Dimensions of I/S Planning and Design Technology

John C. Henderson
Jay G. Cooprider
Dimensions of I/S Planning and Design Technology

John C. Henderson
Jay G. Cooprider

September 1988

Sloan WP # 2077-88
CISR WP # 181

©1988 J. C. Henderson, J. G. Cooprider

Management in the 1990s
Sloan School of Management
Massachusetts Institute of Technology
Management in the 1990s

Management in the 1990s is an industry and governmental agency supported research program. Its aim is to develop a better understanding of the managerial issues of the 1990s and how to deal most effectively with them, particularly as these issues revolve around anticipated advances in Information Technology.

Assisting the work of the Sloan School scholars with financial support and as working partners in research are:

American Express Company
Arthur Young and Company
British Petroleum Company, p.l.c.
BellSouth Corporation
CIGNA Corporation
Digital Equipment Corporation
Eastman Kodak Company
General Motors Corporation
International Computers Ltd.
MCI Communications Corporation
United States Army
United States Internal Revenue Service

The conclusions or opinions expressed in this paper are those of the authors and do not necessarily reflect the opinion of the Massachusetts Institute of Technology, the Management in the 1990s Research Program, or its sponsoring organizations.

Acknowledgements

The authors would like to acknowledge Mark Hunsburger for his contributions to this work.
ABSTRACT

Information technology is increasingly an integral part of the competitive strategies for many organizations. As this trend continues, it is not surprising that there is an increasing emphasis placed on the ability of organizations to plan, design and implement critical information systems. A major strategy to improve the effectiveness of these processes is to utilize computer-based planning and design aids. And yet, there is little empirical evidence that demonstrates a significant performance impact of this technology. One factor limiting research on the impact of technology on planning and design is the manner in which this technology has been conceptualized in order to provide measures of usage behavior. This research develops a functional model of I/S planning and design support technology that distinguishes among three general functional dimensions: Production Technology, Coordination Technology and Infrastructure Technology. An empirical analysis is used to test the robustness of the proposed model and its ability to discriminate between current design aids in a meaningful way. Implications for the use of this model in the study of I/S planning and design processes are discussed.
1.0 Introduction

In today's business environment, a critical management issue is "time-to-market", that is, the length of time it takes an organization to convert a product concept into a viable product that is available in a specific market. The Xerox Corporation, for example, argues that their improved ability to manage time-to-market while retaining or improving quality has been a major factor in their efforts to rebuild their competitiveness. Extending this notion, Hewlett Packard focuses on the "time-to-break even" as a measure of success for product development. This perspective incorporates directly the aspects of quality and maintainability while highlighting the criticality of rapid response.

It is not surprising that the I/S function within a business faces this same challenge. As information technology becomes an integral part of an organization's competitive strategy, the I/S function faces increased demands to improve its ability to manage the "time-to-market" for I/S products and services. In fact, some (Martin 1988) have suggested that the inability of an I/S function to both reduce the backlog of demand for systems products as well as meet an increasing new demand for I/S products represents a serious management failure.

While many factors affect an organization's ability to deliver high quality products in a short time frame (Ancona and Caldwell 1987), one key tool to address this issue involves the usage of computer-aided planning and design tools. We see,
for example, Xerox, Ford and many other organizations focusing on the role of CAD/CAM technologies as one means to radically change their capacity to quickly develop and deliver products to specific markets. Similarly, we have seen the growth of a major industry that seeks to deliver comparable design aid technology to the I/S function. Often referred to as CASE technology (Computer Assisted Software Engineering), this technology is targeted at those who wish to use automation to affect the timing, costs and quality of products and services delivered by the I/S function. Beck and Perkins (1983), for example, found that 56 out of 97 organizations they surveyed used automated tools as a means to improve their I/S planning and design processes.

The impact of these tools, however, on the productivity of software developers and, ultimately, on time-to-market is unclear. Semprevivo (1980) and Necco et al. (1987), for example, have reported that design aid technology improves the productivity of designers. In contrast, Card, et al. (1987) and Lempp and Lauber (1988) found, after controlling for factors such as experience and task complexity, that the use of software development aids did not have a significant effect on productivity and a relatively weak effect on quality.

The explanations for such conflicting results could be attributed to many factors. For example, some of the studies that address productivity impact from narrowly defined tasks such as the encoding of specifications or the development of flow representations (Case 1985). In contrast, other studies focus on the entire system design life cycle (Card et al. 1987). Perhaps more fundamental is the lack of clarity as to the definition of what constitutes usage of the CASE tools. It is often unclear whether usage refers to access, e.g., such technology was available to the team, or, in fact, measures actual usage behavior. Further, it is not clear that the level of aggregation defined by usage variables in most studies provides sufficient precision
to actually predict performance impact. For example, if a macro usage variable is employed, ("did I use this package"), teams may indicate a similar usage level of design aids but utilize quite different subsets of functionality. As a result, the impact of this technology could be easily mixed, leading to an overall assessment across design teams that indicates little or no impact.

The need to better define and measure technology usage behavior suggests a need to develop a model of CASE technology that has a correspondence more closely to key designer behaviors. That is, rather than define CASE technology in economic terms (e.g., costs), technology terms (e.g., PC-based or networked), or in terms of general characteristics (e.g., having an embedded design language or structured code compiler), we must develop a model of CASE technology that is functionality oriented. Such a model would then provide one means to directly relate usage of a CASE tool to design team performance.

The literature on I/S planning and design does offer a starting point. Hackathorn and Karimi (1988) and Welke and Konsynski (1980), for example, differentiate between design methodologies and design tools. The former define the logical disciplines underlying I/S planning and design activities. The latter instantiate the principles in a software application. Hackathorn and Karimi (1988), Beck and Perkins (1983) and others support the notion that software engineering and information engineering involves the application of sound engineering principles to the task of I/S planning and design. Understanding these principles offers one means to map the functions of CASE technology onto key usage behaviors.

The difficulty lies in the diverse set of concepts, principles and subsequent methodologies that could be used to generate a design aid environment. Chikofsky and Rubenstein (1988), for example, claim there is, as yet, no clearly accepted
definition of CASE technology that satisfies this diverse range of design concepts and methodologies. In a similar vein, Osterweil (1981) recognizes this inherent diversity and argues that a research program in software engineering must address the full range of design related activities. He states

"The task of creating effective environments is so difficult because it is tantamount to understanding the fundamental nature of the software processes. A specific environment does not merit the name unless it provides strong, uniform support for the entire process it is intended to facilitate; that is not possible unless the process is fully appreciated and understood." (Osterweil, p. 36)

In the following sections, the development of a functional model of CASE technology that can be used to address a wide range of planning and design activities is described. The results of in-depth interviews with leading academic and industry designers of CASE products concerning the range of possible CASE functionality serves as a starting point for developing this functional model. Past research on CASE functionality is then used to organize these functionalities into six general dimensions of CASE technology. The ability for these dimensions to serve as a model for CASE technology is evaluated empirically through both a Q-sort study with I/S planners and designers (familiar with CASE technology) and use of the dimensions to characterize the strengths and weaknesses of commercially available CASE products. Implications for the use of this functional model for research on the impact of CASE technology is discussed.

2.0 A Functional CASE Technology Model (FCTM)

There are several reviews of the range of functionality found across various CASE environments. Hackathorn and Karimi (1988), for example, categorize CASE technology in terms of the range of the design life cycle addressed and the extent to
which the environment provides for a range of support from conceptual to explicit
design techniques. The functionality of the CASE technology is then implied by the
method(s) incorporated in the environment and the aspect of the planning and
development for which the support environment is targeted. Thus, a tool that
embraces the Gane-Sarson (Gane and Sarson, 1979) method could be expected to
provide features such as functional decomposition or data flow diagram. Of course,
the tool might provide much more in context of communications or analysis. Such
distinctions, however, are not clear.

