

DESIGNING INFORMATION TECHNOLOGY ARCHITECTURES: A COST-ORIENTED METHODOLOGY

Chiara Francalanci, Vincenzo Piuri

Dipartimento di Elettronica e Informazione

Politecnico di Milano

ABSTRACT

This paper proposes a design methodology of information technology architectures tying organizational requirements to technical choices and costs. The primary goal is to provide a structured support to the selection of the minimum-cost architecture satisfying given organizational requirements. Previous empirical studies have attempted absolute cost comparisons of different architectural solutions, primarily relying on the expertise of practitioners and "a priori" beliefs, but have rarely taken into account the impact of organizational requirements on costs. Requirements are modeled as information processes, composed of tasks exchanging information and characterized by varying levels of computational complexity. Different architectural distribution of presentation, computation, and data management applications are compared. The cost implications of organizational requirements for processing intensity, communication intensity, and networking are analyzed. Results show a relationship between structural features of information processes and architectural costs and indicate how architectural design should be based on organizational in addition to technology considerations.

1. INTRODUCTION

Information system design and optimum sizing is the result of a reconciliation of several conflicting requirements, including technical performance and costs, organization impact, and user acceptance. Theoretical research and practitioners often focus on specific techniques for the optimization of individual design phases, usually leading to local technical optima, with little understanding of the relationships among different choices at different design stages.

This paper attempts the definition of a comprehensive design methodology that covers all phases of information system design in order to tie organizational requirements to technical choices and resulting costs. The main goal is the identification of a sequence of design steps that track the cost implications of design choices from requirements analysis to physical implementation. Each step is modeled according to available techniques, borrowing from the literature the modeling tools to represent goals, constraints, and structure of the information system.

The focus is on architectural design tying the information requirements of an organization to the IT components necessary to satisfy those requirements: costs are associated with these components (cf. Zachman 1987, Sowa and Zachman 1992). This process has many degrees of freedom, since different architectures can be built to satisfy a given set of requirements. The proposed methodology guides the system designer through a sequence of choices at different design steps in order to distinguish alternative solutions and evaluate their costs. The primary goal of the methodology is to help choose the architecture that minimizes the overall cost (including hardware, software, maintenance, management and training).

Organizational requirements have been modeled as processes, composed by tasks with a certain level of computational complexity, exchanging information among them and classified as presentation, computation or data processing. At an organizational level, the main cost differentiating factors influencing architectural choices are *intensity of processing* (measured by task complexity), *intensity of communication* (measured by the quantity of exchanged information), and *extent of communication* (measured by the level of networking in the process).

All these characteristics - either directly or indirectly - influence IT costs (cf. Alter 1996, Blyler and Ray 1998, Smithson 1994, Willcocks 1992). Intensity of processing mainly defines the computational capability and related technical features of the processing units. Intensity of communication typically affects the speed characteristics of the interconnection links. Group user interactions constrain the information and computation distribution in the information system. In turn, these requirements influence the choice of the IT architecture in a wide range of possible solutions, from centralized to fully distributed. The optimal selection of the most cost-effective solution needs an accurate analysis and cost evaluation of all technical options satisfying given requirements, since, theoretically, all architectures can provide any high level of processing ability, although at markedly different costs (cf.

Simpson 1997a and 1997b). The proposed methodology facilitates this choice by systematically analyzing technical options and their costs against organizational requirements.

The next Section reviews previous approaches and highlights the main organizational and technical variables of interest. Section 3 presents the phases of the design methodology and the corresponding architectural models. Section 4 discusses how architectural costs, including hardware, software, communication, and management costs change with different technical choices. Section 5 and 6 report and discuss experimental results from a comparative analysis of costs of different architectural choices with varying organizational requirements. Finally, Section 6 draws conclusions and suggests directions for future research.

2. RESEARCH BACKGROUND AND MOTIVATION

Cost analyses of technical choices have highlighted numerous design trade-offs primarily related to the appropriate sizing and location of computing resources within an IT architecture (Guengerich 1992). One of the most widely discussed design trade-offs is described by Grosch's law, recently reformulated as "it is most effective to accomplish any task on the least powerful type of computer capable of performing it" (Ein-Dor 1985). A traditionally lower cost of processing capabilities on smaller computers and a generally reduced software complexity have created the belief that a decentralized IT architecture would involve lower overall costs for an organization.

A number of technical factors would instead work against decentralization. First among them, communication cost can be significantly higher in a decentralized architecture, due to a more cumbersome data retrieval and consolidation in distributed transaction processing (Gavish and Pirkul 1986, Lee et al. 1994). Dynamic file location and file migration techniques have been studied to reduce communication costs and improve transaction performance in decentralized environments (Liu Sheng 1992). In practice, databases are duplicated to limit remote data access through geographical networks, while raising data consistency issues and causing additional communication load to align multiple copies of the same data. Decentralized architectures also increase the variability of processor load and reduce their utilization, which may raise costs particularly in real-time and fault-tolerant distributed systems (Distante and Piuri 1989, Piuri 1994). The most general

technical model answering the research question of this paper is provided by Jain (1987), studying the optimal sizing of a distributed system to minimize the costs of communication for remote transactions, processors, primary and secondary storage, while guaranteeing maximum file availability and minimum response time for queries. Other results concerning optimum sizing of distributed architectures for special applications in real-time and fault-tolerant distributed systems can be found in Distanto and Piuri (1989) and Piuri (1994).

While pointing to the relevant technical cost trade-offs, previous analyses have failed to consider management costs, which can be greatly dissimilar across different IT architectures. Ahituv et al. (1989) distinguish two separate distribution policies, technical and organizational, privileging technology and management considerations, respectively. Management costs represent a relevant dimension of organizational policies for choosing an IT architecture and have been empirically found to shift the balance of technical trade-offs (Moad, 1994, Dec 1996, Simpson 1997a and 1997b). Distributed architectures have in fact been extensively adopted in organizations only in the early 1990s and some of the expected cost reductions have been demonstrated elusive.

