Information & Management 22 (1992) 29–40
North-Holland

Research

An experimental comparison of abstract and concrete representations in systems analysis

Tor J. Larsen
Norwegian School of Management, 1300 Sandvika, Norway

Justus D. Naumann
Carlson School of Management, University of Minnesota, Minneapolis, MN 55455, USA

Keywords: Systems analysis, Requirements determination, Representation, User knowledge, Analyst knowledge, Analyst–user communication.

The process of information requirements determination requires effective communication between systems analysts and users of the system to be developed. The analyst's ability to discover user requirements is partially determined by the analyst's familiarity with and ability to communicate in the user's domain of knowledge and discourse. One such aspect of the user knowledge domain is concrete terminology versus more abstract, conceptual understanding. This paper documents the results of an experiment which compared knowledge representation used by analysts in a systems development discovery task. We hypothesized that the discovery task would be more effective when the analyst's representation was biased toward the concrete. We found that systems analysts whose initial representation was a physical data flow diagram (concrete) made more correct modifications and fewer errors than systems analysts who started with a logical data flow diagram (abstract). The two groups used the same amount of time for each of the sub-tasks. These results indicate that analyst knowledge and use of concrete terms in the user knowledge domain is of more utility in the discovery task than abstract, conceptual domain knowledge.

Justus D. Naumann is Associate Professor of MIS at the Carlson School, University of Minnesota. He has published articles in *Communications of the ACM, Decision Sciences, MIS Quarterly, Information & Management,* and other journals. He teaches software development and telecommunications. His research focuses on the process of system development, especially issues of requirements analysis and system design.

1. Introduction

A crucial prerequisite of a successful computer application is complete and correct systems requirements elicited from the users. These are the basis of all subsequent development. Requirements determination is, therefore, a crucial first step in the process.

On a general level, user requirements are transformed into a *representation* (for example, data flow diagrams with data dictionary, layouts, etc.), which is the initial documentation of the system as it evolves into a complete and accurate description. Users and systems analysts interact in requirements definitions and validation tasks.

The characteristics of the knowledge that forms the evolving representation are one of the many factors that may determine the quality of the communication exchange and therefore the completeness and accuracy of the representation. Experiments in psychology and communications demonstrate that, in some settings, concrete is better than abstract data for understanding. We assume this finding to be significant to the discovery of user requirements. The experiment described in this paper tested the effects of such differences in the requirements determination task.

Tor J. Larsen is Associate Professor at the Norwegian School of Management, Sandvika, Norway. He lectures in the management of information systems and technology based innovation. At present, he is undertaking research on managers use of information technology and innovation, in collaboration with major Norwegian manufacturing companies. He earned his Ph.D. from the University of Minnesota in 1989.

2. Background

The framework for this experiment has been termed the "Representation Model." It may be defined as the outcome of a purposeful mental, linguistic, and social interaction of analyst(s) with user(s) in the process of discovering and validating requirements and transforming them into verifiable system specifications.

Models or representations are central to the development of systems. Representations have been a major focus of attention in data modelling since Herman Hollerith and the 1880 census. Card, report and record layout forms have been employed routinely to communicate information about data. Computer technological development has permitted manipulation of increasingly complex data structures. Flow charts, data flow diagrams, and many other graphical constructs have been devised and used to depict organizations, mechanisms, and processes, and their patterns of intercommunication. CASE tools are now emerging to support them.

Representations may be used as both communication media and reasoning tools. In the simplest case, a representation communicates certain facts among individuals sharing an interest. *Figure 1* shows the role of representations in the process of systems analysis and specification. A representation is central to the process at least in the following ways:

(1) It is initially used by the analyst as a template. Much of the task is discovery of user knowledge that completes a particular template. The Whorfian hypothesis, "language determines thought," suggests that knowledge not required by a representation will not be discovered -- indeed, questions that might elicit such knowledge may not even be generated. Additionally, the user may offer information that cannot be included in the representation because of its syntax and semantics.

(2) During the analysis process, and especially at its completion, the representation must be validated. In information systems development, representation-based validation is a critical process. Only the representation is available for validation until system construction and installation has been completed. Therefore, the representation must promote a clear, comprehensive, and accurate understanding of a system specification by its eventual users for both reasons of validation and detailed design.

(3) The end result of the systems analysis and design process is a specification. This is a representation that can be used by system builders: designers, programmers, and soft-

