International Journal *of* Designs for Learning
2014 | Volume 5, Issue 2 | Pages 1-19

TICCIT: BUILDING THEORY FOR PRACTICAL PURPOSES

Andrew S. Gibbons, *Brigham Young University* & A. F. O'Neal, *Whidbey Associates*

In 1971 the National Science Foundation contracted with the MITRE Corporation (MIT Research Corporation) for the production of an experimental system to test the computer-assisted delivery of information and instruction to homes: the TICCIT system—an acronym for "Time-shared, Interactive, Computer-Controlled Information/Instructional Television". The networking concept of TICCIT linked minicomputers through coaxial cables to color television sets. The theoretical design challenge was that the agreement with NSF specified that the instructional component of TICCIT would be learner-controlled. TICCIT system specifications produced a type of instruction that adapted moment-by-moment to the choices of the learner.

Andrew S. Gibbons completed his doctorate in Instructional Psychology at Brigham Young University in 1974. For 18 years he directed large-scale instructional design projects in industry, He joined the Utah State University Department of Instructional Technology for 10 years and the IP&T faculty at BYU for 11 as Department Chair. Dr. Gibbons' book, *An Architectural Approach to Instructional Design* (Routledge, 2014), expresses a theory of design layers, design languages, and modularity.

A.F. O'Neal founded and directed the Kansas City Missouri Public Schools Computer-Assisted Instruction Laboratory 1966 to 1971. From 1971-74 he pursued his PhD at the University of Texas and Brigham Young University, consulting with Mitre and Hazeltine on TICCIT. Since 1974, when he directed the TICCIT based S-3 aircrew training development in San Diego, he has consulted and developed instructional systems for academia, industry, and military and governments worldwide.

THE LANDSCAPE AND THE CHALLENGE

In 1970, one of the most pressing educational questions was, "Can the newly invented computer (barely 25 years old) carry out major instructional functions in a sustainable commercial-grade way?" In 1957 Sputnik had created a new urgency for exploring educational applications of technology. The goal was to improve instruction, distribute it to a wider audience, and at the same time lower the costs of doing it. Though today we take the instructional computer for granted, in 1970 this was a big question, and there were a lot of doubters. Several problems had to be solved: computer cost, delivery cost, instructional development cost, and instructional quality. In the process theory was born unexpectedly.

COSTS

Computers of the day were large and expensive. A mainframe computer required special air conditioned facilities and a team of trained technicians, programmers, and operators. Input-output systems were primitive—a teletype terminal, a deck of Hollerith punched cards producing a paper print-out, or a monochrome text display with keyboard. Vector graphics were a new technology requiring programming skills. Automated graphic and logic authoring interfaces were just being developed and not ready for general use. Connecting the computer to multiple users required the use of expensive high-quality telephone lines.

The costs of computerized instruction did not reside alone in equipment and delivery. The creation of instruction was expensive because each lesson was hand crafted. A lesson design team normally consisted of either a highly gifted individual with a special combination of skills and training, or a team of specialists, consisting of a writer/editor, a programmer, and perhaps an artist. A subject-matter expert was also needed to supply content knowledge. Specialized programming languages were tailored expressly for instructional purposes, but as expectations of lesson quality grew, new features had to be added to the languages, and they became quite large, detailed and difficult to learn.

INSTRUCTIONAL QUALITY

The quality of computerized instruction had to be seen in a future view, since no one, not even those who were creating it, knew quite what the instructional computer could do. The search for instructional quality was taking place on two levels: theoretical, and applied.

During the 1960s and 1970s, experimental computer-based instructional styles grew up in great variety. Atkinson and Wilson (1969) featured reports of several high-visibility theoretical projects. Chapters outlined futuristic visions of intelligent, conversational instructional systems. Suppes (1969) envisioned a system that could "approximate the interaction [of] a patient tutor" (p. 43) and a dialogue system that could "recognize the spoken speech of a child" (p. 43). Stolurow (1969) proposed that, "a CAI course of any magnitude should teach the student *how* to learn" (p. 90, emphasis in the original).

Wenger (1987) conducted an extensive review of experimental intelligent tutoring systems dating from 1970. He distinguishes between "frame based" and "intelligent" CAI, defining one as being fixed in its delivery sequence and the other as being a "communication" system that adapted its interactions to the moment-by-moment responses of the learner. This distinction became, and remains today, a major fault line between two schools of thinking about the core architecture of technology-based instruction, and it was a major issue in the design of the TICCIT system, as will be described later.

Wenger expressed his own vision of the future of computer-assisted instruction:

> *Now imagine active books that can interact with the reader to communicate knowledge at the appropriate level, selectively highlighting the interconnectedness and ramification of items, recalling relevant information, probing understanding, explaining difficult areas in more depth, skipping over seemingly known material*. . . . (p. 6)

Below the level of these exalted visions, everyday designers were also experimenting with the new technology on a less-formal, less-theoretical, and more practical level. They too were struggling to envision what CAI should be. But their ideas had to be expressed in the immediate reality of the now. Using large experimental systems like Plato, designers wrestled with the difficulties of lesson creation within a confined and confining design space that was growing but still incapable of expressing far-out futuristic dreams. This work took place in an atmosphere of excitement and anticipation (see, for example, http://platohistory.org/). This was the flavor of the time, and this was the context of the TICCIT project.

ORIGINS OF THE TICCIT PROJECT

During this period of great experimentation, every aspect of the technology was churning with rapid change much like what we are experiencing today. Specifically, hardware, software, instructional technique, design technique, and context of use were all fair game for experimentation.

TICCIT was one of two large-scale experimental computer-assisted instruction (CAI) projects; the other was PLATO. TICCIT and PLATO became contenders in a lopsided horse race sponsored by the National Science Foundation. The goal of the competition was to prove two very different configurations of CAI system. One, the PLATO system had been under development at the University of Illinois since 1960. By 1970, the system was mature, and in 1973 the university established the Computer-Based Education Research Laboratory (CERL) under the direction of Donald Bitzer. The configuration of the PLATO system included a powerful central computer communicating with hundreds of distant terminals over telephone line connections with smart terminals, which were themselves small computers.

In 1967 the National Science Foundation began supporting continued development of the PLATO system, which had previously been funded mainly from military sources. Courseware authors programmed instructional materials using a specialized CAI language called TUTOR, which automated many functions common to instructional programming, such as answer processing, graphics, and data management. PLATO became popular among a loyal following, who found it easier to use for developing instruction than general-purpose computer systems. A number of experimental projects explored the system's capability for the creation of simulations, games, networked communications, and user-constructed virtual environments.

A major challenge for the PLATO system was costs: phone line costs and the cost per hour of instructional development. The creation of PLATO products required a programmer, often a subject-matter expert who had taken the time to learn the TUTOR language. Development was by hand, and the specialized built-in TUTOR routines were cost-savers, but building every product from scratch was expensive. A second cost factor was the use of telephone lines for communication with far-flung terminals. At first transmission quality was uneven, and as quality went up, so did the phone bill. This configuration was popular, but without outside support, there was no business revenue plan to make it sustainable.

Emerging technologies were another important factor. Smaller, powerful, and less expensive minicomputers became available in the mid 1960s. Advances in peripheral technologies and equipment were rapidly changing the definition of the computer from monolithic and self-contained to smaller and modular. By 1970 the minicomputer technology was competing comfortably with the mainframe, and it was possible to test the new configuration for its capabilities for delivering instruction.

For this purpose, early in 1971, the National Science Foundation funded the TICCIT (Time-Shared, Interactive Computer-Controlled Information Television) project. This was the first part of a two-part experiment in alternative computer system configuration. The goal of Part I was a system that substituted a minicomputer and its multi-user operating system in place of the large mainframe computer. This system was designed to serve a smaller population (maximum 128 users), but it was designed to be adaptable for use in many local settings. An initial system was installed in Reston, Virginia using the community's cable TV system. It was used to deliver video, text, and audio information to television sets in homes, using for a control the keys of a touch-tone telephone. The smaller, more local, and portable design of the TICCIT system, compared to PLATO's large centralized system, allowed it to be installed in any setting wired with coaxial cable. This would eventually include schools, businesses, the military, and other public and private institutions. The physical learning stations of each of the two systems did not appear greatly different, but, the invisible parts of the system—the engines under the hood, so to speak—were miles apart.

Experiment Part II took place after the launch of the "informational" configuration of the TICCIT system. NSF entered into an additional contract for the creation of an "instructional" version of the TICCIT system. This would allow instruction to take place in the home. The initial target for testing this concept was the creation of junior college-level courses in language, writing, and math instruction. This broadened the interpretation of the TICCIT acronym to include "Time-shared Interactive, Instructional Television". The equivalent of five courses were identified, three in introductory Algebra, and two in English grammar, usage and composition. The selection criteria were mainly economic. Since one of the main goals of the project was to demonstrate lower cost, high-volume junior college courses were chosen whose large enrollments would average costs over larger numbers of users. According to Alderman (1978) this coverage accounted for roughly one-fifth of the courses normally taken by a junior college student.

The goal of making the original TICCIT system instructional added a host of new hardware, software, instructional, and system design challenges. This design case describes how a large design team set about to solve these problems, which were:

> *. . .To develop the process of courseware production to a level more comparable to that practiced in the engineering professions, and in the process provide the student with powerful yet simple and consistent control over the instructional process.* (McWilliams, 1974)

This created seemingly conflicting goals:

- Full learner control over instruction
- "Engineered" courseware production
- Rapid production of a large volume of material

The goal of giving the learner control over instruction was very relevant at the time of the TICCIT project, as previously described, but it was considered beyond the reach of most designers. No such system had reached commercial success.

Also, the concept of "engineered" instruction was not particularly popular at the time of the TICCIT project, despite being promoted by influential theorist Richard C. Anderson (1961). For many educators and designers, the engineering term conveyed a mechanistic, robotic image. Recall that in the context of 1971 the programmed instruction method, which had an early association with the engineering of teaching machines, was only a few years in the past. But in the minds of many at the time, the mechanistic image was hard to shake, and it was growing less popular, not more.

