EFFECT OF STRUCTURE OF ANALOGY AND DEPTH OF ENCODING ON LEARNING COMPUTER PROGRAMMING

Yam San CHEE
Department of Information Systems and Computer Science
National University of Singapore

Abstract

This research addresses the need for effective ways of teaching computer programming. It focuses on two aspects of instruction. First, the research investigates the use of analogy in teaching programming. It extends existing research by investigating what constitutes a *good* analogy. Second, the research investigates the effect of depth of encoding on programming performance.

The factors *analogy* and *encoding* were manipulated in a 3 x 2 factorial design. Analogy was operationalized by varying the clarity and systematicity/abstractness of the analogies used. Encoding was operationalized by varying the frequency with which deep encoding and elaboration of learned material were invoked by the presentation of questions on the learned material. The dependent variables were score obtained on program comprehension and program composition tasks and the time taken to perform the tasks. Research subjects were 15- to 17-year-olds without prior exposure to computer programming. Differences in mathematics ability and age were controlled.

The results provide empirical support for a predictive theory of the relative goodness of competing analogies. They provide only marginal support for depth of encoding (as operationalized) in learning computer programming effectively. *Post hoc* data analysis suggests that good analogies assist the learning of semantics but not syntax. Furthermore, the effect of encoding was only apparent in learning syntax but not semantics.

Introduction

The traditional approach to teaching computer programming by emphasizing programming language statements (Mayer, 1979; Spohrer & Soloway, 1986) has proved unsatisfactory. Such an approach fails to assist in the acquisition of a useful mental model of the notional machine underlying the programming language (Bayman & Mayer, 1983) and to facilitate the transition from programming knowledge to programming behavior (Anderson, Farrell & Sauers, 1984).

Explanatory Analogy

Analogies are a useful tool for learning and instruction (see, for example, Norman, 1980; Rumelhart & Norman, 1981). The validity of this claim has been demonstrated in the domain of learning computer programming (Mayer, 1975, 1976). Analogies can provide the required mental model of the notional machine. They can also facilitate the transition from programming knowledge to behavior as novices attempt to execute their mental model. Simons (1984) posits that analogies assist learning by making abstract information imaginable and concrete, by providing an existing schema as the basis for the formation of a new schema, and by making relevant anchoring ideas available so that new information can be actively integrated with prior knowledge.

Gentner (1982) postulates the characteristics of analogy that contribute to explanatory power. Her postulation is based on a well-defined theory of structure-mapping (Gentner, 1983) that distinguishes between *attributes* and *relations* on one hand and between *first-order relations* and *higher-order relations* on the other.

An explanatory analogy may be viewed in terms of three properties of internal structure: clarity, richness, and systematicity/abstractness (Gentner, 1982). *Clarity* refers to how base nodes are mapped onto target nodes. A violation occurs if one base node maps to two or more distinct target nodes *or* if two or more distinct base nodes map to the same target node. *Richness* refers to predicate density: that is, for a given set of nodes, the average number of predicates per node that can be plausibly mapped from base to target. *Systematicity/abstractness* refers to the degree to which the imported predicates belong to a mutually constraining conceptual system. Higher-order relations that link lower-order relations are the essence of systematicity. Highly systematic mappings are generally also abstract because they contain a greater proportion of higher-order relations.

The Theory of the Structure of Explanatory Analogies is derived, in part, from distinctions drawn by Gentner (1982) between good and bad explanatory analogies. It states that important, regularly occurring structural differences exist between good explanatory analogies and weak explanatory analogies. In particular, (1) good explanatory analogies possess clarity; weak explanatory analogies do not; (2) good explanatory analogies are higher in systematicity and abstractness than weak explanatory analogies; and (3) good explanatory analogies are lower or equal in richness to weak explanatory analogies.

The theory is used as the basis for distinguishing between the explanatory power of alternative analogies in this research. For achievement in both program comprehension and program composition, learning with an analogy that possesses the structural properties of good explanatory analogy is expected to result in a better learning outcome than learning with an analogy that possesses the structural properties of weak explanatory analogy.

Depth of Encoding

Learning outcomes depend not only on the quality of instruction but also on the efficacy of cognitive processing *during* the learning phase. Good learning outcomes are associated with depth of encoding (Craik & Lockhart, 1972). Greater depth implies a greater degree of semantic or cognitive analysis on the part of the student, resulting in superior understanding, recall, and retention of material learned. The initial encoding of learned material can pass through further elaboration whereby more associations are formed between newly acquired knowledge and prior knowledge. The establishment of these associations leads to better integration of new knowledge with old knowledge and improved understanding of learned material.

Depth of encoding also results in better recall because of a more persistent memory trace (Craik & Lockhart, 1972), with deeper levels of encoding associated with more elaborate, stronger, and more lasting traces. In addition, retention is a function of depth of encoding, as well as other factors such as the amount of attention devoted to a stimulus and the time available for processing the stimulus.

Based on the foregoing, the quality of students' learning when acquiring knowledge related to a new and unfamiliar domain should be significantly affected by the depth of encoding they achieve during learning. Deeper encoding should be facilitated by presenting instructional material in relatively short segments followed by questions on the material just learned. The presentation of questions forces students to try to actively understand the instructional material so that they can answer the questions correctly. Consequently, deep encoding and elaboration receive active support. The presentation of questions in short segments also eases the burden of learning because a lighter cognitive load is placed upon memory.