Reifer and Montgomery (1980) provide a more general schema. They begin with
a general model of design as having three components: input, process, and output.
Each component is decomposed until a set of 52 functions are identified. They argue
that this taxonomy permits classification of all current software development tools
(given the time of their study) and allows easy comparison and evaluation of tools.
While one could argue the validity of such an ambitious claim, their taxonomy does
provide a direct linkage to design behavior. For example, they identify features such
as tuning, structure checking, scheduling, auditing and editing. Clearly, such a
model can be linked to the actual behaviors of designers. Similarly approaches are

These models, however, do appear limited. For example, the functionality
associated with teams is not clearly identified. Features such as those found in
COLAB (Stefik et al., 1987) or PLEXSYS (Konsynski et al., 1984) that support
groups through structured processes for brainstorming, communication, voting,
negotiations or the key team behaviors appear to be lacking. To the extent that
"time-to-break even" will involve the use of teams as suggested by Ancona and
Caldwell (1987), Cooprider and Henderson (1988) and others, there is a need to
incorporate these functions into CASE technology.
In this research, we pursue an objective consistent with prior research that attempts to characterize the key dimension of design support technology. That is, we will develop a function model of design support (CASE) technology. To achieve this objective we used a four step process. First, leading designers of CASE related technology were interviewed to generate a set of critical functions that could be of value to an I/S planner or designer. The specific functional definition used was required to correspond to an observable design behavior. Second, this set of functions was reviewed by 25 practicing designers familiar with CASE technology to refine ambiguous items and reduce any obvious redundancies. Third, a classification scheme was developed based on a review of the design literature and used as a basis to sort each specific functionality generated during the interview process. The Q-sort was done by an independent group of 34 I/S designers experienced in CASE technology. The intent of this step in the process was to evaluate the robustness of the model. Finally, the model was used to evaluate currently available CASE products. This step represents one test of the model's ability to adequately represent and discriminate between actual CASE environments.

In the first step, open ended interviews with leading CASE designers (both academics and practitioners) were used to develop a list of possible CASE functionalities. A total of eleven interviews, each lasting from two to three hours, were conducted. Each interview subject had extensive personal involvement in CASE technology research or had actual development experience with a range of commercial CASE products. Subjects included three academics and eight practitioners.

The interviews consisted of providing the subject with a list of functionalities extracted from the literature. To ensure adequate discussion, the lists were divided
into five sections. The subjects reviewed each functional description, noting ambiguity or bias in definition. At the end of each section, problems with definitions were discussed and new functionalities added. The order of presentation of each section was randomized across subjects.

A total of 124 distinct functionalities were generated via the interview process. The second step involved a clarification procedure to combine or eliminate vague and/or redundant functional definitions. In this effort, three to five expert users for each of eight existing CASE products were asked to evaluate their product using the 124 functionalities. Each subject indicated the ease of use of a given function on a one to five Likert scale where one equals very difficult to use or nonexistent and five equals very easy to use or essentially automatic. The reliability of each functional definition can be assessed by analyzing the variance (or correlation) across subjects for a given product. If the definition is unambiguous, subject experts should assign the same ease of use rating to a given functions. Functional definitions receiving high variance or inter-rate reliability below .8 were reviewed and eliminated or refined. As a result of this process, 98 distinct functionalities were defined.

The third step in the process involved developing a model that reflected the scope of these 98 functions. This model, called the Functional CASE Technology Model (FCTM), was developed in a two stage process. First, a review of relevant design literature was used to define a priori a general model. Then, a new set of 34 expert CASE users were given the task of sorting each function into one of the a priori dimensions defined by this model. The extent to which this Q-sort process reflected a consistent sorting pattern across subjects is then taken as evidence that the a priori model is a meaningful abstraction and can be used to represent a wide range of CASE functionality. That is, it is more than a unique artifact of the researchers interpretation of existing literature.
An alternative approach for developing such a model is discussed by Sherif and Sherif. In this approach the subject is asked to manually cluster attributes thereby developing a subject specific model. The models generated by a set of subjects can then be analyzed for underlying similarities and, hence, form the basis for generating an overall model. The strength of this approach lies in the ability to eliminate the bias created by an a priori model. However, such an approach requires extensive time and may result in dimensions that have little theoretical grounding. In this case, the time demand for the clustering task with approximately 100 items exceeded the time subjects were willing to provide. Further, years of both theoretical and empirical research on I/S planning and design provide a basis for developing an a priori model. Given these two factors, a Q-sort testing strategy was utilized.

As we will discuss, the final step then tests this model by using it to discriminate between actual CASE products. In the following section, each dimension of the FCTM is described and the results of the Q-sort process are provided. The section concludes with a summary of the adequacy of this model. Section 3 then uses the model to evaluate actual products and discusses the implications of these results. Finally, Section 4 summarizes the findings of this research and discusses the implications for future research.

2.1 Three Dimensions of CASE Technology

Reviews of the organizational literature on technology (Fry (1982), Fry and Slocum (1984) Slocum and Sims (1980), Withey et al. (1983)) reveal a diversity of approaches to the measurement of technology. Perrow (1967) defines technology as the actions used to transform inputs into outputs. In that context, technology is a
production variable, describing the way inputs are converted to desired outputs. Economists have long characterized technology as production technology concerned with creating, processing, and handling physical goods. Thus, as illustrated in Figure 1, one perspective of CASE technology is to view it as an underlying production technology.

A second concept that has been used to evaluate technology is coordination. Thompson (1967) argues that coordination is needed when interdependence occurs among business processes. Interdependence requires that performance of one or more discrete operations has consequences for the completion of others. The concept of interdependence is a fundamental principle in designing organizations (McCann and Galbraith (1981), Galbraith (1977), Thompson(1967)). Different types of interdependence create different coordination structures between participants involved. Malone (1988) defines "coordination technology" as any use of technology to
help people coordinate their activities. Since a design team consists of multiple agents with a variety of goals and skills, coordination technology may emerge as an important dimension of CASE technology.

A last dimension of technology is infrastructure. Infrastructure technology is viewed as an organizational support technology. Even though there are few who use this term, this is an important dimension of design aid technology. A given design team may interact with other teams in order to obtain resources, coordinate work, make decisions, and exchange inputs and outputs. In this regard, infrastructure technology is concerned with the interaction with persons or units which are outside of a given design team, i.e., key stakeholders. Thus, a major difference between coordination technology and infrastructure technology is the focus of the infrastructure technology on providing an organization-wide design support environment.

Taken together, technology can be conceptualized as production technology, coordination technology and infrastructure technology. In the following we will build from these three perspectives of technology to characterize the dimensions of CASE technology. In each section we will examine relevant research on I/S planning and design aids and define the components of production, coordination and infrastructure technology from this viewpoint.

In this section each major dimension is defined in terms of distinct sub-dimensions (Figure 1). Results of the Q-sort process are provided in Section 2.2. A summary of the FCTM is provided in Section 2.3.
2.1.1 Production Technology: Representation

As discussed above, one perspective on technology is action used to transform input to outputs (Kottemann and Konsynski 1984). At an individual level, Simon (1976, 1981) argues that bounded rationality ultimately limits the capacity of human information processing and, hence, this transformation process. This information processing perspective is often used to characterize the planning and design task (Thomas and Carroll 1979) and provides a basis to characterize the production dimension of design aid technology. The first component of production technology is label representation to emphasize the notion of abstracting or conceptualizing a phenomenon. Schon (1984), Zachman (1986) and others have identified the process of evolving abstractions and presenting them in a communicable form as an essential activity in planning and design. Zachman (1986), for example, lists categories of functionality such as process flow diagrams, functional charting or entity modeling that reflect alternative means to represent concepts or phenomena. Kottemann and Konsynski (1984) identified a hierarchy of knowledge representation that included names or labels, domain set specifications, association or relations mapping and complete meaning that suggest the need for a range of representation functionalities. From our perspective, each of these categories suggests the need for specific functionality to support the process of externalizing and communicates a design concept.

