinguish between resource managers or information managers and the platforms they run on. The former are primarily software objects called "processes". The latter are primarily hardware objects called "computer systems". On many computer systems, it is now possible to run more than one process simultaneously. Furthermore, it is often possible to run the same process on two separate, not necessarily identical, computer systems. The responsibility of the communications management system is to make it possible for processes to communicate with one another no matter where they reside. Communications can be divided into two classes: those WITHIN computer systems, and those ACROSS computer systems. The first type is often referred to as "interprocess communication" while the second is often called "network communication".

Interprocess communication is dependent on features of the local operating system, which vary considerably. On some systems the coordination of multiple resource management and data management activities is extremely difficult. The proper solution for the future is to make local interprocess communication a special case handled by the network communications software. This solution has the added advantage that all communications by a resource management or data management process, regardless of the location of the correspondent process, has the SAME interface.

The accepted paradigm for network communication is the Open Systems Interconnection Reference Model (OSI) [DAY83]. Before the development of the OSI model, "physical" and "logical" communication were equated. Physical communication is the transfer of signals between computer systems. Logical communication is the transfer of information between software processes. The important aspect of the OSI model is that it formalizes and separates the logical process-to-process link (upper layers) from the physical network service considerations (lower layers).

A great deal of flexibility is created by implementing the OSI model. On one hand, a single physical medium can multiplex many separate process-to-process communications. On the other hand, a given process-to-process connection can use several separate physical connections with relays between them. This gives rise to the general network architecture shown in Figure 6. Ideally, all stations on the network implement common OSI protocol suites in the intermediate layers and upper layers to assure that resource and data management process can talk to each other. The choices of protocol suites in the bottom two layers and the connectivity of individual stations will vary. They will depend on the physical arrangement and capabilities of the individual stations, and their functional assignments and performance requirements. There may be one physical network, or many. All of the separate physical networks, however, must be linked together by "bridges" implementing the proper protocols. This results in a SINGLE LOGICAL NETWORK on which any given resource management or data management process can connect to any other process regardless of location. Because the architecture is layered, the physical network and all of its subnetworks become transparent to the logical communication among processes.

3. APPLICATION TO THE LETTER SORTING OPERATIONS

Recall that our objective is to develop a system architecture which integrates all letter sorting operations. These include not only the physical sorting and movement operations, but also the decisions that are required to initiate those operations. Following the approach described in the preceding section, we have divided this problem into three subproblems: an architecture for resource management, and architecture for data management, and an architecture for communications management. In the ensuing sections we outline the issues that must be resolved to design these architectures.

3.1 Resource Management Issues

Based on the discussion in section 2.1, resource management decisions will be made inside a multi-level hierarchy. The following decisions must be made to utilize the resources in the postal facility effectively:

1) start time, end time, and selection of sort schemes
2) the time to be moved and the destination of a tray that has just finished an operation
3) the time to be moved and the destination of every tray in storage
4) route selection for each tray movement
5) processing order at each operation
6) personnel assignments and durations

To determine the structure of the hierarchical resource management architecture we must first specify the number of levels, information flow between levels, and the decisions and functions assigned to each level. It will also be necessary to develop algorithms and scheduling rules for those decisions. We must then identify the information from the world model that is needed to reach such decisions. Since these decisions must be made in "real-time," performance issues related to communications must be resolved at each level. Finally, we must describe how each type of equipment (existing and future) should be integrated into the architecture.

Many of the decisions made by this hierarchy govern the movement of trays around the facility. Since it is likely that the tray transportation system, both hardware and control software, will be procured from an outside vendor, it is important to specify how that system will be integrated into the resource management hierarchy. We note three things. First, it is important to identify the decisions appropriate to transport in general. Second, we must identify additional decisions that may be transportation-system specific. For example, if a monorail system is used, it will be necessary to choose a specific carrier to move each tray. Finally, we must ensure that decisions made by the tray transport controller are clearly separated from those made by higher levels in the hierarchy.