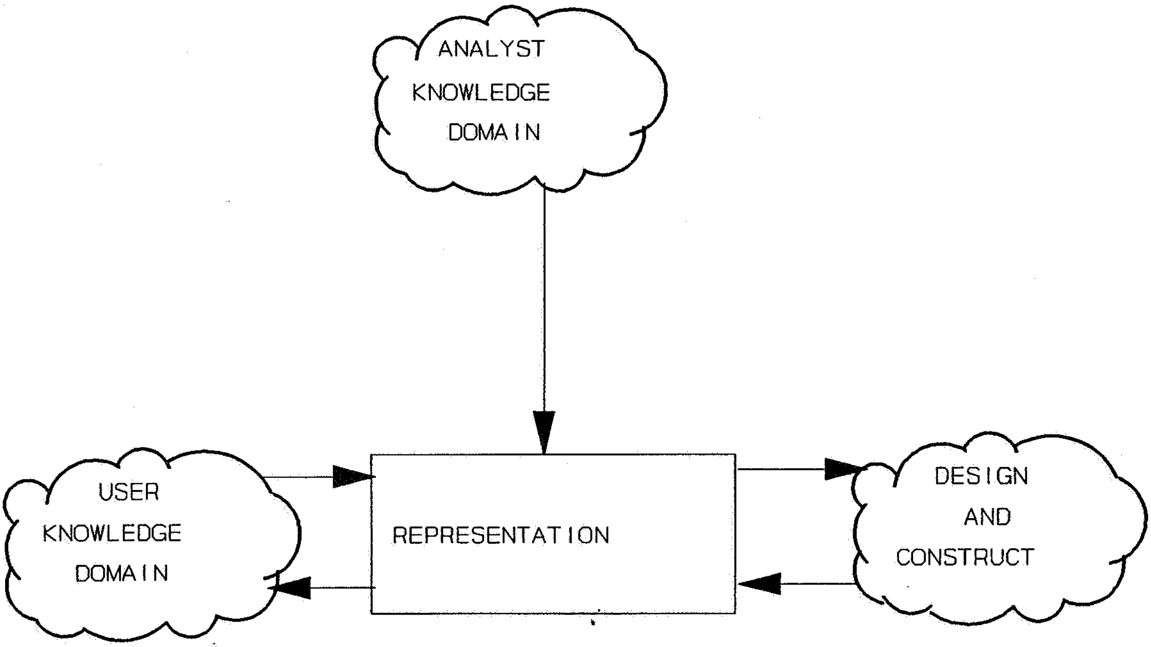

Fig. 1. Representation model.

ware engineers. The critical aspect of specification is that it must be verifiable. That is, in the subsequent stages of design, development, and testing, builders must be able to evaluate and test their work against the representation provided as a specification.

However, representations that facilitate an analyst's task of requirements discovery are not likely to be effective in communicating a clear and comprehensive understanding of the eventual system to users. Equally, the latter are likely to be too imprecise and informal for verification by analysts. Easily understood representations lack sufficient precision and rigor to be useful for either reliable system construction or for verification of subsequent steps.

We assume the analyst begins each new task with knowledge that is "the sum of all his experience." With this as an initial state, the analyst's expectation of the discovery task assignment forms an initial representation [12,20].

Understanding is based on an interplay between concrete and abstract knowledge [9]. Concrete knowledge refers to physical entities and phenomena in the real world, that is, things we can relate to through our senses. In contrast, abstract knowledge consists of models and/or thought patterns that enable us to create order into our diverse surroundings. Consequently, an abstract representation makes it possible for humans to categorize large amounts of concrete information, to see the connection between different pieces of information, and to seek further information effectively. However, abstract information without a rich set of connections to relevant, concrete information has a negative effect on understanding.

The experimental evidence from psychology is derived from studies of storage and recall. Many experimenters document the superiority of concrete over abstract information in simple recall tasks. Concrete words are recalled much better than abstract words in many different populations (children, adults, male, and female) [2,7,15,16,21]. Research has also shown that concrete sentences are easier to recall than abstract sentences [15,18].

Concrete information was remembered much longer than abstract information in an experiment by Anderson [1]. This conclusion also applied when the initial information was wrong and subjects became aware that it was wrong. In Anderson's experiment, subjects exposed to concrete data displayed a tendency to draw causal inferences to a much greater extent than subjects exposed to abstract data. Subjects in the abstract category did not maintain erroneous beliefs as stubbornly as subjects with concrete data.

Researchers have also examined whether the superiority of concrete information holds for higher mental activities such as decision processes. An example of this line of research is the "spy chain" dilemma of Cox [4,5], where the decision is based on the way a pair of spies fits into a larger chain of spies. Each spy can be identified by a synonym, which has either an abstract (e.g., irony) or concrete (e.g., chair) referent. Cox's findings showed that concrete spy-labels were recalled significantly better, and that the time needed to solve the problem was significantly higher for subjects receiving the "abstract treatment."

Researchers within the communication school-of-thought conclude that computer system development success is highly dependent on a process where users and systems analysts share a common frame of reference [10]. Under this communication model, common frameworks are established when two parties have a common perception of meaning, goal achievement, and rapport.

According to De Brabander and Thiers [3], communication may become more effective if a third person, acting as an "active question efficiator," is present during the user-analyst interaction. The added role is intended to stimulate the user to provide more arguments whenever the systems analyst makes a proposition.

Language also has an important role; specific language provokes better reactions (e.g., more concrete responses and a more positive attitude toward the systems analyst as well as toward the systems development task) than vague language. King [13] argues that specific language patterns are crucial to the systems analyst's ability to elicit user requirements.