Where students complete their study of the entire instruction set *before* attempting questions on the materials learned, depth of encoding is less well supported. The absence of questions that evoke deeper processing of instructional material during learning results in more superficial processing and, consequently, in poorer understanding, poorer retention, and poorer recall ability. Furthermore, when students are required to answer questions only at the end of the instruction set, the cognitive load on memory is very great because students have to draw their answers from across the *entire* instruction set.

Thus, deep encoding is expected to result in better understanding, retention, and recall of learned material, and hence in superior programming task performance compared to shallow encoding that occurs when the entire instruction set is studied before questions on the instruction set are attempted.

Hypotheses Tested

The research hypotheses are based on four theoretical constructs: (1) explanatory power of analogy, (2) depth of encoding, (3) quality of program comprehension, and (4) quality of program composition.

Hypothesis 1 The quality of program comprehension when learning with a good analogy is better than the quality of program comprehension when learning with a weak analogy or without an analogy.

Hypothesis 2 The quality of program comprehension when learning with a weak analogy is better than or equal to the quality of program comprehension when learning without an analogy.

Hypothesis 3 The quality of program comprehension when learning with deep encoding is better than the quality of program comprehension when learning with shallow encoding.

Hypothesis 4 The differences in quality of program comprehension when learning with a good analogy, a weak analogy, and without an analogy will be greater when learning with shallow encoding than when learning with deep encoding; that is, there will be an interaction effect between the explanatory power of analogy and the depth of encoding.

Hypothesis 5 The quality of program composition when learning with a good analogy is better than the quality of program composition when learning with a weak analogy or without an analogy.

Hypothesis 6 The quality of program composition when learning with a weak analogy is better than or equal to the quality of program composition when learning without an analogy.

In general, the above hypotheses follow from the preceding discussion. In Hypothesis 4, an analogy is postulated to possess an *integrating* function in addition to the functions of concretizing, structurizing, and active assimilation. Hypotheses 5 and 6 are similar to Hypotheses 1 and 2 and are based on the expectation that mastery of syntax and semantics is an essential component of program coding ability.

Method

Design The factors *analogy* and *encoding* were manipulated in a 3 x 2 factorial design. Analogy comprised three levels: (1) good analogy, (2) weak analogy, and (3) no analogy (a control condition). Encoding comprised two levels: (1) deep, and (2) shallow. The experiment was conducted in two phases: a program comprehension phase followed by a program composition phase. Two dependent variables were used in each phase. The dependent variables in the program comprehension phase were (a) program comprehension score, and (b) time taken to answer comprehension questions. The dependent variables in the program composition phase were (a) program composition score, and (b) time taken to answer composition questions. Both the program comprehension and program composition scores are performance metrics obtained by applying a predetermined scoring template to subjects' responses. The experimental design incorporated two covariates: mathematics ability and age.

Subjects Subjects were school students between the ages of 15 and 17 years. They were unexposed to computer programming. Ninety valid subjects' responses were obtained; 60 were boys and 30 were girls. They were assigned randomly to treatment conditions.

Materials

Treatment Materials. The treatment materials comprised three sets of instruction on programming in BASIC: (1) the good analogy set, (2) the weak analogy set, and (3) the no analogy set. In the good analogy set, the instructional materials were woven around an analogy that dealt with a master processor and his three assistants – the assigner, the reader, and the printer – working together in a room to perform the operations of a notional computer. Data were input either through an input slot or via data cards that came through an input window on an input wall. Data were output via an output window on an output wall. Window boxes in the room stored the values of variables whose names were written on the boxes. In the weak analogy set, the underlying analogy was similar but less elaborate. There was only one window through which both input and output were handled. In addition, the names of the three assistants were generalized to "assistant," "messenger," and "helper" in order to facilitate the one-to many and many-to-one object mappings in the weak analogy. Finally, the no analogy set presented the instructional material without reference to any analogy.

The instructional materials covered the program statements LET, PRINT, END, REM, INPUT, DATA, READ, GO TO, and IF/THEN. Looping constructs were taught using the IF/THEN and GO TO statements.

The exact length of the instructional materials was controlled. To compensate for the additional text required to present the analogy material, filler text (which presented a brief history of computers) was added to the weak analogy and no analogy materials so that the word count for each set of instructional materials was identical.

The good and weak analogy treatment materials instantiated the Theory of the Structure of Explanatory Analogies. The instructional materials contained the base of the analogy (good or weak) woven into the instruction on BASIC.

A sample of the good and weak analogies, depicted in propositional network form, is shown at the end of this paper. Networks 1 and 9 depict those portions of the base of the good analogy and weak analogy respectively that deal with the organization of the computer. The corresponding targets of the good and weak analogies are depicted in Networks 5 and 13.

Object mappings between base and target may be inferred directly by the positions the object nodes occupy in two-dimensional space. In the weak analogy networks, however, this method does not apply if an object participates in a one-to-many or many-to-one mapping. In such instances, the object mapping is specifically shown using a striped arrow.

