PRISM

Dispersion and Interconnection: Approaches to Distributed Systems Architecture

Final Report

June 1986

PRISM
CSC Index, Inc.
5 Cambridge Center
Cambridge, MA 02142
(617)492-1500

DISPERSION AND INTERCONNECTION: APPROACHES TO DISTRIBUTED SYSTEMS ARCHITECTURE

FINAL REPORT

CONTENTS

PRISM

DISPERSION AND INTERCONNECTION: APPROACHES TO DISTRIBUTED SYSTEMS ARCHITECTURE: FINAL REPORT

EXECUTIVE SUMMARY

PRISM (Partnership for Research in Information Systems Management) is a multi-client research service of Index Systems and Hammer and Company. Each year sponsors recommend the topics to be researched, serve as the primary source of data for the research, and participate in conferences at which results are presented.

In 1986, one of the PRISM focused research topics was "Dispersion and Interconnection: Approaches to Distributed Systems Architecture". Fifty sponsors participated in the research. A listing of participating sponsors is given as Exhibit 1. In addition to the architecture topic, PRISM research in 1986 covers the major research topic of "Integrating Information Systems and the Business" and two additional focused topics. These are "Expert Systems: Prospects and Early Development" and "Information Systems Planning in an End User Environment".

Information systems architecture is a topic of significant importance to most organizations. It offers great promise to the I/S function and to the entire organization in solving several persistent and difficult problems. Among the problems for which the concept of architecture offers solutions are the following:

- Rapid change in technology, application requirements, and organization that makes any solution quickly obsolete
- Technologies that are incapable of communicating with each other
- Lack of organizational consensus and shared understanding about technology and its management
- Excessive choice in the marketplace

These problems have all been intensified by the distribution of technology and its management, from a centralized information systems function to end users.

Though all PRISM sponsors had these problems to some extent, and therefore felt that architecture was desirable, there was wide variation in their definitions of architecture. Some mentioned diagrams, others brought up lists of processors in various locations, and some referred to rules and statements of direction. Sponsors usually felt dissatisfied with their situations in these areas, whether they had attemped to create architectures or not. In general, the

EXHIBIT 1

PRISM
SPONSORING FIRMS - 1986

AT&T (I/S)
AETNA LIFE AND CASUALTY INSURANCE
AIR PRODUCTS AND CHEMICALS
AMERICAN CAN
AMERICAN EXPRESS
ATLANTIC RICHFIELD COMPANY
BANKERS LIFE
BELL CANADA
BORG-WARNER CHEMICALS
BURLINGTON NORTHERN RAILROAD
CABOT CORPORATION
CHESEBROUGH-PONDS, INC.
CHUBB GROUP OF INSURANCE COMPANIES
CIBA-GEIGY CORPORATION
CLOROX COMPANY
COCA COLA USA
COLGATE-PALMOLIVE COMPANY
CONNECTICUT MUTUAL LIFE INSURANCE
CONSOLIDATED EDISON COMPANY
CONTINENTAL TELECOM SERVICE
DART & KRAFT, INC.
DIGITAL EQUIPMENT CORPORATION (I/S)
GREAT-WEST LIFE ASSURANCE CO.
HOUSTON LIGHTING & POWER
IBM (I/S)
INTERNAL REVENUE SERVICE
JOHNSON AND JOHNSON PRODUCTS
LIBBEY-OWENS-FORD COMPANY
LOCKHEED CALIFORNIA COMPANY
MCI (I/S)
METROPOLITAN LIFE INSURANCE COMPAN
MONSANTO COMPANY
MORGAN GUARANTY TRUST COMPANY OF N
MUTUAL LIFE INSURANCE COMPANY OF NY
NCR CORPORATION
NEW ENGLAND MUTUAL LIFE INSURANCE
NORTON COMPANY
ONTARIO HYDRO
PACIFIC BELL COMPANY
PACIFIC GAS AND ELECTRIC
PHILIP MORRIS, INC.
PORTLAND GENERAL ELECTRIC
ROCKWELL INTERNATIONAL
S.C. JOHNSON & SONS, INC.
SANDIA NATIONAL LABORATORIES
SAN DIEGO GAS & ELECTRIC
SMITH KLINE & FRENCH LABORATORIES
SWISSAIR
TRW INC.
UPJOHN COMPANY
UTAH POWER & LIGHT COMPANY
WAUSAU INSURANCE COMPANY
WESTINGHOUSE ELECTRIC
WILLIS FABER LTD.
XEROX CORPORATION (I/S)

architectural domain was filled with frustration for sponsors, and many of the architecture projects they had initiated met with failure.

The underlying problem of architecture is that there is no context for making decisions about technology and its use. There are rarely any evaluative criteria or structures in place in organizations. Interest in the outcome of architecture, and change in its terms and inputs, are at higher levels than ever before. The result is either a paralysis and an inability to make decisions, or constant insecurity about making the wrong decision.

In trying to identify a comprehensive structure for architectural decisions, we identified four different types of architecture, and four different domains in which it must be applied. The most important architectural type is the **principles** on which an architecture is based. These principles embody the organization's philosophy of information systems, and its objectives for technology and its management. The other types are **inventory**, a simple structured listing of architectural elements; **models**, or diagrams of the desired architectural state; and **standards** for selecting and using architectural elements. The elements to which architecture is applied are infrastructure, or hardware, communications, and system software; applications, data, and the organization that supports it all.

A complete architecture involves an approach to all combinations of applications elements and types, i.e., sixteen cells of a matrix. Some cells are more important than others, but this will vary across organizations.

Principles are the most important aspect of architecture, and drive all other types. They are based on the technology values of the firm in such areas as user autonomy, degree of risk acceptance, and the overall role of information technology in the firm. They should be relatively few in number, specific enough to drive behavior, distinctive to the organization, and articulated rather than invented. They should be stated by senior management; the I/S role is to assist in the articulation.

Principles are the focus of the recommendations of the project, but we also look more broadly at architecture and the other components of it. In inventory, models, and standards, it is important to do only what is important to the organization. For example, only key applications should be inventoried, key data modeled, and key technologies standardized. Otherwise, the architecture will never be finished. In general, the architectural enterprise should be conducted briefly and quickly. It should be conducted only when the key executive of a business unit feels it is necessary. It should involve users and their management, and should have the building of consensus as a primary objective.

A fictional architecture process included in the report illustrates the ease with which architectural decisions can be made after principles have been determined. Those PRISM sponsors who have started with principles seem to have had greater success than others in completing their architectural activities.

DISPERSION AND INTERCONNECTION: APPROACHES TO DISTRIBUTED SYSTEMS ARCHITECTURE

FINAL REPORT

PRISM

CHAPTER I

DISPERSION AND INTERCONNECTION: APPROACHES TO DISTRIBUTED SYSTEMS ARCHITECTURE

Introduction

In this section we describe the findings and conclusions of our examination and analysis of sponsor activities in distributed systems architecture. Architecture as we discuss it here refers to activities associated with the development of information technology approaches, policies, and strategies as they relate to distributed information systems.

Our objective in this project was to characterize the architectural environment in sponsor organizations, and then to provide an approach for architectural activity. The goal for the approach is not an algorithm specifying all steps in an architecture project in detail, but an overall framework that sponsors with varied architectural environments could use. The remainder of this chapter is organized as follows:

- Sponsor Perspectives and Activities
- A Hypothesis on the Nature of the Problem
- A Framework for Architecture Development
- Domains of Architecture
- Types of Architecture
- The Process of Architecture Development
- Conclusion

Sponsor Perspectives and Activities

Our discussions with sponsors revealed two general facts about architecture:

- Almost every information systems (I/S) organization believes that it has problems which can only be addressed by the development of an architecture.