We noted earlier that a world model is used at each level to 1) aid in monitoring the execution of the tasks assigned to lower levels, and 2) make decisions about what to do next. The model used by a given module consists of an aggregation of the information in the models used by that module's subordinates. This aggregation is based on the feedback provided by subordinates and takes place throughout the entire hierarchy. As the system evolves in time, the state of the systems changes. Consequently, so must these world models. To do this, we must first determine the information contents and data structures for the world models at every level in the hierarchy. Once we know what the models look like, we must then develop real-time analysis software to update the models from the sensory information and other feedback provided by the subordinates.

The bottom level in this hierarchy will contain modules which are responsible for managing

the activities at one or more processing stations. Once the decision has been made about what to do at a given station, the module must implement that decision and monitor its execution. As we discussed, this requires the use of a world model. World models at this level are updated directly from operator inputs and the sensors on the equipment. As indicated, our previous information analysis will tell us what information is needed to complete the world model at the lowest level. Engineering and ergonomics will dictate what data can be gathered from each particular station. From these, appropriate sensory or operator input devices can be designed for each station. Such devices would then be directly connected, via communication link, to the corresponding controller. Algorithms must then be developed which can transform that data into the information that is needed to update the world model.

Similarly, higher-level controllers, primarily engaged in planning and scheduling, must implement decisions by monitoring status reports from subordinates and updating their world models. At these levels, some world model inputs will require residual databases to store chronological information such as truck departure/arrival times and personnel availability.

The final component to be included in the resource management architecture is the human supervisor. In the postal system, this is critical since the entire operation is, and will continue to be, human intensive. Therefore, we must provide the supervisors and superintendents with access to the various levels in the control and information systems. We must decide on appropriate displays and input devices. We must also determine what decisions the operators will be allowed to override. Finally, since most workers have little exposure to computer technology, a training program will be required.

3.2 Data Management Issues

In the preceding section, we identified the need to collect several types of information to support functions in the resource management architecture. Some information units are collected and used locally and need not be shared with other controllers (sensor data for example). Others will be collected and passed on to supervisors and other controllers (mail volumes for example). Still others will be stored and used at some later time (mail volume of incoming trucks for example).

Currently, the Real-time Productivity Management System (RPMS) is capturing personnel data but little else. In addition, the USPS is developing a new technology to reduce the amount of human reading and sorting. This new technology is video-based and may require some type of information storage. We design a data management system to store, update, and disseminate all shared information. The system should perform these functions in a manner which is accurate, timely, and completely transparent to the users (both people and computer programs).

The first step in developing an architecture is to specify the information units that are needed to make and execute decisions and to specify the relationships among these information units. Then we must identify the producers and consumers of those units. We may also have to identify related storage requirements and representation forms for existing systems like RPMS. The next problem is to determine a distribution strategy for the information units including organization, location and access or interchange methods, taking into account limitations imposed by existing systems. The final step is to develop the designed information base and to build or acquire the data access services.

Based on our current findings, we expect the information base for a mail facility to be quite modest in both size and complexity. The only exceptions are the specialized systems like

the OCRs and the new video system. But, the shared outputs of these systems, which must be handled by the global data system, are quite simple.

3.3 Data Communications Issues

As discussed above, a variety of data is used to make real-time decisions. If that data is not present when it is needed, erroneous decisions or no decision may be made, resulting in processing delays and reduced mail throughput. Delivering data where it is needed when it is needed is the job of the communication system. The first step in designing a system to meet those needs is to specify all logical connections - who speaks to whom. Then we must decide how frequently they talk to one another and the amount of the information they exchange during conversations. This can be used to define the physical network architecture, and then point-to-point traffic load. These traffic loads govern the size of the links and the choices for protocols.

In practice, it is desirable to have a flexible, bus-type physical network with excess capacity connecting the major control systems. The terminal and sensory systems, on the other hand, will typically be connected point-to-point to some networked station (which is often one of the controllers). The communication loads for the mail facilities we have seen should be well within the the capacity of a single IEEE802.3 and 9600 bps direct serial links. However, further analysis will be needed to refine these estimates.

4. SUMMARY

These are difficult issues and many questions are unanswered. However, NIST's experience in manufacturing automation, factory data administration systems, and network standards and protocols provides insights and suggests approaches to resolving these issues and questions. Others in the field of automated manufacturing and intelligent control also bring experience to bear on the problems. If all this knowledge and experience can be integrated into a single intelligent system architecture, we believe that a major improvement in productivity can be achieved in postal system operations.