Relation mappings from base to target are inferred via the identical positions that the relations occupy in the two-dimensional space of the propositional networks. Exceptions to this rule again occur in the weak analogy networks, and they occur when object nodes do not map one-to-one from base to target. Unlike object nodes, however, relations that map across always do so with the same name. First-order relations are depicted by normal arrows; higher-order relations are depicted by heavy arrows. Higher-order relations constrain lower-order relations in accordance with structure-mapping theory.

The operationalization of the Theory of the Structure of Explanatory Analogies can be summarized as follows. First, the good analogy possesses clarity because all object mappings from base to target are one-to-one; the weak analogy does not possess clarity because it contains two one-to-three and three one-to-two mappings. Second, the richness (predicate density) of the good analogy is 2.00, and the richness of the weak analogy is 1.85. From a practical viewpoint, richness may be regarded as equal; the closeness of the richness measures is not surprising given that *both* the good and weak analogies aim to explain the operations of the notional computer. Third, the good analogy possesses higher systematicity/abstractness than the weak analogy because 1 third-order relation, 9 second-order relations, and 78 first-order relations were mapped to the target in the good analogy compared with no third-order relations, 6 second-order relations, and 57 first-order relations being mapped across in the weak analogy.

Test Materials. The test materials comprised two sets of questions. The first set was designed to test program comprehension. It was divided into eight parts: (1) Elements of the BASIC language; (2) The replacement statement LET; (3) The PRINT statement; (4) Review: LET, PRINT, and END; (5) The INPUT statement; (6) The DATA and READ statements; (7) The unconditional transfer statement GO TO; and (8) The decision statement IF/THEN.

The second set of questions was designed to test program composition. The set comprised seven questions in increasing order of difficulty and covered the full range of BASIC statements presented.

Procedure The study was conducted in two sessions. Session 1 (the program comprehension phase) commenced at 10:00 a.m. On average, the session lasted 2 1/2 hours. Session 2 (the program composition phase) commenced after a lunch break of about one hour. The session lasted 1 1/2 hours on average.

The encoding factor (deep versus shallow) was operationalized by administering the instructional materials and test questions differently in Session 1. In the deep encoding condition, subjects alternated between reading instructional material on BASIC and answering questions on the material just read. In the shallow encoding condition, subjects read the entire set of instructional materials. They then answered each set of questions in the same order as subjects in the deep encoding condition.

After subjects had been instructed on how the experiment would be conducted, the experiment proper commenced. Subjects were told to begin reading the instructional materials placed before them. As they completed the reading, each raised their hand to indicate to the researcher that they had done so.

If subjects were in the deep encoding condition, they were given the first set of printed questions to answer. The researcher asked them to start work and started the stopwatch. On completion of the set of questions, subjects stopped the stopwatch and raised their hand. The researcher then recorded the time taken on the question sheet. The question sheet was then put away. The next set of instructional materials was then given to the subject. Subjects continued by alternating between reading instructional material and answering questions until the last set of questions was answered.

If subjects were in the shallow encoding condition, they first read the entire set of instructional materials. They then answered each set of questions in the same order and following the same procedure as subjects in the deep encoding condition.

Upon completing Session 1, subjects were released for lunch. When they returned for Session 2, they were instructed on the conduct of the experiment in the program composition phase. They were then given 10 minutes to review the instructional materials they had read in Session 1. The researcher then started each subject on the program composition task and, at the same time, started the stopwatch.

When subjects completed the program composition questions, they stopped the stopwatch and raised their hand. The researcher recorded the time taken on the question sheet. Subjects who completed the experimental task were each paid $20.

Scoring Subjects' responses to the program comprehension and program composition questions were scored according to a template designed by the researcher. The scoring scheme was devised to reward the demonstration of correct knowledge of BASIC and to maximally discriminate between the levels of achievement attained by subjects. An independent check of scoring reliability was performed.

Results and Discussion

Program Comprehension

Figure 1 shows the means for program comprehension score. The program comprehension data were analyzed using ANCOVA and MANCOVA. Hypotheses

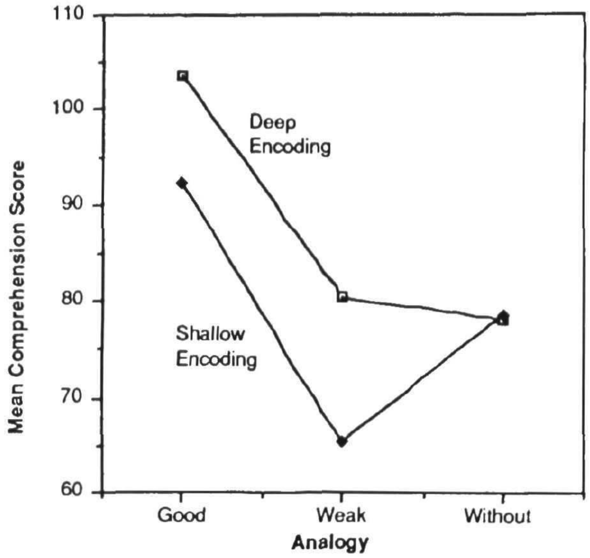


Figure 1

1 and 2 were evaluated using planned comparisons. Hypothesis 1 was confirmed when quality of program comprehension was evaluated in terms of program comprehension score ($p = .001$). It was also confirmed when quality of program comprehension was evaluated in terms of program comprehension score and time ($p = .003$).