One of the claims to explain these failures is that the overall cost of a distributed architecture depends on organizational requirements, which may or may not favor distribution (Guengerich 1992). For example, the reengineering literature has advocated the need of aligning organizational structure and technology and contributed to creating the belief that a flatter, process-oriented and more networked organization would be best served by a distributed IT architecture (Hammer and Champy 1993, Jarvenpaa and Stoddard 1993, Venkatraman 1994). However, the relationship between the overall costs of an IT architecture and the structural features of organizational processes has not been investigated. The few empirical results that have been published present absolute cost comparisons between centralized and client-server architectures and do not discuss the impact of process features on findings (Moad, 1994, Dec 1996, Simpson 1997a and 1997b). Recently, practitioners are considering the re-centralization of data and application functionalities in order to decrease management effort and costs. This seems to indicate that the design of the IT architecture is often guided more by current beliefs and market offer, than by an understanding of organizational requirements and technology costs.

Initial organizational requirements and resulting architectural costs are tied by design methodologies that incrementally translate requirements into technical choices (Aue and Breu 1994). This paper proposes a methodology that models such choices, accounting for technical as well as management trade-offs. Design phases are relatively consolidated (Aue

and Breu 1994), but each one of them has been primarily studied in isolation and often further partitioned into sub-phases (Alter 1996). For example, requirements design has been addressed both in the information system and organizational literature. The latter typically views information processing capacity as a factor of organizational performance, with limited attention to technology costs incurred to support different levels of information requirements (Galbraith 1973 and 1977, Lin and Carley 1997). The information system literature deals with representational issues of organizational requirements, leaving costs as an issue for subsequent design phases more directly involved in technology choices (De Marco 1978, Coad and Yourdon 1991). Aggregate perspectives evaluating the effect of structural properties of processes on costs have rarely been taken.

3. DESIGN METHODOLOGY

The architectural design process ties the information requirements of an organization to the physical components necessary to satisfy those requirements. Acquisition and usage costs are associated with these physical components. The architectural design process offers many degrees of freedom, since in most cases different architectures can be built to satisfy a given set of requirements. These degrees of freedom involve choices at different stages of the architectural design (Aue and Breu 1994). We consider four design phases, each producing a corresponding model of the IT architecture:

1. *Information requirements design*, defining organizational requirements for information processing capacity;
2. *Technology requirements design*, defining information technology resources necessary to satisfy given requirements for information processing capacity;
3. *Infrastructural design*, allocating the required technology resources onto separate and interconnected computers;
4. *Physical design*, associating computers and communication links with minimum-size commercially available devices.

Each model of the technology architecture is translated into the model produced in the subsequent design phase by a specification activity (Figure 1). This specification involves design choices that, in turn, affect architectural costs. The following subsections describe the design phases, the corresponding models, and the design choices involved by specification activities.

3.1 The Information Requirements Model

Organizational requirements for information processing capacity are modeled as *information processes*. Table 1 summarizes the complete set of elements constituting the model and their symbols.

An information process (referred to subsequently as *process*) is described by a set of tasks $\{T_i | i=1..n\}$ and the input-output *interdependencies* among them. We consider processes as information processing entities and limit interdependencies among tasks to information interdependencies. T_i is input interdependent with T_j if T_j produces information I_{ji} that is input to T_i ; conversely, T_j will be output interdependent with T_i .

We consider the information I_{ij} and I_{ji} exchanged between two interdependent tasks T_i and T_j to be measured as an *information exchange rate*, that is the average quantity of information exchanged in a time unit. Information quantities are defined as integer multiples of the smallest piece of information that can be exchanged in a given organization. Records or information objects are examples of information units that are used in this design phase. A task T_i can receive input information from multiple output interdependent tasks and convey output information to multiple input interdependent tasks. The overall input information of task T_i is $I_{input,i} = \sum_x I_{xi}$ and the overall output information is $I_{output,i} = \sum_y I_{iy}$.

Each process has single entry and exit tasks, called *source* and *sink*, respectively. Information interdependencies with source and sink represent information exchanges with the world external to the process.

A task T_i can be repeatedly accomplished by n_i human executors working in parallel, each of them completing k_i activations of the task in a time unit. This keeps the decomposition of a process into tasks at a conceptual level, while separately representing the number of a task instances by the same or different individuals. With these definitions, a task has $n_i k_i$ instances in each time unit.

Tasks are also associated with a factor χ_i measuring their *complexity*, defined as the number of high-level instructions procedurally describing the task. This definition assumes that a task corresponds to a software application, whose procedural high-level requirements are available at this stage of design. This one-to-one correspondence simplifies the design process, but it does not affect the final relationship between structural features of information processes and architectural costs. Possible inaccuracies in the number of high-level

instructions can be reduced if the assessment is obtained for all tasks either from the same designer or by using the same evaluation techniques.

The information requirements model does not make a distinction between different types of information, nor does it provide qualitative descriptions of information interdependencies. This orientation differs from conceptual models of information requirements proposed in the information system literature, traditionally emphasizing descriptive representations of information (DeMarco 1978). Describing information is critical to guarantee the conceptual completeness and correctness of requirements design. Although maintaining many of the representational constructs from the information system literature (e.g., tasks, information interdependencies, and complexity), we aim to the most parsimonious representation of cost-differential characteristics of information processes.

3.2 The Technology Requirements Model

Technology requirements design associates tasks and information interdependencies with required technology resources. Table 2 explains the specification activities leading from the elements of the information requirements model to the corresponding elements of the technology requirements model.

Each task T_i corresponds to a software application A_i , being the total number of applications equal to the total number of tasks. A one-to-one correspondence between tasks and applications also involves an equal number of executors n_i and activations k_i for tasks and corresponding applications in a time unit. Applications are associated with primary and secondary memory requirements PM_i and SM_i , respectively, for data, executable code, and temporary structures supporting execution. Memory requirements are usually related to the application's computational complexity, generally growing with it. However, they are not univocally determined by complexity, since different applications may create local data to varying degrees. Secondary memory requirements are assumed to include the storage of the application itself.

Information interdependencies are translated into quantities of data exchanged in a time unit by specifying a conversion factor α between information and data units (e.g., between records and bytes). Since information interdependencies have been defined as multiples of the same unit of information, α is constant across interdependencies, both between generic tasks (I_{ij}) and with the external world ($I_{source,i}$, $I_{i,sink}$). A label M_{ij} is also

attached to data transfers to indicate the type of data transfer, either in primary (PM) or secondary (SM) memory. Task complexity is translated into computational complexity by specifying a conversion factor β_i from high-level to machine-level instructions. If complexity can be assumed to be homogeneous across tasks, β_i can be considered constant.

