WHY ARE SOME REPRESENTATIONS (*SOMETIMES*) MORE EFFECTIVE?

Jungpil Hahn
Carlson School of Management
University of Minnesota
U.S.A.

Jinwoo Kim
Department of Business Administration
Yonsei University
Korea

Abstract

Graphical representations are used heavily in systems analysis and design without proper verification of their usability. However, different representations have varying impacts on the effectiveness of systems analysts. The objective of this study was to explore the effects of diagrammatic representations on task behavior and performance. A cognitive model of an analysis and design task using diagrams was proposed, from which hypotheses about the effects of diagrammatic representation on task behavior and performance were derived. A laboratory experiment was conducted to test the hypotheses. Results of the experiment show that the representational features of the diagrams induced subjects to adopt different problem solving strategies, which resulted in different task performances.

Keywords: Information presentation, systems analysis, systems design, problem-solving behavior, experimental research, protocol analysis

1. INTRODUCTION

An important issue in information systems research is the development of effective systems development methodologies. The purpose of these methodologies is to assist the systems analysts in developing information systems efficiently (Pressman 1992). Typically, software development methodologies consist of procedures, graphical representation techniques, and tools to support the entire development process (Booch 1994; Jacobson 1995).

Of the various components of a systems development methodology, the graphical representation technique is an area in need of research because of the problem of the methodology jungle faced today (Avison and Fitzgerald 1988), we have an overwhelming number of representation techniques and as well as an overwhelming number of different versions within each technique. For example, there are dozens of object-oriented methodologies, each with different types of diagrams. For instance, the Unified Modeling Language (UML) provides three different types of process model diagrams that represent the same information but differ only in terms of their visual appearance (Rational 1997).

This situation yields interesting problems for both practitioners and researchers in information systems (Wynekoop and Russo 1993). Even though systems development methodologies typically provide a straightforward description of how to use their representation formalisms, using these for systems analysis and design is not as easy a task as it seems—errors often occur, and sometimes even systematically. Thus, practitioners have a wide selection of different representation formalisms to choose from, but do not have good criteria for selecting among the various options. Since most representation formalisms have been merely proposed and/or developed without being evaluated (Vessey and Conger 1994), practitioners cannot possibly know which one is more appropriate in a given situation. On the other hand, the researchers, who are continuously designing and upgrading new versions of representations, do not have adequate guidelines or principles to assist them in producing *good* designs (Tversky 1997). Therefore, we need to understand how and why different representation formalisms affect our behavior when we use these for systems analysis and design.

The objective of this paper is to investigate how and why differences in representation formalisms affect the systems analyst's task behavior and performance. The main theme of the paper is to present an empirically validated view of how people make use of different diagrammatic representations and how these different diagrammatic representations will affect our behavior and subsequently performance in a systems design task. Theories of information visualization and diagrammatic reasoning from cognitive science are examined in order to build a cognitive model of a systems design task; empirical results from an experimental study are presented to illustrate how different modeling formalisms lead to different behaviors.

This paper is organized as follows. Section two reviews prior research on diagrammatic knowledge representation. Section three presents a task analysis model of a diagrammatic problem solving task and discusses the effects of different representation on task behavior and performance. Section four describes our experimental research methodology. The results are discussed in section five. Finally, the paper concludes with implications of the study.

2. DIAGRAMMATIC KNOWLEDGE REPRESENTATION AND REASONING

Even though we are very adept at drawing and working with diagrams, representing knowledge via diagrams is not as trivial a task as it seems. This is especially true when we need to represent information that is inherently abstract, such as many of the constructs we need to represent in systems analysis and design. For example, a *process* is an intangible concept that cannot be touched or adequately visualized in terms of its inherent structure or form. The data flow diagram (DFD) formalism provides us with ellipses to artificially represent the abstract construct of processes. The use of such artificial substitutes is inevitable because a direct isomorphism does not exist between the real-world concept and its graphical representation (Gallistel 1990; Tversky 1997).

In order to represent abstract information, we need to decide how to partition the real-world knowledge into various graphical constructs and how to position the various constructs onto the presentation space so that it is intuitive (Engelhardt et al. 1996). Conceptually, these basic principles for representing abstract information fall into two main categories: information distribution (Tabachneck-Schijf, Leonard and Simon 1998) and spatial organization (Engelhardt et al. 1996). These rules comprise the visual grammar of a modeling formalism.

Information distribution refers to the level of the partitioning of information into meaningful graphical primitives (e.g., rectangles, lines, arrows, etc.), which form the elementary constructs of the modeling formalism. The various types of information may be represented at different levels of granularity. For example, the collaboration diagram (CD) of the UML models entities as boxes, while actions (messages) are represented as arrows where the direction from and to indicates the sending and receiving entities respectively (i.e., each construct has its own graphical representation), whereas the activity diagram (AD) represents actions and receiving entities within a rounded box and uses arrows to indicate the sequence of actions (i.e., multiple constructs are aggregated into one graphical construct). *Spatial organization* refers to the way meaning is conveyed through the layout within the presentation space. The presentation space can be divided into separate areas and graphical constructs assigned to a sub-area to express a categorization (Moher et al. 1993). For example, the AD of the UML partitions the presentation space into "swimlanes" that indicate the ownership of actions (i.e., spatial categorization), whereas the CD does not use such spatial organization.[1]

[1]Examples of the UML diagrams are provided in the appendix.

The above discussion concerning the visual grammar of the modeling formalism suggests that various combinations of information distribution and spatial organization may be applied in representing the same information.[2] Applying different grammatical rules will result in a different representation format of the information. Even though the different representations are informationally equivalent (i.e., represent the same content), they will not necessarily be computationally equivalent (i.e., be equally easy to use) (Larkin and Simon 1987) because different diagrammatic representations provide different perceptual cues that affect the amount of search effort that is required for problem solving with diagrams (Zhang 1997).