Similarly, Hypothesis 2 was confirmed when quality of program comprehension was evaluated in terms of program comprehension score ($p = .499$). The proposition was also supported from a multivariate viewpoint ($p = .395$).

Consistent support for Hypotheses 1 and 2 when evaluated in terms of program comprehension score as well as program comprehension score and time confirms that the quality of program comprehension when learning with a good analogy is significantly better than that associated with learning with a weak analogy or without an analogy, and the quality of program comprehension when learning with a weak analogy and when learning without an analogy are not significantly different. Given the mix of clarity, richness, and systematicity/abstractness operationalized in the experiment, the theory's prediction that these characteristics effectively define good analogy is supported.

Hypothesis 3 was marginally supported when evaluated in terms of program comprehension score ($p = .055$). Despite the lack of statistical significance, the overall (weighted average) program comprehension scores were in the predicted direction (Deep Encoding, $M = 86.5$; Shallow Encoding, $M = 77.4$). The marginal significance of the univariate result may be due the task complexity and constrained experimental learning time that did not allow the expected benefit of deep encoding on the creation and restructuring of knowledge to materialize fully. A deep, semantic appreciation of the notional computer's operations requires that the knowledge acquired be assimilated and restructured over the course of learning. However, restructuring is associated with knowledge understanding but requires time to take effect (Norman, 1978).

Hypothesis 4, the posited interaction between the analogy and encoding factors, was not supported when the quality of program comprehension was evaluated in terms of program comprehension score and also in terms of program comprehension score and time. Thus, the expectation that the analogy would help subjects in the shallow encoding condition more than subjects in the deep encoding condition was not confirmed. Instead, the data suggest that the analogy and encoding factors are independent.

Program Composition

Figure 2 shows the means for program composition score. The program composition data were analyzed using covariance analysis. However, a test of the homogeneity of regression assumption revealed that the assumption was violated for the dependent variable program composition time.

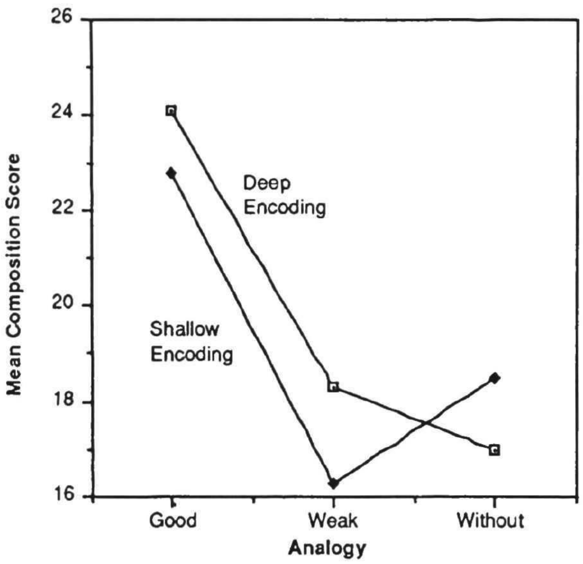


Figure 2

Consequently, the program composition data were analyzed using a covariance model proposed by Searle (1979). *Covariate-adjusted observations* were obtained by rearrangement of the model equation, and the model was estimated using MANOVA.

Using planned comparisons, Hypothesis 5 was supported when evaluated in terms of covariate-adjusted program composition score (p = .014). It was also supported when evaluated in terms of covariate-adjusted program composition score and time (p = .044). Hypothesis 6 was supported when evaluated in terms of covariate-adjusted program composition score (p = .998) and also when evaluated in terms of covariate-adjusted program composition score and time (p = .613). The non-significant result indicates equality between the weak analogy and no analogy treatment conditions.

The consistent program composition results show that the Theory of the Structure of Explanatory Analogies is also supported with respect to program composition. Note that the analogy factor accounted for 6.6% of the explained variance of program composition score but accounted for 12.0% of the explained variance of program comprehension score. A *transfer of learning* from program comprehension to program composition is thus evident. The smaller effect of type of analogy on program composition is consistent with the expectation that the explanatory power of analogy facilitates performance in program composition via achievement in program comprehension.

Post Hoc Analysis

While scoring subjects' responses to the program comprehension questions, it was noticed that the variability of scores on questions that focused on the syntactic rules of BASIC statements was consistently smaller than it was on questions that focused on the conceptual understanding associated with the operations of the notional computer. This phenomenon suggested that it might be fruitful to investigate the data further by distinguishing between scores on syntax-oriented questions and scores on semantics-oriented questions. Accordingly, the program comprehension data were subclassified into syntax scores and semantics scores and analyzed further via a *post hoc* analysis.

The data for the *post hoc* analysis were analyzed using ANCOVA and MANCOVA. The dependent variables were syntax score and semantics score. The experimental design was identical to that used in the main analysis.

Model estimation revealed that for syntax score, the analogy factor was significant (p = .015); the encoding factor was also significant (p = .012). For semantics score, however, only the analogy factor was significant (p = .003). From a multivariate viewpoint, both the analogy factor and the encoding factor were significant (p = .001 and p = .043 respectively).