3. The research question

In our framework, discovery takes place in the user's knowledge domain. Users are involved with the concrete aspects of their work; use of abstract language can have a significantly negative effect:

user-analyst interaction needs to be oriented to references to concrete phenomena in the user domain. As any other form of communication it can syntactically and semantically vary over an abstract to concrete continuum. We assumed that if a systems analyst based discovery on an abstract representation, the work would be less effective than if it were more concrete to help the analyst understand the user domain. Improved understanding should make it easier to ask the right questions and give good comments. The user-analyst interaction will then yield more complete and correct requirements.

Based on this, the main hypothesis of our experiment is:

> Systems analysts starting with a concrete representation will discover requirements more effectively than analysts starting with an abstract representation.

One iteration of a discovery process can be viewed as four (somewhat overlapping) steps. *First*, the analyst must prepare for interaction with the user. *Second*, the analyst/user interaction is conducted. *Third*, the representation is updated with the newly gained knowledge. *Fourth*, the revised representation is validated with the user.

Prior to conducting an interview, analysts prepare for discovery in a variety of ways. The results of this may range from a broad, general strategy to listing specific, detailed questions. We expected the degree of abstraction of the initial representation to influence the preparation step and its result.

H1.1 Analysts using an initial concrete representation for discovery will generate more relevant questions than analysts using an initial abstract representation.

H1.2 Analysts using an initial concrete representation are likely to generate a more complete discovery pattern than analysts using an initial abstract representation.

H1.3 Analysts using an initial concrete representation are more likely to generate a discovery pattern from detailed to general, while an abstract treatment group will exhibit the reverse pattern.

Hypotheses H1.2 and H1.3 were suggested by other research looking for evidence of mental behavior in similar reasoning tasks, especially in [11 and 19].

The second product of discovery is the set of modifications made to the initial representation to incorporate the new state of knowledge.

H2.1 Analysts using an initial concrete representation for discovery will make more correct additions and modifications to the representation than analysts using an initial abstract representation.

H2.2 Analysts using an initial concrete representation for discovery will make fewer errors in their additions and modifications to the representation than analysts using an initial abstract representation.

The initial concrete representation, having helped generate more relevant questions, should facilitate more useful and understandable responses. Analysts in the "concrete treatment" group will make more correct modifications, and will make fewer errors.

We may expect that systems analysts using an initial concrete representation will work faster than analysts using the abstract version. The opposing perspective, however, suggests that detailed knowledge makes analysis more time-consuming, because there are more things to identify and relate to each other.

Conversely, systems analysts using an abstract initial representation may be expected to take a relatively longer time to think through the problem, since the abstract representation provides fewer cues. The opposing perspective suggests that they will use less time, since there is less information to comprehend.

Time measurements relate to the three steps of discovery: (1) preparation, (2) the interview central to the discovery process, and (3) the time taken to modify the initial representation. Hypotheses related to time differences are non-directional, because we had no clear expectation of the treatment effect on time.

H3.1 There is no difference in the time it takes to prepare for the discovery process between systems analysts using an initial concrete representation and analysts using an initial abstract representation.

H3.2 There is no difference in the time it takes to complete the user interview between systems analysts using an initial concrete representation and analysts using an initial abstract representation.

H3.3 There is no difference in the time it takes to complete necessary modifications between systems analysts using an initial concrete representation and analysts using an initial abstract representation.

The utility of a particular representation may be partially dependent upon attitudes toward it. One component of attitude is the perception of ease of use; another is the perception of effectiveness. Subjects acting as analysts were expected to differ in their attitudes depending upon the initial representation they were assigned.

H4.1 The systems analyst using an initial concrete representation will perceive that preparing for discovery is easier than will the analyst using an initial abstract representation.

H4.2 The systems analyst using an initial concrete representation will perceive that he discovers more relevant information than will the analyst using an initial abstract representation.

H4.3 The systems analyst using an initial concrete representation will perceive the task of modifying the initial representation as easier than will the analyst using an initial abstract representation.

Users participate in the discovery process as respondents to questions. They are expected to perceive differences between treatments. Specifically, they were expected to perceive that analysts using the concrete representation asked more relevant questions than analysts using the more abstract treatment. We also anticipated that users would perceive the systems analyst working from the initial concrete representation to be more effective.

H5.1 Users will rank the questioning by the concrete treatment group as more relevant than those of the abstract treatment group.

H5.2 Users will perceive systems analysts using an initial concrete representation to be more effective than analyst using an initial abstract representation.

4. Experimental Design

The design chosen to test these hypotheses was a randomized experiment with two treatment factors [14]. External validity was increased by having the systems analyst obtain information from a peer playing the role of a user, rather than from a paper case.

Systems analyst subjects were split into two groups. Each was given an incomplete concrete or abstract representation. Analyst subjects were informed that a specific, vital section was missing from the representation. Each analyst subject was instructed to prepare a list of initial questions. The second task of the experiment was to complete the representation by interviewing a user to discover the missing information. Finally, analysts were instructed to complete the representation using the acquired information.

