DOCUMENT RESUME

ED 335 018 IR 015 185

AUTHOR Tennyson, Robert D.; Christensen, Dean L.
TITLE Automating Instructional Systems Development.
PUB DATE 91
NOTE 38p.; In: Proceedings of Selected Research Presentations at the Annual Convention of the Association for Educational Communications and Technology; see IR 015 132.
PUB TYPE Reports - Research/Technical (143) -- Speeches/Conference Papers (150)

EDRS PRICE MF01/PC02 Plus Postage.
DESCRIPTORS *Authoring Aids (Programing); Computer System Design; *Expert Systems; *Instructional Design; Instructional Development; *Instructional Systems; Learning Theories; Man Machine Systems; *Models; *Systems Development
IDENTIFIERS *Intelligent CAI Systems

ABSTRACT

This paper presents framework specifications for an instructional systems development (ISD) expert system. The goal of the proposed ISD expert system is to improve the means by which educators design, produce, and evaluate the instructional development process. In the past several decades, research and theory development in the fields of instructional technology and cognitive science has advanced the knowledge base for instructional design theory such that learning and thinking can be significantly improved by direct instructional intervention. Unfortunately, these advancements have increased the complexity of employing instructional design theory, making instructional development both costly and time consuming. It is proposed that through the application of expert system methods, it is now possible to develop an intelligent computer-based ISD expert system that will enable educators to employ instructional design theory for curricular and instructional development. Presented in this paper is a framework for the development of an ISD expert system that will assist both experienced and inexperienced instructional developers in applying advanced instructional design theory. Diagrams of three generations of the proposed model are appended. (38 references) (Author/BBM)

Title:

Automating Instructional Systems Development

Authors:

Robert D. Tennyson
Dean L. Christensen

Automating
Instructional Systems Development

Abstract

This paper presents framework specifications for an instructional systems development (ISD) expert system. The goal of the proposed ISD expert system is to improve the means by which educators design, produce, and evaluate the instructional development process. In the past several decades, research and theory development in the fields of instructional technology and cognitive science has advanced the knowledge base for instructional design theory such that learning and thinking can be significantly improved by direct instructional intervention. Unfortunately, these advancements have increased the complexity of employing instructional design theory, making instructional development both costly and time consuming. We are proposing that through the application of expert system methods, it is now possible to develop an intelligent computer-based ISD expert system that will enable educators to employ instructional design theory for curricular and instructional development. Presented in this paper is a framework for the development of an ISD expert system that will assist both experienced and inexperienced instructional developers in applying advanced instructional design theory.

Automating
Instructional Systems Development

Advancements in cognitive psychology and instructional technology in the past three decades have aided in the building of a literature of instructional design theory that can provide educators with sophisticated means to improve learning in all levels and conditions of education and training (Tennyson, 1990d). However, with this theoretical growth in instructional design has come the problem of instructional developers (i.e., any educator producing teacher-independent instruction) learning how to apply the new knowledge.

In response to this growth in the field of instructional design theory and practice, universities have developed graduate programs to produce instructional design (ID) experts. Even at the masters' level, these graduate programs require at least two years of full-time study. Therefore, if the educational community is to employ this body of knowledge to improve learning, it must either (a) develop inservice training programs to teach instructional design theory and practice or (b) develop a means by which educators can employ the knowledge without necessarily having to become ID experts.

This rapid growth in the instructional design field has also occurred in hundreds of other technical fields. To help maintain high levels of sophistication and to bring into application the most advanced knowledge from their respective fields, many of these other fields have employed expert system methods. An expert system is a computer-based representation of the domain-specific knowledge of an expert in a form that can be accessed by others for assistance in problem solving and decision making. An implication of this definition is that an inexperienced person can with the aid of an expert system perform tasks that would normally require the direct involvement of a domain-expert. Proposed in this paper are framework specifications for the development of an expert system to help instructional developers (i.e., authors) use the most advanced knowledge in the field of instructional design theory when designing and producing curriculum and instruction.

Instructional Systems Development

The process of designing, producing, and evaluating instruction is referred to in the literature as instructional systems development (ISD). The main components of ISD include (a) analysis of the instructional (and/or curricular) problem/need, (b) design of specifications to solve the problem, (c) production of the instruction, (d) implementation of the instruction, and (e) maintenance of the instruction. Embedded in each component of ISD are specific types of evaluation to insure quality control (Tennyson, 1978).