Planned comparisons were performed to evaluate Hypotheses 1 and 2 for syntax, semantics, and both syntax and semantics. For syntax, the good analogy versus weak analogy and no analogy comparison was marginally significant (p = .056), while the weak analogy versus no analogy comparison was significant (p = .027).

For semantics, the good analogy group was significantly better than the weak analogy and no analogy groups (p = .001), and the weak analogy and no analogy groups were not significantly different (p = .841). That is, the comparisons were consistent with the results obtained for the composite program comprehension score.

Some interesting insights are obtained from the above analysis. The significance of the analogy factor on both syntax score and semantics score and the significance of the encoding factor only on syntax score suggest that the analogy treatment affects performance on both syntax and semantics, while the encoding treatment affects performance on syntax only.

Furthermore, it becomes clear that the marginal significance of the encoding factor on the composite program comprehension score (p = .055) was attributable to the effect of deep encoding on syntax (p = .012), not on semantics (p = .117). This result suggests that students learn the technical (rule-like) nature of syntactic knowledge effectively when such knowledge is tested shortly after it is presented. In effect, the quick application of newly-acquired syntactic knowledge assists students in assimilating the rules associated with syntax and helps to drill them in the application of such rules.

By contrast, the lack of significance of the encoding factor on semantics suggests that, contrary to the intended outcome of the encoding treatment, a deep semantic understanding of program statements was not achieved probably because of the limited exposure that students were given to programming. The hypothetical time division associated with complex learning proposed by Norman (1978) suggests that the bulk of knowledge restructuring (and hence deep semantic understanding) occurs during the central phase of learning, after sufficient time has been spent on the accretion of knowledge. Given the restricted learning time in the experiment (approximately four hours), the relatively small amount of time spent on restructuring appears to be due to the use of analogy rather than the use of deep encoding.

Examination of the weighted average means for syntax shows that the good analogy and no analogy groups were more alike than different, while the weak analogy group was unlike both (Good Analogy, M = 25.5; Weak Analogy, M = 20.6; No Analogy, M = 24.1). This result suggests that the weak analogy harmed the acquisition of syntactic knowledge, and the good analogy did not assist the acquisition of such knowledge. However, for semantics, the good analogy group was distinct from the weak analogy and no analogy groups (Good Analogy, M = 72.6; Weak Analogy, M = 52.0; No Analogy, M = 54.6). Thus, the good analogy assisted the acquisition of semantic knowledge but not syntactic knowledge.

Research Conclusions

The Theory of the Structure of Explanatory Analogies was empirically tested. The research supported the theory's prediction that clarity and systematicity/abstractness are structural characteristics of analogy that effectively capture the strength of its explanatory power. *Post hoc* analysis further revealed that good analogy assists the acquisition of semantic programming knowledge but not syntactic programming knowledge.

From the viewpoint of experimental methodology, the *explicit* operationalization and measurement of systematicity and abstractness has shown that these structural characteristics of analogy can be derived objectively and in a manner that possesses empirical validity. Thus, the usefulness of the syntactic perspective on knowledge representation based on the concepts of systematicity and abstractness has been demonstrated.

References

Anderson, J. R., Farrell, R. & Sauers, R. (1984). Learning to Program in LISP. *Cognitive Science*, **8**, 87-129.

Bayman, P. & Mayer, R. E. (1983). A diagnosis of beginning programmers' misconceptions of BASIC programming statements. *Communications of the ACM*, **26**, 677-679.

Craik, F. I. M. & Lockhart, R. S. (1972). Levels of processing: A framework for memory research. *Journal of Verbal Learning and Verbal Behavior*, **11**, 671–684.

Gentner, D. (1982). Are scientific analogies metaphors? In D. S. Miall (Ed.), *Metaphor: Problems and Perspectives*, pp. 106-132. Brighton, Sussex: The Harvester Press.

Gentner, D. (1983). Structure-mapping: A theoretical framework for analogy. *Cognitive Science*, **7**, 155-170.

Mayer, R. E. (1975). Different problem-solving competencies established in learning computer programming with and without meaningful models. *Journal of Educational Psychology*, **67**, 725-734.

Mayer, R. E. (1976). Some conditions of meaningful learning for computer programming: Advance organizers and subject control of frame order. *Journal of Educational Psychology*, **68**, 143-150.

Mayer, R. E. (1979). A psychology of learning BASIC. *Communications of the ACM*, **22**, 589-593.

Norman, D. A. (1978). Notes toward a theory of complex learning. In A. M. Lesgold, J. W. Pellegrino, S. Fokkema & R. Glaser (Eds.), *Cognitive Psychology and Instruction*, pp. 39-48. NY: Plenum.