The subject pool was formed from a class of university level faculty who were resident participants of the Information Systems Faculty Development Institute, sponsored by the American Association of Collegiate Schools of Business (AACSB), during the summer at the University of Minnesota. These adults were more likely to have had experience with real systems in the work place than the typical undergraduate student. Another benefit in using volunteers from this group was that they had completed a graduate level course in systems analysis before the experiment took place. Participants were therefore familiar with the form, construction, and interpretation of the data flow diagrams used as the experimental material.

Subjects were randomly assigned to one of three categories: user role, concrete representation treatment, and abstract representation treatment. One analyst from each treatment group was randomly assigned to interview each user. Sequence effects were mitigated by alternating the order of treatment scheduled for each user.

Before the interviewing process began, users were given documentation about the business and their roles. A one-hour presentation and discussion session was conducted to ensure that users understood their roles and the information available to them. Systems analysts selected for the first interview were trained in two separate treatment groups. A 45-minute walkthrough, based on the appropriate set of data flow diagrams, was carried out to ensure that the systems analysts understood the diagrams as well as possible. Systems analysts assigned to the second time-block were given their introduction and walkthrough while the members of the first cohort were carrying out their interviews. All training was conducted by one of the experimenters, to help assure consistency.

All experiments were conducted in one continuous sequence in one afternoon and evening. This administrative measure was designed to avoid contamination of results that might otherwise have occurred. In particular, the proximity and frequent interaction of participants in a residential program might have resulted in inadvertent transfer from an initial analyst cohort to a later one.

The treatments were designed to emphasize differences on the abstract-concrete continuum, and to minimize all other differences. *Data flow diagrams* (DFDs) are widely used modeling tools that can emphasize such a difference. "Physical" DFDs are intentionally concrete, often representing the current system or the ultimate detailed design. "Logical" DFDs exclude concrete references, sequences, and specific actors in favor of the abstract in an attempt to divorce the present and future operational details from the needs. The two variants use the same syntax: circles and arrows, and semantic units: sources, sinks, data flows, transforms, and data stores.

The literature does not emphasize the difference between physical and logical DFDs explicitly as concrete versus abstract [for example, 6, 8, 22]. A typical definition of the logical DFD is [17, page 160]:

> "One dimension of a system that can be shown in a model is the essential business, or logical, processing... Thus, the logical aspects of a system are those elements that are the same whether the work is done with pencils and paper or by a computer."

The difference between physical and logical DFDs is explicitly stated as:

> "Logical models, then, concentrate on *what* the system does, while physical models stress *how* the job is done."

Physical DFDs explicitly include concrete references. DFD references to "data entry terminal," or "invoice-customer copy" are analogous to "chair," "tree," or "Mary" in the psychological experiments cited. In contrast, logical DFDs *are* abstractions just as are the words "thought," "imagery," and "philosopher" in storage and recall experiments. Broader differences do exist, for example between a prototype implementation and a narrative description. Representations that differ more widely on the concreteness – abstractness dimension were rejected because they differ in many ways more than abstractness – concreteness, and would therefore confound the experiment.

The problem of stimulus equivalency between treatments was a consideration throughout the process of treatment development. While the two treatments describe the *same* underlying system, logical DFDs are much more compact than the corresponding physical DFDs. We considered and rejected an alternative notion of equivalency that would have required us to make each treatment the same number of pages, symbols, levels, etc.

A more subtle equivalency issue was also considered: normally, the logical DFD is derived from the physical in the process of requirements determination. This dependency might introduce new problems, for example, the logical DFD might be a clarification of the physical, as well as a more abstract view of the system it represents. To diminish the potential for error resulting from the dependence of one treatment on the other, both sets of DFDs were derived from a third description of the underlying system. This third view was a more graphical description that included an office layout, examples of forms and documents used, a sample business conversation in the context, etc. This third, non-treatment representation of the system was a basis for training volunteers in the user role.

After the experiment was completed, all data were encoded on paper, and then statistically analyzed using the "RS1" package operating on a PC. Coding was a lengthy and complex process in several respects. First, questions prepared by the analyst subjects were classified into discrete categories and quantified. Second, modifications to the initial DFD representations were identified, decomposed, classified, and quantified according to a predefined set of categories. Coding reliability was examined by independently coding selected samples and comparing results. Formal tests of coding reliability were not considered necessary due to the consistency of results.

5. Results

Since the subject pool was relatively homogeneous, no background factors were expected to affect the results. *Table 1* describes the subjects and their assignments. A one-way ANOVA F-test on age differences between analyst groups showed an F statistic of 2.26 ($p = 0.12$). There was no significant difference between groups with respect to age.

During the debriefing, subjects were asked to evaluate the quality of the material handed out to them. In addition, their perception of the quality of our presentation and walkthrough of the material was solicited. No significant differences between groups were found on these items.

Hypothesis H1.1 predicted that systems analysts using initial concrete representations would ask more relevant questions than the analysts using the initial abstract representations. Subjects wrote their questions on forms that were collected at the end of the experiment. Questions were considered to be relevant if they fit into one of the following categories:

- questions about the business in general, that would help the analyst understand interaction between the user and the rest of the organization;
- specific questions about user-environment interactions
- general questions about the user's role;
- specific questions about tasks carried out by the user, or about documents, data stores and information flows;
- questions about alternative sources of information, such as who else the systems analyst might interview;
- questions about the user's perceived need for changes.