3. EFFECTS OF DIAGRAMMATIC REPRESENTATIONS

A diagram is *well represented* when the representation supports the cognitive processes in reasoning with the diagram (Kulpa 1994). A well-represented diagram is *effective* in presenting information in a way that makes it easy for humans to perceive and reason with (Mackinlay and Genesereth 1985). However, different diagrammatic representations will not necessarily be equally effective (i.e., computationally equivalent) because the representation affects the amount and effort of search required (Larkin and Simon 1987; Zhang 1997).

The effectiveness of a representation depends on the task performed (Benbasat and Dexter 1985; Tan and Benbasat 1993). For example, with charts, representing values as marks on a numerical scale is more effective for accurate value lookup, whereas representing the values by size of graphical entities might be more effective for general magnitude comparison (Kosslyn 1994). This paper investigates the task of integrating multiple business processes through the use of process diagrams.[3] In order to propose hypotheses on the effects of diagrammatic representation, we need to understand how the representation fits the cognitive activities needed to perform the task. We present a task analysis model of process integration using the GOMS model.[4]

3.1 Task Analysis Model of Process Integration

The GOMS model of process integration is presented in Figure 1. The problem solver begins the process integration task with the top level goal of integrating multiple individual processes into a single integrated process (`GOAL: INTEGRATE-PROCESS`). This is accomplished by integrating the individual diagrams that portray the subprocesses into a single diagram that illustrates the integrated process. The top level goal of integrating the processes is further partitioned into two subgoals: analyzing the individual processes (`GOAL: ANALYZE-PROCESS`) and designing the integrated process (`GOAL: DESIGN-PROCESS`).

The second level sub-goal of analyzing the process (`GOAL: ANALYZE-PROCESS`) is accomplished by reading the individual diagrams and integrating the information acquired from reading into an internal representation of the integrated process (`GOAL: READ-AND-INTEGRATE-DIAGRAMS`). The third level sub-goal of reading and integrating the diagrams can be achieved by means of two alternative methods. These methods are (1) first reading each individual diagram fully and then integrating them once all the information has been fully analyzed (`BATCH-METHOD`) and (2) integrating while reading the individual diagrams (`ALTERNATE-METHOD`). The `BATCH-METHOD` integrates the information presented only *after* having investigated all the information provided in the individual diagrams. For example, the problem solver will take a diagram to investigate, read all of the information presented in that diagram, take the other diagram, read all of the information presented there, and then try to

[2]As will be shown with the experimental treatments, most of the diagrams used in systems analysis and design methodologies are a combination of information distribution and spatial organization.

[3]This task was chosen because it involved both analysis and design activities that are hypothesized to be affected by the diagrammatic representation.

[4]The GOMS model describes rational human behavior in terms of a set of **G**oals and sub-goals, a set of cognitive **O**perators, a set of **M**ethods for achieving the goals, and a set of **S**election rules for choosing among competing methods for goals (Card, Moran and Newell 1983). The GOMS methodology was chosen for the task analysis because the diagram integration behavior can not be faithfully described as a sequence of perceptual, cognitive and motor activities as in procedural task analysis models, and has been widely accepted and used in the human-computer interaction (HCI) literature (Butler, Jacob and John 1998; John 1988; John and Kieras 1996).

```
GOAL: INTEGRATE-PROCESS
.  GOAL: ANALYZE-PROCESSS                              }
.  .  GOAL: READ-AND-INTEGRATE-DIAGRAMS                } Analysis
.  .  [select ALTERNATE-METHOD                         }
.  .          BATCH-METHOD]                            }
.  GOAL: DESIGN-PROCESS                                }
.  .  GOAL: DRAW-PROCESS-COMPONENTS                    }
.  .  [select GOAL: TEMPLATE-METHOD                    }
.  .               .  [select DRAW-SEPARATE            } Design
.  .               .          DRAW-BULK]               }
.  .          GOAL: SEQUENTIAL-METHOD                  }
.  .               .  [select DRAW-SEPARATE            }
.  .               .          DRAW-BULK]]              }
```

Figure 1. GOMS Model of Diagram Integration

integrate all of the information into a coherent internal representation of the integrated process. On the other hand, the `ALTERNATE-METHOD` involves a purposeful exploration among the information presented in the different diagrams *during* the process of reading the individual diagrams. For example, the problem solver will first take a diagram and read by following the sequence of process activities. When a reference to another diagram appears in that diagram, he/she will switch to the other diagram and search for the referent. Once the referent is identified, he/she will compare the two diagrams in order to infer the relative order of occurrence between the diagrams, then continue to follow the sequence of process activities until reference to the first diagram appears. Such a process of following process activities and switching diagrams will continue until all the information in all the diagrams has been read.

The other second level sub-goal of designing the integrated process (`GOAL: DESIGN-PROCESS`) consists of externalizing the internal representation of the process constructed from the analysis phase into a valid diagrammatic representation (`GOAL: DRAW-PROCESS-COMPONENTS`). Externalizing the internal representation of the integrated process is basically achieved by drawing all of the components of the process model correctly. Normally, the drawing activity does not proceed mechanically from the top of the page to the bottom without regard to the semantics of what is being drawn. The drawing activity is a more meaningful process of illustrating the objects of interest in an organized manner. In the case of process diagrams, the drawing activity will be a sequence of specifying the components of the process model—the entities and actions performed—in the order of occurrence of the actual process activities. This drawing activity can take the form of (1) drawing the components of the process model in a strictly sequential manner (`SEQUENTIAL-METHOD`) or (2) building a template of all the entities first and then specifying the actions performed in relation to the template already drawn (`TEMPLATE-METHOD`). Furthermore, different methods may apply in drawing the sequence of activities – the components for each process activity (i.e., sender, receiver and action) can be (1) specified sequentially as separate graphical tokens (`DRAW-SEPARATE-METHOD`) or (2) expressed together as aggregated graphical tokens (`DRAW-BULK-METHOD`) depending on the visual grammar of the modeling formalism.

The above task analysis model of process integration through diagram casts important insights as to how different diagrammatic representations can have an effect on the cognitive behavior of systems analysts/ designers. Basically, successful diagram integration will depend upon which methods are selected in order to achieve the lower level sub-goals of analyzing the individual diagrams (`READ-AND-INTEGRATE-DIAGRAMS`) and of designing the integrated diagrams (`DRAW-PROCESS-COMPONENTS`). This paper argues that the selection rules may be guided by the diagrammatic representation of the process diagrams.