5. REFERENCES

[ALB81] Albus, J., Barbera, A., and Nagel, N. 1981, "Theory and practice of hierarchical control, "Proceedings of 23rd IEEE Computer Society International Conference, pp. 18-39.

[ALB88] Albus, J., "System description and design architecture for multiple autonomous undersea vehicles", NIST/SP-1251, National Institute of Standards and Technology Gaithersburg, MD, USA, September, 1988.

[ALB89a] Albus, J., McCain, H., and Lumia, R., " NASA/NBS standard reference model for telerobot control system architecture (NASREM)", NIST/SP-1235, National Institute of Standards and Technology, Gaithersburg, MD, USA, April, 1989.

[ALB89b] Albus, J., Quintero, R., Huang, H., and Roche, M., "Mining automation real-time control system architecture standard reference model (MASREM)", NIST/SP-1261, National Institute of Standards and Technology, Gaithersburg, MD, USA, May, 1989.

[DAY83] Day, J., and Zimmerman, H., "The OSI reference model", Proceedings of the IEEE, Vol. 71, 1983, 102-107.

[JON89] Jones, A., Davis, W., and Barkmeyer, E., " Issues in the design an implementation of a system architecture for computer integrated manufacturing, Internal Journal of Computer Integrate Manufacturing special issue on CIM architecture, Vol. 2, No. 2, 1989, 65-76.

[SIM82] Simpson, J., Hocken, H., and Albus, J., "The automated manufacturing research facility of the National Bureau of Standards", Journal of Manufacturing Systems, Vol. 1, No.1, 1982, 17-21.

