Association for Information Systems
AIS Electronic Library (AISeL)

AMCIS 1998 Proceedings

Americas Conference on Information Systems (AMCIS)

December 1998

An Integrated Approach to Defining Enterprise Computing Architectures

George Nezlek
Loyola University Chicago

Hemant Jain
University of Wisconsin-Milwaukee

Derek Nazareth
University of Wisconsin-Milwaukee

An Integrated Approach to Defining Enterprise Computing Architectures

George S. Nezlek
Loyola University Chicago

Hemant K. Jain
Derek L. Nazareth
University of Wisconsin-Milwaukee

Abstract

This paper describes research into the development of enterprise computing architectures that employ a mix of mainframe, local area network, and cooperative computing paradigms. It outlines a robust approach that permits the incorporation of several different distribution criteria, while accommodating designer preferences.

Introduction

The need to align business strategies and goals with the technological environment is a recurrent theme in I/S research [5] [7], and has featured prominently in recent surveys of critical issues for IS management [1] [2]. This trend reflects a flattening the organization, a common theme in the contemporary literature of many managerial disciplines.

The rapid growth of cooperative computing in the 1990s has transformed both the IS function and its management in many organizations. This is frequently characterized by a downsizing of systems away from mainframe environments to smaller hardware platforms, coupled with network-based approaches to information management. In other cases, growth in the size and sophistication of user developed systems has led to 'upscaling' of departmental or LAN-based computing, as LANs become repositories for mission-critical corporate data [3].

We use the term ***enterprise computing architecture*** to describe the set of computing platforms and the data networking facilities to support an organization's information needs. The benefits of coherent and coordinated architectures include reduction of undesirable redundancy of system components, allocation of information processing functions to appropriate computing platforms, allocation of computing resources to appropriate locations within an organization, and the ability to share information resources across organizational entities. Appropriate architectures allow organizations to meet current as well as projected information needs, and to successfully adopt new information processing paradigms in a cost effective manner.

Prior research in this area has generally addressed different facets of this problem separately. Several techniques are available, summarized in [6], to address problems of file allocation, network design, processor selection and allocation, distributed database design, etc. Most of these techniques consider one aspect of architecture design at a time, and assume that the information on other aspects is available. However, these decisions are really interdependent. Moreover, these techniques are designed to work in environments where an application system can be neatly partitioned and allocated to single processors in the enterprise computing architecture. The traditional approach to architecture design has involved addressing these in a predefined sequence, as illustrated in Figure 1.

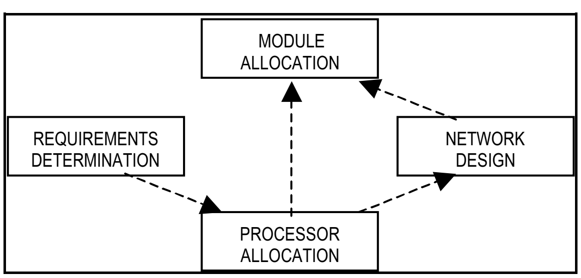

Figure 1. Traditional Architecture Design

Unfortunately, these assumptions are not always valid. As new technology paradigms evolve, the notion of strict partitioning of modules and their allocation to individual processors may not apply. This is particularly true in the case of cooperative computing, as illustrated in the client/server paradigm. This requires a different approach to enterprise computing architecture design. In this environment, the enterprise computing architecture should reflect and be driven by the organization's information requirements. These requirements are specified in terms of functions performed by individuals at various locations. Functions are then aggregated across user groups and locations, as well as propagated to individual users as needed to derive the global enterprise requirements. This in turn will determine the software components needed at each location. This information can then be used to drive the hardware specification at each location, which in turn will be an input to the network design. This integrated approach represents a much more complex problem, and is depicted in Figure 2.