3.3 The Infrastructural Design Model

Infrastructural design allocates the technology resources required to perform the tasks specified by the information requirements model on the computers composing the IT architecture. The three main resources to be allocated are processing capacity, memory, and communication load. A computer, to which a number of users n_i of an application A_i are assigned from the technology requirements model, should provide enough hardware features to support computation and communication requirements. In particular, it must guarantee that:

1. each activation of A_i is attributed PM_i and SM_i shares of primary and secondary memory;
2. $n_i k_i$ activations of A_i can be executed in a time unit.

In order for $n_i k_i$ activations of A_i to be executed in a time unit, a computer's processing capacity should evaluate at least to $n_i k_i \chi_i$ machine-level instructions per time unit. If k_i measures the number of activations per second, $n_i k_i \chi_i$ expresses processing capacity in MIPS (millions of machine-level instructions per second); MIPS can be translated into MHz, a common unit for processing capacity of small computers, through the number of cpi (cycles per instruction) (Patterson and Hennessy 1994). If multiple applications $\{A_1, \dots, A_k\}$ are assigned to the same computer, processing capacity must be at least $\sum_{i=1..k} n_i k_i \chi_i$.

Similarly, total memory requirements are the summation of the primary and secondary memory requirements for all activations, that is $\sum_{i=1..k} n_i k_i PM_i$ and $\sum_{i=1..k} n_i k_i SM_i$, respectively. Secondary memory requirements are supposed to include an estimate of the data created by the application and, possibly, stored in a common database. By referring to data created as opposed to data accessed, the summation of secondary memory requirements for $\{A_1, \dots, A_k\}$ involves minimal redundancy among different applications.

We assume that an application is executed on a single machine, possibly multiprocessor, but never partitioned and remotely distributed for execution to separate machines. If the designer needs to model parallel and cooperative execution of subtasks

within a given task, task partitioning must be explicitly represented in the information requirements model. The main types of cooperative processing can be modeled by classifying applications as presentation, computation, and data management, i.e., as activities related to user interaction, data processing, and data retrieval and storage, respectively.

Interactions among cooperative applications supporting a process are modeled as information exchanges. Applications A_i and A_j exchange data αI_{ij} between each other. If A_i and A_j have been allocated to the same computer, the data transfer αI_{ij} is *local*, while it is *remote* otherwise. Local and remote data transfers involve different memory requirements. A local data transfer in primary memory is supposed to take place in an area of memory of size αI_{ij} shared between the two applications. A remote data transfer in primary memory would instead require a quantity of memory αI_{ij} on both communicating computers. A remote data transfer in secondary memory is supposed to avoid the duplication of stored data, requiring a quantity of memory αI_{ij} on only one of the communicating computers. Figure 2 shows the total amount of data transferred between computer N^{th} and M^{th} . The communication link between them should provide a communication capacity proportional to the total amount of transferred data, the physical distance between the two computers and response time requirements for each remote data transfer.

Similarly, data transfers $\alpha I_{\text{source},i}$ and $\alpha I_{i,\text{sink}}$ with the external world translate into memory requirements for the computer where A_i is allocated. The required memory is primary or secondary as specified by the labels $M_{\text{source},i}$ and $M_{i,\text{sink}}$. Also communication with the external world can be distinguished into local and remote.

3.4 The Physical Model

Physical design associates physical components to the requirements specifications of the infrastructural model, by identifying commercially available computers and communication links that offer at least the required capacity in terms of processing, memory, and communication. This design phase is typically vendor dependent, since each manufacturer offers a different set of computers with distinct basic configurations and expansion components. To support physical design, vendors use computer applications, called *configurators*, that select and analyze the commercial components necessary to satisfy customers' computing requirements. When requirements are imprecise or there exist

multiple options, configurators may compare alternative solutions against cost and performance criteria by adopting optimization techniques to explore the range of possible solutions.

The simulation described in Section 5 is based on a simplified configurator operating on a vendor-independent component database. A limited number of basic computer configurations have been considered, selected as those common to the majority of commercial configurators. Additional components to expand basic configurations and obtain a higher capacity have also been included. 34 basic configurations have been stored in the configurator, ranging from 133 Mhz Pentium to a mainframe with 12 parallel 200 Mhz processors. The primary feature determining the choice of a basic configuration is CPU speed, which has to be equal to or greater than the processing capacity of the corresponding computer in the infrastructural design model. To satisfy memory requirements, it may be necessary to expand the selected basic configuration. Particularly high memory requirements may drive the choice of the basic configuration, given the practical limits to memory expansion.

Communication links are distinguished into *remote* or *local*, ranging from 14 Kbit/s to 100 Mbit/s. The choice of a local vs. remote communication link will depend on the physical location of communicating computers and sources of external information. In both cases, the discriminant feature of a physical communication link is capacity, most often measured in bits per second (bps). A physical link's capacity should be equal to or greater than the capacity of the corresponding link between computers in the infrastructural requirements model.

4. ARCHITECTURAL COSTS

Technology resources involve different costs over their life cycle to be purchased, installed, operated and maintained. Costs are classified into hardware, software, communication, and management, with the definitions reported in Table 3. The literature documents numerous cost classifications, typically targeted to technology cost accounting (cf. Parker et al. 1988, Guengerich 1992). A distinction between technology and management costs is common to most of these classifications. A further specification of technology costs into hardware, software and communication allows a more precise association between costs and physical components of the IT architecture.

Each class of costs is associated with a different set of physical components of the IT architecture. Hardware costs are associated with computers, expansion components, and communication links. Software costs are associated with applications. Management costs are associated with computers, communication links and applications. Physical components may not incur all costs within a particular class. For example, the majority of expansion components do not involve any user training, nor do they require a specific operating system. Being associated to physical components, costs are unitary.

In our experimental component database, hardware costs were obtained as a market average of 6 different vendors, encompassing IBM, HP, Zenith, Digital, Compaq, and Olivetti. Communication costs were calculated as the average among IBM, HP, Digital, Bull, Olivetti, and Telecom. Cost data were collected from vendors' web sites and from benchmarking sites, such as "<http://www.tpc.org>". Software costs were separately gathered for software packages and in-house developed applications. The characteristics for the main software packages by Microsoft, Oracle, SAP, IBM, Digital, HP, and Bull were obtained from vendors' web pages; vendors' web sites provide in fact thorough descriptions of software packages, including functionalities, size, hardware requirements, and costs. The cost of in-house developed applications was calculated as a function of computational complexity. The site "<http://www.davidconsultinggroup.com/dcgindu.htm>" is an example of the relationship between software costs and size, as it has been adopted in this research. Management costs were obtained from benchmarking sites and triangulated with the empirical findings of field surveys (Moad, 1994, Dec 1996, Simpson 1997a and 1997b).