Specifically, the representation dimension is defined as *functionality to enable the user to define, describe or change a definition or description of an object, relationship or process*. The interviews resulted in a range of functionalities that appear to operationalize this conceptual dimension. As shown in Table 1A, these functionalities reflect a general notion of knowledge representation and acquisition.
Table 1A: Functionalities of Representation (Production Technology)

<table>
<thead>
<tr>
<th></th>
<th>% 1st Choice</th>
<th>% 2nd Choice</th>
<th>% 3rd Choice</th>
<th>Prod Coord Infr</th>
</tr>
</thead>
<tbody>
<tr>
<td>96. Represent a design in terms of process or flow models</td>
<td>88%</td>
<td>6%</td>
<td>3%</td>
<td>94%</td>
</tr>
<tr>
<td>5. Represent a design in terms of data models</td>
<td>82%</td>
<td>2%</td>
<td>9%</td>
<td>3%</td>
</tr>
<tr>
<td>74. Construct several types of models (data, process, functional...)</td>
<td>77%</td>
<td>2%</td>
<td>18%</td>
<td>5%</td>
</tr>
<tr>
<td>75. Customize the language or conventions used for representation</td>
<td>70%</td>
<td>4%</td>
<td>15%</td>
<td>6%</td>
</tr>
<tr>
<td>53. Represent relationships between information requirements &amp; goals</td>
<td>68%</td>
<td>2%</td>
<td>21%</td>
<td>4%</td>
</tr>
<tr>
<td>52. Represent authority relationships of target system's organization</td>
<td>65%</td>
<td>2%</td>
<td>12%</td>
<td>6%</td>
</tr>
<tr>
<td>67. Provide the option of drawing diagram lines exactly where wanted</td>
<td>65%</td>
<td>6%</td>
<td>18%</td>
<td>2%</td>
</tr>
<tr>
<td>55. Combine many entities or processes into a single complex object</td>
<td>59%</td>
<td>3%</td>
<td>15%</td>
<td>2%</td>
</tr>
<tr>
<td>44. Show an object's attributes by selecting it in a diagram</td>
<td>52%</td>
<td>2%</td>
<td>27%</td>
<td>6%</td>
</tr>
</tbody>
</table>

62. Maintain descriptions of existing systems to interact with target system (47%)
63. Provide flexible naming conventions (47%)
60. Maintain a single master definition of each process, object, etc. (44%)
92. Move between different types of models (44%)
22. Redraw a diagram so that it is uncluttered and easy to read (42%)
83. Map the existing systems onto a functional desc. of the organization (42%)
97. Combine structurally equivalent processes or objects (35%)
21. Simultaneously display several screens showing different versions (34%)
11. Choose a first-cut model from among stored generic models (30%)

* - Failed Chi-Square test for 3 dimensions.

Table 1B: Functionalities of Analysis (Production Technology)

<table>
<thead>
<tr>
<th></th>
<th>% 1st Choice</th>
<th>% 2nd Choice</th>
<th>% 3rd Choice</th>
<th>Prod Coord Infr</th>
</tr>
</thead>
<tbody>
<tr>
<td>26. Test for consistency between a process model and a data model</td>
<td>85%</td>
<td>6%</td>
<td>4%</td>
<td>91%</td>
</tr>
<tr>
<td>48. Check for the structural equivalence of objects or processes</td>
<td>82%</td>
<td>5%</td>
<td>9%</td>
<td>1%</td>
</tr>
<tr>
<td>25. Check for unnecessary or redundant model connections</td>
<td>79%</td>
<td>1%</td>
<td>6%</td>
<td>4%</td>
</tr>
<tr>
<td>29. Detect inconsistencies in models, definitions, etc.</td>
<td>79%</td>
<td>4%</td>
<td>9%</td>
<td>1%</td>
</tr>
<tr>
<td>76. Identify the design impact of proposed changes in a design</td>
<td>79%</td>
<td>6%</td>
<td>3%</td>
<td>6%</td>
</tr>
<tr>
<td>31. Search the design for similar objects</td>
<td>74%</td>
<td>6%</td>
<td>12%</td>
<td>1%</td>
</tr>
<tr>
<td>1. Use analytical decision aids to measure performance</td>
<td>73%</td>
<td>1%</td>
<td>9%</td>
<td>3%</td>
</tr>
<tr>
<td>2. Detect and analyze system errors from execution of target system</td>
<td>73%</td>
<td>3%</td>
<td>15%</td>
<td>4%</td>
</tr>
<tr>
<td>38. Identify schedule impacts of a proposed design change</td>
<td>70%</td>
<td>4%</td>
<td>18%</td>
<td>1%</td>
</tr>
<tr>
<td>9. Search design for complex relationships</td>
<td>68%</td>
<td>6%</td>
<td>18%</td>
<td>4%</td>
</tr>
<tr>
<td>80. Suggest problem resolutions based on previously used solutions</td>
<td>65%</td>
<td>6%</td>
<td>21%</td>
<td>13%</td>
</tr>
<tr>
<td>93. Estimate the process/performance characteristics of a design</td>
<td>64%</td>
<td>4%</td>
<td>15%</td>
<td>6%</td>
</tr>
<tr>
<td>8. Search design for objects with specified characteristics</td>
<td>59%</td>
<td>6%</td>
<td>24%</td>
<td>1%</td>
</tr>
<tr>
<td>91. Simulate the production environment of the target system</td>
<td>58%</td>
<td>3%</td>
<td>27%</td>
<td>6%</td>
</tr>
<tr>
<td>6. Identify where predefined criteria or rules have been violated</td>
<td>56%</td>
<td>4%</td>
<td>32%</td>
<td>3%</td>
</tr>
<tr>
<td>4. Trace relationships between detailed specs and planning efforts</td>
<td>50%</td>
<td>1%</td>
<td>21%</td>
<td>4%</td>
</tr>
<tr>
<td>18. Identify the differences between separate versions of an object</td>
<td>50%</td>
<td>6%</td>
<td>15%</td>
<td>4%</td>
</tr>
</tbody>
</table>

94. Recommend a gen'l model incorporating many limited perspectives (41%)
87. Perform an operation on only a portion of a design (35%)

* - Failed Chi-Square test for 3 dimensions.

1 Questionnaire available upon request.
### Table 1C: Functionalities of Transformation (Production Technology)

<table>
<thead>
<tr>
<th>3. Generate executable code from a screen mockup</th>
<th>27. Generate executable code in several procedural languages</th>
<th>50. Generate code compatible with a variety of physical environments</th>
<th>71. Generate standard code for generic programs</th>
<th>90. Generate executable versions of a design for testing/evaluation</th>
<th>28. Convert a logical specification into a physical one</th>
<th>51. Transform a high-level representation into a more detailed one</th>
<th>46. Provide documentation as a by-product of design</th>
<th>73. Perform reverse engineering</th>
<th>49. Generate screen mockups</th>
<th>78. Import data from and export data to external files/packages</th>
</tr>
</thead>
<tbody>
<tr>
<td>% 1st Choice</td>
<td>% 2nd Choice</td>
<td>% 3rd Choice</td>
<td>Prod Coord Infra</td>
<td>Prod Coord Infra</td>
<td>Prod Coord Infra</td>
<td>Prod Coord Infra</td>
<td>Prod Coord Infra</td>
<td>Prod Coord Infra</td>
<td>Prod Coord Infra</td>
<td>Prod Coord Infra</td>
</tr>
<tr>
<td>(91%)</td>
<td>2</td>
<td>3%</td>
<td>4%</td>
<td>3%</td>
<td>94%</td>
<td>3%</td>
<td>3%</td>
<td>91%</td>
<td>3%</td>
<td>94%</td>
</tr>
<tr>
<td>(91%)</td>
<td>2</td>
<td>3%</td>
<td>4%</td>
<td>3%</td>
<td>94%</td>
<td>3%</td>
<td>3%</td>
<td>91%</td>
<td>3%</td>
<td>94%</td>
</tr>
<tr>
<td>(79%)</td>
<td>4</td>
<td>9%</td>
<td>1,2,5,6%</td>
<td>3%</td>
<td>85%</td>
<td>12%</td>
<td>3%</td>
<td>94%</td>
<td>3%</td>
<td>94%</td>
</tr>
<tr>
<td>(79%)</td>
<td>6</td>
<td>9%</td>
<td>4%</td>
<td>6%</td>
<td>82%</td>
<td>9%</td>
<td>9%</td>
<td>97%</td>
<td>0%</td>
<td>97%</td>
</tr>
<tr>
<td>(79%)</td>
<td>1</td>
<td>9%</td>
<td>2%</td>
<td>6%</td>
<td>94%</td>
<td>3%</td>
<td>3%</td>
<td>94%</td>
<td>3%</td>
<td>94%</td>
</tr>
<tr>
<td>(74%)</td>
<td>1</td>
<td>12%</td>
<td>2%</td>
<td>12%</td>
<td>97%</td>
<td>0%</td>
<td>3%</td>
<td>91%</td>
<td>0%</td>
<td>91%</td>
</tr>
<tr>
<td>(68%)</td>
<td>1</td>
<td>18%</td>
<td>2%</td>
<td>12%</td>
<td>97%</td>
<td>0%</td>
<td>3%</td>
<td>94%</td>
<td>6%</td>
<td>94%</td>
</tr>
<tr>
<td>(59%)</td>
<td>6</td>
<td>27%</td>
<td>1%</td>
<td>12%</td>
<td>74%</td>
<td>0%</td>
<td>2%</td>
<td>91%</td>
<td>0%</td>
<td>91%</td>
</tr>
<tr>
<td>(59%)</td>
<td>2</td>
<td>21%</td>
<td>1%</td>
<td>12%</td>
<td>97%</td>
<td>0%</td>
<td>9%</td>
<td>97%</td>
<td>0%</td>
<td>97%</td>
</tr>
<tr>
<td>(53%)</td>
<td>1</td>
<td>23%</td>
<td>2%</td>
<td>18%</td>
<td>94%</td>
<td>6%</td>
<td>0%</td>
<td>94%</td>
<td>6%</td>
<td>94%</td>
</tr>
<tr>
<td>(50%)</td>
<td>5</td>
<td>22%</td>
<td>6%</td>
<td>13%</td>
<td>56%</td>
<td>31%</td>
<td>13%</td>
<td>56%</td>
<td>31%</td>
<td>56%</td>
</tr>
</tbody>
</table>

47. Create templates for tasks and deliverables
(38%) 4 38% 1.6 12% 50% 38% 12% *
32. Propagate a change in an object to all places the object appears
(32%) 1 27% 2.4 12% 70% 21% 9% **

* = failed Chi-Square test for 3 dimensions.
** = failed Chi-Square test for 6 components.