- No two I/S organizations mean the same thing by architecture when they express the above belief.

While a wide variety of specific architecture-related problems were identified in sponsor organizations, there were recurring themes such as vendor and product proliferation, incompatible systems, difficulties in sharing application software and data, and general uncertainty not only about which alternative to choose, but also about how to evaluate the choices. Many sponsors hoped that an architecture would shed light on questions such as data location, PC policy, office automation standards, and cabling.

Most sponsors we spoke with had completed architecture projects, had efforts under way, or were planning to begin architecture development in the near future. We found a wide range of approaches in terms of formality, focus (data vs applications vs infrastructure), use of outside consultants, and organizational scope. Products of these efforts varied from multi-volume binder sets containing exhaustive detail about all facets of information technology to occasional memos and position papers which were the subject of informal discussion.

Sponsor architectures also varied widely in format. Some firms showed us an "architecture" that consisted purely of diagrams of processors and their interconnections; slightly simpler versions only listed the processors by number and type. Other organizations considered architecture to be a set of statements about their technology preferences and policies. A few sponsors simply referred to their primary vendor as the basis of their architecture. All of these organizations derived some value from their versions of architecture, yet most still felt that something was missing.

While some sponsors have conducted successful architecture efforts (success being the resolution of the problems the architecture attempted to address) most architecture efforts have fallen short of their promise. Some examples:

- Detailed architecture documents which, while containing valid information, have never been read since they were published

- Completed projects that failed to answer the relevant questions

- Projects that were never completed due to lack of interest and/or direction

- Projects that took so long to complete as to be obsolete at the time of issue

These failures were not for lack of effort. Many unsuccessful projects involved considerable amounts of time by the "Architecture Task Force." Some had included senior line management. Some had involved consultants. Important activities had taken place in many of these efforts, including reviews of corporate strategy, assessment of technical directions, and monitoring of long-term application plans. The great majority of projects, however, no matter how well-intentioned or executed, did not achieve the desired result of providing overarching, long-term answers to architectural questions.

A Hypothesis on the Nature of the Problem

We have concluded that architectural efforts are attempts by I/S organizations to restore a sense of structure and order into a constantly changing world. The advent of powerful, affordable, and interconnectible distributed computing technologies combined in many cases with rapidly changing business conditions and environments are challenging some of the traditional technical and economic assumptions which have guided the actions of I/S organizations in the past. The resulting uncertainty has created confusion and anxiety, and has left organizations with choices which cannot be evaluated in a straightforward and analytical way. While architecture has been put forward as the method to reduce or eliminate this uncertainty, most of the activities conducted in its name do nothing of the sort.

There was uncertainty when systems were centralized, but it is the distribution of information systems that makes architecture difficult. Distributed processors must be chosen and connected to each other and to a central data center, data and applications must be located in their proper processor and then shared, and someone must manage all of this. Distribution, then, provides the incentive and the need for research into architecture. Our insights and recommendations in this research project will be relevant to centralized environments, but they may not be as important or necessary.

Given the uncertainties brought about by distribution, the fundamental problem is a lack of evaluative criteria and decision-making mechanisms which can guide the development and choice of distributed technologies. Traditional evaluation techniques fail for a variety of reasons:

- Applications and their requirements are not yet known
- Future vendor offerings and developments are unknown
- Evaluative criteria are unclear and undefined
- Often the alternatives being evaluated are sufficiently different in nature as to be impossible to compare.

The factors listed above lead us to the conclusion that uncertainty and ambiguity are part of the current environment for I/S organizations, and cannot be utterly defeated by even the most masterful architectural stroke. Architecture is not a formal solution to these problems, but rather a mechanism for identifying and resolving conflicts and building consensus on technology directions and strategies. What is needed is a set of long-term, consensual criteria for evaluating and managing information technology that will persist in the face of change.

Most current architectural efforts deal with technical debates, e.g., the relative merits of local area networks versus minicomputers, or IBM vs. DEC. Consultants and the media exacerbate the problem by operating at this level. Because of the above uncertainty, however, there is no way to resolve technical debates in any meaningful way. The real debate should be about the criteria used to make technical decisions. The terms of the discussion should be recast in those terms, in a way in which senior management can relate to it.

Viewed in this light, it is possible to more clearly articulate a reasonable set of expected benefits of architectural endeavors:

- Avoid "re-inventing the wheel" by capturing the general principles of technology decisions and management

- Over time, to move the technology toward increased compatibility, interconnection, and integration, where appropriate.
- Enable growth, extension, and enhancement of the existing installed base
- Create a consistent and coherent development environment.
- Establish basic guidelines for conducting I/S business.

A Framework for Architecture Development

Given the diverse and fragmented nature of the activities which are included in the category architecture, we believe that an overall framework which describes and relates the various components and approaches is required. Exhibit I-1 depicts a matrix which maps the domains of architecture (those aspects of I/S to which architecture is applied) against the types of architecture (processes which can be exercised). This framework can serve as a starting point for architecture development, as it identifies the range of possible activities and approaches (each of the 16 cells of the matrix is a set of activities in its own right) which are available to I/S organizations in their attempt to illuminate the dark and misty course of information technology development.

Domains of Architecture

There are four domains to which architectural process can be applied:

- Infrastructure - The underlying technological platform which supports data and applications, including hardware, systems software, and communications networks.

- Application software - The code which processes data for the organization, including acquired as well as internally developed programs.

- Data - The information assets of the organization.

- Organization - The people and structures that make it all work.

Types of Architecture

Sponsor activities in the area of architecture can be classified (as they are in Exhibit I-1) into four general types:

- Inventory: A snapshot of the current state showing the architectural items in place today and their relationships.

- Principles: A statement of the organization's philosophy of information systems expressed in terms of objectives and goals in each domain area.

Exhibit I-1

Architectural Variations

	INVENTORY	PRINCIPLES	MODELS	STANDARDS
INFRASTRUCTURE				
DATA				
APPLICATION				
ORGANIZATION				

- Models: Pictures of the desired state, with emphasis on what goes where, and how it is all connected.

- Standards: Specific rules or guidelines for implementing the models.

The relationship among these types of architecture is not static; however, there is a natural flow and sequence of activities which proceeds from an articulation of technology values to the development of principles, and on to models and standards. Inventory serves as an input to the development of models as the current state influences the practicality of various desired states. Exhibit I-2 depicts these relationships.

While each type/domain cell of the architecture matrix has importance in particular contexts, some are generally more valuable than others. We have found that principles in all domains have the most far-reaching and significant impact on an organization's actions, as they form the link between the basic technology values as they are held by the business organization, and technology development and implementation. Data is the most important domain, as it represents the information assets of the organization, and the other domains can be built around it. Data inventories can be of great benefit to many organizations who may not have in one place a coherent picture of the strengths and weaknesses of their information assets. Standards for data are valuable for key information which is broadly shared and exchanged. Two other areas of relatively high value are infrastructure standards which streamline acquisition procedures and create better potential for future compatibility and connectivity, and organization inventory, especially skills assessment as a determinant of the feasibility of introducing technologies at various levels of the organization.