3.2 Information Distribution

The perceptual explicitness arising from information distribution is expected to provide effective visual cues that trigger the recognition of relevant information from different diagrams (Green, Petre and Bellamy 1991). Once such recognition is triggered, the problem solver will be induced into switching diagrams in order to relate and integrate the relevant information among different diagrams (Narayanan, Suwa and Motoda 1995). Therefore, distribution of the process components will induce the problem solver to apply the `ALTERNATE-METHOD` for reading and integrating the processes. Otherwise, if distribution is not supported, the problem solver will apply the `BATCH-METHOD` because, without visual cues, she can hardly recognize the point to switch diagrams. Information distribution will also have an effect in the design phase of the process integration task. If distribution is supported, then the components of the process model can be specified in sequence (`DRAW-SEPARATE-METHOD`), whereas the components need to be specified together if distribution is not supported (`DRAW-BULK-METHOD`).

The effect of information distribution on integration behavior is also hypothesized to affect integration performance. In the analysis phase, the `ALTERNATE-METHOD` will have a higher potential for successful integration than the `BATCH-METHOD`. Since the problem solver does not know a priori how the diagrams of the individual sub-processes should be integrated, deliberately attempting to integrate *while* reading the diagrams will help the problem solver make the critical inferences necessary for successful integration, such as where the process starts and where the different diagrams meet. This information will be attended to and thus located in working memory while reading and integrating the individual diagrams. Furthermore, information distribution will enable visual mapping[5] of the process components that will provide effective perceptual inferences to assist the integration process. Visual mapping only requires recognition whereas conceptual mapping requires search and inference (Smith 1995). Visual mapping of the process components should facilitate the identification of the interface between the diagrams since the interface is represented as two elements present in both diagrams that convey the same meaning (i.e., an output in one diagram would be associated to the input in another diagram). However, if distribution is not supported, the same information may not look similar visually because of the different perspectives between the diagrams. Therefore, conceptual mapping, instead of visual mapping, will be required. Therefore, the `BATCH-METHOD` will have a lower probability of successful integration because this method requires conceptual mapping and a vast amount of information to be retained in working memory in order to induce the critical inferences (Baddeley 1986). Thus,

H-A1: Diagrams with information distribution support will result in higher analysis performance in terms of successful integration than those without information distribution support.

H-A1': Diagrams with information distribution support will result in higher analysis performance in terms of fewer analysis errors committed than those without information distribution support.

In the design phase, the specification of the activities will also have a higher potential for success if components can be specified separately, one at a time, as individual graphical tokens (`DRAW-SEPARATE-METHOD`) rather than in bulk (`DRAW-BULK-METHOD`). With the `SEPARATE-METHOD`, components of the process model have to be separately and explicitly specified. The explicit specification acts as an external constraint on the design activity (Zhang and Norman 1994) and makes the omission of process components highly improbable, whereas with the `BULK-METHOD` these constraints no longer apply and the problem solver is more prone to omit some of the components. Thus,

H-D1: Diagrams with information distribution support will result in higher design performance in terms of successful integration than those without information distribution support.

H-D1': Diagrams with information distribution support will result in higher design performance in terms of fewer design errors committed than those without information distribution support.

3.3 Spatial Organization

If some elements from the components of the process model can be spatially organized, then these will be represented in advance. Therefore, the possibility of spatial organization will induce the problem solver to apply the `TEMPLATE-METHOD`. Otherwise, if the diagrammatic representation does not support spatial organization, the problem solver will follow the `SEQUENTIAL-METHOD` for specifying the process activities and relevant entities because separate space divisions cannot be assigned to entities or activities a priori.

In the design phase, the `TEMPLATE-METHOD` will be easier than the `SEQUENTIAL-METHOD` in terms of processing load and memory load. Processing and memory load will be reduced since the problem solver will not have to keep track of all the components involved in each process activity. This is because many components are already drawn in the template and can easily be identified with the `TEMPLATE-METHOD` (Kalyuga, Chandler and Sweller 1997), thus the probability of omitting necessary components is reduced. On the other hand, the `SEQUENTIAL-METHOD` forces the problem solver to consider all the components

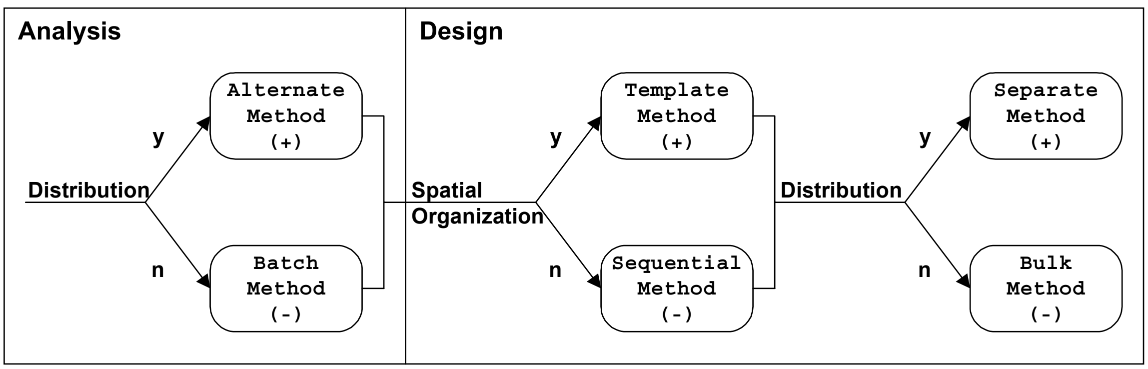
[5]Visual mapping refers to a perceptual inference that two visual elements are the same, whereas conceptual mapping refers to relating two elements by comparing the semantic meaning (Narayanan, Suwa and Motoda 1995).

involved in each process activity to draw the entire process, which increases the probability of omitting important components. Thus,

H-D2: Diagrams with spatial organization support will result in higher design performance in terms of successful integration than those without spatial organization support.