### Table 1D: Functionalities of Control (Coordination Technology)

<table>
<thead>
<tr>
<th>14. Specify who can review various parts of the design work</th>
<th>61. Provide project management information</th>
<th>72. Maintain a record of who is responsible for each part of project</th>
<th>60. Maintain a record of changes made in the design</th>
<th>19. Provide management information for more than one project</th>
<th>37. Specify who can modify various parts of the design work</th>
<th>66. &quot;Freeze&quot; a portion of a design to protect it from changes</th>
<th>17. Manage the quality assurance path for a project</th>
<th>40. Alter rules that control the way certain functions are performed</th>
<th>15. Provide assistance in analyzing project management priorities</th>
<th>57. Estimate how long a specific task or project will take</th>
<th>69. Remind members of team about approaching deadlines</th>
</tr>
</thead>
<tbody>
<tr>
<td>(79%)</td>
<td>5</td>
<td>15%</td>
<td>1%</td>
<td>6%</td>
<td>6%</td>
<td>94%</td>
<td>0%</td>
<td>94%</td>
<td>0%</td>
<td>94%</td>
<td>0%</td>
</tr>
<tr>
<td>(79%)</td>
<td>2</td>
<td>6%</td>
<td>3%</td>
<td>6%</td>
<td>15%</td>
<td>82%</td>
<td>3%</td>
<td>94%</td>
<td>3%</td>
<td>94%</td>
<td>3%</td>
</tr>
<tr>
<td>(71%)</td>
<td>5</td>
<td>12%</td>
<td>6%</td>
<td>12%</td>
<td>6%</td>
<td>82%</td>
<td>12%</td>
<td>6%</td>
<td>82%</td>
<td>12%</td>
<td>6%</td>
</tr>
<tr>
<td>(65%)</td>
<td>5</td>
<td>18%</td>
<td>6%</td>
<td>9%</td>
<td>9%</td>
<td>82%</td>
<td>9%</td>
<td>94%</td>
<td>9%</td>
<td>94%</td>
<td>9%</td>
</tr>
<tr>
<td>(64%)</td>
<td>5</td>
<td>18%</td>
<td>1%</td>
<td>9%</td>
<td>9%</td>
<td>82%</td>
<td>9%</td>
<td>94%</td>
<td>9%</td>
<td>94%</td>
<td>9%</td>
</tr>
<tr>
<td>(64%)</td>
<td>5</td>
<td>21%</td>
<td>1%</td>
<td>9%</td>
<td>9%</td>
<td>82%</td>
<td>9%</td>
<td>94%</td>
<td>9%</td>
<td>94%</td>
<td>9%</td>
</tr>
<tr>
<td>(62%)</td>
<td>1</td>
<td>12%</td>
<td>5%</td>
<td>9%</td>
<td>9%</td>
<td>79%</td>
<td>6%</td>
<td>94%</td>
<td>6%</td>
<td>94%</td>
<td>6%</td>
</tr>
<tr>
<td>(55%)</td>
<td>5</td>
<td>30%</td>
<td>1.6%</td>
<td>6%</td>
<td>9%</td>
<td>85%</td>
<td>6%</td>
<td>94%</td>
<td>6%</td>
<td>94%</td>
<td>6%</td>
</tr>
<tr>
<td>(55%)</td>
<td>1</td>
<td>15%</td>
<td>5%</td>
<td>12%</td>
<td>24%</td>
<td>67%</td>
<td>9%</td>
<td>94%</td>
<td>6%</td>
<td>94%</td>
<td>6%</td>
</tr>
<tr>
<td>(52%)</td>
<td>2</td>
<td>21%</td>
<td>6%</td>
<td>12%</td>
<td>33%</td>
<td>55%</td>
<td>12%</td>
<td>33%</td>
<td>55%</td>
<td>12%</td>
<td>33%</td>
</tr>
<tr>
<td>(52%)</td>
<td>2</td>
<td>27%</td>
<td>6%</td>
<td>12%</td>
<td>36%</td>
<td>52%</td>
<td>12%</td>
<td>36%</td>
<td>52%</td>
<td>12%</td>
<td>36%</td>
</tr>
<tr>
<td>(52%)</td>
<td>5</td>
<td>33%</td>
<td>6%</td>
<td>9%</td>
<td>6%</td>
<td>85%</td>
<td>6%</td>
<td>94%</td>
<td>6%</td>
<td>94%</td>
<td>6%</td>
</tr>
</tbody>
</table>

43. Follow rules in merging separate versions of models, diagrams, etc. (49%) 5 21% 3 15% 27% 70% 3%
82. Produce metrics for comparing projects (complexity, quality, etc.) (49%) 2 21% 6 18% 30% 52% 18% *
39. Maintain list of requirements for design and how satisfied (46%) 2 27% 1 12% 49% 45% 6%
7. Temporarily ignore a problem/inconsistency so work can continue (29%) 6 27% 2 24% 41% 32% 27% *

* = failed Chi-Square test for 3 dimensions.
Table 1E: Functionalities of Cooperative Functionality (Coordination Technology)

<table>
<thead>
<tr>
<th>#</th>
<th>Functionalities</th>
<th>1st Choice</th>
<th>2nd Choice</th>
<th>3rd Choice</th>
<th>Prod Coord Inf</th>
</tr>
</thead>
<tbody>
<tr>
<td>41</td>
<td>Maintain a dialogue with other users of the tools</td>
<td>(91%)</td>
<td>6%</td>
<td>1%</td>
<td>3% 91% 6%</td>
</tr>
<tr>
<td>85</td>
<td>Allow a group of users to work simultaneously on a single task</td>
<td>(91%)</td>
<td>6%</td>
<td>1%</td>
<td>3% 6% 91% 3%</td>
</tr>
<tr>
<td>16</td>
<td>Send messages to others who use the tools</td>
<td>(88%)</td>
<td>6%</td>
<td>4.2%</td>
<td>3% 3% 91% 6%</td>
</tr>
<tr>
<td>42</td>
<td>Allow concurrent use by several users of dictionary/diagram/etc.</td>
<td>(85%)</td>
<td>4%</td>
<td>6%</td>
<td>1,2% 6% 91% 3%</td>
</tr>
<tr>
<td>64</td>
<td>Provide group interaction support (brainstorming, NGT, etc.)</td>
<td>(85%)</td>
<td>2%</td>
<td>6%</td>
<td>4% 3% 9% 88% 3%</td>
</tr>
<tr>
<td>84</td>
<td>Attach electronic notes to objects for others to read</td>
<td>(62%)</td>
<td>6%</td>
<td>15%</td>
<td>1% 12% 20% 65% 15%</td>
</tr>
<tr>
<td>20</td>
<td>Allow giving of anonymous feedback or input</td>
<td>(53%)</td>
<td>34%</td>
<td>6%</td>
<td>6% 6% 88% 6%</td>
</tr>
<tr>
<td>86</td>
<td>Notify designer if a change is made in design that affects his work</td>
<td>(53%)</td>
<td>4%</td>
<td>35%</td>
<td>1% 9% 9% 88% 3%</td>
</tr>
<tr>
<td>59</td>
<td>Build a catalog of macros that other users can access</td>
<td>(50%)</td>
<td>6%</td>
<td>27%</td>
<td>13% 12% 24% 50% 26%</td>
</tr>
</tbody>
</table>

85. Help the designer and end user evaluate design alternatives                | (41%)      | 2%         | 35%        | 6% 9% 47% 44% 9% |

* -- failed Chi-Square test for 3 dimensions.