The following paragraphs describe the role of each of these types of architecture, as well as presenting some examples. We begin with principles, which are the keystone of the architectural endeavor.

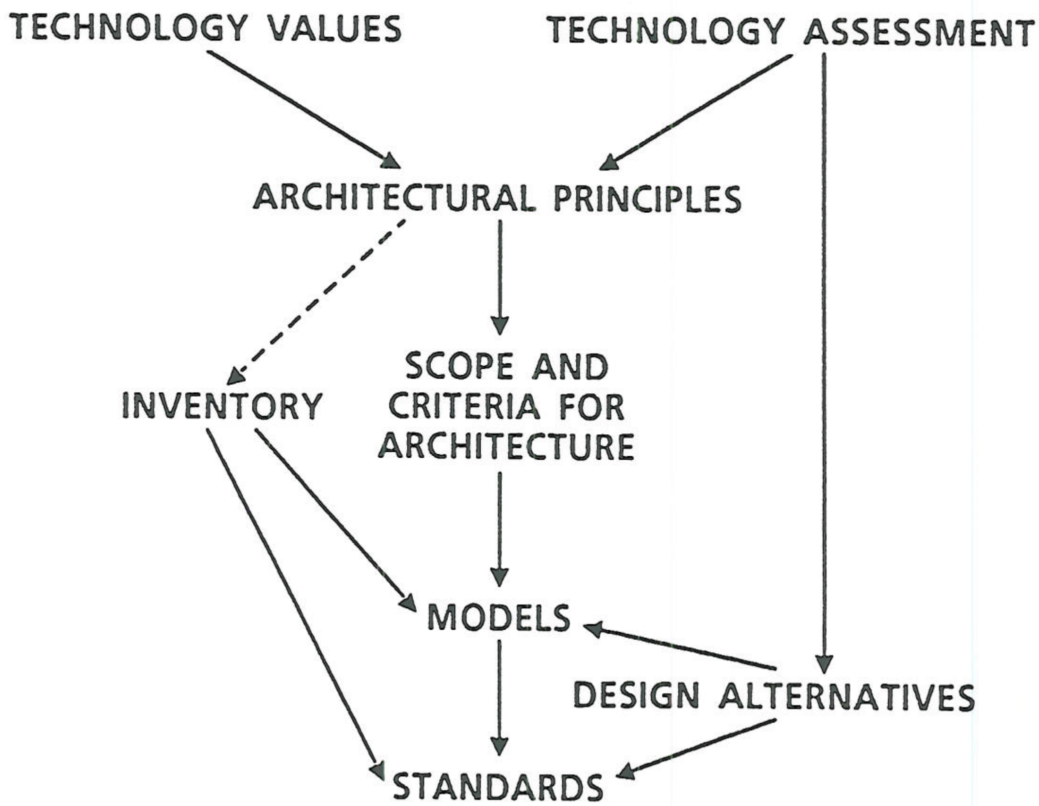
Principles

Principles, as we have stated, are an articulation of basic philosophies of information management in each domain. Principles are the most stable element of architecture. Thus, they are the key element precisely because, in an atmosphere of change and uncertainty, they represent continuity and relative stability. If developed correctly, they should serve as the starting point for most difficult evaluations and decisions. Indeed, in many cases (especially when the alternatives are so different from one another as to defy comparison) principles will be the only rational basis for coming to a decision.

Principles are important in all domains of architecture, and specifying principles in each domain leads to greater concreteness and relevance to actual behavior. Exhibit I-3 describes the major topics which principles should address. Some of these topics cross domains, while others apply only to one domain. For example, principles should specify whether the organization will strive for

Exhibit I-2

Architectural Development



PRISM

Exhibit I-3

Principal Areas of Principle

- CROSS- AND INTER-DOMAIN
  - COST SENSITIVITY
  - TIME HORIZONS
  - DEGREE AND EXTENT OF STANDARDIZATION
  - SIMPLICITY/COMPLEXITY
  - GENERALITY/OPTIMALITY
  - SHARING
  - ORGANIZING THEME
- INFRASTRUCTURE
  - EXTENT AND ACCESS
  - VENDOR STANCE
  - DEGREE OF INTERCONNECTION
- DATA
  - OWNERSHIP
  - RESPONSIBILITIES OF STEWARDSHIP
  - LOCATION
  - ACCESS
- APPLICATION
  - LOCATION OF PROCESSING
  - EXTENT OF INTERFACING AND INTEGRATION DEMANDED
  - RESPONSIBILITIES FOR DEVELOPMENT AND MAINTENANCE
- ORGANIZATION
  - FOCUS OF SYSTEM RESPONSIBILITIES
  - DEFINITION OF ROLES
  - CAREER PATHS

general technology solutions that meet all application needs, or, on the other hand, prefer optimal specific solutions to each need. This principle would apply to all domains. The number of vendors to be considered in an evaluation, however, is an issue which only involves infrastructure. Exhibit I-4 presents some sample principles.

Some PRISM sponsors have articulated principles, but not in all of the domains mentioned above (infrastructure, data, applications, and organization). Applications principles, about how software will be engineered, are common. Many organizations have an infrastructure principle that they will buy from IBM (any other vendor preference is probably a standard rather than a principle!) For most organizations, however, principles are only implicit, and their implicitness leads to dissent and uncertainty. Even when the principles are articulated, they may not represent an organizational consensus, or have enough real content to affect behavior.

The Sources of Principles

The main determinant of architectural principles is the organization's technology values, which are a set of underlying attitudes and perspectives that shape the organization's fundamental approach to information systems. Values cut across technology domains, and are even more long-lived than principles. Technology values exist in all organizations, whether or not they are articulated as such. The key to successful elucidation of principles is the discovery of these values, as opposed to their invention. All technology principles can be constructed from three primary areas of value:

- Orientation to risk: aversion to tolerance
- User Autonomy: From low to high
- Technology perspective: From cost displacement to strategic tool

If an organization is risk-averse, it should formulate principles which minimize experimentation with infrastructure, protect data, keep application development in the hands of professionals, and ensure that all systems are managed by a central, highly skilled I/S function. On the other hand, an organization that most values user autonomy will want principles that encourage users to make technology decisions. The organization's perspective on technology is a value that will determine the aggressiveness, orientation to cost, and integration with business of its principles. Values in one area can obviously impinge on another area; then the organization must decide its value priorities. It is not any particular value, but the aggregate of all technology values, that should determine principles.

Values are not the sole determinant of principles. The organization's business situation and condition, its market position, competitive environment, and other elements of business strategy all impact technology values and principles directly, and in turn therefore underlie the entire architectural endeavor. A principle might specify, for example, that emphasis will be placed on applications which increase market share. Even if the values of the

Exhibit I-4

Sample Principles

- INFRASTRUCTURE
  - WE ARE COMMITTED TO A SINGLE-VENDOR ENVIRONMENT
  - READY WORKSTATION ACCESS FOR EVERY PROFESSIONAL
- DATA
  - DATA WILL BE OWNED BY ITS ORIGINATOR BUT ACCESSIBLE TO HIGHER LEVELS
  - CRITICAL DATA ITEMS IN CUSTOMER AND SALES FILES WILL CONFORM TO STANDARDS FOR NAME, FORM AND SEMANTICS
- APPLICATIONS
  - APPLICATIONS SHOULD BE PROCESSED WHERE DATA RESIDES
  - WE WILL HAVE A STANDARD MRP SYSTEM THROUGHOUT THE ORGANIZATION
- ORGANIZATION
  - EVERY MANAGER WITH P&L RESPONSIBILITY SHOULD MANAGE HIS OWN SYSTEMS
  - THE CENTRAL I/S ORGANIZATION WILL APPROVE ALL HARDWARE AND SOFTWARE ACQUISITIONS

organization suggest principles based on investment orientations to systems, short-term business conditions may force principles stressing conservative spending on technology. Architectural principles must also be informed by, but are not determined by, an assessment of the current state and future direction of technology. This assessment provides the options for technology-based principles; a firm could not state in its principles that all employees should have intelligent workstations on their desks without believing that such workstations will be available and affordable.

