8-16-1996

An Instrument to Assess Client-Server Architecture in an Organizational Setting

Mayur S. Desai
University of North Texas, msd2@msg.ti.com

Richard A. Huff
University of North Texas, huff@cobaf.unt.edu

An Instrument to Assess Client-Server Architecture in an Organizational Setting

Mayur S. Desai (msd2@msg.ti.com) and Richard A. Huff (huff@cobaf.unt.edu)
University of North Texas, Denton, Texas 76203

Abstract

This study reports the final results of a delphi panel procedure used to develop an instrument to assess the client-server system architecture in an organization. At present there is no formal instrument that allows an information systems manager to assess an existing CSS architecture. A delphi panel was used to develop a formal instrument for CSS assessment.

Introduction

The Client-Server System (CSS) environment has become quite popular in a number of organizations because it seems to provide an alternative, cost-effective solution to mainframe applications. At the same time, the integration of CSS into an existing organizational setting has raised concerns for managers at all levels (Whitman and Carr 1994). Desai (1995) provided a theoretical research framework to address CSS issues and guidelines for researchers to investigate the factors underlying CSS issues. Since CSS is one of the categories of distributed computing, it inherits the problems of end-user computing, specifically managing end users (Alavi et al. 1987) and hardware/software resources. As the number of mainframe applications transformed into CSS applications increases, issues of controlling computing resources (Stedman 1994), data security (Johnson 1994), and training IS personnel, as well as end users, become critical. Thus it becomes important for managers to become proactive and foresee problems before they become unmanageable. The CSS is not a new concept, but is perceived as a new concept because there are now commercially available computers capable of performing client-server processing (Sinha 1992, Winter and Dove 1991, Huff 1990).

At present there is no formal instrument that allows an information systems (IS) manager to assess an existing CSS architecture (CSSA). A delphi panel was used to develop a formal instrument for CSS assessment. The CSS instrument will allow IS managers to measure their current CSS environment from which they can forecast the future growth of CSS. Thus the instrument presented in this paper serves as a decision-making tool for IS managers.

Research Purpose and Objective

The purpose of this paper is to report the final results of a delphi study used to develop the CSS assessment instrument (Desai and Huff 1996). Specifically, the study reports how the variables identified in the CSSA framework (Desai 1995) are operationalized to assess the CSS.

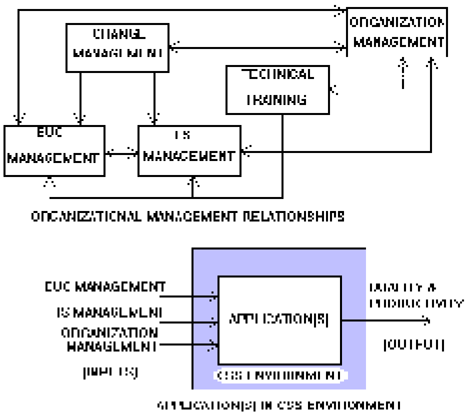
Problem Definition and Significance

Integrating the CSS into an existing information architecture is a challenge for large organizations (Borthick and Roth 1994). In addition, the distributed computing environment of a CSS makes it difficult to manage and control end-user activities. Backup and recovery of data, keeping applications running, training end-users, and maintaining their satisfaction level are some of the common problems in CSS environment. The problems become severe when the demand for new applications and additional data increases. Thus the information system management is faced with making provisions for unforeseen demand and future growth in their CSS strategy (Cafasso 1994, Tsay 1994).

The availability of a variety of computing resources, such as high-speed processors, peripherals, and graphical user interfaces, makes the assessment of an optimum CSS solution a real challenge to the IS personnel. The project development life cycle also poses a threat due to the accelerated growth of technology. Long development life cycles may result in an obsolete application due to availability of new and user-friendly technology. This implies that the skills and efficiency of the software personnel need to be "upgraded" on an ongoing basis to keep pace with the advancing technology (Desai 1996).

The main issue addressed by the development of a CSS assessment instrument is: "how does one assess the existing CSS environment?" There is a need to develop an instrument that will effectively measure the current state of the CSS environment.

This study is significant because it will allow organizational management to assess the CSS technology. A proper assessment will allow management to effectively allocate the computing resources. Organizations planning to move from a mainframe to CSS environment will also benefit by the CSS assessment instrument.



ORGANIZATIONAL MANAGEMENT RELATIONSHIPS

APPLICATION(S) IN CSS ENVIRONMENT

Figure 1

Research Methodology

This study used the Desai (1995) CSS framework to develop the assessment instrument. The key elements of the CSS framework are shown in Figure 1 and Table 1. Figure 1 represents the overall interactions between IS management, EUC management, organizational management, technical training, and change management in a CSS environment. Table 1 presents the independent and dependent variables, surrogates, and measures used to gauge the major constructs in the CSS environment. The independent variables are computing resources, IS personnel, end users and organizational management. Dependent variables are the productivity and the end-user satisfaction with the applications running in CSS environment.