H-D2': Diagrams with spatial organization support will result in higher design performance in terms of fewer design errors committed than those without spatial organization support.

In summary, how information is distributed and spatially organized is hypothesized to affect the selection of methods to perform analysis and design activities as well as performance, since different methods applied have different potential for successful completion. The effects of diagrammatic representation on analysis and design behavior and performance are summarized in Figure 2.



Note: The rounded boxes represent hypothesized analysis and design methods. The (+) and (-) within the selected methods indicate hypothesized relative performance.

Figure 2. Effects of Diagrammatic Representation

4. RESEARCH METHODOLOGY

An experiment was conducted to explore the effects of information distribution and spatial organization on the task behavior and performance of a process integration task. The details follow.

4.1 Subjects

A total of 32 students participated in the experiment for course credit. All subjects were junior or senior undergraduate students enrolled in a systems analysis and design course. The experiment was conducted near the end of the course, by which time all of the subjects had completed several assignments and were comfortable with the modeling formalism used as the experimental stimuli.

4.2 Experimental Design

A 4x4 analysis of variance design (Latin Square Confounded Factorial Design: LSCF 8-4^2) was used as the experimental design (Kirk 1995). Each subject performed four process integration tasks with four different system processes, each modeled with different diagrammatic representations. The sequence in which the diagrams were presented was completely randomized. Ultimately, eight subjects performed each process-diagram combination.

4.3 Procedure

Each experimental session was divided into four sections. First, subjects were given instructions about the general nature of the experiment and that verbal and action protocols would be collected. Second, the subjects were trained to think aloud with several traditional training tasks until they were comfortable (Ericsson and Simon 1993). The experimenter then presented the subjects with a one-page overview of the task. Finally, the subjects were provided with two diagrams for each process integration task.

4.4 Experimental Material

For each task, the subjects were provided with two diagrams on paper depicting a single process modeled from two different systems' perspectives: a company's headquarters and its branch. Each diagram depicted a portion of the whole process with the activities within the other system encapsulated. Both diagrams for each integration task were modeled with the same representation formalism. The subjects were asked to understand the given process as a whole and to integrate the two diagrams by drawing the integrated process into a single diagram with the same representation formalism.

The diagrams used in modeling the system processes were four kinds of process model diagrams[6]: the collaboration diagram (CD), the sequence diagram (SD), the activity diagram (AD) and the activity flow diagram (AFD).[7] The diagrammatic representations differed only with respect to how they distribute and spatially organize the process components (Table 1).

Table 1. Visual Language of the Experimental Diagrams

		Information Distribution	
		Yes	No
Spatial Organization	Yes	Sequence Diagram (SD)	Activity Diagram (AD)
	No	Collaboration Diagram (CD)	Activity Flow Diagram (AFD)

4.5 Hypotheses for Diagram Integration Performance

As discussed in the task analysis model of process integration, diagrammatic representation treatment is hypothesized to have an effect on the task performance. The model of diagram integration predicts that information distribution will support both analysis and design activities, whereas spatial organization will only support the design activity. The prediction is that the subjects who are given the SD and CD will perform better than those with the AD and AFD both in the analysis and design phases (H-A1, H-A1', H-D1 and H-D1'). On the other hand, the prediction is that those subjects given the SD and AD will work better than those with the CD and AFD only in the design phase (H-D2 and H-D2').

---

[6]Examples of the diagrams are provided in the appendix.

[7]Three of the diagrams were adopted from an existing object-oriented software engineering methodology—the Unified Modeling Language (Rational 1997), whereas one (i.e., AFD) was newly designed for the purpose of the experiment to prepare a diagram treatment without information distribution and spatial organization.

4.6 Data Analysis Strategy

The subjects' performance and behavior data were collected from the experiment: performance data was collected from the subjects' final integrated diagrams and behavior data from the subjects' concurrent verbal and action protocols.

The data analysis consisted of two steps. First, the subjects' task performance was compared between treatments to see if different representations incurred different performance. Then, in case of significant differences, the subjects' behavior data would be analyzed to examine why the difference in performance was observed.

4.6.1 Performance Data

The purpose of the performance data analysis was to examine whether the experimental treatment had an effect on the subjects' task performance. Each subject's final designs were evaluated independently by two coders. The evaluation consisted of counting the number of analysis and design errors present in the integrated process diagram. The analysis errors were concerned with the subjects' inability to construct a correct internal representation of the integrated process. Therefore, the incorrect design of the process sequence was coded as an analysis error. The design errors concerned the subject's inability to correctly follow the modeling formalism. Intercoder reliability was assessed with the kappa ratio[8] (Van Someren, Barnard and Sandberg 1994).

4.6.2 Integration Behavior Data

In order to understand how the different diagrammatic representations affected the subjects' analysis and design behaviors, protocol analysis with verbal and action protocols was performed to trace the subjects' cognitive processes (Ericsson and Simon 1993; Van Someren, Barnard and Sandberg 1994).

The behavior data in the analysis phase was coded based on the subjects' verbal protocols. The analysis data was concerned with which part of the diagram the subjects were investigating at a given time. All of the information presented in the diagrams was indexed and the verbal protocols were coded accordingly. The subjects' analysis behaviors were categorized as applying the `ALTERNATE-METHOD` if the subjects showed explicit transitions to another diagram before all of the information on the first diagram was fully analyzed. Otherwise, their behavior was categorized as `BATCH-METHOD` (i.e., the subjects' showed transition only after having fully analyzed all of the information in the first diagram).

The behavior data in the design phase was coded based on the subjects' action protocols. The design data was concerned with which component of the process model the subjects were drawing at a given time. All of the information from the diagrams was indexed and the action protocols were coded accordingly. The subjects' design behavior was categorized as applying either the `TEMPLATE-METHOD`, if the subjects drew all the entities before specifying any process activities, or `SEQUENTIAL-METHOD`, if the subjects specified the relevant entities and actions for each process activity in sequence. Furthermore, if the remaining components were specified in sequence, the design behavior was coded as `SEPARATE-METHOD`. Otherwise, if they were specified in bulk, the behavior was coded as `BULK-METHOD`.