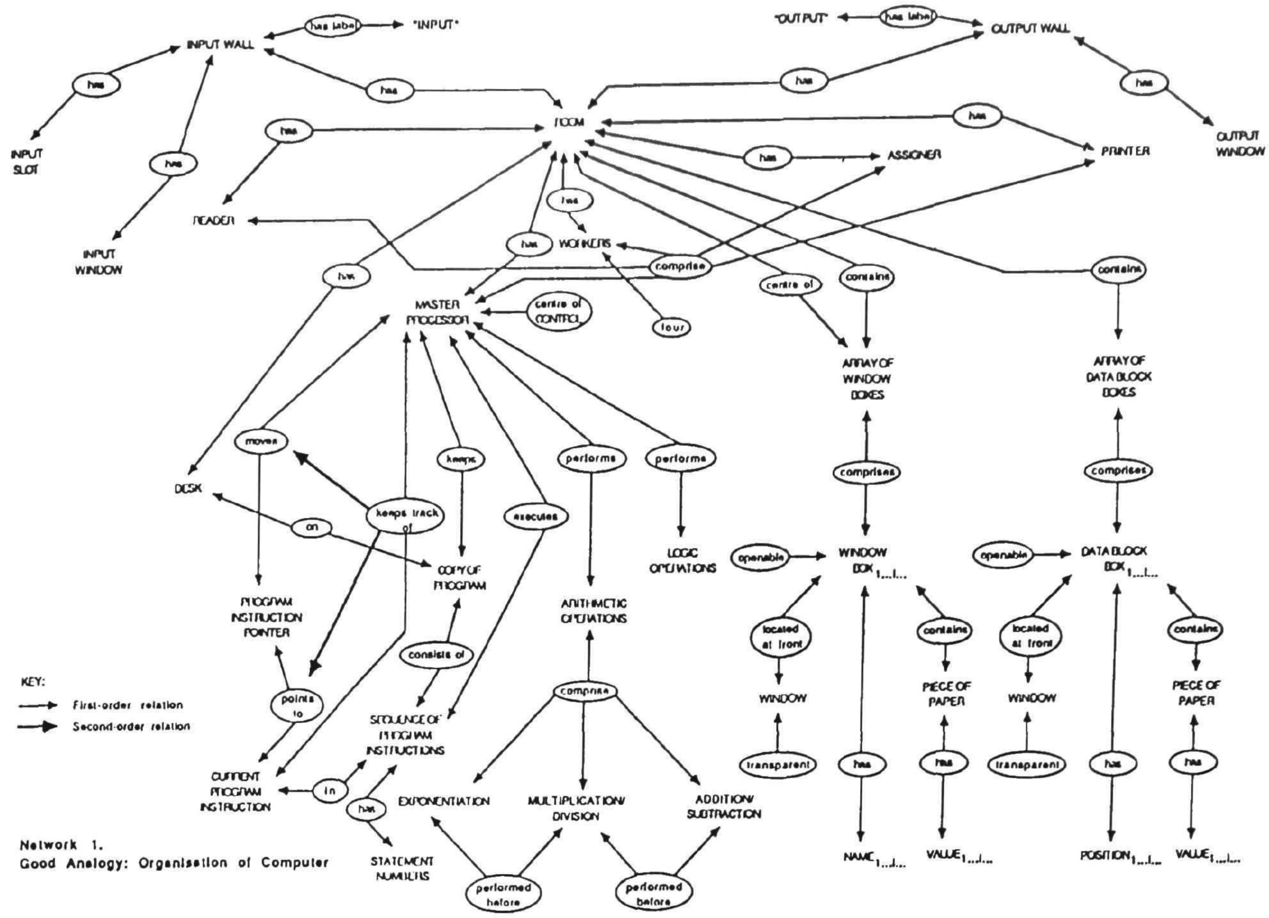
Norman, D. A. (1980). Teaching, learning, and the representation of knowledge. In R. E. Snow, P. A. Frederico & W. E. Montague (Eds.), *Aptitude, Learning, and Instruction*, Volume 2, pp. 237-244. Hillsdale, NJ: Lawrence Erlbaum Associates.

Rumelhart, D. E. & Norman, D. A. (1981). Analogical processes in learning. In J. R. Anderson (Ed.), *Cognitive Skills and their Acquisition*, pp. 335-359. Hillsdale, NJ: Lawrence Erlbaum Associates.