Developing Principles

The nature of principles suggests some guidelines for their development. The effort should be small-scale, emphasizing the participation of key constituencies (typically I/S and representative managers from various user business units). Identification of people within the organization who set and carry technology values may be the most important task in the entire process. The formulation of strawman statements by a sophisticated I/S manager can facilitate the process by providing a starting point for discussion. A series of workshops or other structured discussion formats should be used to refine and expand the strawman principles into an initial set of statements.

Consensus on the principles is essential; indeed, if significant disagreement with the validity of a principle arises, it probably is not valid for the organization. On the other hand, most organizations will arrive at some principles which in some aspects are mutually contradictory. Evaluating the consequences of an agreed upon set of principles as part of the process of developing them helps to avoid disillusionment further down the line. As any set of real principles will inevitably contain contradictions, identifying where they are and what their implications might be in advance of the application of the principles to real decisions reduces the tension that inevitably surrounds major technology decisions.

If there is no consensus from senior management on the technology values of the organization, it will be impossible to develop consensus on principles. The absence of shared values should be addressed directly before attempting an architecture project. We know of "education" programs for senior executives which have successfully articulated, and even changed in more favorable directions, the technology values of the organization. This is, however, a broader problem than just architecture, and we will not treat it further here.

I/S can play several different roles in articulating principles. Depending on its organizational charter, it may want to take a leadership role and shape the firm's values and principles, though this must be done in a subtle and gentle fashion. A more passive I/S should simply help the organization to realize and articulate values and principles. The extent of leadership that I/S should take here is actually a value in itself.

Desirable Characteristics of Principles

Not all principles are created equal. There are some characteristics of principles which make them more effective as guides to architecture. The most simple of these is that principles should be few in number; no organization can implement an architecture based on hundreds of principles. A recommendation for any particular number would be arbitrary, but as a guide we suggest that an organization's principles should fit on one page.

Principles should also avoid advocating "motherhood" - obviously true statements which have no real effect on behavior. For each principle to be valuable, it should be possible to argue the opposite point if circumstances were different. This might not be true, for example, of the widely-stated principle, "Data is a valuable corporate asset." Motherhood can also be avoided by developing principles which are specific to the organization. It should not be possible for an entirely different organization to use your principles, and have them fit. If generic principles are the only type that fit, perhaps the architectural effort should take place at a lower level of the organization, where there is less variation in I/S philosophy, and the "lowest common denominator" principles are not necessary.

Inventory

Taking inventory is not a glamorous or inspiring undertaking, but in the context of architecture development it has concrete value. Any attempt to further develop information resources and technology must be informed by the current state of affairs. Particularly, a consideration and evaluation of alternatives must account for the starting point; an alternative which is extremely attractive in the abstract may be less so when viewed in the context of the path required to get from here to there. Another potential value of inventory is the identification of hidden assets and gaps. The process of producing a data inventory can, for instance, reveal that information which was assumed to be present in a useful form is in fact absent. An applications inventory which includes the age and functionality of applications is important input for determining systems renewal and replacement priorities. We have already discussed the importance - and the neglect - of the inventory of organizational skills and capabilities.

Many sponsors already conduct inventories in the various domains of architecture. The data dictionary, for example, is a structure for developing a data inventory. Most architecture projects include a list of key infrastructure components; this is most valuable when the list is too long, or too complex, to remember in one's head. And many planning projects - particularly those involving the BSP approach - have an applications inventory as a key step.

In approaching inventory efforts, it is important to realize that exhaustive, complete inventories are both impossible to perform and an enormous waste of time and money. In most organizations, the items which are subject to inventory change too rapidly to ever create a "complete" catalogue. More significantly, many items are not worth counting. Architectural principles can guide the differentiation of key assets and unimportant ones. For example, in the context of an infrastructure principle that states that all professional employees will have ready access to intelligent workstations, it may be far more important to understand the population and location of PC's than to count all the data entry terminals in the organization.

Models

Models can be viewed as the first step in the implementation process. Principles determine the scope and time horizon for implementation, and point out the key domains and establish priorities. A principle that states that information technology applications are best developed in an entrepreneurial fashion by individual business units might point to a relatively short planning horizon, and would produce models which would be general and non-restrictive. A model for departmental systems, for example, would leave great latitude as to the type of processor to be employed. Principles are also the main consideration in establishing evaluation criteria for alternate models. The degree of acceptance of new technology, for example, is a principle which would help to determine the place of local area networks in an organization's architecture. At the same time, new principles may be discovered or old ones modified as the process of developing and evaluating alternative models proceeds.

Types of Models

The nature of the models produced in architecture development may vary considerably and may emphasize different domains. Some general features are often applicable, however. Infrastructure is the integrating element, defining the potential connections and dependencies among the other domains; many sponsors already model their infrastructures. This does not necessarily place infrastructure as the determining force. The demands for data integration may determine the infrastructure model; as an example, large, centralized databases may restrict the use of departmental minicomputers. Data modelling is a well-known activity, and activities such as entity-relationship modelling are structures for producing data models. Organizational models are the most often neglected element of architecture, and are often the most important element in determining the success or failure of technical development projects. An organizational model should include the roles and responsibilities for information systems activities in a business unit or function.

Developing Models

Given the variety of forms that models can take, the process of developing them also takes many forms. Some necessary steps are:

- Identify and specify known application requirements and, to the extent possible, corporate strategy and directions

- Identify potential technical options. This process should account for principles which describe acceptable risk and technology restrictions.

- Explicitly formulate evaluation criteria based on the principles in the domain in question (for example, a preference for technologies already used within the organization as a model constraint)

- Evaluate the alternative models and select the winner (or winners, depending on whether principles call for single or multiple endorsed approaches).

Standards

Standards follow from models as the concrete guidelines to particular development efforts. They are the most detailed aspect of architecture, and the primary activity in which names are named - of vendors, databases, applications, and people filling support and management roles. Infrastructure standards specify component selection and connectivity for particular environments. Data standards describe structures, data definitions, redundancies, and security considerations for databases. Application standards prescribe tools and environments, and can also mandate programming practices and structures for developed software. Organization standards describe support and management structures and staffing requirements for the delivery of information services.

An organization's architectural principles, of course, determine many aspects of its standards. The principles will influence the criteria used for selecting between vendors, approaches, and other general alternatives. They should prescribe who actually creates the standards - whether it is I/S or the user organization. The extensiveness of the standards is also a function of the level of detail specified in principles. With principles that are consensual and understood, standard-setting should be a relatively straightforward activity.

The Scope of Architecture

We have shown that there are several types and domains of architecture. Filling out the matrix they create (though all cells may not be necessary) is an extensive activity. It is important to point out, however, that for architecture to be feasible, its scope should be limited. This statement applies to almost all cells of the matrix, with the exception of principles.