Table 1
Subjects and treatment groups.

Subjects	Representation			
	Abstract	Concrete	User	Total
Number participating	12	12	14	38
Mean age	43.4	37.0	42.5	41.6
Standard deviation of age	6.26	4.27	6.79	6.79

Table 2
Classification of prepared interview questions.

Questions asked	Representation			
	Abstract	Concrete	t-value	p-value
Mean number asked	14.8	14.5	−0.00	0.53
Mean number relevant	7.3	11.8	2.08	0.02
Relevant/total	0.517	0.855	2.64	< 0.01

Questions about the organization that were clearly not connected with the user's role, questions containing unknown terminology given the facts of the case, and incomprehensible questions were categorized as not relevant. One example of an irrelevant question was a request for an explanation of the analyst's DFD: "Please describe all bubbles and data stores in diagram 0."

The analysis is presented in *Table 2*. All t-tests were one-tailed, physical greater than logical.

These results show that the two groups of analysts asked the same total number of questions of the user. After questions were screened for relevance, however, significant differences were apparent. Analyst subjects using the initial concrete representation asked more relevant questions than did those in the abstract treatment group.

Relevant questions were also categorized into the four groups:

(1) General business questions
(2) Interrelationships between the organization and the user's role
(3) General questions related to the user's role
(4) Detailed user's role questions and other specifics.

A pattern of reasoning might be revealed by the sequence of questions among these categories. If the pattern is categories 1, 2, 3, and 4, the analyst's reasoning is characterized as top down. In the opposite sequence, the reasoning pattern is characterized as bottom up. An additional consideration is the depth of the reasoning pattern. If all 4 categories are present the pattern

Table 3
Category of the first prepared question.

Question category	Abstract		Concrete	
	Frequency	Percent	Frequency	Percent
(1) General business	3	25.0	2	16.7
(2) Business/role interrelation	2	16.7	0	0.0
(3) General user role	5	41.7	10	83.3
(4) Detailed	0	0.0	0	0.0
No discernable pattern	2	16.7	0	0.0
Total	12	100.1	12	100.0

has a depth of 4. If only two are found, the depth is 2. In this study, depth was measured from the sequence of questions until (1) four categories were included, (2) a particular category repeated itself, or (3) all questions the systems analyst had written down had been scored.

The depth of the patterns resulted in a significant difference between groups. The group using an initial concrete representation had a mean depth of 2.9 while the abstract representation treatment group had a mean depth of 2.0 ($t = -2.11$, $p = 0.04$). This result supports hypothesis H1.2: the concrete initial representation resulted in a more complete discovery pattern.

The first category in the sequence of questions strongly indicates the pattern. This first selection is documented in *Table 3*.

Although general questions related to the user's role (category 3) were dominant in both groups, there was some difference between the patterns of the two treatment groups. The initial abstract representation led to an initially broader pattern (categories $1 + 2 = 41.7$ percent compared to 16.7 percent for the concrete treatment group). Although the number of subjects per category was small, the treatment difference between the combination of categories 1 and 2 with category 3 was significant (chi-squared $= 2.79$, $p = .095$) (Note that the chi-square was performed on a collapsed version of the table: the sum of rows 1 and 2, and row 3. The test did not include empty cells). This result provided some empirical support for hypothesis H1.3. Subjects in the concrete treatment group seemed to begin their interviews with more focused questions.

In hypothesis H1.3, we predicted that the reasoning pattern induced by an initial concrete representation would be from detailed to general, and the reverse pattern would hold for the abstract treatment group. The results documented in *Table 4* support this hypothesis. The number of subjects was small but a comparison of the treatment difference on the pattern "from detailed to general" was significant (Chi-squared $= 4.44$, $p = .04$).

In addition to directly confirming the hypothesis, our data indicated an additional pattern in these recorded questions. Of the 14 patterns discerned, all but one repeated or cycled. For example, the pattern of questions of eight subjects cycled from detailed to general and back to detailed again.

The end result of the discovery process was modifications and additions to the initial representations. These end results of the experiment were evaluated by scrutinizing every atomic item of addition, change, or deletion made to the initial representations. Each atomic item was coded:

- As an addition or a deletion. A change to an existing portion of a diagram was coded as two changes: a deletion followed by an addition.
- As relevant or irrelevant.
- As a violation of data flow diagramming syntax. These errors were defined and documented as:
  (1) an error in connecting a symbol or a flow;
  (2) a symbol or flow with no name;
  (3) a physical name on a logical diagram;
  (4) an unrecognizable symbol;
  (5) the wrong name used, i.e., the subject invented a name when appropriate terminology was available;
  (6) an existing part of the DFD was duplicated by the subject;

Table 4
Pattern of reasoning in prepared interview questions.

Pattern of reasoning	Abstract		Concrete	
	Frequency	Percent	Frequency	Percent
General to detailed	3	25.0	2	16.7
Detailed to general	2	16.7	7	58.3
No discernable pattern	7	58.3	3	25.0
Total	12	100.0	12	100.0

Table 5
Data flow diagram modifications resulting from discovery.