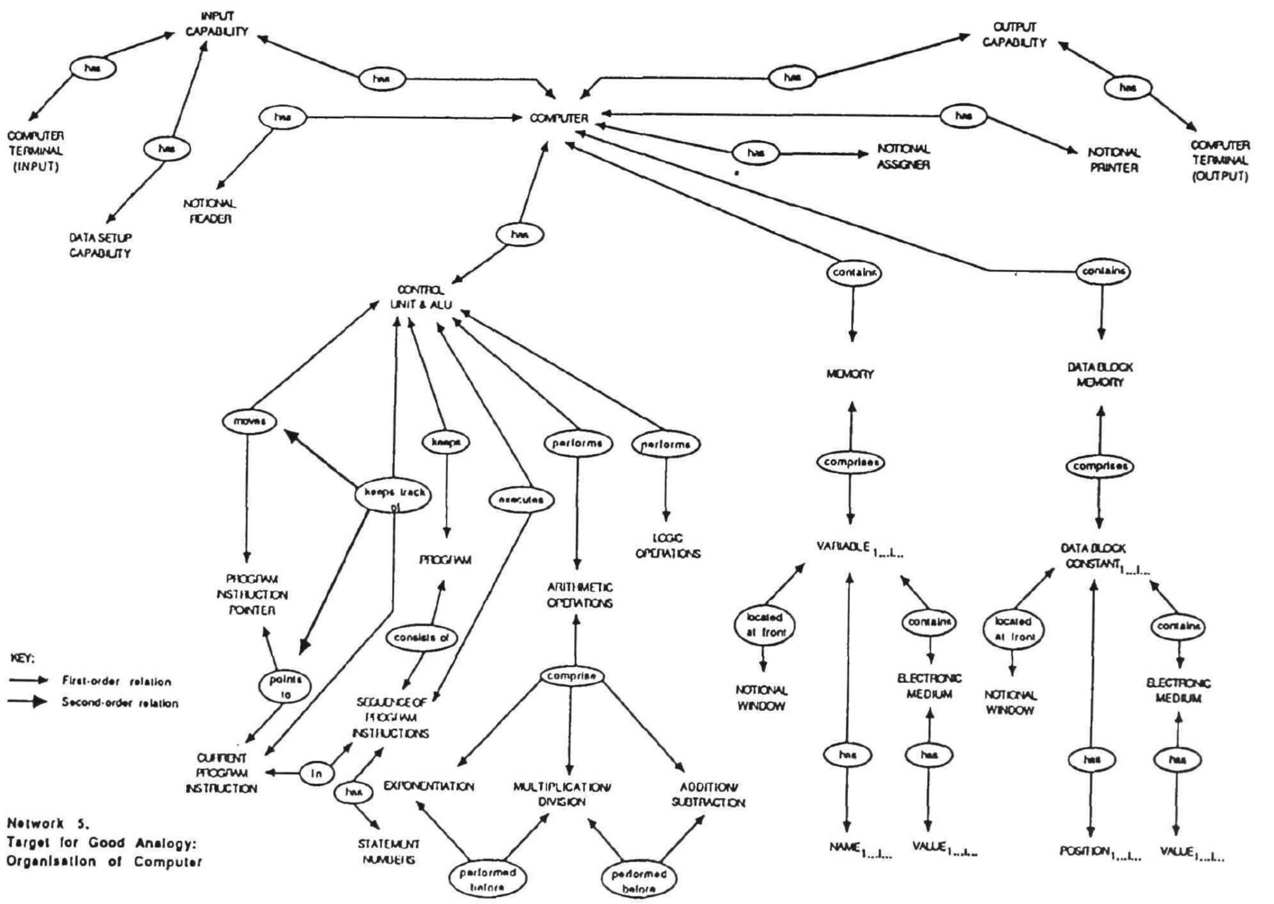
Searle, S. R. (1979). Alternative covariance models for the 2-way crossed classification. *Communications in Statistics – Theory and Methods*, **A8**, 799-818.

Simons, P. R. J. (1984). Instructing with analogies. *Journal of Educational Psychology*, **76**, 513-527.

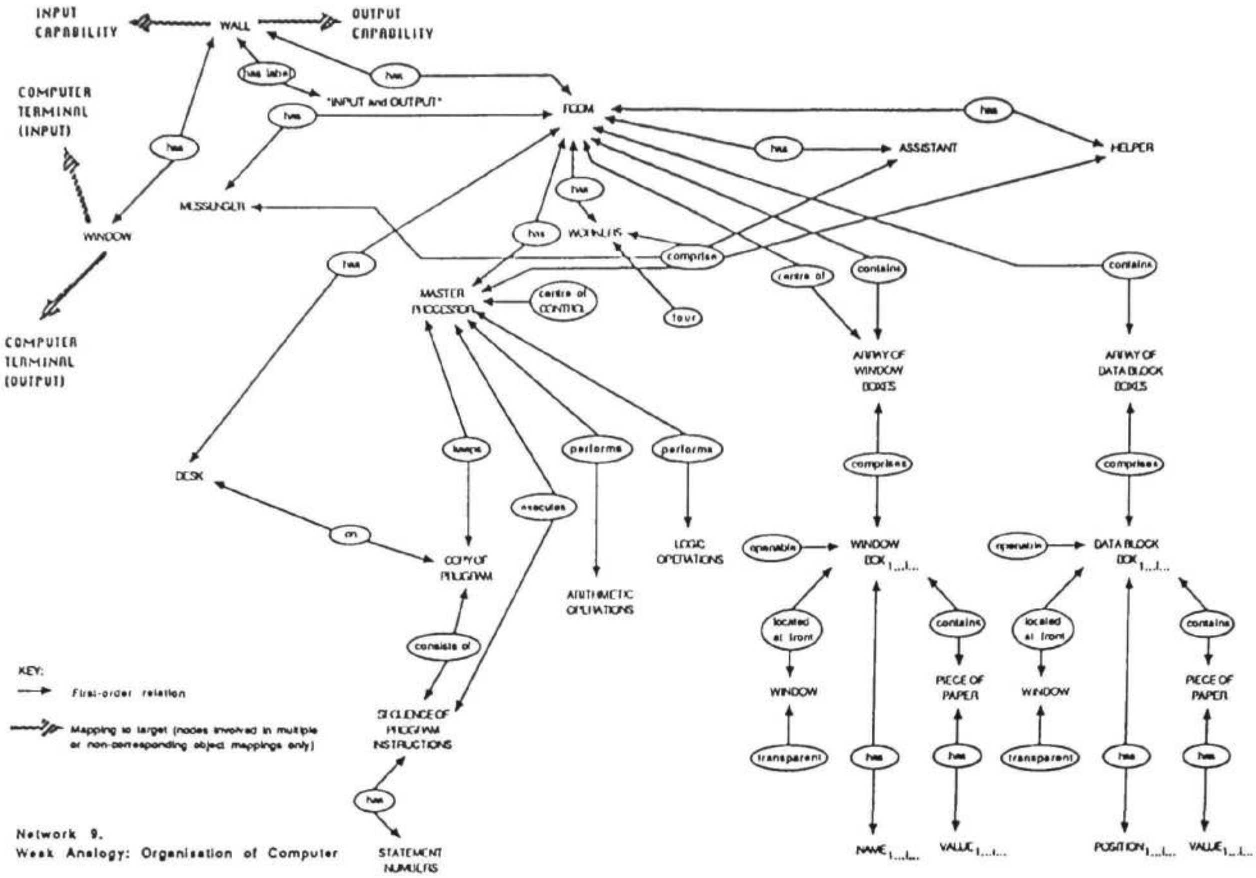
Spohrer, J. C. & Soloway, E. (1986). Novice mistakes: Are the folk wisdoms correct? *Communications of the ACM*, **29**, 624-632.