What was meant by the "engineering" goal of the TICCIT project? This was not certain at the beginning, but the feature of learner control was what occupied the spotlight. However, given the volume of instruction to be created and the limited time, engineering that took advantage of the efficiencies of common structures could not help but be a critical part of the plan. Engineering came to mean that instead of hand crafting each lesson as a unique creation there would be an underlying architectural concept that defined a pattern of behavior and operation common to all lessons. Built into this pattern would be a degree of underlying structure and discipline. It would provide for both exposition and practical application during learning. The evolution of this kind of a design and its impact on high-volume development is described in this paper.

STAKEHOLDERS

The principal stakeholders of the TICCIT project were:

- The National Science Foundation (NSF) – Funding
- Educational Testing Service (ETS) – Evaluation services
- The MITRE Corporation – Hardware/Software system developer
- The TICCIT design research team
  - C. Victor Bunderson – Project Lead
  - M. David Merrill – Design lead
- Primary implementation test sites
  - Northern Virginia Community College
  - Phoenix College

The authors of this design case were in 1971 graduate students in the new Instructional Psychology program at Brigham Young University. Both were employed by the TICCIT project. In retrospect, it is amazing how much they were able to learn, but also how much they were able to contribute, given the right assignments and the confidence of their mentors.

For MITRE and NSF, what was at stake was proving an alternative concept of hardware and software deployment using minicomputers, color television displays, computer-controlled random access video streams, and secure, high-bandwidth connections— a formula for lower cost and higher performance with a smaller footprint and a local range of operation. This would be an alternative configuration to the wide-range mainframe delivery system configuration pioneered by Bitzer (1971) in the Plato system.

Bitzer's description of Plato is divided into two parts: the "science" of computer-based education, and the "engineering" of computer-based education. It is clear that engineering as Bitzer uses the term refers to the engineering of computers and peripheral hardware and software systems into a working delivery platform from which a great variety of instructional forms determined by individual designers could be delivered. This sense of the term engineering pertained also to the development of the TICCIT system. TICCIT involved a massive hardware and software integration effort, the details of which are not covered in this design case. However, the TICCIT project used the term engineering in a broader sense to refer to the structuring of the instructional experience, as noted above.

What was at stake for the TICCIT design team was the challenge of meeting goals similar to those being explored by intelligent tutoring system developers: (1) making possible instructional experiences adapted for or by the learner, (2) the engineering of an architecture that was scalable, and (3) high-volume production techniques that shortened development time. In 1971, this was an audacious goal.

THE CONTEXT OF DESIGN PRACTICE

The context of instructional design practice in the early 1970s was evolving. The empiricist approach of make-try-revise-repeat that characterized the programmed instruction movement (Markle, 1964) was familiar and intuitive enough for most designers to accept. The concept of the instructional objective or goal was also well known, so it had become a standard tool of the designer's craft (Tyler, 1949).

But in 1971 the concept of instructional design as a formal process had not been defined, except within design communities involved in large cold war systems engineering projects such as the Pine Tree Line, the Mid-Canada Line, and SAGE—all complex air defense systems that combined radar, computers, and complex communications with human operators and maintainers. These projects called for more disciplined design processes that included detailed analyses of several kinds and the coordinated efforts of multiple design teams.

Robert Gagné participated in projects of this complexity, and he produced an edited volume dealing with the more involved systematic processes they entailed (Gagné, 1965). These ideas laid the groundwork for the systematic instructional design movement. Briggs also published a monograph for designers on media selection processes (Briggs, 1967), which he followed soon after with his first version of what he termed a "model" of procedures for instructional design (Briggs, 1970). Each military service promulgated its own instructional design process standards, but the Interservice Procedures for Instructional Systems Development (IPISD) that consolidated them and gave us the term ISD was not published until 1975 (Branson et al, 1975).

None of these works had acquired broad influence by 1971, when the TICCIT project was initiated, so the systematic design formalisms commonly known today as ISD or ADDIE did not provide a beginning point for the project. The design process guidelines that had more influence were related to Tyler's doctrine of the alignment of instructional goals, instructional activities, and instructional measurement (Tyler, 1949). Tyler's student, Bloom (1956), had published a system for categorizing learning outcomes, as had Gagné (1965, 1985). The taxonomic principle clearly did have an influence on the approach to TICCIT: certainly more than the then-emerging systematic design, or ISD, principle. This was generally the context of design practice among the principals of the design team at the time of TICCIT.

THE CONTEXT OF DESIGN THEORY

There was also a context of theory in 1971, and it was in a state of flux. At that time, interest in behaviorism was waning in favor of cognitivism. Publications by Bruner et al. (1956), Neisser (1967), and Simon and Newell (1971) and others had taken the discussion of human cognition and learning back inside the mind, to study its internal processes.

This was a critical point for the use of the term "theory" by designers. Up to that point, designers had been used to thinking exclusively in terms of scientific learning theory. Simon (1969) opened new directions for thinking about theory by describing the importance of *technological theories* of design in contrast to scientific theories, as did Vincenti (1990).

The phrase "instructional theory"—referring to technological theory applied to the creation of instructional experience—became increasingly common, beginning in the 1970s (Bruner, 1964, 1966; Gibbons & Rogers, 2009; Merrill & Twitchell, 1994; Reigeluth, 1983; Reigeluth, 1987; Reigeluth, 1999; Reigeluth & Carr-Chellman, 2009; Snelbecker, 1974;

Snow, 1977) Glaser and Resnick (1972) also wrote about theories of instruction in addition to theories of learning. It is interesting to note that since then, *learning* theory has taken a direction that inclines it toward application to the point where it is getting harder to distinguish learning from instructional theory (see Bransford et al, 2000; Lave & Wenger, 1991; Richey, Klein, & Tracey, 2010).

The distinction among the kinds of theory is important because in the course of the design of the TICCIT system and for many years afterward, instructional theories emerged from the project. These are described in the narrative of the design process below.

SOLVING THE DESIGN PROBLEM

The most practical design problem of TICCIT was drawing together a number of disparate engineering concerns. On the one hand, there was the need to satisfy the computer programmers. They needed orderly, algorithmic structures to work with, because algorithms are the operational principle of computer programs. On the other hand, there were the demands of an effective instructional strategy to consider. This, it was clear, would be messy and non-algorithmic, but somehow it had to become more structured. On yet another hand, there was the need to create an interface or a mechanism that would unite these algorithmic and strategic concerns together in a way that allowed the learner to exert control over events. Finally, there was the requirement that the design be producible within time and resource constraints. This was perhaps the most difficult challenge, as the redesign of the original TICCIT system, the conception of the unique instructional approach of TICCIT, the design of lessons, the production and testing of five courses worth of lessons, and the installation of TICCIT systems on two junior college campuses had to be accomplished in just over three years (1971-1975), with none of the conveniences of the mouse, the touch panel, or the menu system of a user-friendly graphical interface.

This was the design problem, and it took some time for the large and diverse team to arrive at a unified understanding of it. The team grew quickly, striving to absorb the magnitude of the problem and their unique contribution to it in a short time.

There were several reasons why it was a challenge for individuals to adapt to this new environment: the diversity of the team, the depth of their expertise in their individual specialties, their intense focus on their own requirements, their own preconceptions about possible solutions, and their lack of understanding of the requirements and constraints of others. None of the team members had experienced a challenge the size of TICCIT, and most of them were used to the more relaxed pace of an academic setting. All of these factors turned the team at first into an effective Babel: everyone spoke a different, specialized design language. Of course, this was not apparent to the group at the time; people were focused on the design itself and not on the linguistic nature of designing: this "omigosh!" sunk in only years later. [See, for example, how Rheinfrank & Evenson, (1986; 1996) brought formalization to the notion of design languages. See also Dubberly & Evenson, 2010.]

How did work on the design proceed? Not systematically, it turned out, according to any particular design model. Things worked inward from different ends of the problem, beginning from two critical sub-problems that had to be solved right away: the programmer's logic problem and the instructional designer's strategy problem. Neither group had prior experience in team design, especially not team design that brought hardware and software engineers together to work with instructional designers. Bringing the two specialty worlds together was more difficult than anyone had imagined. Without realizing it, each group was solving its own sub-problem by creating a design language that eventually could be integrated into a more inclusive and project-wide shared design language that represented the needs of all design team members.

The Logic Problem

The logic problem was a programmer's problem. It turned out to be a problem of creating a language that both the computer and humans could understand: one that directed the computer what to do under different circumstances. It was a matter of creating for the computer a language of things that needed to "happen" and a set of rules for determining when to make them happen.

None of us realized at the time this is what we were doing. Mitchell (1990) describes a similar problem architects encounter when they try to create a common vocabulary for the computer as a partner in the design of buildings. Mitchell characterizes this problem as finding "the logic of architecture", and he proposes that this entails "both the practical and poetic uses of architectural languages" (p. ix). Any designer who engages the computer in designing encounters similar problems, including computer designers, computer chip designers, and computer network designers. Many design disciplines have encountered this problem and solved it successfully. Each case involved the invention of a design language that could be understood by both the computer and the human. Simon (1999, p. 153-54) refers to this as the problem of "representation", referring to the manner of representing the problem to the computer.

The solution to the logic problem was the invention of the "base frame." Figure 1 is taken from early work notes and shows the basic mechanism of a base frame.