Modifications to the DFD	Representation			
	Abstract	Concrete	t-value	p-value
"Ideal" number possible	25	61		
Mean number of changes made	22.3	36.2	2.16	0.02
Mean number correct	1.8	15.6		
Correct/ideal	0.073	0.255	4.44	< 0.01
Adjusted number correct	4.3	15.6		
Adjusted number/ Concrete ideal	0.071	0.255	4.36	< 0.01

(7) use of the wrong symbol;
(8) an atomic item was connected to an existing symbol on the wrong level;

- Whether the change was semantically valid; i.e., whether the change could be judged relevant to the case.

The total number of atomic units of change and the number of correct atomic units of change were computed for each analyst subject, and a correctness proportion was calculated. The denominator of the correctness proportion was an "ideal" (correct) version established by the researchers. *Table 5* shows the results by treatment group.

Concrete representations contain a higher number of atomic unit symbols than abstract representations of the same underlying system. During the discovery process, however, analysts in both treatment groups might be eliciting and processing units of information of a similar level of abstractedness. It is of course possible that the initial abstract representation led to discovery at a similarly abstract level. We considered the more likely case to be the opposite: analysts using the initial abstract representation would elicit units of information that were relatively more concrete than their initial representations. If that was the case, then the modification process might be much more difficult for subjects in the abstract treatment group. They would have to convert the knowledge they discovered to a more abstract representation before modifying their data flow diagrams.

An additional consideration is that error number 3, "a physical name on a logical diagram," could only be made by analysts in the abstract treatment group. To see if this was true, we adjusted the score for analysts in the abstract group by ignoring this type of error.

This adjustment, however, effectively permits concrete representation units as modifications to the initial abstract representation. The ratio "correct:ideal" is therefore less valid under this condition. To compensate, we tested the ratio "adjusted:concrete ideal." As Table 5 shows, performance between the two treatments was significantly different even after this adjustment.

Hypothesis H2.1 is clearly supported. Systems analysts using initial concrete representations out-performed analysts using initial abstract representations by recording a higher score of correct additions, deletions, and modifications to the DFDs. We expected analysts using an initial concrete representation to make relatively fewer errors because communication during the discovery process was expected to be dependent on the relative concreteness of the terminology used. Differences in the proportion of errors, therefore, provided additional evidence that the initial concrete representation facilitates more effective discovery.

Analysts in the two treatment groups committed the same mean number of errors, even though the analysts in the concrete treatment group made substantially more changes than did their counterparts in the abstract treatment group. Proportionally, however, analysts using the initial concrete representation made significantly fewer errors, as *Table 6* shows. As in the previous test, we adjusted the scores of the abstract treatment group by ignoring error 3, physical references on logical diagrams. Hypothesis H2.2 is supported by the experimental results.

Hypothesis group H3 predicted three treatment differences in the time it took to prepare,

Table 6
Data flow diagram modification error analysis.

Modification errors	Representation			
	Abstract	Concrete	t-value	p-value
Mean number of errors made	20.0	20.6	0.12	0.55
Error/number of changes made	0.871	0.555	−5.98	< 0.01
Adjusted errors/ changes made	0.735	0.555	−1.99	0.03

Table 7
Task time analysis.

Measures of task time (minutes)	Representation			
	Abstract	Concrete	t-value	p-value
Preparation	38.2	41.2	−0.42	0.68
Discovery (interviewing)	33.6	31.3	0.44	0.66
Revision (to the DFDs)	44.4	44.5	−0.01	0.99

interview, and modify the initial representation. We were not able to predict the direction of the expected difference.

Table 7 shows that no significant treatment differences on experimental task time were detected. We do not interpret this result to mean there is no treatment effect on analyst's time in these three tasks. Rather, our results are likely to have been artifacts of our administration of the experiment: it would have been difficult for the subjects to take much more time than the mean due to the experiment schedule.

We anticipated that the systems analysts using an initial concrete representation would find each of the three discovery tasks easier than would analysts in the abstract treatment group. Each construct was measured by a single item with a five-point scale. Each item had "very easy" (coded +2) as one extreme and "very difficult" (−2) as the other. *Table 8* shows the results.

Hypothesis H4.3 was the only significant finding. Subjects in the concrete treatment group found it easier to change the initial DFDs than did those in the initial abstract representation group. However, both groups perceived the task as relatively difficult on the absolute scale. On completion of the experiment, subjects were asked if they found the DFDs useful as a basis for discovery. Seven of the 12 analysts using the initial abstract representation said they were useful, while nine of the 12 in the concrete treatment group stated that the data flow diagrams were useful. These differences were too small to be significant. The overall conclusion is that hypothesis group H4, analyst subjects' perceptions, is only partially supported by the experimental results.

Table 8
Analyst subject perceptions of difficulty and relevance.

Analyst's perceptions	Representation			
	Abstract	Concrete	t-value	p-value
Difficulty of preparation	0.55	0.18	−0.84	0.80
Difficulty of modifying DFDs	−1.42	−0.75	1.97	0.03
Relevance of information	0.91	0.83	−0.15	0.56

Table 9
User perceptions of the distribution of questions.