We have already stated, for example, that not all architectural items are worth inventory. This point also applies to standards and models. What should be counted, modelled, and standardized is only that which is important. No one cares about anything else (at least in the abstract); even if they did there would be no time to deal with it all.

There are several ways to focus on what really matters. The following are three approaches to focusing:

- Doing what is really important to the business. The Critical Success Factors methodology is well-suited for identifying these components, particularly with regard to applications and data. Value-chain approaches are used for the same purpose.

- Doing what needs to be done now. "What needs to be done now" may refer to a whole domain, or to aspects of all domains. Taking this approach in the data arena, for example, would lead an organization to model only the data used in new applications. Similarly, standards for an infrastructure component would only be developed when that component needs to be acquired somewhere within the organization.

- Doing what is easiest to do. This is a last resort, but sometimes it is better than doing nothing at all. In the data domain, for example, it is better to start with modelling and standardizing the data environment for business units with simple data needs, than to begin with the hardest environment and fail, or do nothing at all. This strategy to limiting scope is best employed early on; success with easy architectural tasks will allow assuming more difficult problems later.

The I/S world is littered with architectural efforts that failed because they were too ambitious. Even if architectures of extensive scope can actually be done, they are rarely worth the effort.

The Process of Architecture Development

Architecture for distributed systems must be initiated as a result of a real need on the part of an interested party outside the I/S area. An effort initiated by I/S will have no meaningful result outside I/S. As discussed above, the key product of architecture is consensus, and an effort internal to I/S will never produce an effective consensus. A successful architecture effort should have a sponsor who understands that there is an important problem to be solved and that the architecture effort will solve the problem. Thus, the sponsor has a vested interest in the outcome of the process, and will lend authority to both the effort and the results. The sponsor should be the highest-level manager in the business unit with the architectural problem. If no sponsor can be identified, or if the potential sponsor cannot be convinced that an architecture project is necessary, it should not be undertaken.

Once launched, an architecture project should be conducted by a small group from I/S under the active leadership of the I/S executive. This leadership is especially critical during the development of principles, as these should most directly reflect senior management views and values. The I/S executive is part of senior management him or herself, and also has the best access to other senior managers.

Consultants can play a valuable role both by bringing experience with architecture and as facilitators of process who hold no direct stake in a particular outcome of the project. Different types of consultants are relevant for the different types of architecture; working with senior managers on values and principles demands very different skills than directing an entity-relationship modelling effort. Consultants are valuable not only for their expertise, but also for their objectivity and their "outside" nature. That is, especially during the articulation of principles, it is important for senior managers to speak freely about their values and preferences. Their view about the appropriate roles for I/S, for example, may be better expressed to a consultant than an I/S manager.

Last but by no means least, the architecture process should involve representatives of the relevant user communities. Again, the relevant parties from the user community differ by type of architecture. Senior user management should articulate principles, but models and standards should be worked on by actual users - with or without help from I/S, depending on principles. User involvement is easy to obtain on a well sponsored and motivated project; if the users don't come to the meetings, they don't think they need an architecture. This state of affairs is most often a comment on shortcomings of the architecture process, not on the users.

Building consensus requires authentic and open communication, especially between I/S and the distributed user community. Real concerns of all constituencies must be recognized and addressed. One thing that is almost guaranteed to accompany this process is conflict and disagreement, based on true conflicts of interest which exist among varied business units and departments. These conflicts must be welcomed and managed by the architecture team, as the exposure of these disagreements and contradictions is a great part of the value of the architecture process. The resulting tradeoffs, compromises, and codified antitheses represent important resolutions for the organization which then may not have to be relived in new forms every time and information technology decision is made.

The level at which architecture should be done in the organization will vary. In general, architecture should be done at the highest level for which there is consensus on principles. If all divisional managers cannot agree on principles, there should not be a corporate architecture (unless the CEO can convince them to agree!) Overly generic principles are a sign that the organizational level of the architectural effort needs to be lowered. If the business units of a corporation are in different businesses, with different value chains and bases for competition, there is no reason why they should have the same architecture.

The amount of time involved in an architectural effort should be short. Otherwise, its conclusions will be outdated upon issue, and all interest in the project will be gone. As a rule of thumb, few projects should take more than one year, and most should be done in less than six months. The key to accomplishing these timeframes is to limit the scope.

A word about the level of ambition for architectural projects is in order. All architectural conclusions will be imperfect, inaccurate, and out-of-date. Wrong decisions will be made. It is most important to remember that some architecture is better than none, and that the information systems environment of any organization will never be fully rationalized. This is clearly a situation where the 80/20 rule (80% of the value in 20% of the time) applies.

Conclusion

In this report, we have attempted to clarify the purpose and nature of architecture in its many manifestations and incarnations. Architecture has the potential to provide an invaluable mechanism for I/S organizations to formulate and implement information technology strategy in the most difficult and challenging environment it has ever faced. In closing it is appropriate to mention some overall factors and limitations which must be considered along with architecture.

Architectures have phases and milestones, but are never done. While this point about architecture has been made in many places many times, it is nevertheless true. Continuous adaptation of the architecture, led by the refinement of principles, is necessary to its ongoing value. Also, the true test of an architecture is its continuing relevance to the process of technology implementation and development. The implementation of the architecture never ends.

Architecture can be extremely helpful in dealing with business change; however, change of too great a magnitude in application, technology, or organization can invalidate even the best architecture. Perhaps the clearest example of this is a merger or acquisition, or some other dramatic and far-reaching organizational change. Such a precipitous and sudden change in the business environment and the information technology inventory cannot be anticipated by an architecture. This is not a call to stop doing architecture; rather it is to note that there is no use in trying to structure an architecture to deal with precipitous and unpredictable change since it cannot be done. If such levels of change are expected, architectural work should be postponed until after the dust settles.

What can be done is the most important message. Principles provide an anchor in a sea of change. They provide guidelines and rationales for the constant examination and re-evaluation of technology plans. They also point out those questions which simply cannot be answered. With continuous refinement and adjustment, architecture can separate the manageable from the unknown, which ultimately is the first step in the successful discharge of the information systems function.

The second half of this report is a report on a fictional architecture project, in a fictional company. Because no one, to our knowledge, has ever explicitly and formally employed this complete approach to architecture, only a fictional case could illustrate its use. The events and situations in the case, however, are drawn from a variety of real sources. The case is realistic and plausible, and illustrates the steps, potential problems, and value of a distributed systems architecture as we have proposed it.

CHAPTER II

HYPERCORP ARCHITECTURE CASE

Hypercorp is a $3 billion consumer products manufacturing and marketing company. It is organized by product and by region, although information systems have always been developed and managed largely from headquarters. Roger Rambyte, Vice President of Information Systems, is beginning to notice a number of problems with the centralized approach they have taken.

Product and regional managers, for example, have complained more lately about their lack of systems capabilities. They have said that they would like greater flexibility at their levels, and that the centrally-developed systems no longer meet their needs. A further problem is that the communications costs for data entry and access from the field have always been high, and are growing at too fast a rate. The computers used to support terminals in the field were becoming obsolete, and were no longer really supported by the vendor.

Many field marketing offices had acquired personal computers. So far they had primarily been used as terminals or in standalone mode, but users were beginning to request help in connecting local area networks. But even despite this personal computing, the applications backlog for centralized applications was still growing, and the corporate data center would need a major expansion if the portfolio grew much larger.