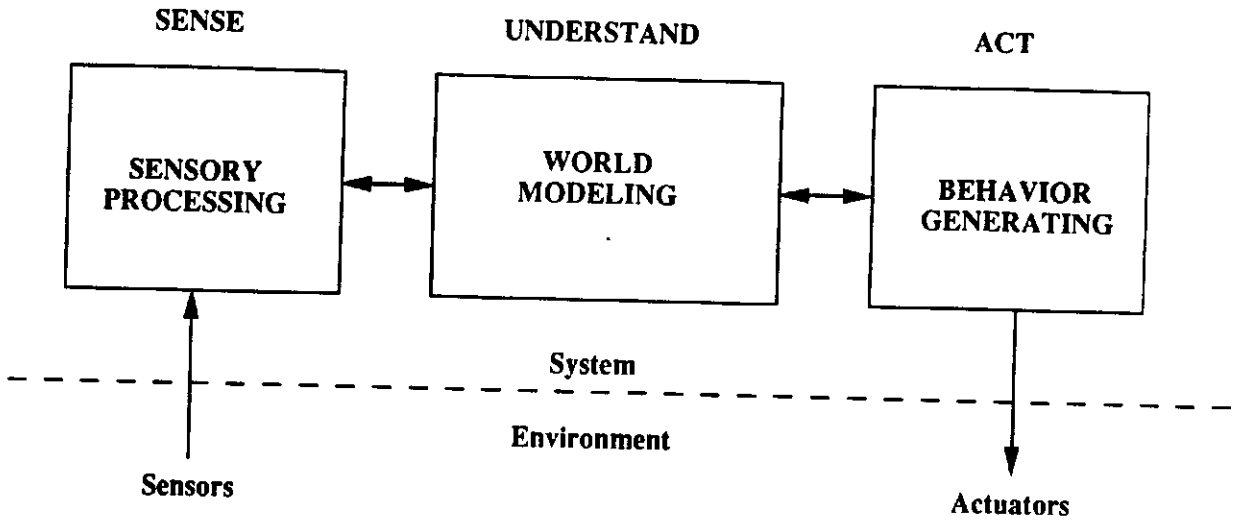

Figure 1. General View of an Intelligent System

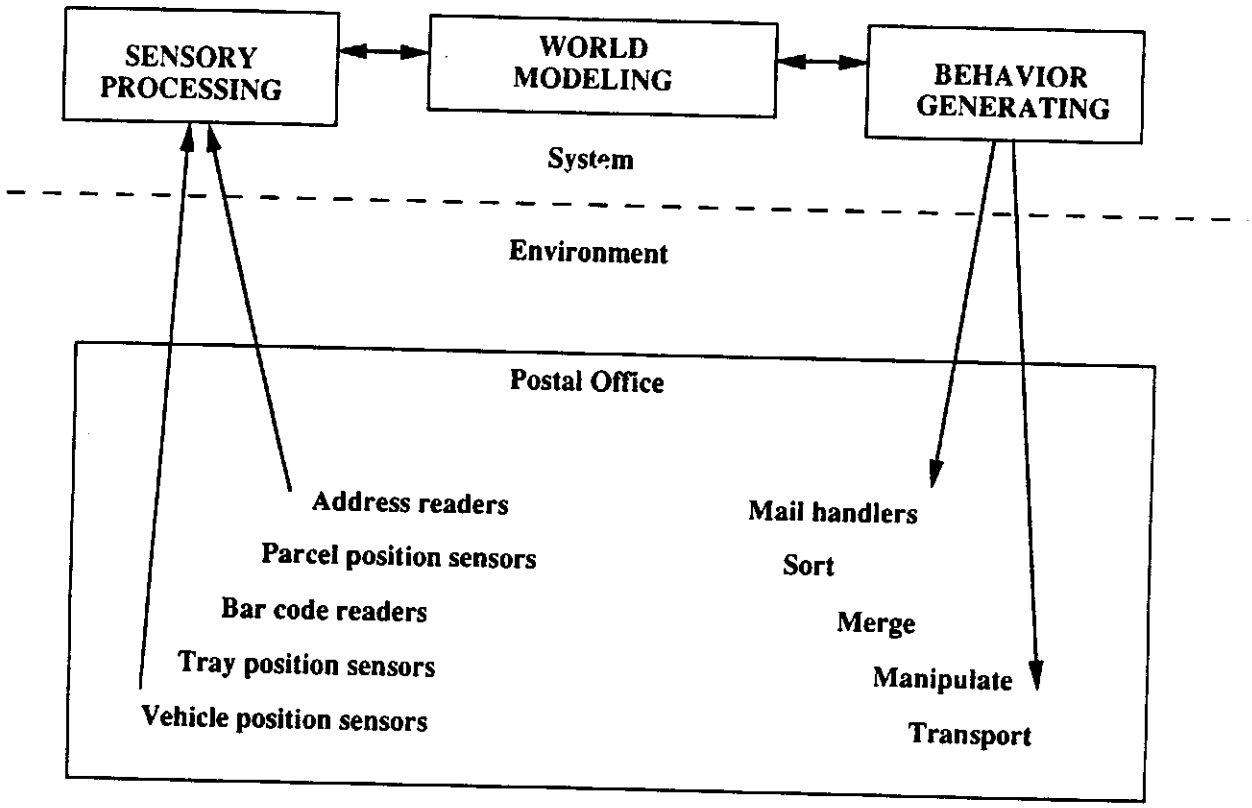