A base frame was defined as a chunk of self-contained computer logic for governing changes to the display. During

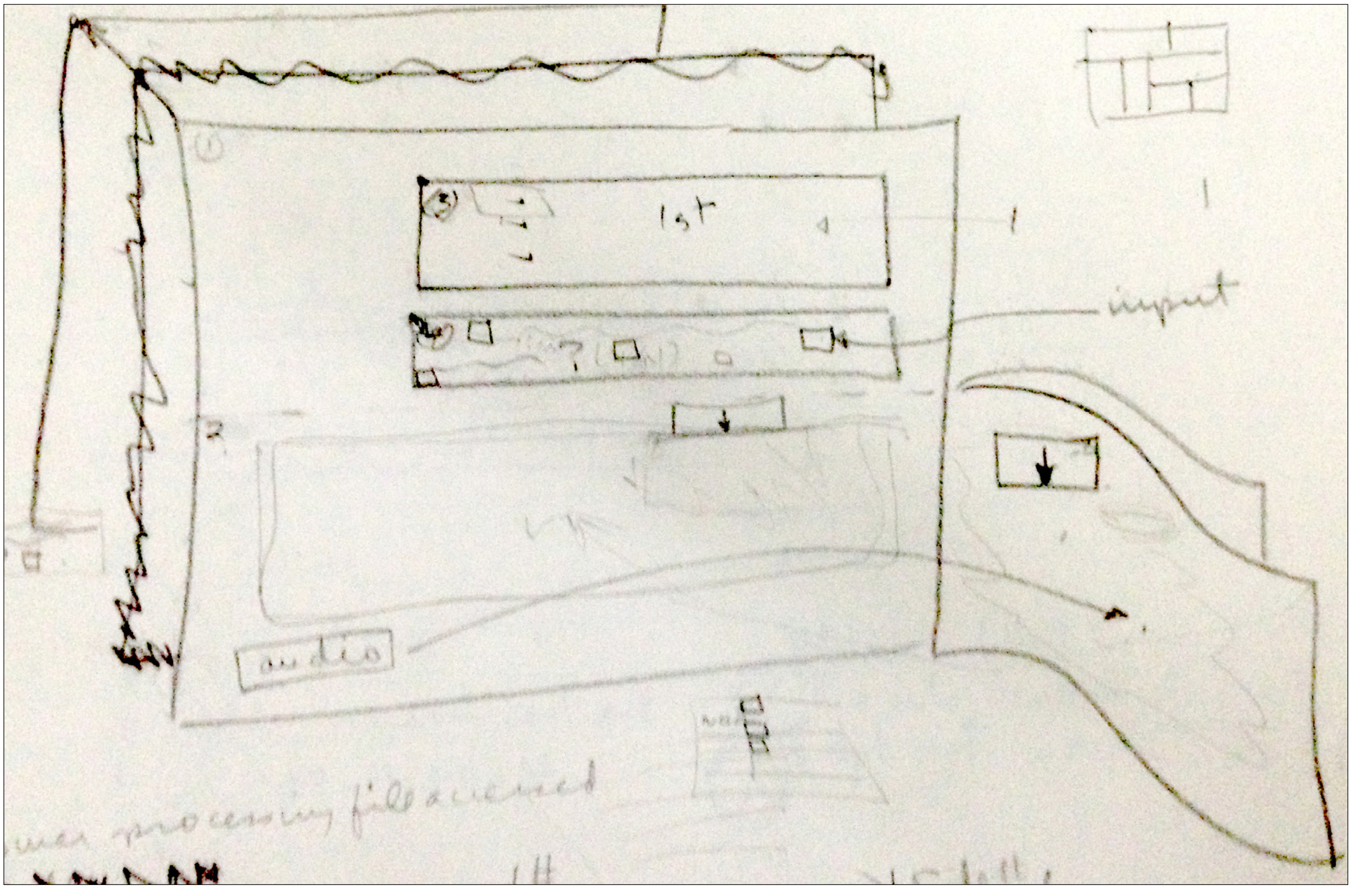

FIGURE 1. An early representation of the base frame concept that bridged computer logic concerns with instructional (display representation) logic concerns.

a single strategic exchange with the learner, the interaction might consist of multiple changes to the display, without a complete change of context (e.g., erasure). A base frame provided a way of defining the boundaries of a visual context. The concepts of window and overlay today make changing just a portion of the screen easy, at the time of TICCIT this was a more complicated challenge, especially because of the non-linear display sequences made possible by learner controls.

The base frame made it possible to maintain some elements of the display (such as a math graph related to a particular problem), while allowing other elements (such as explanatory or emphasis material, enhanced explanations, or practice-related feedback) to change, based on interactions with the learner. The base frame provided the interface between display (representation) content and the execution of strategic logic.

There were many styles of base frames. Base frames could be called and executed as needed, and new base frames could be created as needed. Instantiating a base frame with specific content was a matter of data entry, relieving lesson authors of the task of programming. Specifying screen coordinates identified changeable display areas; the various content elements that could populate an area were contained in one or more stored files. Figure 2 shows a page of designer notes exploring the partitioning of computer logic (the flowchart) into base frames (the boundaries shown by dotted lines). The computer did not have to know what content was already displayed: it only had to know what content file to place on the display at given coordinates at a given moment, defined by learner control sequences.

The base frame became a design language term that bridged the interests of different factions of our design team. It gave computer programmers a set of logical functions that they could program, and it gave designers a way to describe media elements to activate based on the control operations of the learner, and it allowed the two groups to communicate across disciplinary boundaries.

The base frame from TICCIT is an early example of the "frame" concept that in later years later became a fundamental structure in several authoring tools, including Authorware, Director, Flash, WICAT'S WISE, Allen Communications' Quest, and many others (Gibbons & Fairweather, 1998, Chapter 4). It is doubtful that this was the first use of the frame (or window) concept; many people were experimenting with basic structures that could underlie instructional software designs. We just found it useful for our purposes. Later, the frame was also used as the fundamental construct underlying card systems such as Hypercard and Supercard. In the card systems, a "stack" of individual "cards" could each be populated with logic and display events selected from a menu. This was the equivalent of selecting an existing base frame and then populating it with content.

The Strategy Problem

The strategy problem was also a language creation problem. The language in this case was made up of terms that had meaning to the designer in the context of a coherent instructional plan. The solution of the strategy problem began with intense and sustained discussions, led by Dave Merrill, centered on research into concept instruction strategies that he had conducted beginning with his dissertation study and extending through the period of the TICCIT project and beyond (see Tennyson et al, 1972; Merrill, et al, 1992).

The two basic strategic structures in Merrill's research were: (1) the concept definition and (2) the contrasting example/non-example pair. Merrill and his associates had learned through much research (Tennyson et al., 1972) that learning concept classification behavior was facilitated not only by the presentation of concept exemplars, but by the simultaneous presentation of very similar non-exemplars as well. This is called the matching principle. Moreover, it had been shown by this research that when building sequences of exemplar/non-exemplar pairs, successive pairs should be as different as possible from each other. This is called the divergence principle.

The strategy design team started to think of instructional strategy in terms of these basic elements: pairs of matched examples and non-examples and divergent sequences of these pairs that showed the broad range of exemplars. This was a good place to begin—with a design hypothesis that could be tested by building hypothetical prototypes and imagining what that would be like for the learner.

Many design sessions were spent experimenting with stringing combinations of these basic elements (definition, exemplar, non-exemplar) into fixed, algebra-like formulas. One might represent what we called an "all-american" strategy (one of everything); another might represent a "lean" strategy (minimum number of examples); yet another might be biased heavily toward "practice". There could also be different sequences of definitions and examples, and so the number of strategy pattern varieties could multiply through combination and recombination (see Figure 3).

From the list of strategy patterns, the idea was that a learner would choose a pattern they desired and the system would execute that pattern. It was not a great plan for learner

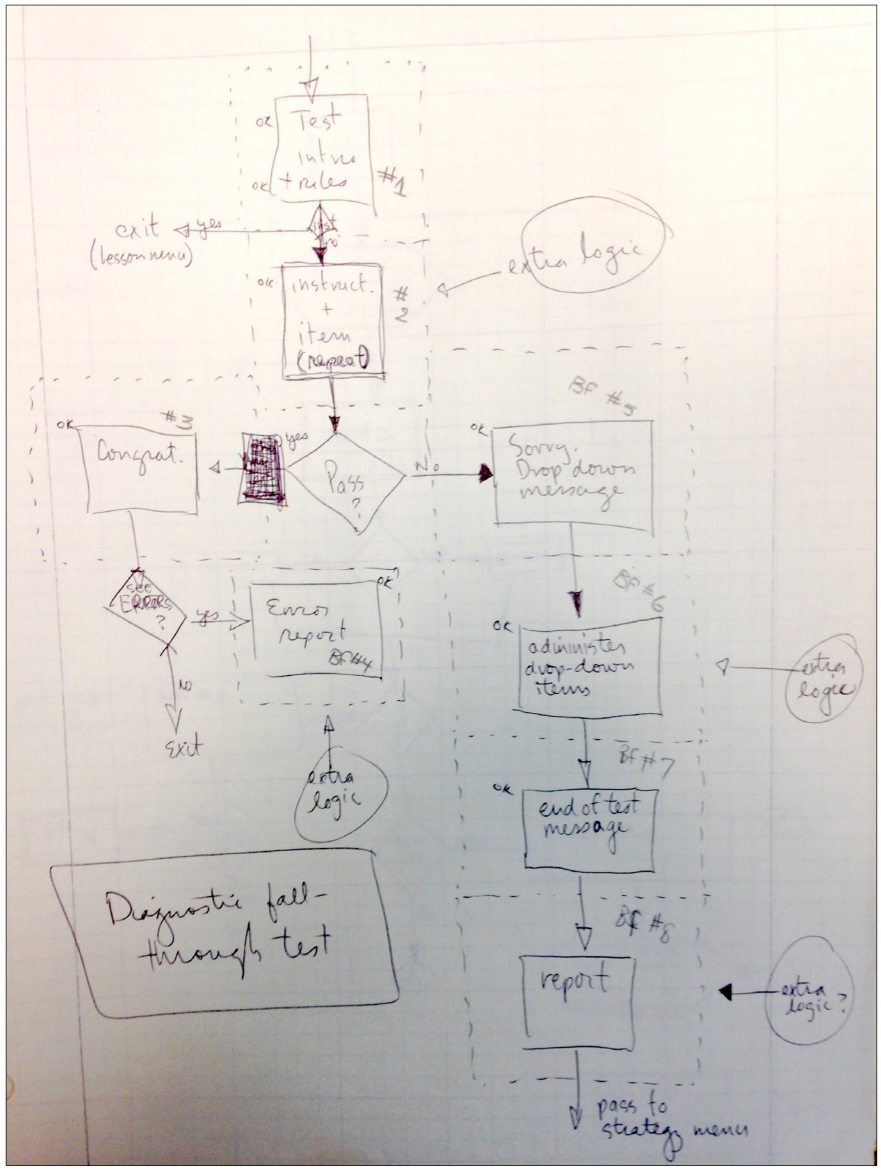

FIGURE 2. The segmentation of computer logic (flowchart) into base frames that relate display content to interaction and computer logic. For example, the "test intro rules" base frame might involve the presentation of multiple representation elements, but during their presentation and the interactions related to them, there would be an element that did not change (e.g., background, visual, etc.). Only upon exiting from one base frame to another did the entire display change and provide a new context for the display.

control, as a small number of options would end up being interesting, and they would be at a relatively large level of granularity. Over time a kind of symbol system evolved, where multiple repetitions of example/non-example pairs were represented with a "Δ" symbol and sequences of definitions and example sets were concatenated with "+" or "-" signs. Only "-" is shown in Figure 3. Figure 4 shows how long strategy sequences could be collapsed using a sub-index notation.