5. RESULTS

5.1 Performance Results

The subjects' task performance is summarized in Table 2, which shows the number of subjects who made errors vs. the number of subjects whose overall integration was correct as well as the average number of errors committed during both the analysis and the design phases of the task. The results show that the group that used those diagrams supporting information distribution (SD

[8]The results of the reliability test show that the two independent codings were consistent (kappa = 0.937).

and CD) had more correct subjects than those with diagrams not supporting distribution (AD and AFD) for both analysis and design phases (analysis: $CMH = 6.79$, $df = 1$, $p < 0.01$; design: $CMH = 45.11$, $df = 1$, $p < 0.001$) (H-A1 and H-D1 supported), and that the group that used diagrams supporting spatial organization (SD and AD) had more correct subjects than those with diagrams not supporting spatial organization (CD and AFD) only in the design phase ($CMH = 2.75$, $df = 1$, $p < 0.1$)[9] (H-D2 supported).

Similar effects were obtained comparing the average number of errors between groups (analysis: $F(3,39) = 4.04$, $p < 0.05$; design: $F(3,39) = 53.46$, $p < 0.0001$). When the average analysis error was compared with contrasts of information distribution and spatial organization, we could see that distribution had an effect on the number of analysis errors ($F(1,39) = 11.24$, $p < 0.005$) (H-A1' supported), while spatial organization did not have a significant effect ($F(1,39) = 0.18$, ns). Information distribution also had an effect on the number of design errors ($F(1,39) = 131.61$, $p < 0.0001$) (H-D1' supported), and design performance was also affected by spatial organization ($F(1,39) = 22.51$, $p < 0.0001$) (H-D2' supported).

The above results suggest that diagrammatic representations supporting information distribution resulted in fewer analysis errors than those not supporting information distribution, and those supporting distribution and spatial organization resulted in fewer design errors than those not supporting them.

These results of the subjects' analysis and design performance confirm our hypotheses on task performance. The next section presents the results of the protocol analysis of the subjects' behavior in an attempt to understand how the information distribution and spatial organization affected the actual task behavior, resulting in different analysis and design performance.

Table 2. Task Performance Results

Information Distribution	Spatial Organization	Treatment	Analysis		Design	
			Error/Correct	Average	Error/Correct	Average
Yes	Yes	SD	3 / 29	0.0938	11 / 21	0.531
Yes	No	CD	2 / 30	0.0625	16 / 16	0.938
No	Yes	AD	8 / 24	0.2813	30 / 2	2.156
No	No	AFD	8 / 24	0.3750	32 / 0	3.468

5.2 Process Results[10]

5.2.1 Behavior in the Analysis Phase

The subjects' behavior in the analysis phase based on their verbal protocols is summarized in Table 3. The GOMS model predicted a correct selection of the methods in the analysis phase for 75% of the representative subjects. The results indicate that information distribution resulted in more subjects applying the `ALTERNATE-METHOD` rather than the `BATCH-METHOD` ($CMH = 4.00$, $df=1$, $p < 0.05$).

---

[9]The Cochran-Mantel-Haenszel statistic was used to test the independence of the observed frequencies of errors for the 2x2x2 contingency table (Agresti 1996).

[10]Although comprehensive analysis of the verbal and action protocols most accurately shows the dynamic nature of the diagram integration behavior, it is impractical to analyze all of the data in detail for all of the subjects because of the tremendous amount of protocol data. Therefore, for each diagram, four representative subjects were chosen for detailed protocol analysis. For an objective selection of representative subjects, the average error rate was calculated for each process-diagram treatment and the subject closest to the mean was selected as the representative subject for that treatment.

Table 3. Method Selection for Read and Integrate Diagrams

Distribution	Diagram	Hypothesized Selection	Correct	False	Prediction (%)
Yes	SD	ALTERNATE	4	0	100
	CD	ALTERNATE	3	1	75
No	AD	BATCH	2	2	50
	AFD	BATCH	3	1	75
	Total		12	4	7

5.2.2 *Behavior in the Design Phase*

Table 4 shows the methods selected for designing the integrated process. The model predicted a correct selection of the methods in the design phase for 75% of the subjects. The results indicate that information distribution resulted in more subjects applying the `TEMPLATE-METHOD` rather than the `SEQUENTIAL-METHOD` ($CMH = 3.73$, $df = 1$, $p < 0.1$).

In summary, the results of the protocol analysis indicate that subjects did (in most cases) perform the methods predicted by our task analysis model. In other words, the diagrammatic representation, in terms of information distribution and spatial organization, induced subjects into applying the hypothesized methods. This could account for the observed difference in task performance.

Table 4. Method Selection for Design Integrated Business Process

Spatial Organization	Information Distribution	Diagram Treatment	Hypothesized Selection	Correct	False	Prediction (%)
Yes	Yes	SD	TEMPLATE SEPARATE	4	0	100
Yes	No	AD	TEMPLATE BULK	2	2	50
No	Yes	CD	SEQUENTIAL SEPARATE	3	1	75
No	No	AFD	SEQUENTIAL BULK	3	1	75
		Total		12	4	75

6. CONCLUSION AND DISCUSSION

6.1 Summary of the Results

The results of the experiment indicate that diagrammatic representation does have an effect on task performance. As hypothesized by the cognitive model of diagram integration, information distribution reduced the number of errors in the analysis and design activities. Spatial organization also had a positive effect on the design activities in that design errors were reduced.

Detailed protocol analysis of the representative subjects suggests how and why information distribution and spatial organization might have affected the task performance. First, in terms of *information distribution*, the subjects' analysis behaviors show that, when the information was distributed, the subjects undertook an alternating integration behavior which had a higher potential for success than the non-alternating batch behavior. The perceptual explicitness arising from the distributed information representation was expected to provide effective visual cues to alternate between different diagrams to look for relevant information in the other diagram (Narayanan, Suwa and Motoda 1995). This alternating behavior is expected to have reduced working memory load because the critical information in integrating the diagrams was more likely to be readily available as a result of the alternation. At the same time, the subjects' design behaviors show that information distribution played an important role in keeping track of important pieces while drawing the integrated diagram. Information distribution is hypothesized to have forced the subjects into explicitly specifying all the pieces, making it hard to omit important information. When distribution was not supported, the subjects committed more design errors by omitting certain components in their designs. Second, in terms of *spatial organization*, the design behaviors show that the subjects were induced into specifying the spatially organized components first as a basic template and then completing the remaining process components later. We infer that the basic template provided effective external memory aids for keeping track of necessary components that reduced unnecessary errors.