Figure 2. Specific Application to Post Office

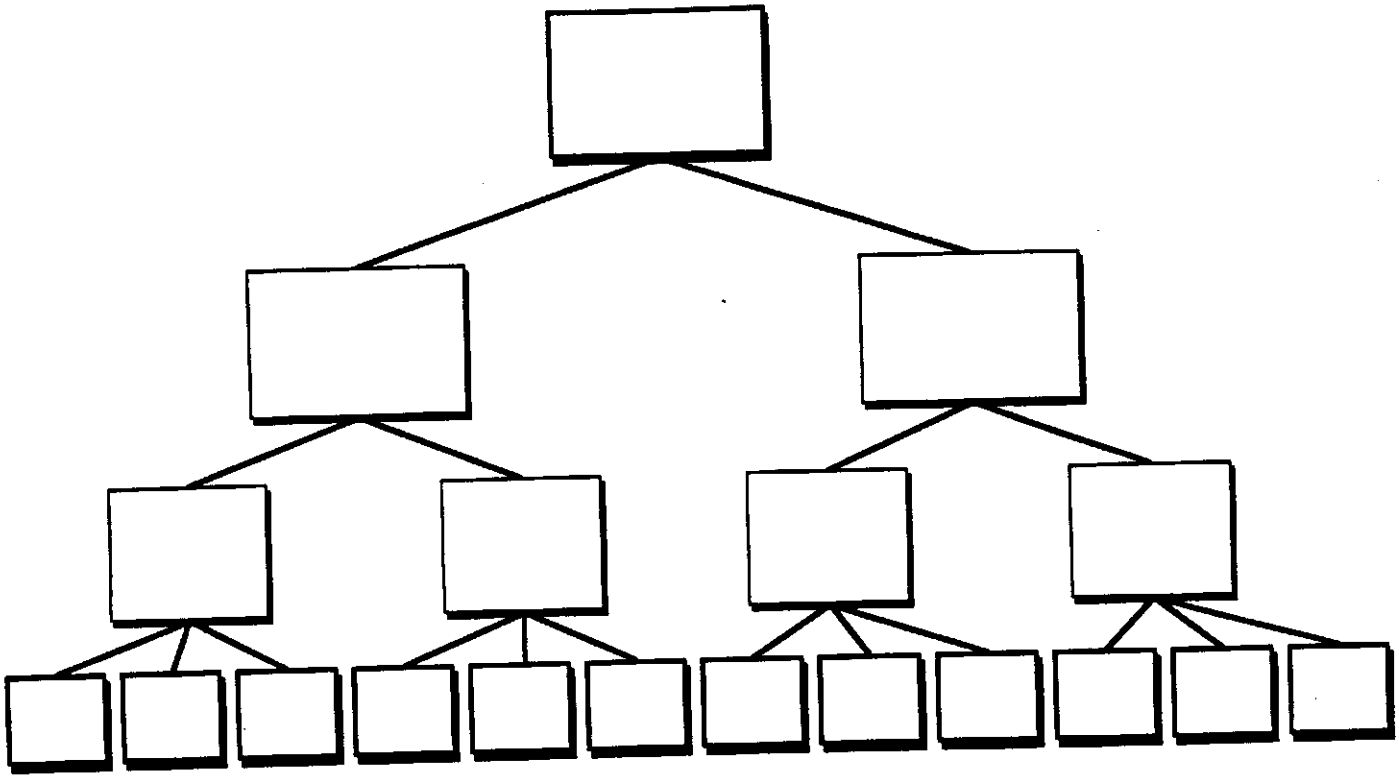

Figure 3. General Hierarchical Tree Structure