Roger's Dilemma

All of these signs pointed to greater decentralization of computing at Hypercorp, but to Roger it wasn't that simple. There seemed to be a number of obstacles to decentralization. For one thing, all of the skills in information systems seemed to be within the I/S function. Roger was worried that the users would create some disasters; although he felt powerless to prevent these, he was sure he would be blamed for them. Furthermore, Roger thought that hardware would provide a further obstacle. In the past Hypercorp had been a one-vendor organization. In Roger's opinion, that vendor was quite weak at the middle or departmental level.

Roger also feared that decentralization would raise the overall costs of computing at Hypercorp. Some senior managers thought costs were already too high as a percentage of sales. On the other hand, some managers were pushing for expensive new technologies, such as expert systems. Roger felt somewhat paralyzed by these contradictions.

A final obstacle to centralization was the company's key database. It contained linked information on products, customers, and distributors. Breaking it up seemed both technically infeasible and a security and integrity risk.

Light in Roger's Tunnel

Roger was a PRISM sponsor, and he liked its approach to architecture. He was particularly attracted by the emphasis on architectural principles, which he felt Hypercorp was lacking. He decided to implement an architecture project. His first task was to put Mark Megaboy, his trusted planning manager, in charge of the project.

Together, Roger and Mark decided that the project would have three phases. These are outlined below, and displayed graphically in Exhibit II-1.

- Phase 1: All principles and organizational inventory; 2 months
- Phase 2: Remaining inventory and all modelling; 3 months
- Phase 3: Standards; 2 months

The timeframes for these phases were quite ambitious, but Roger and Mark felt that they would never finish the architecture otherwise. They also felt strongly that there should be a high level of user involvement in the architecture, so they appointed a different task force of I/S and user managers for each phase.

Phase 1

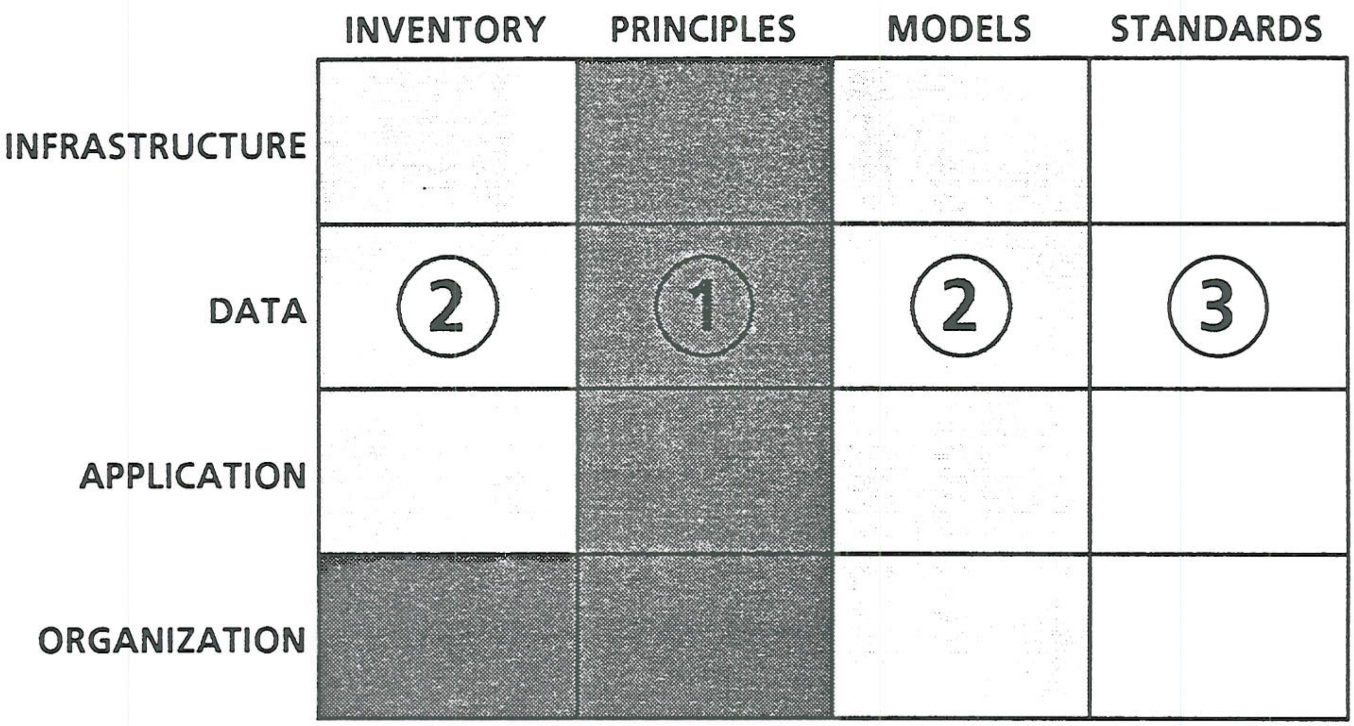
Phase 1 of the architecture had three major objectives. First and most important was to articulate and achieve consensus on values and principles. Secondly, the phase should involve the assessment of interests and skills in the I/S area by key managers and professionals. Thirdly, as a by-product of the first phase, the task force hoped that Phase 1 would make managers aware of the architectural process and get their buy-in and participation.

The best way to achieve these objectives, it seemed to the group, was an education program. The senior management team would be informed as to the need for architecture and the consequences of architectural alternatives. Somewhat secretly, however, the group hoped that this "education" would lead to changes in the managers' values.

The format of the education program called first for an assessment. The assessment was ostensibly to help in the design of the education program, but would also be the input for the organizational inventory. The task force planned to use the assessment interviews to assist them in proposing a set of values and principles during the program. Then the managers would discuss and modify these. The task force hoped that the discussion of the values and principles would center on implications of the statements, and the tradeoffs between various values and principles. The program was scheduled to last for two days, and was held off-site.

Exhibit II-1

Hypercorp's Architecture Project



Hypercorp's Proposed Values

Roger and Mark (and the task force, though attendance lagged toward the end) read all the interview summaries from the assessment and decided that the statements in Exhibit II-2 embodied Hypercorp's information technology values. They are conservative compared to many organizations, and still include a strong central orientation. Most of the interviewing had been done by Mark, and no one had told him that they thought I/S should play a lesser role than the values implied.

The Program Begins

Roger kicked off the program by talking about the overall matrix and the importance of values and principles. He then showed the group the values which were proposed. There were some grumbles from the group that corporate I/S certainly played a strong role in these values. Roger, feeling that things might get out of hand, proposed that the group move on to the principles the task force had come up with, since they were less abstract than the values. One person agreed, and Roger took that as a mandate.

When the discussion of the principles began, it quickly became clear that the actual wording of the principles could not be decided "on line". Rather, Roger and Mark recorded the overall sense of the group on each issue, and promised to consult members of the group about wording.

Hypercorp's Principles

The principles as they emerged from the managers are listed in Exhibits II-3 through II-6. Footnotes indicate points that were particularly controversial. The most interesting aspects of the principles involved the strong role by user managers in the determination and management of applications. It was clear that the values of the firm as a whole did not involve strong control by central I/S. A strong centralized orientation had prevailed in the past, but the architectural work in Phase 1 had revealed a major values change.

Hypercorp remained conservative with respect to technology, choosing to use already-tested technologies. In terms of organization, however, the firm was anything but conservative. Roger and Mark, of course, had strong doubts about how these principles would work out in practice, but they at least felt that they had achieved consensus.