The constructs and variables used for assessment of the CSS environment are derived from the CSS framework proposed by Desai (1995). A delphi approach was used to develop the CSS assessment instrument. The following section briefly discusses the constructs.

Constructs and Variables

Constructs are abstract concepts and variables are partial representations of constructs (Kidder and Judd 1986). Causal constructs delineate independent variables (IV) and affect constructs represent dependent variables. Since variables cannot be directly measured, surrogates are used as representative measures of them.

The major constructs, variables, and surrogates shown in Table 1 were used to develop the instrument. The CSS framework identifies two major constructs --

Table 1: Variables, Surrogates, and Measures (Source - Desai 1995)

Variable	Surrogate	Measures Used
Client (IV)	Hardware Software	Work Station type, vendor, cost, storage, speed configuration, software package, software type
Network (IV)	Hardware Software	OS environment type, GUI interface type DBMS, Application sw, Novell etc.
Server (IV)	Hardware Software	transmission speed and other hardware related parameters Communication protocols
IS Professional (IV)	Skills Experience Training	sw, hw, network, applications software # of years, type type, length of time
EUC/ end user (IV)	Skills Experience Training Level of Authority	sw, hw, application # of years, type type, length of time develop applications, expertise, influence, power
Organizational Management (IV)	Budget Skills/Style CSS Control CSS Planning	$ allocation, $ variance/overruns human relations, sw/hw education department and IS budgets schedule, long/short term

Productivity (DV)	Productivity	system down time, hw failure errors, overtime
End-user	Ease of Use	user friendliness, ease of use
Satisfaction	Timeliness	on time & correct information
(DV)	Accuracy	satisfaction with accuracy of information

CSSA (causal construct) and performance (affect construct). The CSSA is measured by the set of independent variables in the framework. The performance construct is measured by the set of dependent variables.

Instrument Description

The instrument is divided into two parts. The first part addresses the measures for the independent variables. The second part assesses the measures for the dependent variables. For each variable, the researcher prepared a series of questions to address the surrogate measures for that variable. The questions are based on the measures suggested in the CSSA framework as shown in Table 1.

The independent variables are operationalized by selecting appropriate surrogates described in the table. For example, an independent variable client may be measured by the suggested surrogates - the hardware and the software. They may be measured by the type of platform, the cost, the speed of the system, software configuration, type of graphical user interface and so on. The independent variables server and network may be measured in a similar manner. To measure the independent variables IS professional and EUC/end user, the appropriate surrogates such as skills of the individuals, experience level -- educational and professional -- of individuals may be used.

Performance, the dependent variable, is divided into two components -- quantitative and qualitative. The quantitative component represents productivity, while the qualitative refers to end-user satisfaction. The productivity component was addressed by preparing a series of questions similar to the first part. The productivity depends upon the specific client-server setup within an organizational setting. Productivity also refers to how efficiently hardware and software components interact. For example, an organization may use a one-tier architecture, i.e. the three key components of an application - the user interface, business rules, and database access -- all reside on the same platform or computer system. In such a situation, productivity could be measured by the "internal efficiency" of the single computer system. However, in another example, when an application is used in a distributed environment, a two- or three-tier client-server architecture is appropriate. In a two-tier architecture, the user interface and the business rules may reside on the client platform, and the database access may reside on the server platform. In this scenario, the productivity measure includes the internal efficiency of the client and server platforms, as well as the communication software interface between the platforms. The distributed environment further necessitates assigning the levels of "access rights" to the database. In summary, in a distributed environment the efficiency of the server, network, and client stations determine the productivity measure.

The questions measuring end-user satisfaction were based on the instrument developed by Doll and Torkzadeh (1988). The end-user satisfaction is qualitative because there is a certain level of subjectivity involved in measuring the satisfaction of end users. Again, the type of questions to include in the measure of end-user satisfaction depends on the complexity of the client-server setup within an organization. For example, in a distributed environment, the end-user satisfaction largely depends upon the network speed, the availability of the database access, and the level of support provided to end users.

A delphi approach was utilized to face validate the questions used in the CSS assessment instrument. The delphi panel for this study had several years of experience with the CSS applications. This study took two delphi rounds to arrive at a reasonable consensus among panel members.

Discussion and Future Directions

No current means exists to formally assess the state of a CSS environment, which leaves a void that must be addressed. The instrument developed in the course of this study may be used to fill that void. The research suggests that the instrument will allow the IS personnel to measure the state of an existing CSS environment. A possible research methodology would be to study the CSS environment at a single or multiple sites in the form of a case study. A case study is deemed appropriate as a first step because the purpose of the study is to identify and validate the key variables identified in the CSSA (Yin 1989).

The CSS is an evolving concept. A future research agenda would be to provide this instrument to IS personnel within an organization and have them answer the questions as they apply to their CSS environment. IS management will then have an opportunity to obtain a "snapshot" of their CSS environment. This "snapshot" will allow the IS management to foresee the future growth and the evolution of CSS applications and assist them in their decision making. The instrument also allows organizations operating in non-CSS environments, such as a legacy system, to evaluate and transit to a CSS environment.

(References and instrument available upon request)