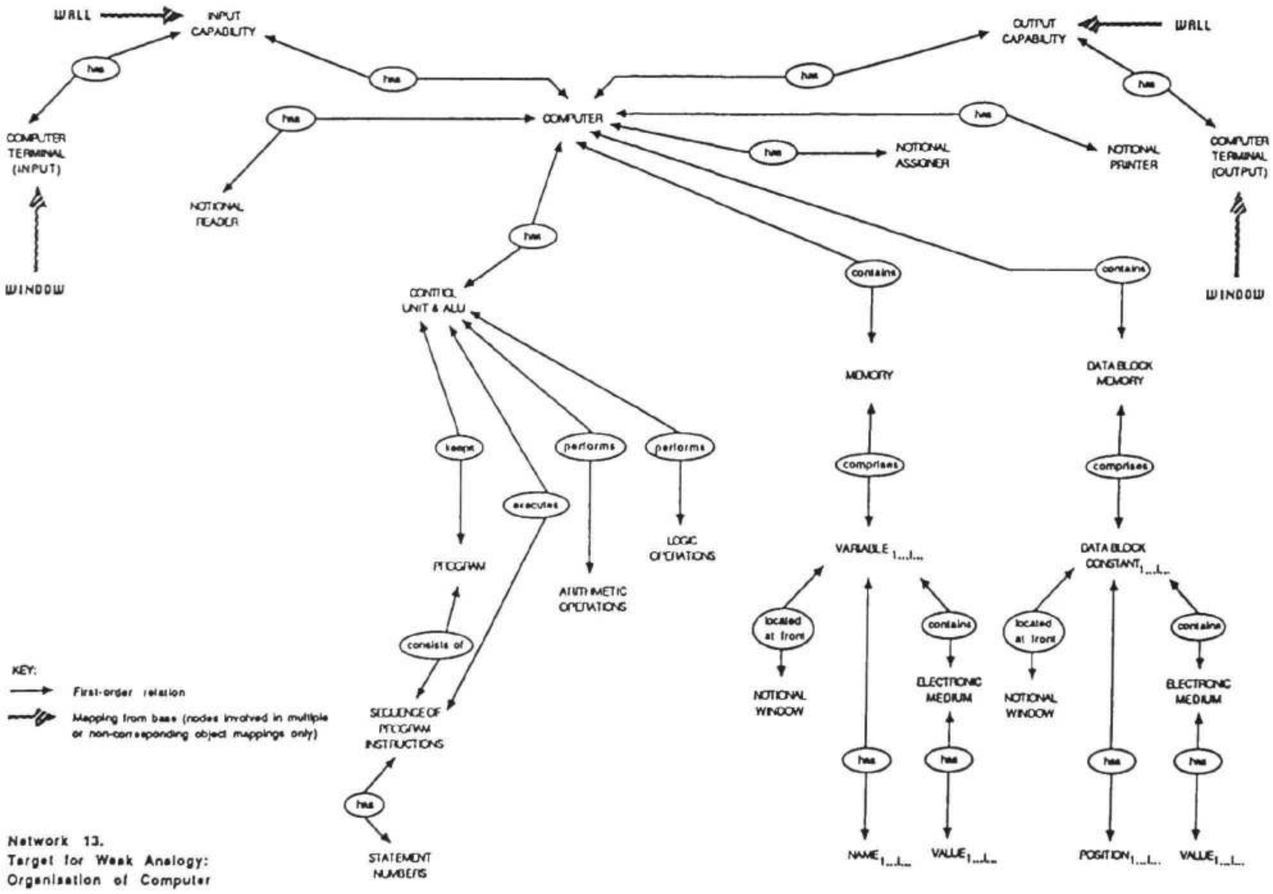
Network 1.
Good Analogy: Organisation of Computer



Network 5.
Target for Good Analogy: Organisation of Computer



Network 9.
Weak Analogy: Organisation of Computer



Network 13.
Target for Weak Analogy:
Organisation of Computer