The integrated architecture specification problem is NP-complete, and susceptible to computational intractability as the number of users and locations increases. The growing number of technology options for clients and servers, as well as communication alternatives, only serves to complicate the problem further. It would be nearly impossible to include all conceivable permutations of computing hardware in any consideration of architectures. Consequently, complete enumeration techniques (e.g. optimization), are inappropriate. Instead we adopt an approach based on heuristic classification [4].

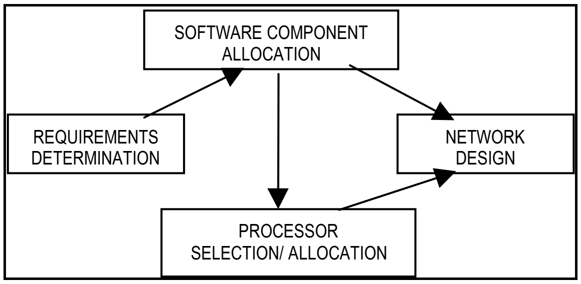

Figure 2. Integrated Architecture Design

The technique consists of three stages. In the first stage, the problem and potential solutions are abstracted, based on predetermined dimensions, thereby hiding unnecessary detail. For example, all application systems for individual users are specified in terms of generic software modules, data, and processor requirements. Similar abstractions are performed for other aspects of the problem. In the second stage, the abstracted problem is solved and expressed in terms of an abstracted solution. This entails the specification of generic processing capacities and data storage requirements at the desktop, local server, and global server levels. In the final stage, the abstracted solution is restored to the necessary or desired level of detail. This includes mapping the processing requirements to a specific platform, including specification of processor type, memory and secondary storage requirements. In addition, the network bandwidth requirements can be used to select appropriate type and capacity of appropriate communication links.

Methodology

Our approach is shown in Figure 3. This technique allows the designer of an enterprise computing architecture to generate a consistent, technology independent model of the information management requirements that the enterprise computing architecture must ultimately satisfy.

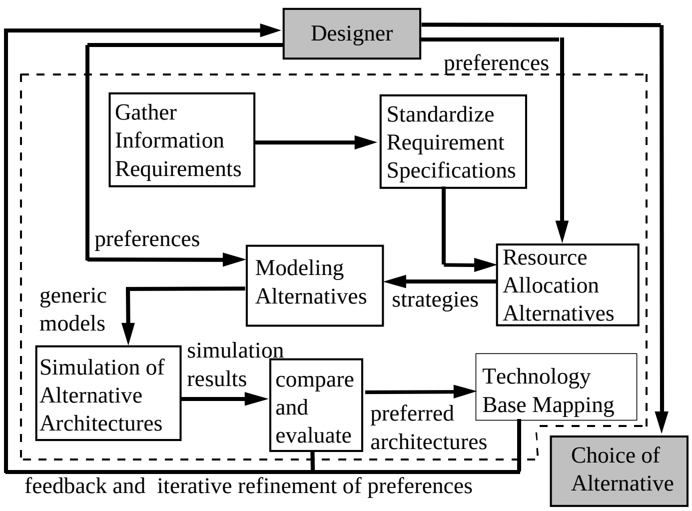

Figure 3. Integrated Design Methodology

These requirements are then considered in the context of various alternatives for resource allocation. The approach is general enough to accommodate traditional applications (including legacy and 3G/L systems) as well as more contemporary applications (including 4G/L or OO languages, JavaScript, pre-packaged software, etc.); running against a wide variety of data stores, (including flat files, RDBMS, and OODBMS).

The approach supports allocation based on one of two strategies: minimization and localization. With the minimization approach, instances of programs or data components are allocated to the fewest locations practical. In the localization strategy, software and data components are allocated as close to their potential users as practical. The modeling technique offers several dozen pre-specified allocation options, as well as allowing designer-supplied values, forced replication, sharing restrictions, among others.

Several different client/server options are considered (e.g. centralized, thin client, fat client, etc.) to create a set of generic cooperative computing alternatives. These form the basis for a simulation-based capacity planning exercise. Promising alternatives from the simulation are retained and mapped to generic architectures. One or more candidate architectures are then selected for further analysis, and mapped to specific sets of technology components. These are evaluated on the several criteria, including acquisition and operating costs, total capacity, etc., for final selection or refinement by the designer.