Phase 2

Phase 2 of the architecture project was not nearly as controversial as Phase 1. With the principles determined, the other decisions seemed to fall into place easily. There were two primary tasks in Phase 2: inventory of data, applications, and infrastructure, and model development for all four architecture domains. To accomplish these two tasks, however, there were several specific tasks:

Exhibit II-2

Hypercorp's Values

(AS PERCEIVED BY ROGER & MARK)

RISK:	GENERALLY RISK-AVERSE WITH REGARD TO TECHNOLOGY AND ITS MANAGEMENT.
USER AUTONOMY:	GROWING ROLES FOR USERS, BUT I/S STILL RETAINS PRIMARY CONTROL.
ROLE OF I/S:	INCREASINGLY STRATEGIC, WITH BUSINESS-CRITICAL SYSTEMS DEVELOPED BY I/S.

Exhibit II-3

Data Principles

- DATA SHOULD BE COLLECTED IN A MACHINE READABLE FORM AS CLOSE TO ITS ORIGINAL SOURCE AS POSSIBLE, AND DISTRIBUTED ELECTRONICALLY TO OTHER USERS.[1]

- DATA IS OWNED BY, AND IS THE RESPONSIBILITY OF, ITS CREATOR. THE EXCEPTION TO THIS RULE IS WHEN DATA ARE USED BY TWO OR MORE ORGANIZATIONAL UNITS, IN WHICH CASE IT BECOMES CORPORATE DATA.

- THE LOCATION AND OWNERSHIP OF DATA WILL DRIVE THE LOCATION AND APPROACH TO APPLICATION DEVELOPMENT.

[1] THIS PRINCIPLE WAS LOBBIED FOR BY THE VP OF FINANCE, WHO BELIEVED THAT HIS ANALYSTS SPENT INORDINATE AMOUNTS OF TIME REKEYING DATA FROM PAPER.

Exhibit II-4

Infrastructure Principles

- WE PREFER TO USE TECHNOLOGY ALTERNATIVES THAT HAVE BEEN TESTED WITHIN THE ORGANIZATION; IF THIS IS IMPOSSIBLE, WE PREFER THAT THEY HAVE BEEN USED IN A PRODUCTION CONTEXT IN OTHER COMMERCIAL ORGANIZATIONS SIMILAR TO OURS.

- WE HAVE A PREFERENCE FOR INCREASING THE LEVEL OF USER ACTIVITY IN COMPUTING OVER INVESTING IN A NEW CORPORATE DATA CENTER.

- ALL PROFESSIONAL OR MANAGERIAL EMPLOYEES SHOULD HAVE EASY ACCESS TO COMPUTING IF THEY WANT IT.[1]

- ALTHOUGH WE WILL NO LONGER BE A ONE-VENDOR ORGANIZATION, THE CORPORATE I/S ORGANIZATION WILL CREATE A SHORT LIST OF VENDORS WHICH IT WILL SUPPORT.

- SIMPLICITY OF INFRASTRUCTURE, AND SOLUTIONS WHICH ARE GENERAL, ARE TO BE PREFERRED OVER OPTIMAL SOLUTIONS TO PARTICULAR COMPUTING PROBLEMS.

[1] ORIGINALLY WORDED AS "ALL EMPLOYEES," BUT THE VP OF MANUFACTURING FELT STRONGLY THAT THE RANK-AND-FILE IN PLANTS DID NOT NEED ACCESS TO COMPUTING.

Exhibit II-5

Applications Principles

- APPLICATIONS WILL BE SHARED ACROSS BUSINESS UNITS AND FUNCTIONS AS MUCH AS POSSIBLE.
- MANAGEMENT APPROACHES WILL BE PRIMARILY BY APPLICATION TYPE (E.G., ORGANIZATIONAL, DEPARTMENTAL, INDIVIDUAL) RATHER THAN BY THE TECHNOLOGY ON WHICH APPLICATIONS RUN.
- APPLICATIONS WILL HAVE A STANDARD STRUCTURE TO FACILITATE SHARING OF CODE.
- THE USER MANAGER WHO IS SPONSORING THE APPLICATION WILL BE THE PRIMARY DECISIONMAKER IN DETERMINING THE LOCATION AND MANAGEMENT APPROACH FOR THE APPLICATION.[1]

[1] ROGER POINTED OUT THAT THIS PRINCIPLE MIGHT CONFLICT WITH THE DATA PRINCIPLE SPECIFYING THAT THE LOCATION OF DATA SHOULD DETERMINE APPLICATION BY LOCATIONS. OTHER MANAGERS, HOWEVER, SEEMED UNCONCERNED BY THIS POTENTIAL CONTRADICTION.

Exhibit II-6

Organization Principles

- BUSINESS MANAGERS WHO SPONSOR I/S APPLICATIONS HAVE THE PRIMARY RESPONSIBILITY FOR MANAGING THOSE APPLICATIONS.
- MANAGING INFORMATION SYSTEMS IS A CAPABILITY THAT WILL BE EXPECTED OF ALL MANAGERS, SIMILAR TO MANAGING PEOPLE OR CAPITAL.
- I/S MANAGERIAL POSITIONS WILL PRIMARILY BE TEMPORARY ROLES, AND MANAGERS WILL ROTATE THROUGH THE I/S FUNCTION. THERE MAY BE "DEDICATED" PROFESSIONALS IN THE I/S FUNCTION, BUT ON BECOMING MANAGERS THEY WILL BE MOVED TO OTHER FUNCTIONS WITHIN THE ORGANIZATION.

[1] NONE OF THE PRINCIPLES IN THIS SECTION WERE ORIGINALLY PROPOSED BY RAMBYTE OR MEGABOY. ALL WERE CONTROVERSIAL AND THEY REPRESENTED A RADICAL CHANGE FOR HYPERCORP.

- Key data and applications were to be inventoried. Roger and Mark decided that they should use the Critical Success Factors methodology to determine what data and applications were critical to Hypercorp managers' success. Roger and Mark decided to hire a consultant to perform this task.

- The outcome of the data and applications inventory would be groupings by subject area, e.g., product and customer. User managers would then help to determine subject data base and application locations using "rules" developed by the task force (many of these would be drawn from the 1985 PRISM project on dispersed systems).

- On the infrastructure modeling task, the company was divided into functional sites, and processing and communications needs were to be assessed for each site.

- Using functional site requirements as input, the task force would describe typical generic configurations for workstations, departmental systems, networks, etc., for the various types of functional sites.

- To model the organization, the task force for this phase would describe typical roles, relationships, and necessary skills for information systems management in business units.

The infrastructure model for a typical Hypercorp regional marketing and sales office is displayed in Exhibit II-7. Although the model was very simple, it had been fairly arduous for the task force to achieve consensus on the model, and the group felt a strong sense of achievement. As an interim deliverable of the architecture project, the model was sent out to key managers at Hypercorp along with a simple, nontechnical explanatory letter.