Focus of questioning (percent)	Representation	
	Abstract	Concrete
User's role and duties	45.9	48.5
Inputs and outputs	25.5	26.9
Files and archives	10.3	12.0
Other organizational entities	15.0	11.1
Other	4.3	4.0

During their debriefing, users were asked to categorize the questions they had been asked. This was done with a forced allocation among five predefined question categories. The choices were:

- The user's role;
- documents used as input to or output from the user;
- files or archives used;
- external organizational units with which the user interacted;
- other.

The category means of the resulting are shown in *Table 9*.

A chi-square test on Table 9 (chi-square value $= 0.821$, $p = 0.936$) shows that the results of the two treatments are not independent. As perceived during user debriefing, the two treatment groups asked questions in the same frequency pattern.

Users were also asked to estimate relevancy by question. A relevancy score for analyst subject was calculated by multiplying category score by relevancy proportion and summing the result over the five categories. The weighted mean for subjects using the initial concrete representation was .866 and for the abstract treatment group .731. The two scores are significantly different ($t = 1.98$, $p = 0.03$). Hypothesis H5.1, which expected the user to perceive that analysts using an initial concrete representation asked more relevant questions, is supported by this evidence.

Hypothesis H5.2 predicted that the user would rate the analyst using an initial concrete representation as more effective. This was tested with two separate questions. The mean absolute effectiveness score (on a 5-point, +2 to −2 Likert scale) of this group was 0.40, while the mean for the abstract treatment group was 0.17. This result was not significant ($t = 0.44$, $p = 0.44$) and the hypothesis was consequently not supported by this test. The second debriefing question asked users which of the two analysts, "abstract" or "concrete," they preferred. Two users (of the 7 usable pairs) preferred the "abstract" systems analyst while five chose the "concrete." Using the binomial distribution, the result is not statistically significant ($p = .23$).

The results of hypothesis group H5 indicated that subjects assigned to the user role did perceive a treatment difference in question relevancy but did not reach a similar conclusion regarding either effectiveness or preference.

6. Discussion and conclusions

We predicted and found significant treatment effects. All of the hypotheses except those related to time and attitude were confirmed by the results of the experiment. The evidence suggests some guidelines for practice. Of the three experimental tasks, only one, the interview itself, was clearly in the domain of interpersonal communications. Both preparing for the interview, and later, adding interview results to the representation, were individual, mental processes.

In the first task, generating discovery questions from the initial representation, we observed that subjects using an initial abstract representation did not do well in the task of generating questions relevant to the portion of the system under study. In the final task, they were less able to represent what they had discovered as modifications to their representations.

The psychology literature more generally suggests this result. We think that analysts starting with the abstract representation had too few specifics to generate questions. Similarly, in the third task, they were less able to connect the information gained to their abstract model.

The limited inferences we can make from the results of hypotheses H1.2 and H1.3 are also intriguing. These very exploratory hypotheses resulted in evidence of patterns of reasoning, and significant treatment effects. Analysts using an initial concrete representation began with questions clearly relevant to the user's role and had more detectable patterns of questioning than the abstract treatment group.

Hypothesis group H2 compared the products of systems analysis in the form of modifications to data flow diagrams. The results actually are a composite of performance on the three tasks in the experiment. Analyst subjects used the initial representation they were assigned to generate questions and as the primary guide to their user interviews. But the treatment differences may not have become apparent until analysts had to convert the information they discovered to modifications at the treatment level of abstractedness.

The treatment differences are striking. Analysts in the concrete treatment group made more modifications with a lower proportion of error than their counterparts. The differences remain significant in spite of adjustments we made to compensate for difference in treatment size.

We interpret the combined results of H1.1 and H2.1 and 2.2 to be support for the representation model as an explanation of both internal, reasoning behavior and interpersonal, communications behavior on the part of systems analysts.

The mixed results on perceptual measures are explained in terms of the abstract to concrete continuum. The significant findings are measures of concrete phenomena. The two examples are: (1) that the systems analysts using an initial concrete representation perceived the task of modifying data flow diagrams to be easier than did the analysts using the initial abstract representation, and (2) that the users perceived the questions asked by the analyst in the concrete treatment group as more relevant than the questions asked by the abstract treatment group analysts.

The initial representation formed by a systems analyst influences subsequent behavior and the outcome of the discovery process. When the initial representation is a graphical model on paper, its concreteness, detail, and references to specific real-world artifacts, activities, and events result in: more relevant questions by the analyst (H1.1), more effective interviews as perceived by users (H5.1), and more success in completing the discovery task (H2.1, H2.2).

While our subjects were much more mature and broadly experienced than the usual student subjects, they were still immature as systems analysis. Similarly, our "users" were role-playing and had no "real," on-the-job experience in the setting of the experimental treatments. Our results therefore cannot be interpreted as applying to all systems analysts, but only to novices, or trainees. A replication of this experiment with real systems analysts, real users, or both would more strongly demonstrate the effect of differences on the abstract to concrete continuum.