There are in the current literature several examples of computer-based tools intended to improve the productivity of the ISD process. Hermanns (1990) describes Computer-Aided Analysis (CAA), a computer program which aids in job task analysis. Based on a hierarchically-organized list of job tasks entered by the instructional designer, CAA produces as output a set of preliminary terminal learning objectives that can be further reviewed and edited by the developer. Ranker and Doucet (1990) describe SOCRATES, which allows the user to fill in information that is used by SOCRATES to create an instructor's lesson outline including objectives, events of instruction, samples of student behavior and test questions. Perez and Seidel (1990) present an overview of their specifications for an automated training development environment that will be based on the Army Systems Approach to Training (SAT) model of instructional design. The main features of the environment are a set of tools for developing the components of instruction and an expert design guide for assisting the designer in using the tools. Merrill and Li (1990) propose ID Expert, a prototype rule-based expert system for instructional development that makes recommendations about content structure, course organization, and instructional transactions (tutor/student interactions) based on information supplied by the designer.

The systems just referred to differ greatly in function and scope and in the degrees to which each makes use of expert system methods to reduce the level of knowledge required of instructional designers. This paper proposes framework specifications for an ISD expert system that would employ intelligent interface techniques to allow even the most inexperienced author to immediately begin to develop quality instruction. Labeled ISD Expert, the proposed system would make expert knowledge about the most sophisticated ISD methods readily available to potential authors, thus minimizing or eliminating the need for formal instructional systems development training.

The ISD model proposed for ISD Expert (see Figure 1) was developed to reflect an application model rather than a teaching model. That is, most ISD models are based on learning ISD, thus they resemble a linear process that attempts to include all possible variables and conditions of ISD. The result is that they do not take into account any other ISD situations other than complete start to finish instructional development. The assumption is that in all ISD situations, ISD starts at the analysis phase and proceeds step-by-step to the final completion of the implementation phase. The proposed ISD model, in contrast, views the author's situation as the beginning point of any possible ISD activities. For ISD Expert, the proposed ISD model is an associative network of variables and conditions, that can be addressed at any point in instructional development depending on the given situation.

This paper does not provide complete specifications for ISD Expert: instead, it provides a framework from which specifications can be designed and developed. The content of our ISD Expert includes both the philosophy of the proposed ISD Expert and the framework specifications. Given the complexity of ISD and the effort necessary to develop an expert system, hopefully, this chapter will also serve as a means for extending the dialogue on the concept of automated ISD systems and tools (e.g., see Merrill, 1990).

Philosophy of ISD Expert

Expert systems are designed for domain experts to aid them in dealing with complex processes that are either time consuming or which they do not have specific experience (e.g., in a sub-domain). In practice, expert systems have been successful when the content is narrowly focused and when the situations have clear rules for decision making (Smith, 1984). Because of the range of experience and training in ISD among instructional developers, I am proposing an expert system that will be designed for authors who are content domain experts but not necessarily ISD domain experts. This is not a contradiction of previous expert system efforts, but a reflection of the fact that the user of ISD Expert will not be an ISD expert initially; rather, the proposed ISD Expert would take into account a range of expertise and experience in instructional design theory and practice.

To accomplish this goal, we are further proposing an expert system that would employ intelligent human-computer interface techniques. The intelligent ISD Expert would operate at two basic levels: First, a coaching expert that would direct inexperienced authors through the acquisition of ISD skills while helping them deal with their specific situation; and, second, an advising expert that would assist experienced authors by making recommendations for their specific situation. For example, for an inexperienced author, the coaching function would deal with basic ISD skills and direct the development effort. In contrast, ISD Expert would function as an advisor to an experienced author, making recommendations while the author controlled the actual ISD decision making. In this environment, both inexperienced and experienced authors will be exposed to opportunities to increase their individual expertise through a process of learning ISD while using the system (Schiele & Green, 1990).

The importance of the distinction between the coaching and advisement functions is based on a review of research findings in expert systems. An example from this body of research is Clancey's (1979; 1983) work with MYCIN, a medical diagnosis consultant program, and GUIDON, a tutorial program designed to make use of MYCIN's rule base for teaching purposes. Clancey found that the rules encoded in MYCIN were inadequate for teaching because the knowledge required for justifying a rule and explaining an approach was lacking. He found it necessary to add additional components to GUIDON to help organize and explain the rules (Clancey, 1989). In a similar fashion, ISD Expert will have the ability to support and explain its recommendations and prescriptions in the language of ISD, not merely by enumerating the rules applied to make a recommendation. An example of one approach to providing this ability can be found in Swartout (1983). Swartout combined declarative and procedural knowledge, in the form of domain principles, to create the knowledge base for XPLAIN, a drug prescription consultant which provides detailed justification of its prescriptions.