The First Real Test

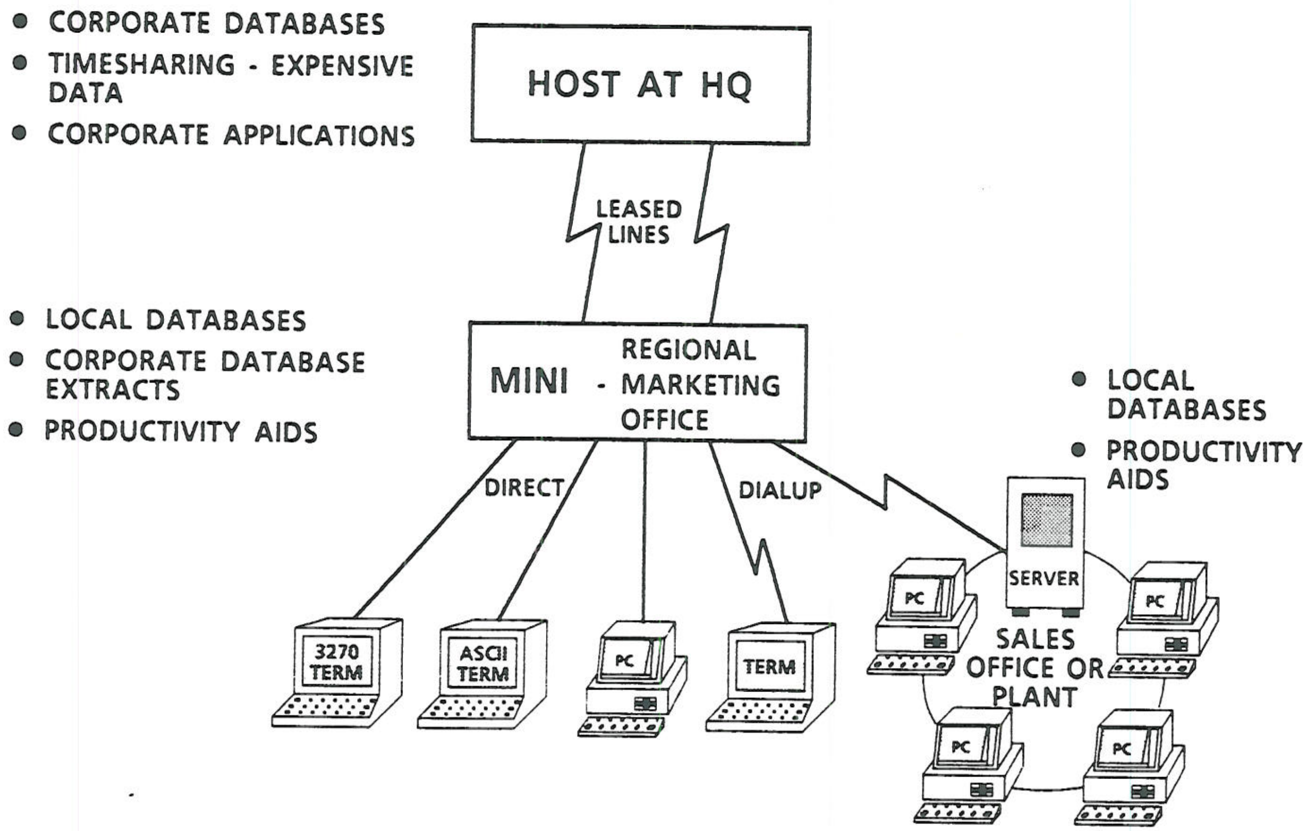
Near the end of Phase 2, the VP of Manufacturing who had denied his workers terminals was the victim of a strike, and the only way to make peace with the union was for him to leave the company. His replacement felt that a new manufacturing system was needed immediately, and this seemed to be an early opportunity to put the architectural principles to work.

Roger explained the principles to him in detail. He accepted the principles, which had been developed without his participation, as valid. However, he insisted on adding cost and productivity enhancement to the list of principles. "We've got to compete with the Japanese or we'll be dead," he said.

A minor crisis ensued. Roger discussed the proposed new principles with several senior executives. Finally, there is agreement that individual managers

Exhibit II-7

Infrastructure Model



can add noncontradictory principles to the list for use in their own business units.

By the time the system for manufacturing is finally chosen, the third phase of the architecture is underway, and a tentative vendor shortlist for manufacturing systems has been compiled by the Phase 3 task force. The system favored by the manufacturing VP is not on the short list, and Roger and Mark persuade the task force to add it. Their feeling is that it is more important not to alienate manufacturing than to preserve the integrity of the architecture process. "Architecture is a process of structuring compromise anyway," Roger and Mark reassured themselves.

Phase 3

The overall objective in Phase 3 was standards, which meant several different activities depending on the domain area. In the infrastructure domain, the task was to identify standard vendors for hardware and software: departmental processors, local area networks, departmental databases, etc. The task force for Phase 3 also decided to review Hypercorp's standards for personal computing hardware and software, and to review the choice of a mainframe database management system.

In the data area, the task force began to create data naming and management conventions for new databases; they ignored the existing databases. For the important product/distributor/customer databases, the task force queried business units about their need for data extracts.

In the business units, managers were asked to create their own I/S support plans. They also began to determine the locations - either in their own units or at corporate I/S - of the applications that were already in the development queue.

The process the task force used to determine a vendor list for departmental processors is displayed in Exhibit II-8. Two principles from Phase 1 were relevant, so the candidate processors were ranked according to their fit with the principles. Some of the model decisions in Phase 2 were also relevant, and the processors were ranked accordingly. Finally, two of the workstation decisions already decided in Phase 3 were used as evaluation criteria for the processors. Because the senior management group had decided on a list of vendors to support rather than only one choice, the first two vendors were chosen to be on the list. All parties involved felt that the process was quite simple and straightforward - a big improvement over their previous efforts at selecting processors.

The End of the Architecture Project

Near the end of Phase 3, when Mark Megaboy was beginning to wonder what he would do when the architecture project was finished, Hypercorp was acquired by the Megadata Corporation. All architecture work stopped, and

Exhibit II-8

Vendor Selection Process - Departmental Processors

	VAX	36	WANG
PHASE 1 CONSIDERATIONS:			
• TESTED TECHNOLOGIES	+		+
• SIMPLICITY OVER OPTIMALITY	+	+	
PHASE 2 CONSIDERATIONS:			
• DBMS DEVELOPMENT AT REGIONAL LEVEL	+		
• HOST INTERCONNECTION - SNA	+	+	
• 3270 PASSTHROUGH	+	+	
• EXTRACTION AND DOWNLOADING EASE			+
PHASE 3 CONSIDERATIONS:			
• ETHERNET CONNECTION	+		
• IBM PC INTERFACE		+	+
TOTAL	6	4	3

Mark took advantage of an early retirement program. One of Hypercorp's senior managers told the new chairman from Megadata about the architecture work that had taken place at Hypercorp. The Chairman was impressed by the firm's principles, models and standards; he told his VP of IS that the principles should be adopted with few modifications, and that Roger should be put in charge of planning and architecture for the new organization.

Although Megadata's infrastructure used a different vendor and was almost totally incompatible with Hypercorp's architecture, the work done in Hypercorp's architecture project was valuable even beyond principles. Because Megadata had no formal architecture, in constructing a new one Roger was able to draw extensively on many models and standards that had been developed for Hypercorp.

Although Roger was glad that the project had been of some use after the acquisition, he spent many nights in bed wondering about whether he had done the right thing. Could he have anticipated the merger, and adjusted the architecture accordingly? At Mark's retirement dinner, he and Roger decided that there was nothing they could have done about it. Had they tried to factor in a merger, they would have been unable to make any progress at all. Mark told Roger that he thought Roger's chances were good for taking over the Megadata vice presidency soon.