In summary, the diagrammatic grammar of information distribution and spatial organization induced subjects into performing different methods of analysis and design. The selection of different methods resulted in different task performance.

6.2 Discussion

On the whole, the results of the study showed that how knowledge is represented in diagrams has an effect on how we use these diagrams for problem solving. However, we did observe some results that were somewhat less conclusive (e.g., low prediction percentage for AD). Several factors may have produced these results. First, since the sender entities were organized as swimlanes in the AD, the specification of the senders was separated, but the action and receiver components remained aggregated, thus causing a partial distribution of the process components. The partial nature of the distribution may have caused the subjects to choose either method for analysis and design. Second, other diagrammatic factors such as color coding of the graphical primitives and spatial orientation have been found to influence reasoning with diagrams (Kalyuga, Chandler and Sweller 1997). Thus, diagrammatic factors other than information distribution and spatial organization investigated in this paper may also have influenced the subjects' problem solving behaviors. For example, an unanticipated consequence of the visual grammar can make some of the diagrams more horizontally or vertically oriented, whereas others do not have any specific orientation. Such factors would affect the subjects' problem solving behavior, since they treat the time dimension differently, which is a very important factor in process diagrams. In addition, diagrams with the same vertical or horizontal orientation may become very similar (e.g., AD and SD). Such visual similarity arising from the salience of orientation may also affect the subjects' behavior. These factors were not controlled in our experiment. However, the effects of these additional factors should be investigated in future research on visual representations.

Despite the above limitations, this study has both strong theoretical and practical implications. Theoretically, this research not only confirms but also extends the current line of research in diagrammatic reasoning. Previous research on diagrammatic reasoning focused on the advantages of the diagrammatic representation over propositional representation (e.g., text) (Larkin and Simon 1987), or on the cognitive processes of reasoning with one type of representation. This research extends the current framework of diagrammatic reasoning by comparing the computational efficiencies *between* different diagrams based on the visual grammar of the representation. This paper identified important diagrammatic representation factors (i.e., information distribution and spatial organization) and proposed a cognitive model of a process integration task, which was useful for hypothesizing about the effects of different representations.

In addition, this research did not compare diagrams, per se, but design dimensions for diagrammatic representations. This is an important distinction between this study and previous research on information presentation, which have, in most cases, compared final representations (e.g., charts and tables). The focus on design dimensions enables us to investigate how the design of the modeling representation will affect our problem solving behavior and performance (Green 1989; Green and Petre 1996). A complete taxonomy of representation design dimensions will also prove useful in the design of new representation formalisms by combining elementary design dimensions. It is our belief that future research on information presentation should continue to focus on design principles to scrutinize the effects on behavior and performance and to explore new innovative ways of representing knowledge.

Practical contributions of this research hold for users and designers of diagrammatic representations in systems analysis and design. As mentioned in the introductory section, widely used systems analysis and design methodologies such as the UML provide a wide variety of different modeling formalisms that represent the same information in a diagrammatically different form and organization. The systems analysts—the users who work with these diagrams—will be able to select particular diagrammatic representations based on the task at hand (Good 1996; Green et al. 1991; Vessey and Galletta 1991). Some tasks may be easier to perform or less error-prone with a particular diagrammatic representation. As seen in our study, the use of less effective representations may result in more analysis and design errors. Thus, the ability to select the appropriate diagrammatic representation will be of great assistance to the systems analysts. Furthermore, the results of this study may be applied in designing the training material for using the systems analysis and design methodologies. The users should not only be taught how to use the different diagrams but also how different diagrammatic representations might systematically cause errors and faults in using these diagrams so as to foster an awareness that will help reduce analysis and design errors. From the designer's perspective, the findings of this research may be used to develop guidelines for designing diagrams that are cognitively compelling (Tversky 1997). These guidelines may be used in the design of diagrams in systems analysis and design methodologies and also in CASE (Computer-Aided Software Engineering) tools (Vessey and Glass 1998).

7. REFERENCES

Agresti, A. *An Introduction to Categorical Data Analysis*, New York: John Wiley & Sons, 1996.

Avison, D. E., and Fitzgerald, G. *Information Systems Development Methodologies, Techniques and Tools*, Oxford, England: Blackwell Scientific Publications, 1988.

Baddeley, A. D. *Working Memory*, Oxford, England: Clarendon, 1986.

Benbasat, I., and Dexter, A. S. "An Experimental Evaluation of Graphical and Color-Enhanced Information Presentation," *Management Science* (31:11), 1985, pp. 1348-1364.

Booch, G. *Object-Oriented Analysis and Design with Applications*, Redwood City, CA: Benjamin/Cummings Publishing Company, 1994.

Butler, K. A.; Jacob, R. J. K.; and John, B. E. "Human-Computer Interaction: Introduction and Overview," *Proceedings of the CHI'98 Human Factors in Computing Systems*, 1998, pp. 105-106.

Card, S. K.; Moran, T. P.; and Newell, A. *The Psychology of Human Computer Interaction*, Hillsdale, NJ: Lawrence Erlbaum Associates, 1983.

Engelhardt, Y.; de Bruin, J.; Janssen, T.; and Scha, R. "The Visual Grammar of Information Graphics," *Proceedings of the Artificial Intelligence in Design (AID'96), Workshop on Visual Representation, Reasoning and Interaction in Design*, Stanford University, Stanford, CA, 1996.

Ericsson, K. A., and Simon, H. A. *Protocol Analysis: Verbal Reports as Data*, Cambridge, MA: The MIT Press, 1993.

Gallistel, C. *The Organization of Learning*, Cambridge, MA: The MIT Press, 1990.