The literature documents numerous cost classifications, typically targeted to technology cost accounting (Parker et al. 1988, Guengerich 1992). A distinction between technology and management costs is common to most of these classifications. A further specification of technology costs into hardware and software allows a more precise association between costs and physical components of the IT architecture.

5. EXPERIMENTAL RESULTS

The proposed design methodology has been applied to compare the costs of different architectures with varying structural features of information processes in the information requirements model. In particular, the following structural features of processes have been studied:

- the *intensity of processing*, measured as tasks' computational complexity χ_i ;
- the *intensity of communication*, measured as the rate of information exchanges I_{ij} , $I_{i,sink}$, and $I_{source,i}$;
- the *extent of communication*, measured as the level of process networking.

Processes defined with the information requirements model can be considered generic graphs and analyzed with graph analysis techniques (Harary 1969, Foulds 1992, Goutis 1995). The mathematical definition of process networking was then borrowed from graph theory. For the purposes of this paper, it is sufficient to know that networking is a number ranging between 0 and 1, measuring the number of tasks input or output interdependent with one another relative to the total number of tasks.

Several physical architectures have been considered, which are differentiated by the level of distribution of presentation, computation and data management applications among client and server computers. Figure 3 shows 8 different architectures obtained with different allocations of applications. This view extends the discretization of the architectural choices commonly used in the practice (Faye, Borthick and Roth 1994, Blyler and Ray 1998). The distribution choice is practically useful since it defines different levels of client-server architectures, as processing migrates from the server to the client side (Blyler and Ray 1998). The central architectures in Figure 3 are in fact generically referred to as *client-server*, as opposed to the extreme centralized and stand alone solutions.

The cost analysis and comparison among these architectural choices was supported by an Excel-based tool developed in Visual Basic helping creation, modification, and versioning of projects according to the methodology explained in Section 3.

Experimentation ranged between 50 and 250 tasks in the information requirements model. Networking was increased from 0.086 to 0.25 (obtained when, on average, each task is interdependent with about 25% of the total number of tasks). The order of magnitude of information interdependencies between task pairs was limited between 100 and 10000 bps. Architectures with 1, 3 and 6 servers were tested with different values of structural properties of processes.

Table 4 summarizes how cost trade-offs among different architectural choices vary with organizational requirements for intensity of processing, intensity and extent of communication. As an example, Figure 4 shows the minimum-cost solution with the distributed data and computation architecture between 50 and 500 users. The required computational complexity and number of clients are reported on the horizontal and vertical axes, respectively. Each area is labeled by the identifiers of the chosen client and server, whose increasing value

corresponds to greater computing capacity. Similar charts have been built for the distributed presentation and distributed application architectures. Figure 5 shows the minimum-cost architecture with varying requirements for computational complexity and number of users. Grey areas in Figure 6 represent values of organizational requirements for which the minimum-cost solution depends on the specific data point.

6. DISCUSSION

Findings highlight a dependence of costs on the structural features of processes. Both networking and information interdependencies modify the overall costs of different architectural choices, changing the minimum-cost solution (*Findings 3 to 8*). On the contrary, computational complexity involves a growth of costs comparable across different architectures and does not affect the cost differences among them (*Finding 2*).

The most interesting finding from an organizational perspective is the impact of networking on costs. A client-server architecture seems to minimize IT costs for information processes with high levels of networking (*Finding 5*). This is basically due to the opportunity of exploiting parallelism in process execution with a limited amount of shared data. However, different client-server solutions show a high variance in their overall costs. While the Distributed Data and Computation architecture has minimum costs, the Remote Data Management would involve maximum costs, significantly higher than a traditional centralized architecture. Variance is mostly related to communication costs that are minimized by local data management on client computers in Distributed Data and Computation architectures.

Intuitively, high networking raises the need for a direct communication among client computers. A low interaction among users' activities suggests the decentralization of application functionalities (i.e., presentation and computation) and the centralization of data storage to facilitate information sharing. The decentralization of application functionalities reduces network traffic, while splitting either computation or presentation between the client and the server would increase the amount of data to be transferred and, as a consequence, communication costs. Conversely, strong data dependencies among users' activities suggests the centralization of application functionalities to reduce the amount of data to be exchanged over the network. The optimum solution consists in decentralizing computation and data management as long as processes of different users do not involve frequent interactions and data sharing.

More specifically, maintaining presentation, computation and data management on centralized computing systems simplifies the operation of the IT architecture and reduces the number of system managers and operators and, thus, management costs. Moreover, the computational demand to support presentation for a high number of users requires high-performance architectures and, consequently, high technology acquisition and maintenance costs.

Distributing presentation over clients both reduces computational requirements on the server and enables the development of interactive and friendly user interfaces. Graphic interfaces can in fact be efficiently supported by local processing capacity on client computers, cheaper and more abundant than on servers. Transferring only the information to be presented and not the whole screen limits the quantity of data to be transmitted over the network, which may otherwise become a bottleneck to processing performance. Similarly, distributing either part or the whole computation over distributed machines is feasible when users' activities are not closely interrelated and the computation moved onto clients uses a limited quantity of data from the server.

Data management can also be distributed if data subsets accessed by processes allocated on different clients have a limited overlap. The quantities of information that are exchanged through the network have been found to be a driver of cost, which is minimized with Centralized architectures (*Finding 4*). Centralization reduces management and software costs by granting the IS function with scale economies primarily related to human resources (*Findings 7 and 8*). These economies seem to offset the higher hardware costs of centralized computing. Distributed architectures require both additional management personnel and software, for example to deal with distributed software versioning, database consistency, network management and tuning.

As networking increases, higher communication requirements shift this trade-off towards client-server solutions with a distributed allocation of Computation and Data Management (*Findings 5 and 6*). Note that the number of servers has been found to be a purely technical choice independent of structural features of information processes (*Finding 1*), as long as user activities can be split on different servers and servers can be connected by high-performance links.