Although ISD Expert can not be considered a means for teaching ISD, the very nature of the system's philosophy which assumes that authors will gain knowledge with experience, will result in continuing improvements in ISD applications. That is, as authors gain experience in ISD, the system would exhibit the characteristics of a conventional expert system. Therefore, it should increase the efficiency of instructional development and help in those areas where even experienced ISD authors initially lack specific expertise.

ISD Expert intelligent author-computer interface. ISD Expert, as proposed, would operate as an expert system employing intelligent author-computer interface (ACI) methods between the author and the system (Anderson, 1988). The ISD Expert intelligent ACI model (see Figure 3) would consist of four modules: the author's model of instruction, an ISD tutor (with both coach and advisor capabilities), an ISD knowledge base, and an instructional content knowledge base. Both knowledge bases would have knowledge acquisition capabilities. The ISD Expert tutor would be responsible for the interface between the individual authors and specific activities associated with developing their respective instructional needs.

ISD Expert system. In addition to the intelligent ACI component, the ISD Expert system would have three functions (see Figure 2). The first function would be to aid in the diagnosis of a given author's situation. This diagnostic function would evaluate the current situational condition(s) of the author (e.g., does the author want to prepare a computer-based graphic program for use in a lecture; does the author want to design a new course?). Following the situational evaluation, the second function would recommend prescription(s) along the lines associated with the level of author experience. That is, instead of trying to force all situations into a single solution, the prescription(s) would be individualized, based on situational differences and ISD experiences of the author. And with the third function, ISD Expert, through the system's tutor, would help the authors in accomplishing the prescriptions. As authors become increasingly more sophisticated in using ISD

Expert, they will be ready to accept increasingly more advanced variables and conditions of instructional design theory and practice.

ISD model. The proposed content for ISD Expert is the fourth generation ISD model (Tennyson, 1990). This ISD model is designed to adjust to future growth in instructional design theory and therefore does not become obsolete as new advancements are made, unlike earlier models. Figure 1 presents an illustration of the fourth generation ISD model. Briefly, the four generations of ISD models can be described as follows:

-First generation (ISD 1, Figure 1, Appendix A). The main focus of the first generation model was the implementation of the behavioral paradigm of learning (Glaser, 1966). The system consisted of four components: objectives, pretest, instruction, and posttest. The system was completed with an evaluation loop for purposes of revision.