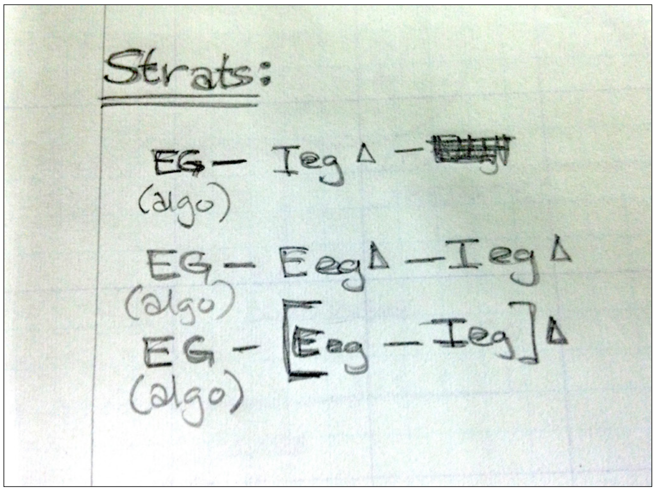

FIGURE 3. Early notes on strategy patterns produced by combination and recombination of basic elements derived from Merrill's concept learning research.

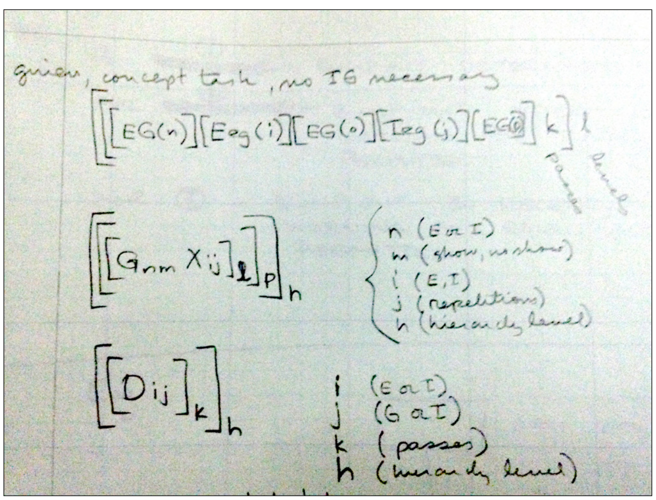

FIGURE 4. Notes on an attempt to represent strategy patterns made of definitions (G) and exemplars (eg) of different kinds into Algebra-like expressions.

The problem with the solution at this stage was that we were thinking in terms of one type of learning objective (concept learning) and that the strategies consisted of long sequences of small events. Though this did give the learner choices to make, it did not seem to be a very good implementation of the ideal of learner control. This imperfect approach might have technically solved the problem, but it was not true to the spirit of creating a system for learner control. A more granular approach soon became apparent.

Dave Merrill arrived one morning in a state of high excitement and called together the TICCIT design team. He explained how the display types could be applied to a number of objective types at the level of the individual display. This not only provided the learner control at a much higher level of granularity, but it could be applied across multiple learning objective types. Dave was presenting what eventually became published as Component Display Theory. It was adopted as the fundamental strategic plan for the TICCIT system.

Component Display Theory, when combined with a typology of instructional objectives, created a language of instructional strategy elements that was needed to afford the TICCIT system a fine-grained form of learner control. It defined a system that was inert until the learner executed a control. To describe how these elements came together, it is necessary to briefly describe: (1) Component Display Theory, (2) the objective typology and its interaction with display types, and (3) the impact this had on the design of control system the learner would use to order up instruction. These things are described in the next few sections.

COMPONENT DISPLAY THEORY

The central premise of Component Display Theory (Merrill, 1983) is a combination of two assumptions: (1) that a given display is either expository (providing information) or inquisitory (requesting a response), and (2) that a given display may work at the content level of a generality or an instance. A basic set of displays is derived from crossing these dimensions (see Figure 5).

The design team used the following abbreviations for the display types:

- EG – Expository Generality
- IG – Inquisitory Generality
- Eeg – Expository Instance
- Ieg – Inquisitory Instance

Additional display types were added, but they were auxiliary types defined relative to this basic set. The complete set will be described below.

The Objective Typology

Assumptions were made: (1) that instructional strategies would correspond to instructional objective types, and (2) that TICCIT would support only a limited number of instructional objective types. Gagné (1965) had connected types of learning outcomes not only with strategic moves, but with "the implications…advances in knowledge have for the formation of what has come to be known as instructional theory" (Gagné, 1985, p. xi). Gagné realized that objectives categories were based on informed assumptions:

> *Eight different classes of situation in which human beings learn have been distinguished—eight sets of conditions under which changes in capabilities of the human learner are brought about…. From the standpoint of the outside of the human organism, they seem to be clearly distinguishable one from another in terms of the conditions that must prevail for each to occur. Might there actually be seven, nine, or ten, rather than eight? Of course.* (Gagné, 1965, p.57)

Figure 5 shows how the design team envisioned their selection of instructional strategies within Merrill's adaptation of Gagné's instructional objective types.

For the purposes of the TICCIT design, three instructional objective types were adopted: concept-using, procedure-using, and principle-understanding. The list of objective types deliberately excluded memory objectives in order to oppose the common tendency of computerized instruction designers to create drill and practice instruction instead of putting emphasis on more complex types of learning.

Why was this particular set of learning objective types chosen? Why not some other set? Gagné's statement above shows that the taxonomic principle as practiced at the time (by a large number of taxonomists) was both subjective and objective. A review of Gagné's categories over the twenty years of the publication of *The Conditions of Learning* makes it plain that the learning objective categories he chose were

	EXPOSITORY	**INQUISITORY**
GENERALITY	(EG) Presentation of a generality	(IG) Request for the generality
INSTANCE	(Eeg) Presentation of an instance (example/non-example single or pair)	(Ieg) Request for a response (regarding an example or a non-example)

FIGURE 5. The basic display set obtained by crossing two assumed dimensions: expository/inquisitory and generality/instance (see Merrill & Twitchell, 1994; Merrill 2008).

sensitive to changes in the learning theory landscape. Over that period the categories evolved from behavioristic, to cognitive in their basis.

At the time of TICCIT, Dave Merrill, who was influenced by Gagné, was experimenting with his own ideas, producing multiple versions of learning objective taxonomies over a period of years. His concern was to represent the influence of content separately from behavior. That is, he wanted there to be a concept, procedure or a principle content that could be the subject of multiple types of behavior. This point of view led him over time to produce a version of his taxonomy that took the form of a matrix, rather than a list. The taxonomy used in the TICCIT system was one version of this evolving idea.

OBJECTIVES HIERARCHIES

Subject matter experts and designers worked together to create TICCIT instructional goals using Gagné's learning hierarchies method (Gagné, 1968; 1977), an experimental method at the time that provided guidelines for analyzing intellectual skills. The result of learning hierarchy analysis—a specialized variety of task analysis—was a set of objectives organized in what was thought to be prerequisite order. Objectives lower in the hierarchy were considered necessary for the performance of objectives at higher levels.

Hierarchical analysis provided TICCIT designers a systematic way to sequence objectives into "maps" that could be organized into Units, Lessons, and Segments. Lessons and units were used as testing points. Instruction took place within segments. TICCIT tests at the unit and lesson levels assessed mastery and were also diagnostic, pointing to remediation that might be needed at lower hierarchical levels. A color code was used on course, unit, and lesson maps to indicate which tests had been passed at lower levels. This turned the course, unit, and lesson maps into a status display. Learners were allowed to enter segments and browse units and lessons at will. This constituted a form of learner control over content. "Mini lessons" were created for each segment to allow learners to survey content.

Designing the Full Set of Display Types and the Control System

Display types resided within segments. The four basic display types were pared to three, based on the assumption that the Inquisitory Generality (IG), a request for memory-level behavior, did not represent a desirable form of practice. It was a concern that designers might fall back to memory-level performance for concept definitions and rules, rather than asking the learner to make classifications and exercise their knowledge of procedures and processes.

Several display types were added to the basic set. These were logically derived from the basic display types to support learning in a variety of ways. It is easiest to explain the logical and consistent relationships among the displays by examining the controls provided to learners for navigation within a segment. These controls were located on a special keypad placed on the right side of the custom TICCIT keyboard (see Figure 6).

The **RULE**, **EXAMPLE**, and **PRACTICE** keys on the keypad correspond to the EG, Eeg, and Ieg display types respectively (refer to Figure 6).

When a learner entered a segment, control keys provided access to the displays within the segment. The basic displays included:

RULE: Expository generality (EG)

For concepts, the concept definition
For rules, a statement of the rule

EXAMPLE: Expository Instance (Eeg)

For concepts, a medium-difficulty example or non-example

For rules, a medium-difficulty demonstration of the procedure or process

PRACTICE: Inquisitory instance (Ieg)

For concepts, a medium- difficulty classification problem

For rules, a medium-difficulty request to apply the procedure/process

One set of controls regulated the technical level of explanations and the difficulty of practice items:

- **HARD**: A harder, more technical version of either the generality or the instance
- **EASY**: An easier, less technical version of either the generality or the instance
- (The **EASY** level of technicality was the default, pressing **HARD** or **EASY** ratcheted the level of difficult one level up or down.)