Figure 5 shows how no IT architecture has absolute lowest costs, even if management costs are accounted for. Traditional technical variables modeling architectural choices remain cost discriminating and management costs shift technical trade-offs towards centralization, but have been found not to determine the minimum-cost solution. Grey areas

in Figure 5 suggest significant variations in the architectural solution minimizing costs even for small changes in organizational requirements. When design is limited to a particular type of IT architecture (e.g. distributed data and computation in Figure 4), these variations seem to become particularly frequent for certain values of requirements. Furthermore, as requirements increase along one of the axes in Figure 4, minimizing costs may require to implement and then to roll back architectural changes. For example, as requirements for processing capacity increase from 9 to 10 MIPS, server 22 should be replaced by server 23 and then restored. On one hand, these results confirm that architectural design needs a systematic approach to be tied to organizational requirements. On the other hand, other considerations, such as scalability of the IT architecture, seem to be necessary to avoid local optima and frequent architectural changes.

6. CONCLUSIONS

The main contribution of this paper is to show a relationship between structural features of information processes and architectural costs. With respect to previous research, findings are interesting as they allow a finer comparison across different architectures by accounting for process requirements. Previous empirical studies attempted an absolute cost comparison across architectures and rarely discussed the impact of process features on their results. The optimum architectural choice from the point of view of costs and organizational requirements has been shown to be not unique, but strongly dependent on the specific application requirements since hardware, software, maintenance, training and management costs have conflicting behaviors among the different structural solutions.

While the computer and cost database that supported experimentation is time dependent, the methodology to translate process requirements into physical choices is general. Results may partly depend on the high costs of centralized hardware in state of the art technology, favoring client-server with respect to centralized architectures. As hardware and communication costs continue to decrease, the minimum-cost architecture may change. However, the same methodology and support tool would allow the amendment of cost analyses, once technical parameters are updated.

Future research and experimentation will be directed to validate the proposed approach with practical large-scale cases. Additional testing for a broader range of process features will be also required in order to generalize results. In this paper we focused our attention on costs and performance of the IT architecture supporting the IS. However, these

aspects are not the only ones relevant to choose the most suitable solution for the application environment and the organization. A few conceptual extensions would be required to measure scale effects on costs linked to the size of the IT architecture, currently captured only when related to software and data sharing among tasks. Scalability should also be considered since it might affect costs when IS's are evolving rapidly. Constraints on the IT components could be added to deal effectively with hardware and software legacies when their expected life time is not expired and their acquisition costs are still to be amortized. User acceptance and system usability should also be evaluated since they affect the actual productivity of the IS and, in turn, of the whole organization. Besides, costs related both to organizational changes and to IT organizational maturity should also be accounted for, although extremely difficult to quantify.

An interesting development would complete the architectural analysis by accompanying the analysis of costs with performance considerations. Notwithstanding the emphasis on costs in the professional literature, the lowest-cost solution does not necessarily represent the best alternative. Technical and organizational performance reflections would improve the understanding of costs and benefits of long-term choices for the information technology architecture.

REFERENCES

- Ahituv, N., Neumann, S., Zviran, M. (1989), "Factors Affecting the Policy for Distributing Computing Resources," *MIS Quarterly*, Dec., 389-400.
- Alter, S. (1996), *Information Systems: A Management Perspective*, The Benjamin/Cummings Publishing Company, Inc.
- Aue, A., Breu, M. (1994), "Distributed Information Systems: An Advanced Methodology," *IEEE Trans. on Software Engineering*, 20(8), 594-605.
- Blyler, J. E., Ray, G. A. (1998), *What's size got to do with it? Understanding computer rightsizing*, IEEE Press, Understanding Science & Technology Series.
- Coad, P., Yourdon, E. (1991), *Object-Oriented Analysis*, Yourdon Press Computing Series.
- Dec, K. (1996), "Gartner View: Client/Server Payoff," *CIO*, April 15, 72-76.
- De Marco, T. (1978), *Structured Analysis and Systems Specification*, Englewood Cliffs, N. Y. Prentice Hall.
- Distante, F., Piuri, V. (1989), "Hill-climbing heuristics for optimal hardware dimensioning and software allocation in fault-tolerant distributed systems," *IEEE Trans. on Reliability*, 38(1), 28-39.
- Ein-Dor, P. (1985), "Grosch's Law Revisited: CPU Power and the Cost of Computation," *Communications of the ACM*, 28(2).
- Faye Borthick, A., Roth, H. P. (1994), "Understanding Client/Server Computing," *Management Accounting*, Aug., 36-41.
- Foulds, L. R. (1992), *Graph Theory Applications*, Springer Verlag, NY, Inc.
- Francalanci, C., Piuri, V., (1998), "Designing Information Technology Architectures: a Cost-oriented Methodology," *Proceedings 6th European Conference on Information Systems*, Aix-en-Provence, France, June 4-6, 1998, 953-968.
- Galbraith, J. R. (1973), *Designing Complex Organizations*, Addison-Wesley Publishing Company Inc.
- Galbraith, J. R. (1977), *Organization Design*, Addison-Wesley Publishing Company Inc.
- Gavish, B., Pirkul, H. (1986), "Computer and Database Location in Distributed Computer Systems," *IEEE Trans. on Computers*, 35(7), 583-590.
- Goutis, C. (1995), "A Graphical Method for Solving a Decision Analysis Problem," *IEEE Trans. on Systems, Man, and Cybernetics*, 25(8), 1181-1193.
- Guengerich, S. (1992), *Downsizing Information Systems*, SAMS-Prentice Hall Computer Publishing.

Hammer, M., Champy, J. (1993), *Reengineering the Corporation*, HarperCollins, New York:NY.

Harary, F. (1969), *Graph Theory*, Addison-Wesley Publishing Company.

Hayes, R. H., Wheelwright, S. C., Clark, K. B. (1988), *Dynamic manufacturing: Creating the learning organization*, New York: Free Press; London: Collier Macmillan.

Jain, H. K. (1987), "A Comprehensive Model for the Design of Distributed Computer Systems," *IEEE Trans. on Software Engineering*, 13(10), 1092-1104.

Jarvenpaa, S., Stoddard, D.B. (1993), *Business Process Reengineering: Tactics for Managing Radical Change*, working paper, Harvard Business School, Boston, MA.

Kwang, K., Briggs, F. A. (1984), *Computer architecture and parallel processing*, McGraw Hill.

Lee, H., Shi, Y., Stolen, J. (1994), "Allocating Data Files over a Wide Area Network: Goal Setting and Compromise Design," *Information & Management*, 26, 85-93.

Lin, Z., Carley, K. M. (1997), "Organizational Response: The Cost Performance Tradeoff," *Management Science*, 43(2), 217-234.

Liu Sheng, O. R. (1992), "Optimization of File Migration Policies in Distributed Computer Systems," *Computers Ops. Res.*, 19(5), 335-351.