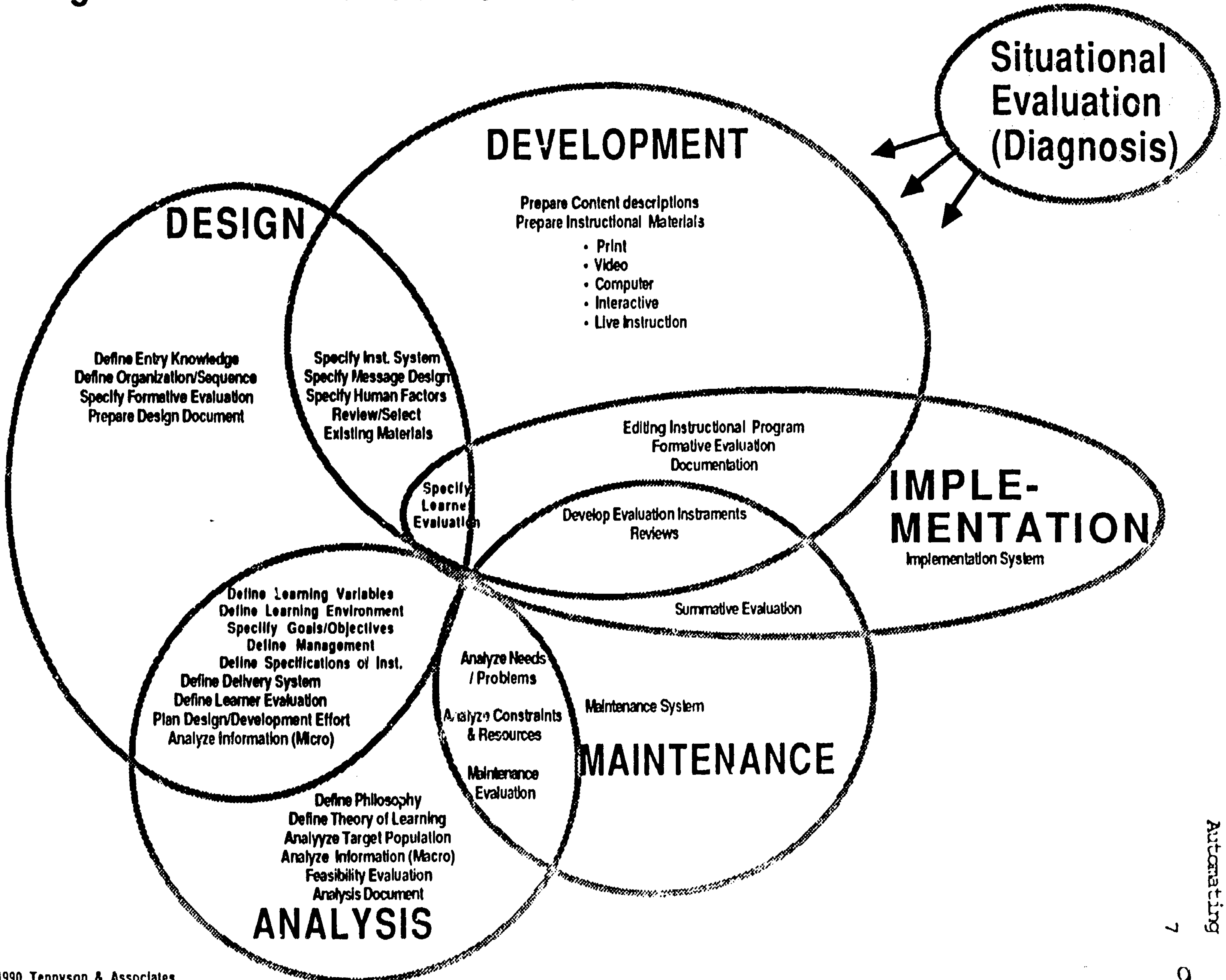
-Second generation (ISD 2, Figure 2, Appendix A). Advancements in instructional technology led to the need to increase the variables and conditions of the ISD model. The second generation adopted systems theory to control and manage the increasingly complex ISD process (Branson et al., 1976). The behavioral learning paradigm remained, but was of secondary importance to the focus of the system: developing instruction.

-Third generation (ISD 3, Figure 3, Appendix A). In practice, the ISD process was too linear and did not account for situational differences among applications (Tennyson, 1977). To account for situational differences, the external control of the system (i.e., the boxes and arrows) gave way to phases of ISD, that could be manipulated in any order by the instructional designer. This model assumed that ISD was an iterative process that could be entered at any point depending on the current state of the author's situation (Allen, 1986). Learning theory was still considered behavioral but cognitive theory was making some appearances (e.g., use of simulations for acquisition of cognitive skills in decision making).

-Fourth generation (ISD 4, see Figure 1). Advancements in cognitive psychology have made major changes in many of the ISD variables (e.g., content analysis, objectives, measurement, instructional strategies), making the ISD model yet more complex (see Tennyson, 1990a,b; Tennyson & Rasch, 1990). Employing technological developments from the field of artificial intelligence, the fourth generation model handles the complexity of ISD with a diagnostic/prescriptive system. Extending from the second and third generations, the ISD 4 model provides the knowledge base for the proposed ISD Expert system.

Cognitive theory. With growth of research and theory in cognitive psychology (Bonner, 1988), ISD Expert will exhibit a strong cognitive learning theory basis in both its ISD content and its approach to author-computer interaction. Early ISD models had a strong behavioral paradigm as their

Figure 1 4th Generation ISD Model



168

learning theory foundation (e.g., the first generation ISD model). The instructional strategies embedded in the first generation ISD models followed closely the behavioral paradigm of small incremental steps with emphasis on reinforcement of correct responses. For example, a task analysis in the ISD 1 paradigm favored a sequential approach that included student exposure to all possible attributes in a domain of information. For the most part, situational context and higher order cognitive knowledge and strategies were not considered because they did not fit the behavioral paradigm that dealt only with temporal content and observable performances (Brown, Collins, & Duguid, 1989).