Table 1F: Functionalities of Support (Infrastructure Technology)

<table>
<thead>
<tr>
<th>#</th>
<th>Functionalities</th>
<th>1st Choice</th>
<th>2nd Choice</th>
<th>3rd Choice</th>
<th>Prod Coord Inf</th>
</tr>
</thead>
<tbody>
<tr>
<td>56</td>
<td>Provide aids for quick references to basic commands/functions</td>
<td>(97%)</td>
<td>3%</td>
<td>3%</td>
<td>0% 3% 6% 97%</td>
</tr>
<tr>
<td>77</td>
<td>Provide on-line help for a specified command/feature</td>
<td>(94%)</td>
<td>1%</td>
<td>3%</td>
<td>4% 3% 3% 94%</td>
</tr>
<tr>
<td>10</td>
<td>Provide instructional materials for learning the tools</td>
<td>(91%)</td>
<td>1%</td>
<td>3%</td>
<td>4.5% 3% 3% 6% 91%</td>
</tr>
<tr>
<td>12</td>
<td>Provide context-specific on-line help</td>
<td>(88%)</td>
<td>1%</td>
<td>6%</td>
<td>4% 6% 6% 88%</td>
</tr>
<tr>
<td>38</td>
<td>Identify external sources of information on specific topics</td>
<td>(84%)</td>
<td>2%</td>
<td>9%</td>
<td>1% 6% 16% 0% 84%</td>
</tr>
<tr>
<td>13</td>
<td>Provide options about how to interact with the tools</td>
<td>(76%)</td>
<td>1%</td>
<td>9%</td>
<td>3% 9% 21% 3% 76%</td>
</tr>
<tr>
<td>33</td>
<td>Build templates or examples of work for use in tutorials/demos</td>
<td>(59%)</td>
<td>5%</td>
<td>21%</td>
<td>1% 9% 18% 23% 59%</td>
</tr>
<tr>
<td>95</td>
<td>Explain why an action or alternative has been recommended</td>
<td>(55%)</td>
<td>2%</td>
<td>30%</td>
<td>1% 1,3% 6% 42% 3% 55%</td>
</tr>
<tr>
<td>45</td>
<td>&quot;Browse&quot; in other segments of the tool while using graphics mode</td>
<td>(50%)</td>
<td>1%</td>
<td>27%</td>
<td>2% 21% 47% 3% 50%</td>
</tr>
<tr>
<td>54</td>
<td>Explain why part of a design has been identified as inconsistent</td>
<td>(50%)</td>
<td>2%</td>
<td>41%</td>
<td>1% 6% 47% 3% 50%</td>
</tr>
<tr>
<td>58</td>
<td>Anticipate user’s mistakes from his pattern of previous errors</td>
<td>(50%)</td>
<td>4%</td>
<td>18%</td>
<td>2% 12% 32% 18% 50%</td>
</tr>
<tr>
<td>79</td>
<td>Allow the undoing of a series of commands</td>
<td>(50%)</td>
<td>2%</td>
<td>18%</td>
<td>3% 18% 44% 9% 50%</td>
</tr>
<tr>
<td>89</td>
<td>Generate outputs in a variety of media</td>
<td>(50%)</td>
<td>5%</td>
<td>21%</td>
<td>3% 15% 26% 24% 50%</td>
</tr>
</tbody>
</table>

35. Incorporate new command "macro" into command structure                    | (49%)      | 3%         | 24%        | 1% 15% 39% 12% 49%|
| 88  | Generate presentation-quality printed reports and documents                     | (47%)      | 3%         | 27%        | 1% 12% 35% 15% 47%|
| 70  | Provide individual change pages of documents                                   | (43%)      | 5%         | 18%        | 2% 4% 15% 24% 33% 43%|
| 22  | Graphically magnify a model to see greater levels of detail                    | (38%)      | 1%         | 38%        | 2% 18% 5% 29% 3% 38%|
| 34  | Build a general access library of customized models                            | (35%)      | 5%         | 27%        | 1% 24% 35% 30% 35%|
| 68  | Prepare, edit, store, send and retrieve documents                              | (32%)      | 1%         | 29%        | 2% 23% 5% 13% 32%|
| 81  | Store versions of a design for later "roll-back"                               | (32%)      | 4%         | 30%        | 2% 21% 38% 30% 32%|
| 98  | Link a design to a library of models/systems for testing                       | (30%)      | 2%         | 24%        | 5% 21% 49% 21% 30% |
| 24  | Develop, run & store completely customized reports                             | (29%)      | 1%         | 24%        | 3% 21% 59% 12% 29% | ** |

* -- failed Chi-Square test for 3 dimensions.
** -- failed Chi-Square test for 6 components.
Functionalities such as an ability to maintain a single master definition or the ability to describe a process in terms of an information flow reflect basic requirements to represent knowledge.

A second aspect of the representation dimension reflects requirements for adapting or changing representations, and for storing or retrieving representations. For example, the ability to propagate a change through a model supports a user in an adaptation or change task.

Finally the ability to use alternative modes of representation, e.g., text versus visual representation, is reflected. In fact, as suggested by Konsynski et al. (1984), our subjects viewed the ability to shift between alternative representations as an important type of functionality.

Several observations seem appropriate. As we will discuss in Section 4, a distinction often made between design support environments is the ease of use of a functionality. For example, two design aid environments may support data flow diagramming. They may differ significantly, however, in terms of the ease of use of this functionality. Ease of use can be viewed as a measure of effort required to exercise the functionality and, thus, a relative measure of cost. Combining a functional model with the notion of ease of use will permit the researcher to explore the usability of CASE technology.

Secondly, the level of specificity of the functionality reflects the goal of creating a correspondence between the functional model and usage behavior. For example, interviewees rejected as too general the use of "documentation" as a type of functionality. Rather, discussions indicated that documentation is a form of representation (a passive form) that requires particular functionality. The need to
develop a parsimonious model in a research setting (particularly one that requires users of a system to describe their usage behaviors) argues against a micro model. The functionality described herein reflects the subjects' judgment as to an appropriate level of aggregation.

Finally, there is no claim that the functionality listed in Table 1A constitutes an exhaustive set for the representation component. Rather, this functional set is viewed as spanning or reflecting the scope of this component. As we will discuss, the convergence found in the Q-sort process and the ability to discriminate across actual products support the conclusion that these functionalities can be meaningful group under the proposed definition of representation.

2.1.2 Production Technology: Analysis

This dimension of analysis reflects the problem-solving and decision-making aspects of planning and design. Simon (1981), for example, portrays design as a problem-solving process and emphasized the criticality of tasks involving evaluation of multiple alternatives and choices made by the designer. In a similar vein, we define the analysis dimension to be functionality that enables the user to explore, simulate, or evaluate alternate representations or models of objects, relationships or processes.

We see this requirement reflected in the functionality listed in Table 1B. Similar to the functional building block of a decision support system (Keen and Scott Morton (1978), Treacy (1981), Sprague and Carlson (1982)), these functionalities reflect the need to compare, simulate, evaluate, ask "what if" with respect to a criteria, and choose or optimize. It is interesting to note that some functional definitions imply an embedded intelligence in the design aid. For example, the ability to explain why a
design decision is best reflects the use of expert system and AI concepts in the development of design aids.

In each case, the functionality in this dimension (Table 1B) assumes the existence of a knowledge base (often a model) and seeks to manipulate this knowledge in order to investigate alternatives, resolve conflicts or support a choice. It is a proactive analysis process that builds upon or adds to knowledge. Thus, we would expect the result of using analysis functionality to be the enhancement or adjustment of a given representation (i.e., the use of modeling functionality). The significant interaction between these two dimensions suggests that they constitute components of the more general dimension of Production Technology.

2.1.3 Production Technology: Transformation

The nature of planning and design has been conceptualized as a process or series of transformations (Kottemann and Konsynski 1984, Zachman 1986). A transformation is an internally complete and consistent change in design concept or artifact. The need for completeness and consistency reflects the attribute that a transformation is a non-random purposeful activity and, hence, is repeatable. For example, converting a logical data model into a set of definitions represented in the language of a given data base system constitutes a transformation.