A practical conclusion is that the process advocated by Yourdon, Constantine, DeMarco, and others should not be arbitrarily shortened. It is very tempting in practice to conduct a systems study at the "logical model" level, bypassing the detail and tedium of the "physical model of the existing system." The results indicate strongly that the more abstract "logical DFD" is not as effective as the concrete "physical DFD" as a basis for effective discovery in the initial phase of systems development. We agree that the representation the analyst eventually manipulates into a new design must be abstract. The process of abstracting, whether carried out using external tools, or as a purely mental activity, is prerequisite to a manipulable representation. Design or redesign requires manipulation of a representation. But the systems analyst's abstract representation does not relate to the user's domain of knowledge. The tools and techniques used for discovery can benefit from these experimental results. Tool-builders as well as analysts need to provide both concrete and abstract representations to support systems development.

References

[1] Anderson, C.A., "Abstract and Concrete Data in the Perseverance of Social Theories: When Weak Data Lead to Unshakable Beliefs," *Journal of Experimental Social Psychology*, Vol. 19, 1981, pp. 93–108.

[2] Borges, M.A., Stepnowsky, M.A., and Holt, L.H., "Recall and Recognition of Words and Pictures by Adults and Children," *Bulletin of the Psychonomic Society*, Vol. 9, 1977, pp. 113–114.

[3] Brabander, B., and Thiers, G., "Successful Information System Development in Relation to Situational Factors which Affect Effective Communication Between MIS-Users and EDP-Specialists," *Management Science*, Vol. 30, No. 2, 1984, pp. 137–155.

[4] Cox, W.F., Jr., "Problem Solving as Influenced by Stimulus Abstractness–Concreteness," *Psychology*, 1976, pp. 37–44.

[5] Cox, W.F., Jr., "Problem Solving as a Function of Abstract or Concrete Words," *Contemporary Educational Psychology*, Vol. 3, 1978, pp. 95–101.

[6] Demarco, T., *Structured Analysis and Systems Specification*, Yourdon, New York, 1978.

[7] Friedman, A., and Bourne, L.E., Jr., "Encoding the Levels of Information in Pictures and Words," *Journal of Experimental Psychology: General*, Vol. 105, 1976, pp. 169–190.

[8] Gane, C., and Sarson, T., *Structured Systems Analysis*, Prentice-Hall, New York, 1979.

[9] Goldstein, K., and Scheerer, M., "Abstract and Concrete Behavior: An Experimental Study with Special Tests," *Psychological Monographs*, Vol 53, No. 239, 1941, pp. 1–31.

[10] Guinan, P.J., and Bostrom, R.P., "Development of Computer-Based Information Systems: A Communication Framework," *DataBase*, Spring 1986, pp. 3–16.

[11] Juhn, S.H., *Information Acquisition Behaviors of Experienced and Novice Systems Analysts During a User Requirements Determination Task: An Empirical Investigation*, Unpublished doctoral dissertation, University of Minnesota, Minneapolis, MN, 1988.

[12] Juhn, S.H., and Naumann, J.D., "The Effectiveness of Data Representation Characteristics on User Validation," *Proceedings of the Fifth International Conference on Information Systems*, University of Indiana, Indianapolis, IN, 1985.

[13] King, W.R., "Feedback-Methodological Optimality in Operations Research," *OMEGA, The International Journal of Management Science*, Vol. 4, 1976, pp. 9–12.

[14] Kirk, R.E., *Experimental Design*, Brooks/Coole Publishing Company, Belmont, CA, 1982.

[15] Paivio, A., and Begg, I., *Psychology of Language*, Prentice-Hall, Englewood Cliffs, NJ, 1981.

[16] Parker, J.F., and Bass, D., "Pictures versus Words as Stimuli in Paired-Associate Transfer," *American Journal of Psychology*, Vol. 88, 1975, pp. 635–642.

[17] Powers, M.J., Adams, D.R., and Mills, H.D., *Computer Information Systems Development: Analysis and Design*, South-Western Publishing Co., Cincinnati, OH, 1984.

[18] Rowe, E.J., Schurr, B., and Meisinger, D., "Comprehensibility Ratings of Concrete and Abstract Sentences," *Bulletin of the Psychonomic Society*, Vol. 11, 1978, pp. 49–52.

[19] Watson, R.T., *A Study in Group Decision Support System Use in Three and Four-Person Groups for a Preference Allocation Decision*, Unpublished doctoral dissertation, University of Minnesota, Minneapolis, MN, 1987.

[20] Wells, C.E., and Nauman, J.D., "System Flow Charts and Data Flow Diagrams," *Proceedings of the Annual Meeting of the American Institute for Decision Sciences*, Las Vegas, NV, 1985.

[21] Yarmey, A.D., "Hyperamnesia for Pictures but not for Concrete or Abstract Words," *Bulletin of the Psychonomic Society*, Vol. 8, 1976, pp. 115–117.

[22] Yourdon, E., and Constantine, L.L., *Structured Design: Fundamentals of a Discipline of Computer Program and System Design*, Prentice-Hall, Englewood Cliffs, NJ, 1979.