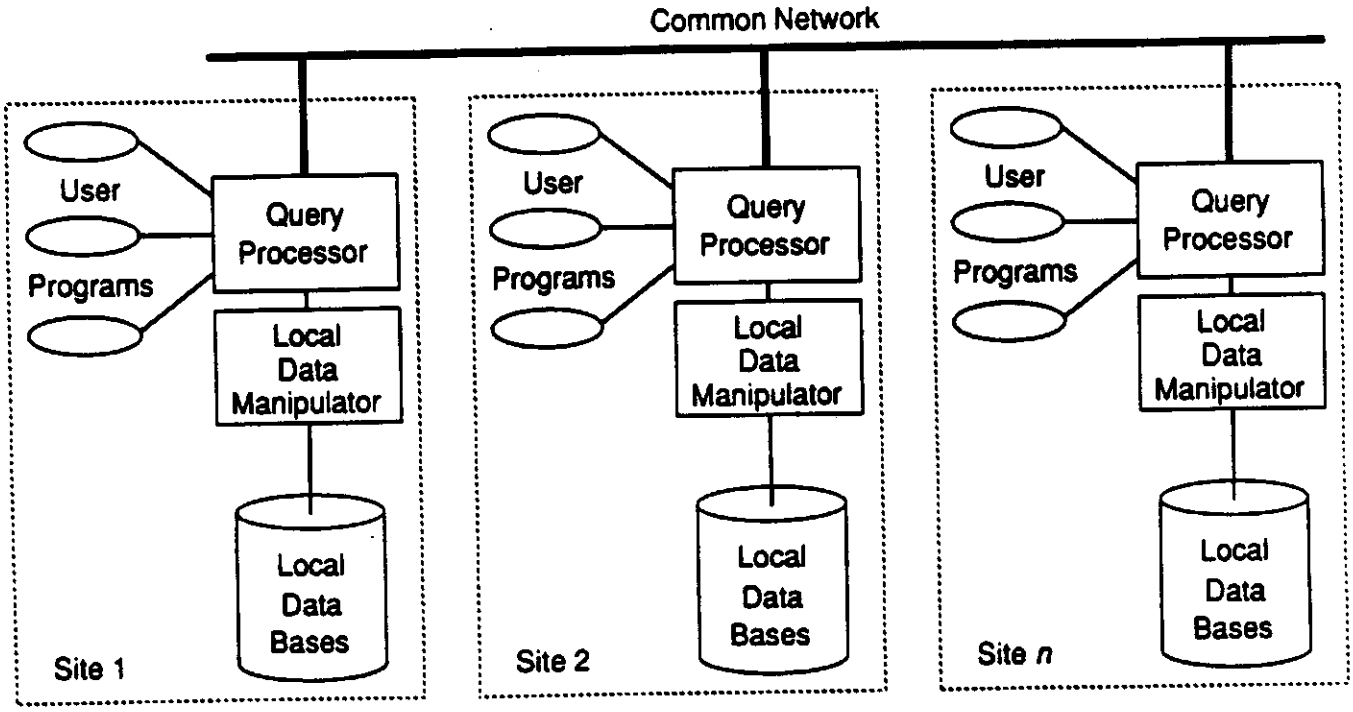

Figure 4. Distributed Data System with Distributed Control

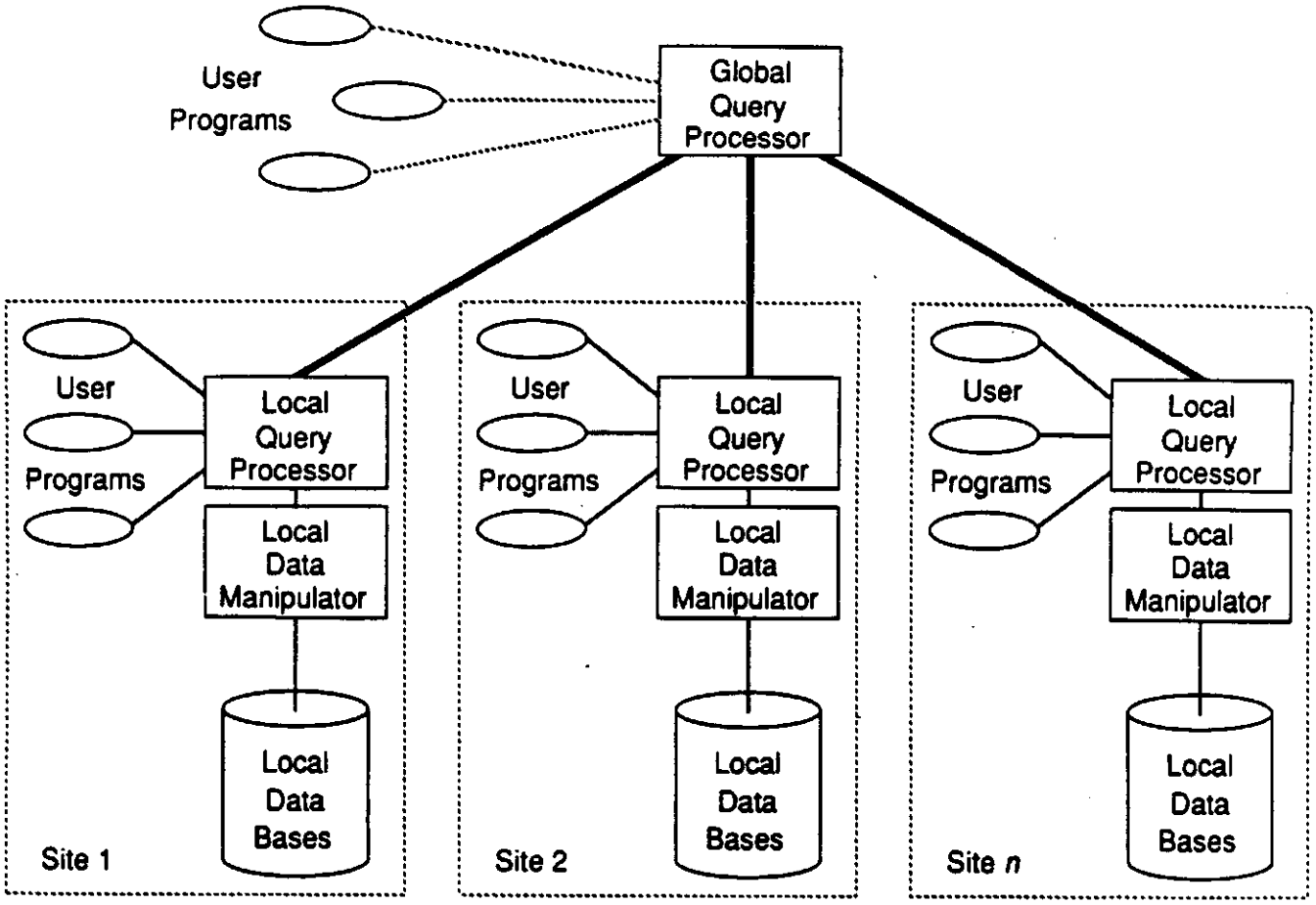