In general, the notion of transformation has been the mechanism to represent important aggregates or chunks of design activity. At a macro-level, the system design life cycle describes a series of design transformations. Researchers such as Zachman (1986) and Hackathorn and Karimi (1988) have suggested a range of transformations that are central to I/S planning and design processes. We define the dimension of transformation to be functionality that executes a significant planning or design task, thereby replacing or substituting for a human designer/planner.
This dimension of CASE technology reflects a straightforward capital/labor substitution. It differs from analysis in that it replaces human activity rather than providing support. In this sense, it is analogous to the distinction between decision support systems and process automation. Of course, transformation technology can enhance the overall performance of humans by allowing redistribution of human resources. Still, at task level, the intent of transformation functionality is direct substitution for the human resource.

The functionalities listed in Table 1C correspond to the transformation dimension. Several observations are appropriate. As might be expected, the bulk of these functionalities address activities late in the design cycle, e.g., code generation. As such, these functionalities often depend on a minimum set of functions being available in the representation component. However, as we will discuss in Section 4, current technology often does not effectively link these two functional components.

A second observation is that the ability to deliver transformation functionality often implies embedding intelligence into the CASE technology. For example, the ability to automatically normalize a data model is an emerging type of transformation functionality that makes extensive use of expert systems and AI technology. As we see increased use of intelligent CASE technology we might expect to see new types of functionality emerge for this dimension. Thus, the set of functionality shown in Table 1C should be viewed as a current benchmark.

2.1.4 Coordination Technology: Control

The focus of the dimensions of technology discussed thus far has been production-oriented. That is, the technology has provided a direct impact on the ability of an individual to produce aspects of the design. In this capacity, the technology
represents a classic productivity-enhancing investment, i.e., a capital/labor tradeoff. Through the investments in technology the task of a design team is accomplished with less resources.

Williamson (1975) notes, however, that the constraints on human information processing can arise from both bounded rationality of a particular agent and from the communication requirements stemming from interaction between agents. Bakos and Treacy (1986) also identify the need to reflect both bounded rationality of individuals and communication costs in a general model of information technology.

Malone (1988) defines coordination as "the additional information processing performed when multiple, connected actors pursue goals that a single actor pursuing the same goals would not perform". The use of technology to reduce the cost of coordination can enable an organization to utilize alternative organizational structures in pursuit of their strategies, and, thereby, achieve new levels of efficiency and effectiveness. For example, Applegate, et al. (1986) and Stefik, et al. (1987) describe technology that is intended to improve the productivity of meetings in part through enhanced communication functionality. Such technology can not only affect the efficiency or effectiveness of a given meeting but also enable organizations decision making or problem solving processes that maximize the use of teams or task forces.

The interviewees also identified a range of technology that focused on the need to effectively coordinate individuals. It was interesting to note that during the interviews subjects seemed to shift from conceptualizing the planning or design process as an individual activity to one involving a group or team. When this shift occurred the design aid functionality discussed reflected issues such as the need to
exchange information, enforce policies or security measures, or understand or resolve conflicts.

It is not surprising that one aspect of design aid technology that emerges from the design literature reflects a component of coordination: control. This component reflects a notion of a manager/employee or principal/agent relationship in a planning or design process. That is, design activities often involve an explicit contract to deliver a product or service to a customer for a given price. In order to ensure that the contract is fulfilled, a control system or monitoring system is required. Similarly, with the activities of a design team, a project leader may contract with an individual. Again, the project leader requires some information to ensure that this individual does, in fact, carry out the contract in the intended way.

In addition to the need to monitor, the principal or manager may want to impose restrictions on the activities of a given agent or employee. For example, he/she may want to restrict access to particular data or prevent changes to some aspect of an existing or proposed system. At a more abstract level, the project leader needs an ability to communicate project goals (even the means to achieve goals) and to ensure that the resources of the teams are allocated in a manner that best achieves the goals.

Of course, requirements to control the activities of a group have long been recognized by the developers of computer-aided design technology. Houghton and Wallace (1987), Reifer and Montgomery (1980) and others identify a range of functionality spanning notions of project management, configuration control, and access control. We define the control dimension to be the functionality that enables the user to plan for and enforce rules, policies or priorities that will govern or restrict the activities of team members during the planning or design process.
The functionality of this dimension identified in the interviews is shown in Table 1D. There appear to be two general types of relations to this dimension: resource management and access control. Resource management pertains to that functionality that enables a manager to ensure that the behavior of individuals and hence, resource utilization by the team is consistent with organization goals. The capability to budget, to identify a critical path or set of activities, to monitor progress or service levels, or to communicate or enforce appropriate goals are examples of this type of functionality. In essence, it is functionality that supports a range of traditional control activities. As will be discussed later, the potential for CASE technology to enable effective internal control, i.e., substitute individual control behavior for managerial control, has major implications for performance.

A second type involves access or change control. This functionality assumes that issues of security and access must be carefully managed. As shown in Table 1D, this functionality includes configuration control, authorization management, and the ability to identify and audit the activity of designers, particularly when these activities change existing work or directly pertain to a team policy. In essence, these types of functionality assume that the design team utilizes and produces a valuable asset. Hence, access to or changes to those assets must be monitored and controlled.

2.1.5 Coordination Technology: Cooperative Functionality

The control dimension addresses the need to establish and enforce goals, policy, procedures, standards and priorities during a design process. It is the traditional concept of manager/employee that assumes the need to enforce a work contract. Information is required both to ensure effective execution of task and to monitor the contract.
An alternative mode of coordination assumes that the participants operate at a peer to peer level. In this mode, the interaction among individuals is based on a shared set of goals and a perception of mutual gain from a given interaction. Thus, cooperative behavior is not enforced by a set of rules. Rather, such interaction reflects a sense of peer involvement where exchange is often voluntary.

Davis and Smith (1983), Henderson (1988) and Malone (1988) describe the concept of cooperative behavior in this manner. For example, Davis and Smith (1983) argue that the need for cooperation among experts arises from both shared goals and knowledge interdependence among the experts with respect to these goals. In this research we will define the dimension of cooperative functionality as functionality that enables the user to exchange information with another individual(s) for the purpose of influencing (affecting) the concept, process or product of the planning/design team.

The interview process generated a range of functionalities that are modeled as cooperative functionality (Table 1E). These functionalities reflect a role of CASE technology both as a communication channel and as a facilitation aid. Reifer and Montgomery (1980) identify communication functionality as an important aspect of computer-aided design technology. Certainly in a group context communication is a key issue. The basic communication functions in Table 1E address the need for a range of communication functionality from basic messaging to enhancements such as the ability to attach a note to a diagram. In essence, this functionality provides a platform for electronic interaction among members of a team.

The second class of cooperative functionality uses technology to help facilitate group interaction. This includes functionality that provides for electronic brainstorming or manages the degree of anonymity of input (i.e., votes). Applegate,
et al. (1986) describe technology that provides this type of functionality. The user of PLEXSYS technology can choose between several structured group processes and adapt the technology to facilitate the execution of the particular approach used. The technology has an impact on the process both through efficiency, e.g., the ability to capture the output of a brainstorming session, and also by changing parameters of the group process within an efficiency level. For example, the technology can permit significantly larger group size than is often associated with a brainstorming session. To the extent that participation and involvement affects the success of a project, this increased capacity could have significant benefits. These functionalities, particularly those that implement structured group process, have aspects of control embedded in them. For example, electronic brainstorming enforces an interaction protocol on members of the team. This association between control and cooperative functionality is to be expected since they are both components of the common dimension of coordination. The key distinction is that cooperative functionality assumes a peer relationship among participants and is based on a concept of sharing. The technology functions primarily as a conduit or enabler of information exchange. Control functionality, in contrast, assumes that a hierarchical relationship exists and provides a mechanism to exchange information necessary to establish, monitor and enforce this hierarchy. Each relates to coordination but does so from a different perspective.

2.1.6 Infrastructure Technology: Support

Simon (1976) notes that bounds of rationality can be increased not only by increasing individual computational power, but also by institutionalizing organization-wide standards to help individual performance. This capability, we term infrastructure technology, can be defined as organization-wide mechanisms through which an organization provides "institutionalized help" to individuals and
groups to overcome their cognitive burdens of information processing. March and Simon (1958) argue that by establishing organization infrastructures, which they call standard operating procedures, the organization can reduce burdens of information processing because search procedures are automated in the standard operating procedures to some extent. Similarly, Galbraith (1977) argues that implementing a vertical information system and the implied standards of data and language associated with such a system is one strategy to increase the information processing capacity of the firm. Malone (1988) extends this notion to describe a range of organizational structures enabled by the use of coordination technology.