The **HELP** control provided expanded explanations for instances and practice items at every level of technicality.

The **OBJECTIVE** control provided access to the instructional objective of the segment:

Several keys performed administrative functions, such as session control:

- **ATTENTION**: Get the attention of the system or a monitor
- **EXIT**: Leave the current session
- **REPEAT**: Repeat the segment just completed

Other controls were for navigation through topic hierarchies:

- **MAP**: Move up one level of the hierarchy to view the map at that level
- **GO**: Enter segment for instruction
- **SKIP**: Skip an item
- **BACK**: Go back an item

An **ADVICE** key provided learners access to strategic assistance. This function was partially implemented and was planned to grow into an intelligent coach. The Advisor function as implemented gave learners access to status information, strategy suggestions, and directions for using the system.

Figure 7 shows the major navigational paths through the displays of a TICCIT segment. Not all paths are shown because doing so would make the diagram unreadable. The numbers in the boxes of Figure 2 are used by Merrill et al (1980) to demonstrate the progress of two fictive learners through a TICCIT segment. The non-circled numbers show the path of Learner #1, and the circled numbers trace the path of Learner #2.

For example: Learner #1, while looking at a lesson map:

1. Presses **GO** to automatically view the mini lesson
2. Presses **MAP** to return to the lesson map
3. Presses a number and **GO**, to highlight the segment box on the lesson map
4. Presses **OBJECTIVE** to view the segment objective and enter the segment
5. Presses **RULE** to view the generality (concept definition or rule expression)
6. Presses **HELP** to see a simplified explanation of the generality
7. Presses **EXAMPLE** to see an easy example (concept instance or worked rule)
8. Presses **HELP** to see a simplified explanation of the example. And so forth…

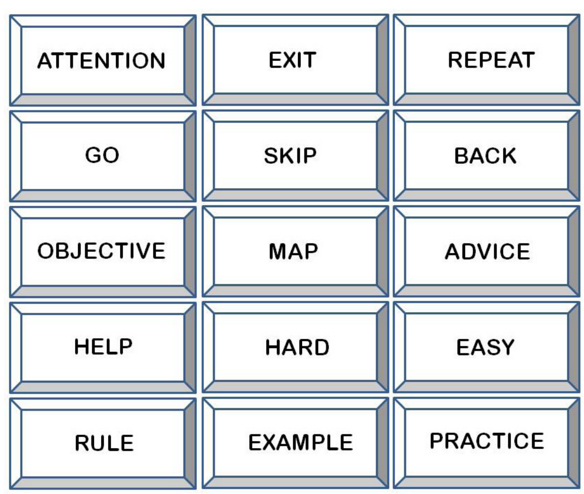

FIGURE 6. The custom controls for learner navigation of TICCIT segment displays (after Merrill et al, 1980).

This example shows that despite the complexity of the Figure 3 diagram, from the learner's point of view, the keypad controls, combined with the consistent definition of each display type created a simple language the learner could use to interrogate the system and chart a personal strategy. Control choices from the learner's point of view are best understood in the context of where the learner was at any given moment and the question the learner might have in mind, which would determine a useful next move. Learners did not advance until they requested a display with a keypress.

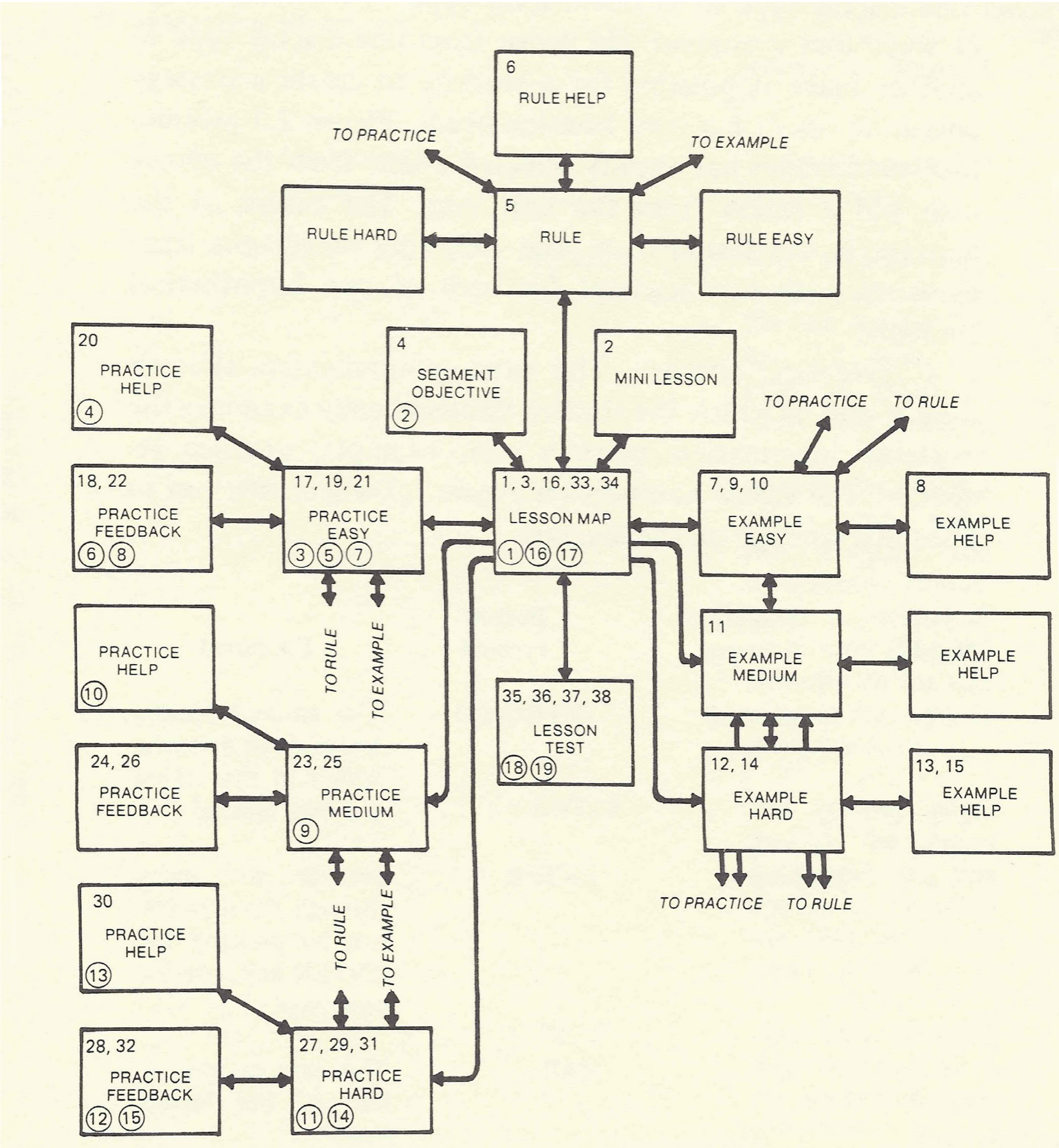

FIGURE 7. Navigational paths among TICCIT display types from Merrill et al, 1980, used with permission).

INTEGRATION OF THE STRATEGY AND LOGIC LANGUAGES

The purpose of this design case is not to defend a particular choice of display types, objective types, or controls. Neither is it written in support of a particular style or philosophy of instruction. The purpose is to show how strategic design assumptions, and a set of logic primitives represented by base frames, created an engineering meta-language that made possible a unified, coherent design for TICCIT system instruction. This language allowed design team members from different specialties to discuss the design itself and create functional interfaces among conceptual strategic entities and practical software elements. We believe that a diverse design team must either implicitly or explicitly form such a language. We propose that design teams can come together more efficiently if they realize that language formation is an important aspect of design work.

The focal point of the different languages was the display. A display type, the basic atom of the TICCIT design, represented the integration point of: (1) the strategic design, (2) the goal structure represented by the objectives, (3) the content structure implied by the objectives, (4) the control structure, and (5) the message and representation structures, which will be defined in the next section.

The display functions chosen— which could easily have been a different set in a different designer's hand— made it possible to correlate the languages of all the functional areas of the design into a single design meta-language. The comprehensive TICCIT instructional design was expressed in terms from this meta-language. We feel this provides a testable design hypothesis: whether this concept of convergent languages might apply as well in the design of non-direct forms of instruction, such as intelligent tutoring, as well as it did to a direct form of instruction represented in TICCIT. We hypothesize that this is the case but that the size of the atomic unit in a more adaptive system will be smaller. In any design, regardless of the granular level of its adaptivity, certain functional decisions must be made (Gibbons, 2014), and the number of alternative paths ahead at any point during instruction cannot be assumed to be infinite, meaning that there is likely to be a finite, and perhaps not too large, number of design options at the heart of any design, around which other design decisions must center.

AUTHORING

The TICCIT design became firm over time, but that did not occur all at once. The best term to describe the firming up of the design might be "settling in". How the disparate parts of the TICCIT design came together in the minds of the entire design team—the subject matter experts who had to do the authoring described in this section, the programmers, the artists, the writers, the editors, the instructional designers, the quality control people, the data entry personnel, and the formative evaluators—may have been best described by Bucciarelli (1994):

> *Shared vision is the key phrase: The design is the shared vision, and the shared vision is the design—a (temporary) synthesis of the different participants' work…. Some of this shared vision is made explicit in documents, texts, and artifacts—in formal assembly and detail drawings, operation and service manuals, contractual disclaimers, production schedules, marketing copy, test plans, parts lists, procurement orders, mock-ups, and prototypes. But in the process of designing, the shared vision is less artifactual; each participant in the process has a personal collection of sketches, flowcharts, cost estimates, spreadsheets, models, and above all stories—stories to tell about their particular vision of the object…. The process is necessarily social and requires the participants to negotiate their differences and construct meaning through direct, and preferably face-to-face exchange.* (p. 159)

The shared TICCIT design vision emerged through negotiation. Leading minds may have seeded and catalyzed the core concepts of the design, but independent specialists on the design team then had to give detail to the design within their area of responsibility, and as conflicts in the details of the design occurred, team members had to negotiate a way through. (See Baldwin & Clark, 2000 for an extensive discussion of the idea of the setting of design rules, followed by negotiation of details in the computer design industry).