Figure 5. Distributed Data System with Hybrid Architecture

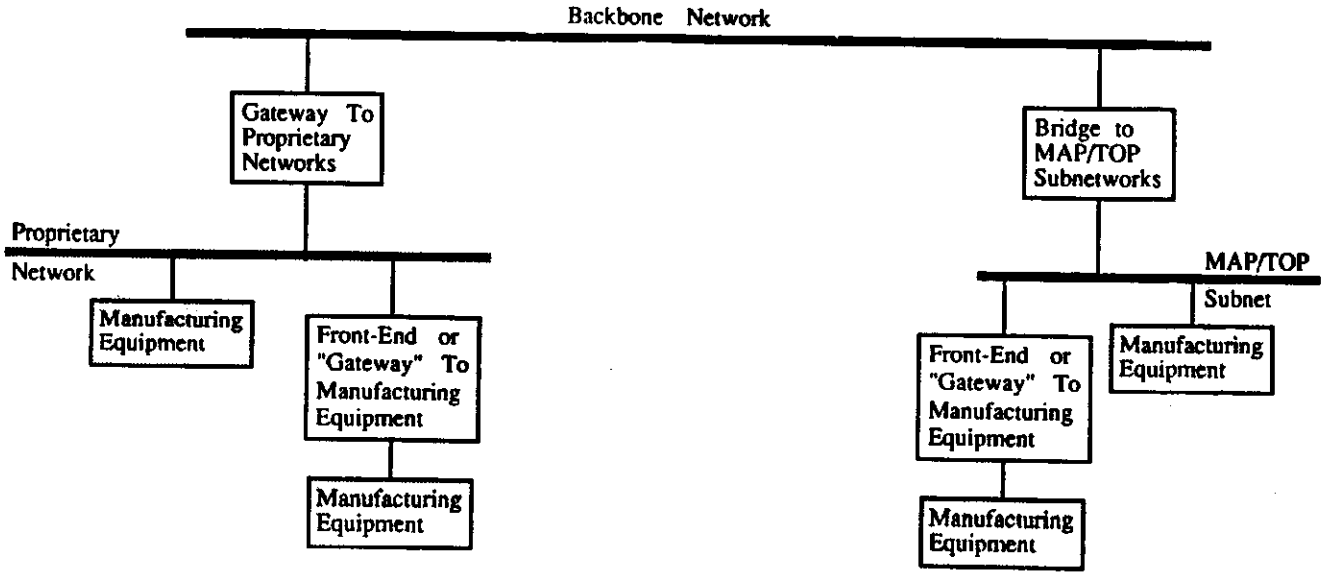

Figure 6. Single Site Physical Network Architecture