Computer based design tools can also provide organization-wide infrastructure for the development of complex software. Often, complex software is built module by module by several design teams. If the teams do not proceed carefully, the idiosyncrasies of an undisciplined team can lead to expensive failure. Design aid tools help the design team manage complexities of development by providing a common foundation for the development of I/S. As a result, the organization gains the potential to introduce parallelism as well as time share scarce talent among teams. The design aid tools also help train designers in advance techniques and enforce consistent techniques usage throughout the organization.

However, because enforcement of organization-wide infrastructure comes primarily by limiting what design teams can do with the tools, there is the potential that an inflexible infrastructure can stand in the way of designing effective systems. Therefore, while the ultimate power of infrastructure technology lies in the ability to widen as far as possible the range of solutions and approaches that can be handled by infrastructure technology, the actual impact on the development process is unclear.
One component of the infrastructure dimension addresses the skills to use technology rather than the task of planning and design. At issue is the range of support required to help the design aid user learn about and utilize the design aid in the most effective way possible. We define this dimension to be the \textit{functionality to help an individual user understand and use effectively a planning and design aid}.

Table 1F lists the range of functionality relating to this dimension. These functions range from passive functionality, e.g., an on-line help function, to describe parameters of a function, to proactive functionality that uses domain knowledge or past user behavior patterns to diagnose or recommend appropriate action, e.g., the ability to explain why a particular functionality should be used.

Many characteristics of "user friendly" systems incorporate these types of support functionality. For example, Houghton and Wallace (1987) describe a range of support functions that reflect the range of skills (expert to novice) of a typical user population. It should be noted that the general interface technology is not incorporated as a support function. For example, the use of a mouse or point-and-click is a feature that affects the effort necessary to exercise a functionality (either physical or mental). As such this aspect of the design environment should be incorporated into the measure of ease of use of a set of functions.

2.1.7 Infrastructure: Standards

Ultimately, the need to develop and sustain an organizational infrastructure demands attention to the need for standards. As suggested above, standards offer the potential both to increase organizational flexibility and to limit the creative process of planning and design. For example, Lempp and Lauber (1988) have argued that the issue of emerging standards for computer aided design technology and
practice is a strategic concern to organizations that depend upon information technology.

A major function of the standard component functionality is to provide portability of skills and data. Portable skills and data will be promoted through standardized relationships between various activities of design life cycle. The ability to introduce simultaneous design processes is enhanced. For example, adopting a standard structure for representing the knowledge generated in a design process increases the ability to share this knowledge with other teams. Similarly, it provides a basis to train designers as to what knowledge is available and how other teams function. As a result, increased organization performance can be achieved by a given team's ability to anticipate when coordination is required.

The interviewees generated few examples of functionality that could be thought of as standards. In general, there is a potential standards issue in many of the elements of the coordination functionality. However, during debriefing with organizations, the issue of standards was highlighted. The discussion of standards functionality often reflected system utilities and architectures. For example, one functionality focused on the ability to port between technology platforms. Another focused on the ability to function in a highly distributed environment. The issue of the consistency of the structure used to store data definitions with the emerging standards for a central repository was also highlighted.

In essence, the feedback was to incorporate a dimension of design aid technology that reflects a potential to support organization change and flexibility. As such we define the standards component as functionality that promotes portability of skills, knowledge, or methods across the organization(s).
2.2 Summary

A final concern in the development of the functionality items is the ability to reliably associate a particular functionality with actual CASE product. As discussed in Section 2, a reliability check resulted in a total of 98 functions forming the pool with which to define the functional dimension described above. In the following section, a Q-sort test used to examine the robustness of the proposed model is discussed. The test consists of giving independent experts in CASE technology the definitions of each component\(^2\) and asking them to sort the 98 functions in these categories. The listing of functions by each component shown in Table 1A-1F is the result of this Q-sort exercise. To the extent that the subjects sort the functions in the same way, there is evidence that the proposed model is a meaningful characterization of wide range of CASE technology. A second test examines the extent to which the model actually discriminates between CASE products in an interesting and useful way. The following section presents the results of this Q-sort and the application of the model to evaluate eight commercially available CASE products.

3.0 Evaluating the FCTM

The results of the Q-sort test are shown in the right-hand columns of Table 1. A total of 34 subjects (not involved in previous development of this model) sorted the 98 functionalities according to the definition described in Section 2. The results are tabulated based on the categories receiving the most frequent assignments.

\(^2\) As discussed in Section 2.1.7, the standard component resulted from feedback during debriefing. Thus, it is not included in the Q-sort test.
Functionality is listed in order of declining frequency among the 34 subjects. Each column has two numbers. The first indicates the specific component most receiving most assignments, the second indicates the percentage of the total assignments following in that component. The first, second and third frequency are shown. This accounts for almost 100% of assignment in all cases.

A second aspect of the model can also be examined with this data. Even if assignments do not indicate agreement as to a primary component, there may be agreement at the more general dimension of production, coordination or infrastructure. If this is true then there is support for that these more general dimensions adequately reflect current CASE technology.

A simple chi square test is used to test the hypothesis that assignments are random. The results of this simple test in Table 1 can be evaluated at both the component level, i.e., the six component that were used in the sort, and at the dimension level, i.e., production, coordination and infrastructure. At the component level, there are only two functions for which a chi square test of uniform distribution is not rejected (transformation, #32 and support, #24). Although this is a weak test, it does support the conclusion that the six component do differ significantly. At the dimension level, seven functions failed to reject the test of a random assignment. Again, this supports the conclusion that these dimensions differ.

A review of the assignment pattern is more revealing. For representation, only nine of the eighteen items received more than 50% as a primary sort. However, as indication in the comments section, five of the functions below 50% appear to have consensus as a general production functionality.

The sorting results for analysis appear more consistent with seventeen of nineteen function receiving more than 50% primary assignments. Again the two
items below 50% appear to reflect a general production functionality with a fairly uniform distribution across representation, analysis and transformation.

The transformation component has eleven of thirteen functions exceeding 50%. In general the functions appear to be clearly within the production dimension. The two functions that are below 50% is ambiguous. Both functions 47 and 32 fail to reject the chi square test at the dimension level, suggesting that there is significant overlap with between the production and control implications for these functions.

In the control component, twelve of sixteen functions receive more than 50% primary assignments. The distribution of assignment suggests a support of this component and a consensus with respect to a coordination dimension. For the four functions not receiving more than 50% assignment, #43 appears to reflect a general coordination perspective, while functions 83, 39 and 7 fail to reject the chi square for differences at the dimension levels. These functions appear to have overlap with support and analysis, suggesting a significant level of ambiguity in the functional description.

The cooperative functionality component receive only total ten assignments but nine of the ten received more than 50% as a primary assignment. In general, these functions appear to reflect a coordination perspective but subjects distinguished them from the control component. Function 85 did not receive more than 50% primary assignment and also failed to reject the chi square test at the dimension level. The component shows significant overlap with both analysis and support.

Finally, the support component had 22 functionalities with only thirteen of the 22 receiving primary assignment. This component appears to be difficult for subjects to clearly differentiate. Although there are six functions that have strong agreement as support, the remaining functions reflect both aspects of production and
coordination. Two of nine functions receiving less than 50% primary assignment fail the chi square test. Function 98 fails to reject the test at the dimension level and function 24 fails to reject at the weaker component level test. The sort pattern across those assignments with less than 50% primary sort appears to reflect significant overlap with at least one other dimension. These results suggest a need to refine the definition for the support component.

In summary, the sorting results provide support for each of the component concepts. Only thirteen of the 98 functions fail to reject the chi square test at the dimension level or component level. Twenty-seven functions receive less than 50% as a primary sort. Again, fourteen of these 27 have support as the first or second choice, reflecting the difficulty with the definition of this component. Of the remaining thirteen functions, six reflect a general production perspective and one a general coordination perspective thereby providing additional support for the dimension level concepts.

As a next step in the analysis, the seven components are use to compare eight commercially available CASE products. The comparison will be used to determine if the FCTM provides a useful tool to evaluate potential CASE environments.

3.1 Comparison of CASE Products

In this section, the FCTM is used to characterize eight commercially available CASE products. The products were selected in an attempt to cover the full span of the system development life cycle. The life cycle was divided into three general categories: planning, design and construction. Two products that appear to target each of these were selected for comparison. In addition, two products that purport to provide integration across all three components were selected for evaluation. To ensure the products did in fact reflect these components, 25 experts users were asked
to indicate the level of support provided by the product for the seven tasks shown in Table 2. These perceptions support the conclusion that the tools selected for evaluation both span the life cycle and have distinctive product features.