Moad, J. (1994), "Client/Server Costs: Don't Get Taker for a Ride," *Datamation*, Feb. 15, 34-41.

Parker, M. M., Benson, R. J., Trainor, H. E. (1988), *Information Economics*, Prentice-Hall International Editions.

Patterson, D. A., Hennessy, J. L. (1994), *Computer Organization and Design: The Hardware/Software Interface*, Morgan Kaufmann

Piuri, V. (1994), "Design of fault-tolerant distributed control systems," *IEEE Trans. on Instrumentation and Measurements*, 43(2), 257-264.

Simpson, D. (1997a), "Cut Desktop Management Costs!," *Datamation*, Jan., 102-105.

Simpson, D. (1997b), "Will NCs Save you a Bundle?," *Datamation*, May, 100-105.

Smithson, S. (1994), "Information Retrieval Evaluation in Practice: A Case Study Approach," *Information Processing & Management*, 30(2), 205-223.

Sowa, J. F., Zachman, J. A. (1992), "Extending and Formalizing the Framework for Information System Architecture," *IBM Systems Journal*, 31(3), 590-616.

Venkatraman, N. (1994), "IT-Enabled Business Transformation: From Automation to Business Scope Redefinition," *Sloan Management Review*, Winter, 73-87.

Willcocks, L. (1992), "Evaluating Information Technology Investments: Research, Findings, and Reappraisal," *Journal of Information Systems*, 2, 243-268.

Zachman, J. A. (1987), "A Framework for Information System Architecture," *IBM Systems Journal*, 26(3), 276-292.

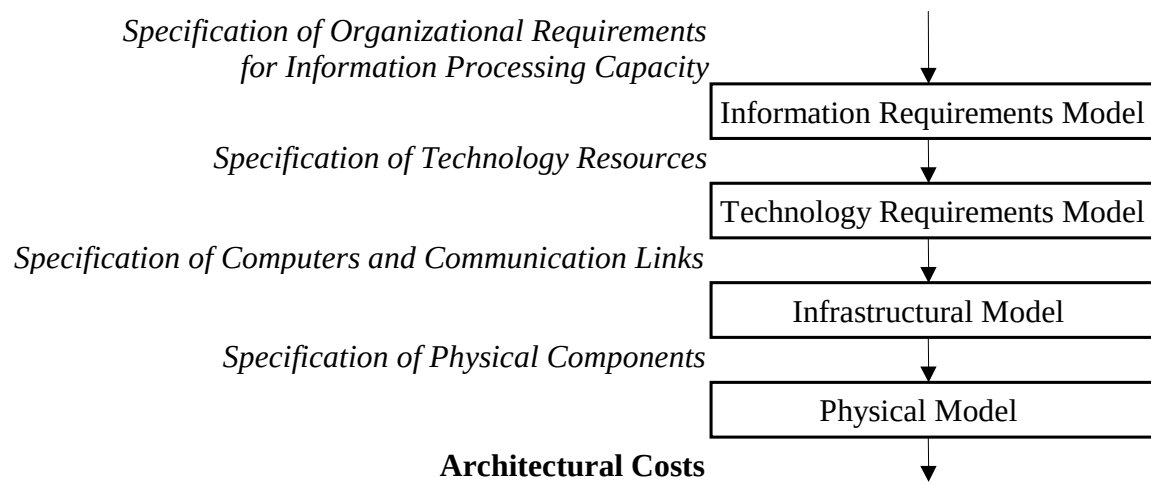


Figure 1. Design models and specification activities

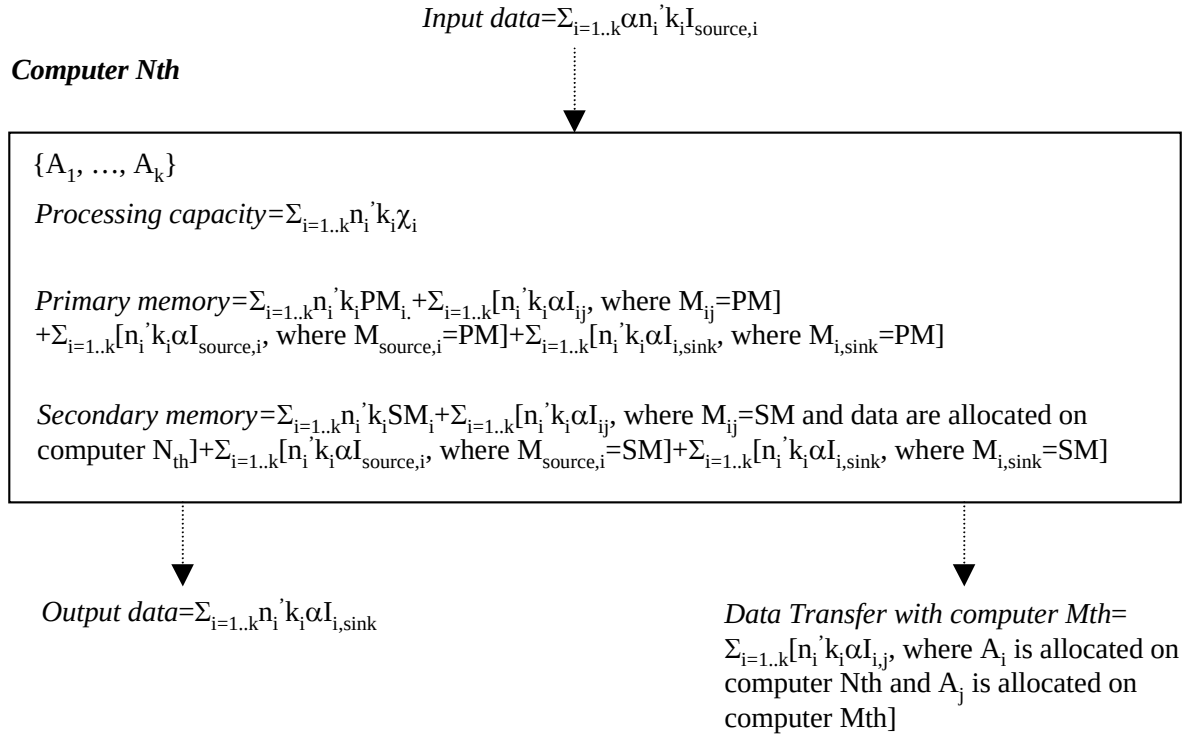


Figure 2. Definition of a computer in the Infrastructural Model (dotted lines represent information exchanges with other computers)