Just as the design process began from opposite poles of the problem (logic and strategy) it was finished beginning at the center and moving outward. The core of the design, once it was firm, began to discipline the details at the outer edges of the design. One of the areas where this had to happen was in a set of authoring standards that ultimately described in great detail what could and couldn't be included in each of the defined display types. Standards were created for artists, writers, editors, data entry personnel, and other production team members defining the points of quality for their step in the assembly process.

This was an engineering necessity because of the large volume of instructional material that had to be created in a relatively short period of time at a high level of consistent quality. What was called for was an assembly line, and in order for that kind of production to take place, standards had to be set and maintained for each producible element. Moreover, as tweaks became necessary or as errors in the details of the production standard were spotted, changes had to be made and communicated to the entire production staff. The best description of what happened can be found in the *Lean* processes described by Womack and Jones (2003).

The assembly line metaphor, just like the engineering metaphor communicates to some a cold, perhaps mechanical, sense to most instructional designers, and it is not a popular topic in the design literature. However, the members of the TICCIT assembly line were intelligent, educated, and creative people. Their job was to work within a framework of constraints, producing a continuous flow of product, but to do it without giving up the creative edge of professionals. This requirement created a kind of living oxymoron: academics and knowledge workers defining their own mass production machinery. As competitive forces from entrepreneurial and commercial organizations apply new standards of quality-at-volume to the design marketplace, the issue of volume production at high quality levels will become increasingly important in the training of instructional designers. Stokes (2005) describes how a number of leading creative minds from a variety of design disciplines imposed constraints upon themselves specifically to challenge themselves to higher levels of creative insight. Perhaps in the future the competitive requirements of a mass educational market will have a similar effect.

IMPLEMENTATIONS AND EVALUATIONS

The subject of this case study is the evolution of the TICCIT design, not the specific courses implemented on TICCIT. However, the implementations and evaluations of TICCIT make it possible for us to speculate about the theoretical impact of the design.

Implementations

Several implementations of the TICCIT system design were made. These consisted of course, or partial-course, developments and productions, followed by field trials:

- Junior College courses in Mathematics and English grammar were created and implemented with over 5,000 Junior College students at Northern Virginia Community College and Phoenix College.
- Course materials for instruction in foreign languages, English, and general academic skills were developed and tested at the Model Secondary School for the Deaf at Gallaudet University.
- A course in Oceanography was created by the U. S. Navy for use in the training of anti-submarine aircrews.
- An experimental course in aircraft systems operation was created and tested for use in S-3A sensor operator training at the Naval Air Station, North Island, California.
- Courses or partial courses were developed at Brigham Young University in Critical Reading, English as a Second Language, French Grammar, Spanish Grammar, and Nursing. Segments were created and tested in German Phonetics and Italian Grammar. Much of the course development was In service to BYU's interest in language instruction, and smaller projects were carried out for research purposes. Pedersen (1985) notes that the BYU TICCIT system was still in use 12 years after its development, and the authors have ascertained that a version of TICCIT that has been ported to a new operating environment multiple times is still in regular use today.

Evaluations

Alderman (1978) reported an extensive evaluation of the TICCIT system in the Junior College implementations. Alderman, Appel, and Murray (1978) published a shorter report that describes the TICCIT and PLATO evaluations funded by the NSF. Evaluation reports from the private educational institutions and military organizations could not be obtained. A personal reminiscence about the TICCIT project by the project principal C. Victor Bunderson (2008) provides a valuable documentation of lessons learned from the TICCIT project. M. David Merrill (2008) also provides some insight into the impact of the TICCIT design.

The Junior College Evaluation

The implementation of TICCIT with 5,000 students at the two junior college sites in Virginia and Arizona was bumpy. It experienced about every irregularity one would normally expect in a real-world environment. This might raise anxieties that the main effects would be hidden, but in fact this was the real environment in which the system was designed to operate, and if the desired real effects were strong enough, they should be detectable amid the noise. In the final analysis, two main effects seem to dominate the results of the trial: an instructor effect, and a practice effect.

The Instructor Effect

Instructors used the TICCIT instruction in ways that suited their own teaching style. They had their own views and expectations of how it would work, and they had never experienced computer-assisted instruction themselves. Therefore, they had no familiar usage patterns to fall back on. Teachers were inventing how to use TICCIT as they went. This unexpected—at least unanticipated—outcome underscores the importance of current research and training in the area of blended learning that can support an instructor in incorporating diverse media forms into an instructional plan.

A variety of usage patterns were observed. Byerly reports, "results were better when the programs were used as a supplement to class instruction. The effect on student morale was quite positive, and there was a 5% increment in student achievement in one English course" (p. 282).

Alderman (1978), the primary NSF TICCIT evaluator, expressed the opinion that the instructor effect was probably more powerful than the effect of the TICCIT instructional design. He observes:

> *Instructor investment in these courses varied from direct supervision of all student work to supplementary assistance provided upon student request. In English courses instructors tended to choose the TICCIT lessons appropriate for their classes and to take an active role in assigning and correcting written exercises; instructors in mathematics courses, where department policy set the TICCIT coverage according to curriculum requirements, had responsibility for managing classes sometimes three times the size of usual lecture sessions…* (p. 41)

Alderman reported that some college administrators used the TICCIT system as an opportunity to increase lecture course sizes, making comparison with traditional class sizes difficult. Alderman further notes that, "most often, faculty indicated that they were unsure about the probable impact and significance of computer-assisted instruction" (Alderman, 1978, p. xxiv). However, he notes that: "completion rates as well as student attitudes improved for the TICCIT program as the teacher's role under the program expanded" (Alderman et al, 1978, p. 45).

The Practice Effect

Alderman also notes a PRACTICE key effect:

> *PRACTICE was the one system feature that received high ratings by students both in comparison to its closest counterpart in lecture classes (i.e., homework assignments) and contrasted with ratings of other components of TICCIT's learner control. The practice problem appeared to be the cornerstone of the TICCIT system.* (p. xxiii, emphasis in the original)

Students using the TICCIT system wanted to press the PRACTICE key early and often. There could be many different explanations. Perhaps they wanted:

- To see what the test was going to ask for.
- To see if they already knew the subject matter.
- To move sooner to a hands-on learner role.
- To avoid being placed in a receptive role.
- To obtain data for making strategy decisions.

Whatever the reason a particular learner had, it is clear that the learners who used the TICCIT system wanted frequent interaction. Even within the structured world of TICCIT, this leads to speculation about the many possible alternative configurations of the display definitions and the controls that might have been designed, which could be focused more centrally on practice activities and the scaffolding of practice. We could ask: what would the control keys for such a TICCIT configuration look like? Current research literature provides many ideas ready-made for implementation in a learner-demand system that would allow the user to participate in determining the most useful control options, rather than the imagination of the designer alone. Koedinger's Knowledge-Learning-Instruction Framework (2012) reviews practice-centered instructional methods from learning sciences literature. These suggest a range of learner controls that could be made available during instruction.

Early versions of the TICCIT control keys, some of which were dropped from the design, anticipated some of these initiatives: HELP at one point was to be augmented with SUPER HELP, which would have provided deeper explanations of concepts and processes; the NOTE key, which did survive, would have provided the opportunity for student reflections and insights to be recorded. At one point a SO WHAT? key was considered and then rejected. However, the most interesting key that never made it to the keyboard was the WHY? key, which "why?" Koedinger (2012) suggests is essential to the attainment of the highest and most complex forms of knowledge. A later section describes the WHY? key issue in more detail.

WHO PROFITED MOST FROM TICCIT?

In general, the results of the TICCIT trials in Junior Colleges (Alderman, 1978) showed that students who were self-directed and self-controlled users of the TICCIT system scored higher on independently administered posttests, as did students who had prior familiarity with the subject matter.

Course completion rates were significantly lower for TICCIT classes than for their control group counterparts (16% versus 50% for math, and 55% versus 66% for English). However, those who did compete the TICCIT courses tended to score significantly higher on posttests. The TICCIT design appears to have been effective for a portion of the learners.

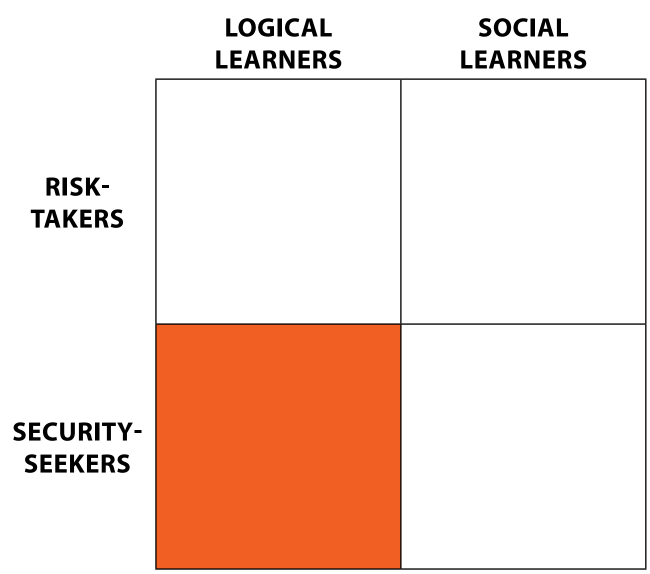

FIGURE 8. Bunderson's assessment of the factors defining learners most compatible with TICCIT instruction.

Bunderson (2008) speculates that learners who were likely to feel most comfortable with the highly structured TICCIT instruction were those in the shaded portion of Figure 8, which includes learners who approach their task in a logical manner and who also prefer the predictability created by consistent, logical structuring of the courseware.