Table 2
Life Cycle Coverage by Products

<table>
<thead>
<tr>
<th>Design Activity</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>I/S Planning</td>
<td>4.5</td>
<td>3.71</td>
<td>1.88</td>
<td>2.8</td>
<td>2.0</td>
<td>1.1</td>
<td>3.5</td>
<td>2.4</td>
</tr>
<tr>
<td>Requirement Definition</td>
<td>3.25</td>
<td>3.86</td>
<td>4.0</td>
<td>3.3</td>
<td>2.8</td>
<td>1.8</td>
<td>4.5</td>
<td>3.2</td>
</tr>
<tr>
<td>Conceptual Design</td>
<td>3.0</td>
<td>3.57</td>
<td>4.50</td>
<td>3.67</td>
<td>2.8</td>
<td>2.3</td>
<td>4.8</td>
<td>4.0</td>
</tr>
<tr>
<td>Detailed Design</td>
<td>2.0</td>
<td>2.29</td>
<td>3.63</td>
<td>3.0</td>
<td>4.0</td>
<td>3.4</td>
<td>4.6</td>
<td>4.6</td>
</tr>
<tr>
<td>Implementation</td>
<td>1.33</td>
<td>1.86</td>
<td>2.1</td>
<td>1.6</td>
<td>4.6</td>
<td>4.7</td>
<td>3.1</td>
<td>3.1</td>
</tr>
<tr>
<td>Testing</td>
<td>1.0</td>
<td>1.43</td>
<td>1.6</td>
<td>1.0</td>
<td>3.2</td>
<td>4.3</td>
<td>2.2</td>
<td>1.7</td>
</tr>
<tr>
<td>Maintenance</td>
<td>2.0</td>
<td>1.43</td>
<td>2.5</td>
<td>1.8</td>
<td>4.2</td>
<td>4.8</td>
<td>3.9</td>
<td>2.5</td>
</tr>
</tbody>
</table>

Scale: 1 very little, 2 some, 3 adequate, 4 good, 5 extensive

Table 3 provides a summary of the product evaluations. In each case, three to five expert users of the product were asked to evaluate the ease of use with respect to the 98 functions for a specific product. A five point Likert scale (see Table 3) was used for evaluation. A function was considered to exist if the average response of the subjects was greater than 3.0.

Several observations can be drawn from Table 3. First, the model at the component level does differentiate across products in an expected way. For example, the products that target planning and conceptual design have the focus of their
The user can instruct the tools to redraw a diagram on the screen so that it is uncluttered and easy to read.

<table>
<thead>
<tr>
<th>Example</th>
<th>Total Used</th>
<th>H</th>
<th>G</th>
<th>F</th>
<th>E</th>
<th>D</th>
<th>C</th>
<th>B</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>17</td>
<td>10</td>
<td>13</td>
<td>3</td>
<td>2</td>
<td>9</td>
<td>11</td>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td>2</td>
<td>94</td>
<td>56</td>
<td>72</td>
<td>17</td>
<td>11</td>
<td>50</td>
<td>61</td>
<td>28</td>
<td>39</td>
</tr>
<tr>
<td>3</td>
<td>9</td>
<td>3</td>
<td>5</td>
<td>0</td>
<td>2</td>
<td>26</td>
<td>16</td>
<td>11</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>47</td>
<td>16</td>
<td>26</td>
<td>4</td>
<td>1</td>
<td>21</td>
<td>7</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>11</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 3</th>
<th>Summary of Product Evaluations</th>
</tr>
</thead>
<tbody>
<tr>
<td>CASE 18</td>
<td>Representation Possible %</td>
</tr>
<tr>
<td>CASE 19</td>
<td>Analysis Possible %</td>
</tr>
<tr>
<td>CASE 13</td>
<td>Transformation Possible %</td>
</tr>
<tr>
<td>CASE 16</td>
<td>Control Possible %</td>
</tr>
<tr>
<td>CASE 10</td>
<td>Functionality Possible %</td>
</tr>
<tr>
<td>CASE 22</td>
<td>Support Possible %</td>
</tr>
</tbody>
</table>
functionality on representation while being relatively weak on transformation. Similarly, those products targeting construction provide transformation functionality and are weaker on representation.

Secondly, only one product provides significant coverage for control functionality. Further, all products are weak on cooperative functionality. This results suggests the current products may have limited impact on team performance issues. This point will be discussed in more depth in the next section.

Finally, the products do provide support functionality but there is, in fact, significant variation across product. As we will discuss, a more detailed analysis shows there exists general level of support in the form of basic help commands but advanced, intelligent support functionality is quite limited.

The detail analysis by component is shown in Table 4. At this level, one can compare functionality across products. For example, in support, the function of "provide on line help" and "quick reference to basic commands" (#56 and 77) is generally available across the life cycle. More sophisticated and intelligent based support such as "the ability to anticipate user mistakes based on past errors" (#58) is totally lacking.

A final observation is reflected in the summary total used for Table 4. This row indicates the number and percentage of the total possible functionality that appears in at least one product. The results suggest that claims for integration and coverage by CASE products are at best limited to notions of production technology. There is a significant gap between possible and available functional in coordination, analysis and intelligent forms of support. Furthermore as the detail analysis suggest the degree of support even within production can vary significantly.
These results suggest the FCTM is a meaningful way to characterize design aid technology. While clearly not the only possible perspective, this model does appear to reflect a reliable and valid model for a wide range of functionality. The model does differentiate across products. In the following section, the implication for use of this model of technology to study the impact of I/S planning and design are discussed.

4.0 Implications and Future Research

This research has led to the development of a model of CASE technology that has three general dimensions: production, coordination and infrastructure. The FCTM appears to be a useful mechanism to assess the range of functionality available in a given design support environment. A more general issue relates to the implications of the model for studying the impact of CASE technology on I/S planning and design teams. Figure 2 provides one model that suggests how the CASE may result in a range of performance impacts.

**Figure 2**

**Impact of Technology on I/S Planning and Design**

At individual level, the production technology is hypothesized to directly impact the process efficiencies (e.g., one measure often used is function points/person/yea}
and product quality (e.g., one measure often used is number of defects/function point). As discussed in Section 2, these measures reflect a task perspective and may be associated with only a marginal impact on overall performance in the life cycle. One source of this limited impact may be reflected by the fact that current tools have limited analysis functionality. The tools evaluated in this research reflect potential for a broad coverage of representation functionality (17/18) and transformation (11/13). However, examination of these functions suggest a relatively passive design aid environment. That is, the functionality enables a designer to capture and present an idea or to transform a well defined design concept. Functionality to aid the critical thinking processes that often constitute a major contribution of the designer appear to be lacking. Thus, we might expect emerging functionality in this area to have a major impact on the efficiency and effectiveness of individuals.

At the team level, coordination technology can help to effect synergy among team members (or at least reduce the loss in productivity often associated with group interactions) and increase the validity of the product. Synergy might occur through both production efficiencies, (as measured by increased number of alternatives considered) and social/political impacts such as increased involvement of key organization roles (as measured by participation or influence in the design process). The potential for an increased validity arises from the ability of the design process to better meet an actual organizational need. This hypothesis argues that if coordination technology increases the ability of the team to effectively manage relationships with key stakeholders, this will increase the likelihood that a valid need (as perceived by the organization) is met.

An important interaction effect between the individuals and team level can also occur. The use of production technology may effectively empower a key organizational role or stakeholder by reducing the skill level or time required to participate and influence the design process. As such, production technology may have a significant impact in that it can change both the composition of the team and
the way in which roles on a team interrelate. Both of these impacts hold promise for significant performance improvement.

Finally at the organization level, the ability to use CASE technology to build an infrastructure could increase the flexibility of the product development process and enable the organization design products across teams to offer a significant performance impact. This potential arises in part from the potential to decentralize the knowledge necessary to coordinate the activities of multiple teams. Decentralizing the coordination knowledge requires individual teams to know or have access to information about goals, critical procedures and resource employed or required by a team (Durfee, et al. 1987). The potential for a CASE environment to provide access to such knowledge via shared design knowledge bases, through the use of standards design practices or by creating the means to time sharing key human resources across projects offers the potential for a major performance impact.

Of course, the ability to attribute performance impact to CASE technology becomes increasingly difficult as one moves from the individual unit of analysis to the organization. However, the ability to map usage behavior of the technology to both individual and team processes suggests the use of the FCTM may help to better understand the performance impact of CASE at these two levels. The functions reflecting an infrastructure dimension extend the model from the team to the organization and require further refinement before its potential explaining organization level impacts can be explored.
References