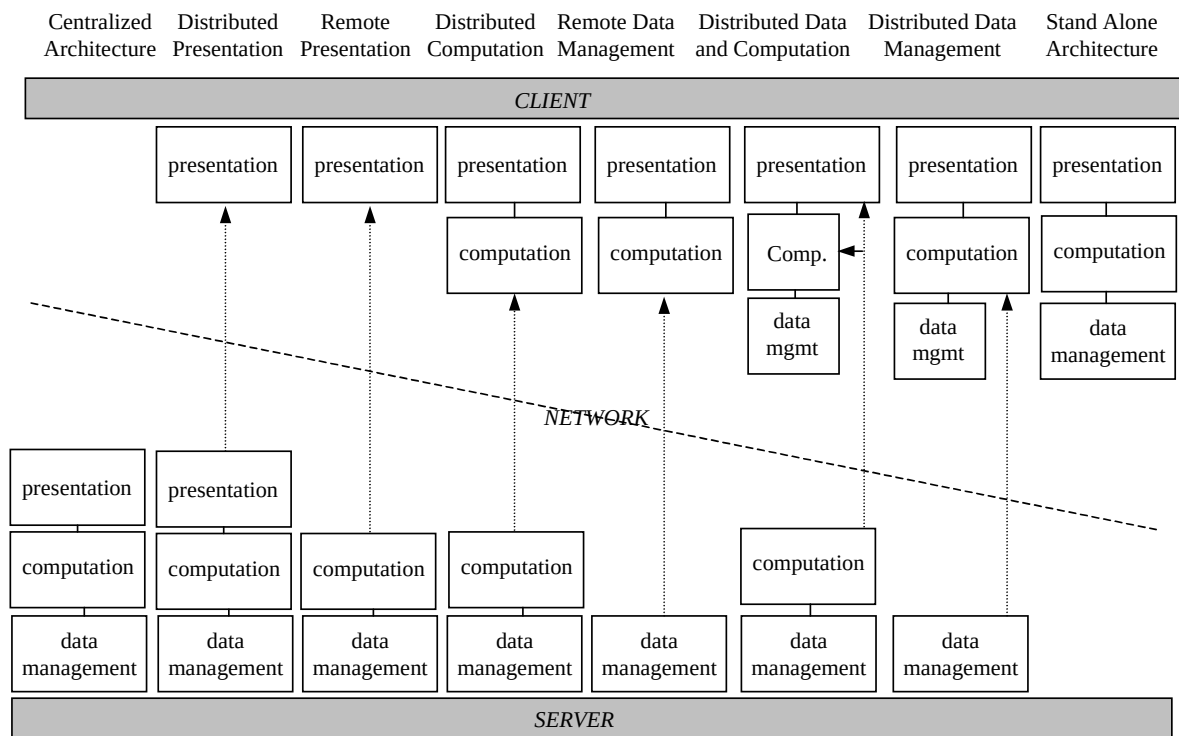


Figure 3. Discretization of architectural choices based on distribution of presentation, computation and data management applications

DISTRIBUTED DATA AND COMPUTATION

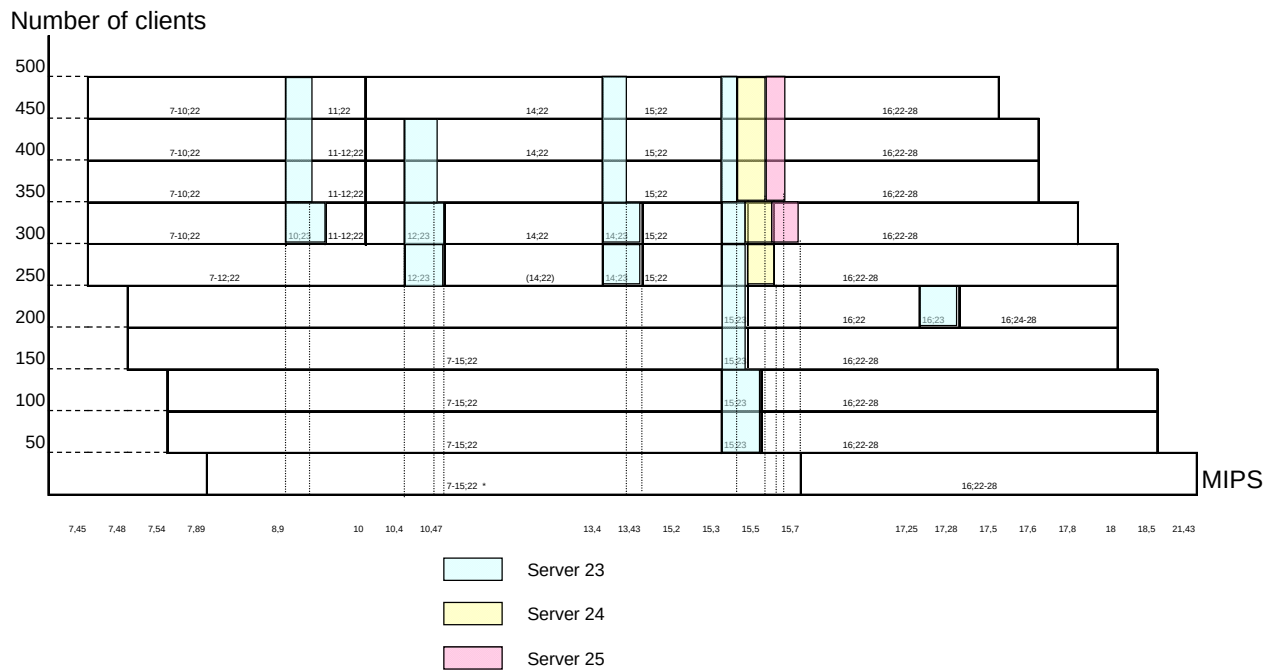


Figure 4. Client-server solutions with minimum cost for the Distributed Data and Computation distributed architecture (pairs of numbers marking different zones in the figure indicate the number of client and server minimizing costs; for example, "7-10;22" indicates a client between 7 and 10 and server 22)