One significant finding came from a test of TICCIT at BYU subsequent to the formal evaluation at the Junior Colleges. A first trial of the BYU system placed no constraints on students relative to semester boundaries. Students could finish when they wanted to. In this trial the usual low completion rates were noted. However, Byerly (1978) describes the results of a second test conducted at BYU that constrained students to finish the course within semester boundaries, according to a set schedule:

> *Not surprisingly, the TICCIT programs have been more apparently successful at Brigham Young University, the institution funded by the National Science Foundation to prepare them, where several hundred freshmen use the system each semester. The completion rates are 10-20% better than those of standard courses, and a survey found 9% more TICCIT than regular students passed a departmental achievement test.* (pp. 282-283)

The Orientation Effect

Bunderson (2008) has speculated about a potential negative effect of the hierarchical system of maps, which represented the subject matter to the learner in a fragmented manner:

> *It is clear that the controls provided and the information displayed on the maps was not sufficient to achieve the broader vision of what learner control is all about For one thing, it did not span an adequate range of preferences as shown in different models of thinking and learning preferences.* (Bunderson, 2008, p. 13)

Bunderson proposes that, "what some of the students needed was to see the big picture more completely than the TICCIT map hierarchies conveyed. He cites as an alternative the concepts of "work models" and "elaborations" that grew out of TICCIT design discussions. Work models (Bunderson, et al, 1981; Gibbons, et al, 1995) progressively group performance goals into more complex performances for the integration of learning in increasingly challenging steps. This concept is closely related to the idea of increasingly complex microworlds described by Burton, Brown, and Fischer (1984) and Vygotsky's Zone of Proximal Development (1978). Elaboration Theory (Reigeluth, 1999) is also a direct response to the fragmentation of subject matter by the TICCIT hierarchical maps and it grew out of a search for an alternative.

The WHY? Key

Many discussions during TICCIT design centered on the WHY? control key, which was omitted from the final design but which existed in keypad designs up to the last moment. Discussions of the WHY? key always opened a Pandora's Box. It may have been due to the fact that it represented a watershed issue in the construction of adaptive, learner controlled, and intelligent instructional systems.

The classical approach to adaptive tutoring system design (See Wenger, 1987) relies for the most part on a semantic representation of the subject matter. In such systems, models of the learner, the learner's knowledge state, and the subject matter are used to inform instructional decisions, which are made using strategic rules that "understand" the semantic of the content and can therefore construct explanations on the fly.

In the TICCIT design, as with other frame-based designs, there was only a strategic semantic, which corresponded directly to the control keys. The system "understood" the categories of strategic frames, but the specific content and representation resources within those frames was pre-composed and fixed, sitting in a data base, waiting to be called up. TICCIT did not have the ability to tailor the subject matter or message in response to a learner query. The learner was not allowed to ask "Why?" because the system had no way to understand and answer the question..

The WHY? key had to be abandoned because it represented a step too large for the architecture of the TICCIT system. But for many of us, it still poses a practical question about how tools for making adaptive, intelligent instructional designs can be placed in the hands of the average designer and how the instruction created can be placed under the control of the learner. If an instructional artifact can answer a "WHY?" question, then truly conversational instruction becomes a possibility.

SUBSEQUENT HISTORY OF THE TICCIT DESIGN

The design of the TICCIT system was sold after the trial period to Hazeltine Corporation, then to Ford Aerospace, then to Loral, a French company. The turbulent computer and software landscape produced tectonic forces that over time shredded the original system. The minicomputer gave way to the microcomputer. Operating systems changed radically in their nature. Programming tools advanced and became more powerful, obsoleting their predecessors. The Internet became a distribution channel capable of exporting courseware easily and widely. In response to changes, the TICCIT system was ported several times to new hardware and software platforms, each time losing some of its functionality, including modifications to the control set. Today, the only known operating TICCIT system exists at Brigham Young University, where it is used to teach languages.

CONCLUSION

We believe the TICCIT project was revolutionary in its time and that its effects are still being felt in many ways. A project as visible, ambitious, and innovative as the TICCIT project was sure to leave behind a legacy of new learning about design. Ironically, the answer to the question of whether computer-based instruction could be used in a cost-effective way for instruction became a foregone conclusion, as prices for hardware, software, development, and delivery fell precipitously. The instructional computer was inevitable, but few realized it at the time of the TICCIT project. According to Suppes (1979, quoted in Bunderson, 2008, p. 8):

> *It is rather as if we had had a similar test of automobiles in 1905 and concluded that, given the condition of the roads in the United States, the only thing to do was to stay with horses and forget about the potential of the internal combustion engine.*

Today, when it is almost always assumed that the computer will play some part in instruction (even if only to project PowerPoint slides), the desperate struggle that filled our eyes with smoke and our ears with the noise of battle during TICCIT looks more like a dust devil on the far horizon. Today horses are not allowed on the freeway, and the instructional computer is a fact of life.

As stated at the beginning, the goals of the TICCIT system were:

- Full learner control over instruction
- "Engineered" courseware production
- Rapid production of a large volume of material
- Lower costs for development and delivery of computer-based instruction

Without quoting statistics, we will say that these goals were reached, and a design was produced that was applicable to a wide range of subject matters. Costs of development and delivery were drastically reduced (to about 10% of usual CAI costs), and a new computer system configuration was tested—a local-system concept that stood in clear contrast to the monolithic system style of the day. At the time of the project, these were major accomplishments, and they initiated a revolution in thinking smaller about the design and delivery of computer-based instruction. Ironically, the systems we use today are both local in the power they provide each user and monolithic in the access they provide to an immense worldwide network.

Other TICCIT outcomes included:

- The team design and production concept showed that it was possible to specialize development functions to achieve mass production of a quality product.
- The creation of specialized design languages and non-programming interfaces allowed a team of designers and producers to speak to the computer in a new human-computer pidgin.
- Control over instruction was placed in the learner's hands to an extent beyond the standard definition of "learner-controlled" at the time. Today, a similar, more advanced control system is available in the Web browser.
- A new componentized definition of instructional strategy was originated that focused increased attention on the analysis of content structures, strategic structures, and the architecture of instructional designs in general at a whole new level of detail.

TICCIT was a bold departure from the existing norm. It generated new theory (e.g., component display theory, elaboration theory, work model theory). It laid the groundwork for several spin-off instructional design firms that carried the TICCIT ideas forward, changing and abstracting them to produce new conceptions of design architecture (Gibbons, 2014).

The perspective of time has made it possible to see some things now that weren't apparent at the time. It was surprising, for example, the extent to which the strategic design was based on a number of key *assumptions* and that a number of assumptions together created a complete design hypothesis that could be (and was) tested. Also, it has become apparent that it was not a single theory that informed the design but many theories, each acting to inform a specific part of the design independently: theories of representation, of message structuring, of control optioning, of data management, and system logic.

The making of these assumptions and the application of many engineering theories was an essential part of the design effort, without which design could not have proceeded. Anderson (1961) points out the importance of engineering knowledge by stating that scientific study results are insufficient to produce a design:

> *Engineering plays a critical role in the application scientific principles in any area. Take the development of rockets, for example. One could not deduce a rocket or even a blueprint for a rocket solely from the principles of physics or chemistry.... The product of basic research is a statement about the relationship among variables. It is my contention that these statements are never of very much direct value to practical educators, even when the statements are perfectly understood and every effort is made to apply them.* (p. 377)

No amount of research and data could have told the TICCIT designers what to include in the design. Research and "science" can only inform designs. They cannot determine designs, because other kinds of knowledge are used during design that science does not provide (Vincenti, 1990).

The testing of the system by instructors showed that instructors with the proper vision could adapt TICCIT lessons within a larger instructional plan that enhanced the ability of the teacher and allowed instruction by teacher and computer to reach a higher standard of achievement than either one alone.

A language of strategic events was successfully mapped to a language of computer events in a seamless meta-design language that integrated computer and strategic functions into a coherent design. The concept of the internal structure of an instructional design at several levels became apparent, but only over time and much additional experience. It should be noted that the assumptions made to form the basic display types and the instructional objective typology were arbitrary, even though they were based to some extent on a research foundation (generality and instance) and a reasonable premise (expository and inquisitory). It should be obvious that the four basic display types were just the core of a larger, abstract set of display types that could have been created by any combination of matrixed dimensions. The number of display types is not absolute, nor is the number of dimensions. However, the ability to categorize messages that populate a display is a powerful concept, related in a reverse way to the ontological analysis of documents. This is an important theoretical take-away.

It would be very neat, and yet untrue, to say that there was a single grand principle behind the TICCIT design from the start that unfolded inevitably. On the contrary, as each part of the TICCIT design unfolded, it was through the discovery of a way forward for one part of the design, and that almost always opened the way for another important discovery. The base frame innovation came early because it was required by the programmers in order to meet their development schedules. The strategic innovation came as the extension of Dave Merrill's research until the pieces fell together with Dave's aha-moment.

It would also be neat theoretically to be able to say that once the pieces fell together the grand conception of design languages, sub-function interfaces, and abstract layers of the design (Gibbons, 2014) emerged immediately in everyone's mind. This would also be an exaggeration. The serendipity that led a creative team out of the wilderness was delivered a bit at a time, and only the perspective of over forty years is bringing the bigger picture into focus.

Most of all, what the TICCIT system represents to those who experienced its design is boldness. The designers of the system moved on to instructional design positions in industry and eventually to universities and research laboratories. Their views of instruction changed, but the TICCIT experience had revealed to them that there is value in radical innovation and that everyone is empowered to have bold ideas. This knowledge is a gift that the team members received from the visionaries who initiated the project, dealt with its enormous demands, and led it to a successful conclusion. Just as we need more researchers and designers today, we need more visionaries to challenge our thinking and show us new ways ahead.

REFERENCES

Alderman, D. L. (1978). *Evaluation of the TICCIT computer-assisted instructional system in the community college. Final report*. Volume I. Washington, DC: ERIC Clearinghouse [ERIC ED167 606].

Alderman, D. L., Appel, L. R. & Murray, R. T. (1978). PLATO and TICCIT. *Educational Technology, 18*(4), 40-45.

Anderson, R. C. (1961). The role of the educational engineer. *Journal of Educational Sociology, 34*(8), 377-381.

Atkinson, R. & Wilson, H. (1969). Computer-assisted instruction. In R. Atkinson and H. Wilson (Eds.) *Computer-assisted instruction: A book of readings* (pp. 3-14). New York, NY: Academic Press.