Good, J. "The 'Right' Tool for the Task: An Investigation of External Representations, Program Abstractions and Task Requirements," in *Empirical Studies of Programmers: Sixth Workshop*, W. D. Gray and D. A. Boehm-Davis (eds.), Norwood, NJ: Ablex Publishing, 1996, pp. 77-98.

Green, T. R. G. "Cognitive Dimensions of Notations," in *People and Computers V*, A. Sutcliffe and L. Macaulay (eds.), Cambridge, England: Cambridge University Press, 1989, pp. 443-460.

Green, T. R. G., and Petre, M. "Usability Analysis of Visual Programming Environments: A 'Cognitive Dimensions' Framework," *Journal of Visual Languages and Computing* (7), 1996, pp. 131-174.

Green, T. R. G.; Petre, M.; and Bellamy, R. K. E. "Comprehensibility of Visual and Textual Programs: A Test of Superlativism Against the 'Match-Mismatch' Conjecture," in *Empirical Studies of Programmers: Fourth Workshop*, J. Koenemann-Belliveau, T. G. Moher and S. P. Robertson (eds.), Norwood, NJ: Ablex Publishing, 1991, pp. 121-146.

Jacobson, I. *The Object Advantage: Business Process Reengineering with Object Technology*, Reading, MA: Addison Wesley, 1995.

John, B. E. *Contributions to Engineering Models of Human-Computer Interaction*, Unpublished Doctoral Dissertation, Carnegie Mellon University, 1988.

John, B. E., and Kieras, D. E. "The GOMS Family of User Interface Analysis Techniques: Comparison and Contrast," *ACM Transactions on Computer-Human Interaction* (3:4), 1996, pp. 320-351.

Kalyuga, S.; Chandler, P.; and Sweller, J. "Levels of Expertise and User-Adapted Formats of Instructional Presentations: A Cognitive Load Approach," *Proceedings of the User Modeling: Proceedings of the Sixth International Conference (UM 97)*, 1997, pp. 261-272.

Kirk, R. E. *Experimental Design: Procedures for the Behavioral Sciences*, Pacific Grove, CA: Brooks/Cole Publishing Company, 1995.

Kosslyn, S. M. *Elements of Graph Design*, New York: W. H. Freeman and Company, 1994.

Kulpa, Z. "Diagrammatic Representation and Reasoning," *Machine Graphics and Vision* (3:1/2), 1994, pp. 77-103.

Larkin, J., and Simon, H. A. "Why a Diagram is (Sometimes) Worth Ten Thousand Words," *Cognitive Science* (11), 1987, pp. 65-99.

Mackinlay, J., and Genesereth, M. R. "Expressiveness and Language Choice," *Data and Knowledge Engineering* (1), 1985, pp. 17-29.

Moher, T. G.; Mak, D. C.; Blumenthal, B.; and Leventhal, L. M. "Comparing the Comprehensibility of Textual and Graphical Programs: The Case of Petri Nets," in *Empirical Studies of Programmers: Fifth Workshop*, C. R. Cook, J. C. Scholtz and J. C. Spohrer (eds.), Norwood, NJ: Ablex Publishing, 1993, pp. 137-161.

Narayanan, N. H.; Suwa, M.; and Motoda, H. "Hypothesizing Behaviors from Device Diagrams," in *Diagrammatic Reasoning: Cognitive and Computational Perspectives*, J. Glasgow, N. H. Narayanan and B. Chandrasekaran (eds.), Menlo Park, CA: AAAI Press, 1995, pp. 501-534.

Pressman, R. *Software Engineering: A Practitioner's Approach*, New York: McGraw-Hill, 1992.

Rational. "The Unified Modeling Language ver. 1.0," Rational Software Corporation, 1997.

Smith, E. E. "Concepts and Categorization," in *Thinking: An Invitation to Cognitive Science* (3), D. N. Osherson and E. E. Smith (eds.), Cambridge, MA: MIT Press, 1995, pp. 3-33.

Tabachneck-Schijf, H. J. M.; Leonard, A. M.; and Simon, H. A. "CaMeRa: A Computational Model of Multiple Representations," *Cognitive Science* (21:3), 1998, pp. 305-350.

Tan, J. K. H., and Benbasat, I. "The Effectiveness of Graphical Presentation for Information Extraction: A Cumulative Experimental Approach," *Decision Sciences* (24), 1993, pp. 167-191.

Tversky, B. "Cognitive Principles of Graphic Displays," *Proceedings of the AAAI 1997 Fall Symposium on Reasoning with Diagrammatic Representations*, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1997, pp. 116-124.

Van Someren, M. W.; Barnard, Y. F.; and Sandberg, J. A. C. *The Think Aloud Method: A Practical Guide to Modeling Cognitive Processes*, San Diego, CA: Academic Press, 1994.

Vessey, I., and Conger, S. A. "Requirements Specification: Learning Objects, Process, and Data Methodologies," *Communications of the ACM* (37:5), 1994, pp. 102-112.

Vessey, I., and Galletta, D. "Cognitive Fit: An Empirical Study of Information Acquisition," *Information Systems Research* (2:1), 1991, pp. 63-84.

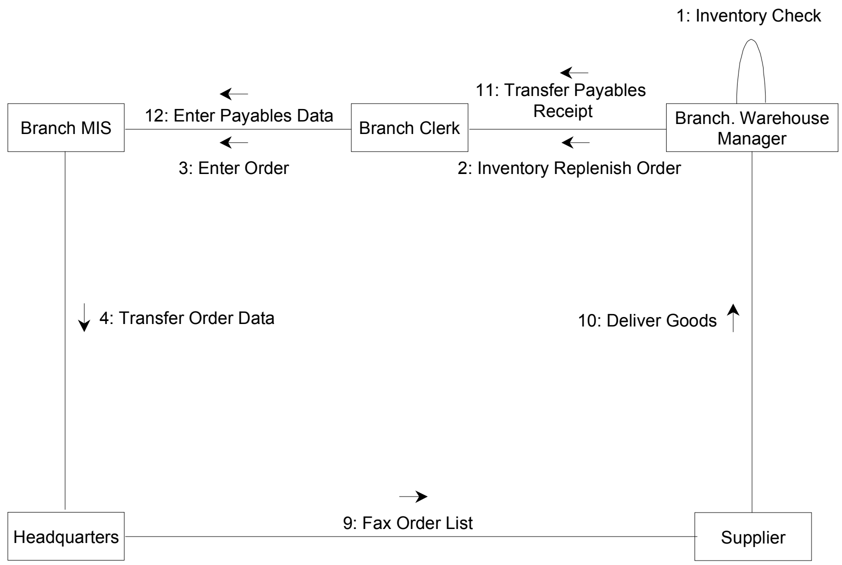
Vessey, I., and Glass, R. "Strong vs. Weak Approaches to Systems Development," *Communications of the ACM* (41:4), 1998, pp. 99-102.