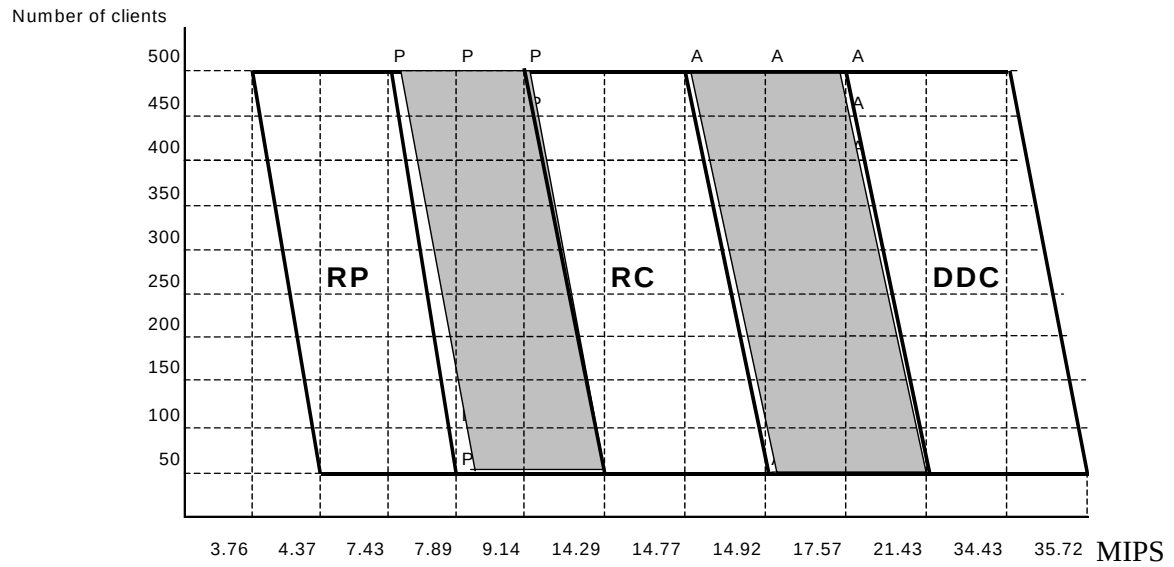


Figure 5. Comparative evaluation of client-server architectures with minimum cost for Remote Presentation (RP), Remote Computation (RC), Distributed Data and Computation (DDC); inside grey areas, the actual choice of the optimum solution depends on specific requirements

**Table 1. Elements and symbols
of the Information Requirements Model**

<i>Model Element</i>	<i>Symbol</i>
Task	T_i
Total number of tasks	n
Information interdependence	I_{ij}
Number of executors of task T_i	n_i
Number of activations of task T_i in a time unit by a single executor	k_i
Task complexity	χ_i
Input information of task T_i	$I_{input,i}$
Output information of task T_i	$I_{output,i}$
Input information from external world	$I_{source,i}$
Output information to external world	$I_{i,sink}$

Table 2. Elements and symbols of the Technology Requirements Model

<i>Element and Symbol of the Information Requirements Model</i>	<i>Specification Activity</i>	<i>Element and Symbol of the Technology Requirements Model</i>
Task, T_i	1. Mapping between tasks and software applications 2. Classification of applications as presentation, computation or data management 3. Specification of primary and secondary memory requirements in bytes	Application, A_i Class of application, $\{P, C, D\}$ Primary Memory, PM_i Secondary Memory, SM_i
Total number of tasks, n		Total number of applications, n
Number of executors of task T_i, n_i		Number of executors of application A_i, n_i
Number of activations of task T_i in a time unit by a single executor, k_i		Number of activations of application A_i in a time unit by a single executor, k_i
Information interdependence, I_{ij}	4. Specification of conversion factor α from organizational units of information to bytes 5. Specification of the type of memory, primary or secondary, used for the data transfer	Data Transfer, αI_{ij} Class of data Transfer $M_{ij} = [PM, SM]$
Task Complexity, χ_i	6. Specification of conversion factor β_i from high-level to machine-level instructions	Application computational complexity, $\beta_i \chi_i$
Input information from external world, $I_{source,i}$	7. Specification of conversion factor α from organizational units of information to bytes 8. Specification of the type of memory, primary or secondary, used for the data transfer	Input data transfer, $\alpha I_{source,i}$ Class of data Transfer $M_{source,i} = [PM, SM]$
Output information to external world, $I_{i,sink}$	9. Specification of conversion factor α from organizational units of information to bytes 10. Specification of the type of memory, primary or secondary, used for data transfer	Output data transfer, $\alpha I_{i,sink}$ Class of data Transfer $M_{i,sink} = [PM, SM]$

Table 3 - Classification of IT costs.

CLASS OF COST	COST ITEMS INCLUDED
<i>Hardware and Operating System</i>	hardware and operating system acquisition hardware and operating system installation hardware and operating system testing
<i>Application Software</i>	requirements analysis acquisition and customization <i>or</i> development and verification integration and installation testing
<i>Communication</i>	network lease (for wide area networks) network installation (for local area networks)
<i>Management</i>	hardware and operating system maintenance and upgrade software maintenance and upgrade network management user training on operating system user training on software applications user technical support (e.g. help desk)

Table 4. Main findings about the relationship between structural properties of processes and costs of different IT architectures

<i>Finding 1</i>	A higher number of servers increases the overall cost of architectures with all computation applications on the servers.
<i>Finding 2</i>	Computational complexity does not represent a cost-differentiating factor across architectures.
<i>Finding 3</i>	When networking and information interdependencies are low, the Remote Data Management architecture has the lowest overall costs, while Distributed Computation has maximum costs.
<i>Finding 4</i>	When networking is low and information interdependencies are high the Centralized architecture has the lowest overall costs, while Remote Data Management has maximum costs.
<i>Finding 5</i>	When networking is high, the Distributed Data and Computation architecture has the lowest overall cost, while Remote Data Management has maximum costs.
<i>Finding 6</i>	Centralized architectures have maximum hardware costs with low networking, while minimum hardware costs as networking increases. Among client-server architectures, hardware costs are maximum for the Distributed Presentation architecture and minimum for Distributed Data Management, with low networking. As networking increases, hardware costs become maximum for the Distributed Data Management architecture and minimum for Distributed Computation.
<i>Finding 7</i>	Software costs are always minimum in Centralized architectures. Among client-server architectures, they are maximum for Distributed Computation and minimum for Distributed Presentation, with low networking. As networking increases, they become maximum for Distributed Data Management and remain minimum for Distributed Presentation.
<i>Finding 8</i>	Management costs are always minimum in Centralized architectures. They are equivalent for the different client-server solutions with low networking. As networking increases, they grow significantly only in Distributed Data Management architectures.