Ultimately, the designer of an enterprise computing architecture must decide which, if any, of the candidates are appropriate. It is probable that a recommended solution, while not exactly what the designer had in mind, represents a good starting point in the overall solution space. The solution space around this point can be explored by modification of requirements data, alternative resource allocations, extended simulation study, or substitution of alternate technology base components, to yield a preferred solution.

Implementation and Test Cases

The approach is implemented in a prototype decision support system called ***The Information Architect.*** The prototype was developed in Microsoft Visual Basic, and is described in detail in [6].

The prototype was used in two real-world case studies. The first case involved a medium-sized manufacturing and distribution company in the mid-western United States. The second dealt with a regional processing facility for a large, non-profit organization, also in the mid-western United States.

The first case involved 157 users at nine locations, dispersed across the Unites States and Canada. The typical processing activities were low volume data update transactions, with the majority of work performed at the central headquarters. The functions involved 268 modules in 16 application areas and included 242 data stores. Data collection for this case spanned a month. Application of the prototype yielded several viable alternative architectures. An interesting facet of the selected architecture was that it yielded more logical placements of data than currently in use.

The second case involved 189 users at five locations, in a single metropolitan area. Typical processing activities involved high volume query processing, with moderate effects of data partitioning, with the majority of work performed at the central headquarters. The functions involved 380 modules in 17 application areas and included 70 data stores. Data collection for this case spanned six weeks. In this case, the prototype was able to recommend several alternative architectures with varying server requirements, based on the allocation of individual tasks to user or group levels.

Several criteria used were employed in assessing the quality of alternative architectures. They included degree of replication of application objects, proximity of data and module objects, acquisition costs for server and user nodes, inter-site data transmission capabilities and costs, and a three-year estimate of operating costs for the aggregate architecture. In both cases, the dominant cost factors were user node acquisition and operations. Some differences were noted in server requirements, and the nature of the differences observed suggests that, for larger organizations, there might be more noticeable differences.

Finally, there were instances where various application partitioning strategies would allow the organizations to take advantage of excess capacity at the user level, consistent with the frequently claimed advantages of client server computing.

Conclusions

An appropriate enterprise computing architecture is a necessary prerequisite for successful deployment of information systems. Defining such an architecture is one of the key issues facing I/S management. The methodology presented in this paper describes an approach that translates logical information requirements into appropriate information architectures. Given the inherent complexity and size of this problem, our research shows it is possible to give system designers a means to manage the process in a controlled and coordinated fashion.

References

1. Benjamin, R.I. and Blunt, J. "Critical IT Issues: The Next Ten Years", *Sloan Management Review,* 33 (4), pp. 7-19, Summer 1992.
2. Brancheau, J., Janz, B., and Wetherbe, J. "Key Issues in Information Systems Management: 1994-95 SIM Delphi Results", *MIS Quarterly*, 20 (2), pp. 225-242, June 1996.
3. Christiansen, C. "*Upsizing Server Environments with Digital's Alpha AXP and Windows NT*" IDC#8770, International Data Corporation, Framingham, MA, 1994.
4. Clancey, W.J. "Heuristic Classification", *Artificial Intelligence*, 27 (3) pp. 289-350, December 1985.
5. Keen, P.G.W. "Information technology and the management difference: A fusion map", *IBM Systems Journal,* 32 (1), pp. 17-39, 1993.
6. Nezlek, G. S. "Architectures for Cooperative Computing: A Knowledge-Based Approach", *Ph.D. Dissertation,* University of Wisconsin - Milwaukee, June 1997.
7. Zmud, R.W., Boynton, A.C. and Jacobs, G.C. "The Influence of IT Management Practice on IT Use in Large Organizations", *MIS Quarterly,* (18) 3, pp. 299-318, September 1994.