Wynekoop, J. L., and Russo, N. L. "System Development Methodologies: Unanswered Questions and the Research-Practice Gap," *Proceedings of the Proceedings of the Fifteenth Annual International Conference on Information Systems*, J. I. DeGross, R. P. Bostrom, and D. Robey (eds.), Orlando, Florida, 1993, pp. 181-190.

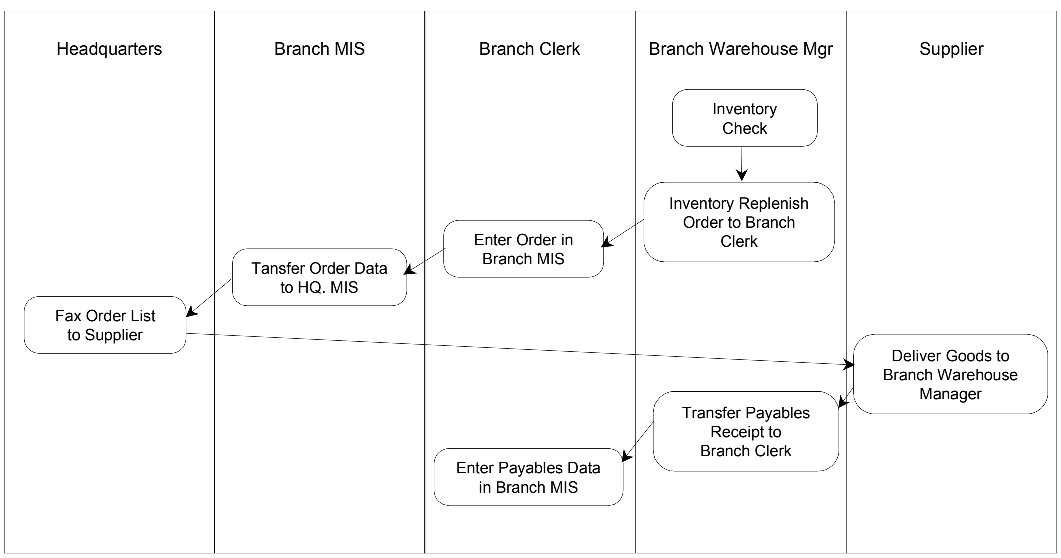
Zhang, J. "The Nature of External Representations in Problem Solving," *Cognitive Science* (21:2), 1997, pp. 179-217.

Zhang, J., and Norman, D. A. "Representations in Distributed Cognitive Tasks," *Cognitive Science* (18), 1994, pp. 87-122.

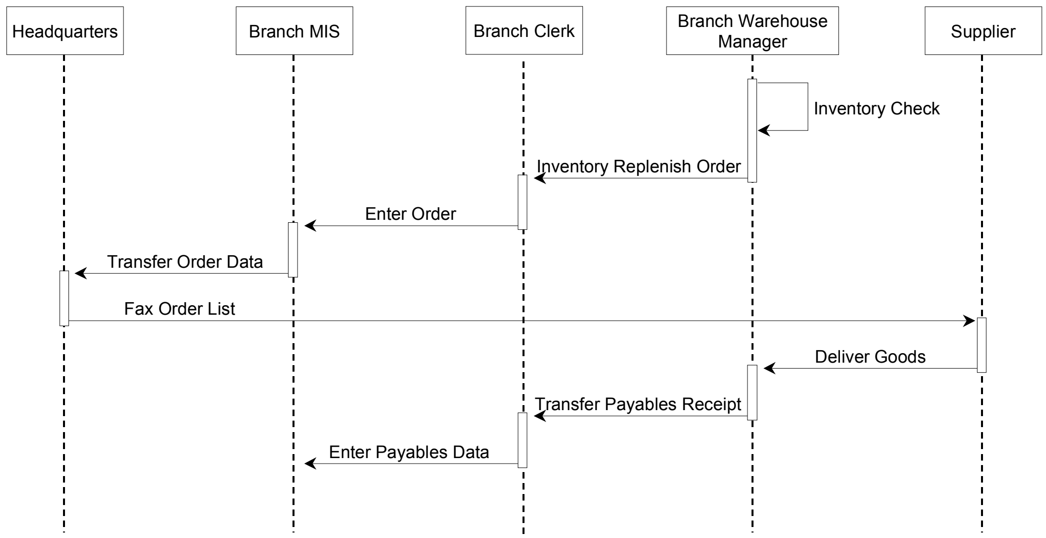
Appendix



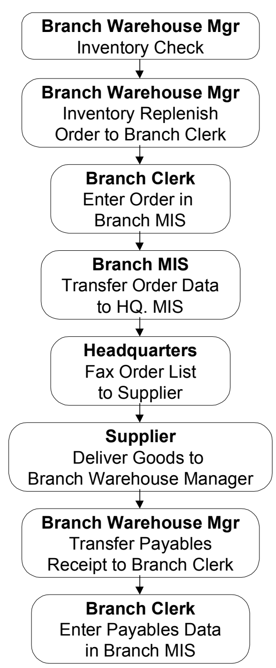
Example of the Collaboration Diagram (CD)



Example of the Activity Diagram (AD)



Example of the Sequence Diagram (SD)



Example of the Activity Flow Diagram (AFD)