Baldwin, C. & Clark, K. (2000). *Design rules: The power of modularity.* Cambridge, MA: MIT Press.

Bloom, B. S., Engelhart, M. D., Furst, E. J., Hill, W. H., & Krathwohl, D. R. (1956). *Taxonomy of educational objectives: The classification of educational goals. Handbook I: Cognitive domain.* New York, NY: David McKay Company.

Bransford, J., Brown, A. & Cocking R. (2000). *How people learn: Brain, mind, experience, and school: Expanded edition*. Washington, DC: National Academies Press.

Branson, R. K., Rayner, G. T., Cox, J. L., Furman, J. P., King, F. J., & Hannum, W. J. (1975). *Interservice procedures for instructional systems development* (5 vols.) (TRADOC Pam 350-30, NTIS Nos. AD-A019 4860, AD-A019 490). Ft. Monroe, VA: U.S. Army Training and Doctrine Command.

Bitzer, D. L. & Johnson, R. L. (1971). PLATO: A computer-based system used in the engineering of education. *Proceedings of the IEEE, 59*(6), 960-968.

Briggs, L. J. (1967). I*nstructional media: A procedure for the design of multi-media instruction, a critical review of research, and suggestions for future research*. Washington, DC: American Institutes for Research. (Preceded by: Briggs, L. J., Campeau, P. L., Gagné, R. M. & May, M. A. (1966). *A procedure for the design of multimedia instruction* (Abbreviated title). Pittsburg, PA: American Institutes of Research).

Briggs, L. J. (1970). *Handbook of procedures for the design of instruction* (Monograph No. 4.). Washington, DC: American Institutes for Research.

Bruner, J. S. (1964). Some theorems on instruction illustrated with reference to mathematics. In E. R. Hilgard (Ed.), *Theories of learning and instruction: The sixty-third yearbook of the society for the study of education* (pp. 306-335). Chicago, IL: National Society for the Study of Education.

Bruner (1966). *Toward a theory of instruction*. New York, NY: Norton.

Bruner, J., Goodnow, J. & Austin, G. (1956). *A study of thinking*. New York, NY: Wiley.

Bucciarelli, L. L. (1994). *Designing engineers.* Cambridge, MA: MIT Press.

Bunderson, C. V., Gibbons, A. S., Olsen, J. B. & Kearsley, G. P. (1981). Work Models: Beyond instructional objectives. *Instructional Science, 10*, 205-215.

Bunderson, C. V. (2008). Reflections on TICCIT. In M. Allen (Ed.), *Michael Allen's e-Learning Annual, 2008* (pp. 1-30). San Francisco, CA: Pfeiffer.

Burton, R. R., Brown, J. S. & Fischer, G. (1984). Skiing as a model of instruction. In B. Rogoff and J. Lave (Eds.), *Everyday Cognition: Its Development in Social Context* (pp. 139-150). Cambridge, MA: Harvard University Press.

Byerly, G. (1978). CAI in college English. *Computers and the Humanities, 12*(3), 281-285.

Dubberly, H. & Evenson, S. (2010). Design as learning—or "knowledge creation"—the SECI model. *Interactions, 18*(1), 75-79.

Gagné, R. M. (Ed.)(1965). *Psychological principles in system development*. New York, NY: Holt Rinehart & Winston.

Gagné, R. M. (1968). Learning hierarchies. *Educational Psychologist, 6*, 1-9.

Gagné, R. M. (1977). Types and capabilities of learning hierarchies in instructional design. *Journal of Instructional Development, 1*(1), 8-10.

Gagné, R. (1985). *The conditions of learning* (4th ed.). New York, NY: Holt, Rinehart, and Winston.

Gibbons, A. S. (2014). *An architectural approach to instructional design.* New York, NY: Routledge.

Gibbons, A. S., Bunderson, C. V., Olsen, J. B., & Robertson, J. (1995). Work models: Still beyond instructional objectives. *Machine-mediated learning, 5*(3&4), 221-236.

Gibbons, A. & Fairweather, P. (1998). *Computer-assisted instruction: Design and development*. Englewood Cliffs, NJ: Educational Technology Publications.

Gibbons, A. S. & Rogers, P. C. (2009). The architecture of instructional theory. In, C. M. Reigeluth and A. Carr-Chellman (Eds.), *Instructional-design models and theory: Volume III, Building a common knowledge base.* New York, NY: Routledge.

Glaser, R. & Resnick, L. B. (1972). Instructional psychology. *Annual Review of Psychology, 23*, 207-76.

Koedinger, K. R., Corbett, A. T. & Perfetti, C. (2012). The knowledge-learning-instruction framework: Bridging the science-practice chasm to enhance robust student learning. *Cognitive Science, 36*(5), 757-798.

Lave, J. & Wenger, E. (1991). *Situated learning: Legitimate peripheral participation.* Cambridge, UK: Cambridge University Press.

Markle, S. M. (1964). *Good frames and bad: A grammar of frame writing.* New York, NY: Wiley.

McWilliams, E. (1974). *Hard core CAI: TICCIT system progress report on field testing*. Retrieved on May 20, 2014 from: http://www.atariarchives.org/bcc1/ showpage.php?page=33

Merrill, M.D. (1983). Component display theory. In C. Reigeluth (ed.), *Instructional-Design Theories and Models: An overview of their status* (pp. 397-424). Hillsdale, NJ: Erlbaum Associates.

Merrill, M. D. (2008). Reflections on a four-decade search for effective, efficient, and engaging instruction. In M. Allen (Ed.), *Michael Allen's e-Learning Annual, 2008* (pp. 141-169). San Francisco, CA: Pfeiffer.

Merrill, M. D., Schneider, E. W. & Fletcher, K. A. (1980). *TICCIT* (Instructional Design Library, Vol. 40). Englewood Cliffs, NJ: Educational Technology Publications.

Merrill, M. D., Tennyson, R. D., & Posey, L. O. (1992). *Teaching concepts: An instructional design guide* (2nd ed.). Englewood Cliffs, NJ: Educational Technology Publications.

Merrill, M. D. & Twitchell, D. G. (Eds.) (1994). *Instructional design theory*. Englewood Cliffs, NJ: Educational Technology Publications.

Mitchell, W. J. (1990). *The logic of architecture: Design, computation, and cognition*. Cambridge, MA: MIT Press.

Neisser, U. (1967). *Cognitive psychology*. Englewood Cliffs, NJ: Prentice-Hall.

Pedersen, E. L. (1985). TICCIT will gladly learn and gladly teach composition skills. *Computers and Composition, 4*(SI), 233-241.

Reigeluth, C. M. (1983). *Instructional-design theories and models*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Reigeluth, C. M. (1987). *Instructional theories in action: Lessons illustrating theories and models*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Reigeluth, C. M. (1999). *Instructional-design theories and models, Volume II: A new paradigm of instructional theory.* Mahwah, NJ: Lawrence Erlbaum Associates.

Reigeluth, C. M. (1999). The elaboration theory: Guidance for scope and sequence decisions. In C. M. Reigeluth (Ed.), *Instructional-Design Theories and Models, Volume II: A New Paradigm of Instructional Theory.* Mahwah, NJ: Lawrence Erlbaum Associates.

Reigeluth, C. M. & Carr-Chellman, A. (2009). *Instructional-design theories and models, Volume III: Building a common knowledge base.* New York, NY: Routledge.

Rheinfrank, J. & Evenson, S. (1986). On the semiology of object, space and behavior: The design of strategic design languages. In *Proceedings of the human factors and ergonomics society 30th annual meeting* (pp. 1059-1062).

Rheinfrank, J. & Evenson, S. (1996). Design languages. In T. Winograd (Ed.), *Bringing design to software* (pp. 63-85). Reading, MA: Addison-Wesley.

Richey, R., Klein, J. & Tracey, M. (2010). *The instructional design knowledge base: Theory, research, and practice*. New York, NY: Routledge.

Simon, H. (1969) *The sciences of the artificial* (1st ed.). Cambridge, MA: MIT Press.

Simon, H. (1999) *The sciences of the artificial* (3rd ed.). Cambridge, MA: MIT Press.

Simon, H. A. & Newell, A. (1971). Human problem solving: The state of the theory in 1970. *American Psychologist, 26*(2), 145-159.

Snelbecker, G. (1974). *Learning theory, instructional theory, and psychoeducational design.* New York, NY: McGraw-Hill.

Snow, R. (1977). Individual differences and instructional theory. *Educational Researcher, 6*(10), 11-15.

Stokes, P. (2005). *Creativity from constraints: The Psychology of breakthrough*. New York, NY: Springer.

Stolurow, L. M. (1969). Some factors in the design of systems for computer-assisted instruction. In R. Atkinson and H. Wilson (Eds.) *Computer-assisted instruction: A book of readings* (pp. 65-94). New York, NY: Academic Press.

Suppes, P. (1969). Computer technology and the future of education. In R. Atkinson and H. Wilson (Eds.) *Computer-assisted instruction: A book of readings* (pp. 41-48). New York, NY: Academic Press

Suppes. P. (1979). Current trends in computer-assisted instruction. In M. C. Yovits (Ed.), *Advances in computers, Vol. 18* (pp. 173-229). New York, NY: Academic Press.

Tennyson, R. D., Wooley, R. R., & Merrill, M. D. (1972). Exemplar and non exemplar variables which produce correct classification errors. *Journal of Educational Psychology, 63*, 144-152.

Tyler, R. W. (1949). *Basic principles of curriculum and instruction*. Chicago, IL: University of Chicago Press.

Vincenti, W. (1990). *What engineers know and how they know it: Analytical studies from aeronautical history*. Baltimore, MD: Johns Hopkins Press.

Vygotsky, L.S. (1978). *Mind in society: The development of higher psychological processes*. Cambridge, MA: Harvard University Press.

Wenger, E. (1987). *Artificial intelligence and tutoring systems: Computational and cognitive approaches to the communication of knowledge*. Los Altos, CA: Morgan Kaufmann Publishers.

Womack, J. & Jones, D. (2003). *Lean thinking: Banish waste and create wealth in your corporation*. New York, NY: Productivity Press.