In many other aspects of instructional development, the first generation ISD models also incorporated the behavioral paradigm, especially in the evaluation of learners. The focus of learner evaluation was on attainment of performance objectives that were isolated from meaningful applications or situations. By the 1970s, however, the ISD models exhibited more characteristics of systems models, the result being a separation of ISD procedures from a given learning theory paradigm. This growth in the "systems" of ISD was referred to above as the second generation ISD model.

Although the learning theory foundation of the ISD 2 models remained basically behavioral, the inflexibility of the flow-chart nature of the models limited their utility. In response to this inflexibility, the ISD 3 models actually proposed the elimination of the linearity of ISD by including phases of development that could be manipulated according to the unique conditions of given situations. The third generation ISD model identified phases of ISD that included direct links to specific forms of evaluation (see Figure 3, Appendix A), and therefore allowed user control of the procedures based on situational need. The third generation focused on an increased emphasis on evaluation without basic changes in the learning theory foundation, although the flexibility of the model made it possible to include the growing literature in cognitive psychology.

By the end of the 1980s, there had been sufficient empirical and theoretical work in cognitive psychology and instructional technology to once again see the possible effects of learning theory on ISD (Glaser & Bassok, 1990). The effects can be seen in such things as the importance of macro (i.e., curricular) level activities in ISD, contextual analysis of the information to be learned, evaluation of the learners, employment of interactive media, instructional strategies for higher order thinking, employment of structured and discovery instructional methods, effect of the affective domain on the cognitive, influences of group interactions on learning, and context and situational variables on knowledge acquisition (Tennyson, 1990b). The result has been the development of fourth generation ISD models that resemble a schematic structure (see Figure 1) and have a cognitive learning paradigm foundation for the various procedures of instructional development. As stated earlier, Tennyson and Christenson's (in press) fourth generation ISD model is proposed for the knowledge base for ISD Expert.

Along with a cognitive learning foundation for the ISD content, we are proposing that the human-computer interface of ISD Expert exhibit a cognitive approach as opposed to a behavioral one. The contrast between the two approaches is the assumption made in regard to the interaction between the author and ISD Expert. In a behavioral approach the interaction between the author and the ISD expert system would be made at a reductionist level, that is, small incremental steps in linear sequence of instructional development in which the author is simply, and constantly, filling in requests for information without understanding the individual ISD tasks in relationship to the given situation. This is a common approach employed in expert systems for novices where the task is relatively concrete and the user is simply filling in information. However, it must be assumed that the ISD task is complex and requires an author who can intelligently use the system more productively as he/she gains experience. Therefore, a cognitive approach assumes, even initially, that the author can connect the individual ISD tasks with his/her given situation.

To summarize, a cognitive psychology implication for ISD Expert is an expert system that assumes that the author can from the start function in the role as an instructional designer. This implies that even at an initial level of ISD, the author will have a real instructional problem/need and that he/she will be able to solve the situation with the prescription(s) offered by the ISD Expert system. And, as the author becomes more experienced with ISD Expert, he/she will be able to make increasingly sophisticated use of the system. ISD is a complex process, but the complexity is in part due to the given situation. Thus, for the initial, inexperienced author, the potential employment of ISD Expert will focus on noncomplex situations, but with the author feeling that he/she is participating in real ISD decision making.

This approach to the author should limit training for ISD Expert to a set of basic software functions and activities. Instead of viewing training on ISD Expert in the conventional linear fashion where the author works through a set of meaningless practice situations, the training will be embedded in the initial individualized ISD situation. For example, if the author wants to develop a test, his/her initial entry into ISD Expert will deal with test construction. In other words, training and gaining experience will be driven by the individual author's situation. Rather than a two year graduate program as prerequisite to being an instructional designer, the author will be an instructional designer with ISD Expert beginning with his/her first time situation. Because ISD is a complex environment and the needs of individual authors will vary at any given time, over an extended period of time, the individual authors will acquire more ISD knowledge as situational needs occur.

Computer technology. Because ISD Expert is intended to improve the performance of instructional designers, rather than to advance the state of the art in expert systems techniques and methods, it is most productive to make use of existing, standard computer hardware and software architecture whenever possible in the development of ISD Expert.

Certain restraints are imposed on the hardware and software choices by the requirements of the environment in which ISD Expert will most often be applied. These requirements are summarized as follows:

-Support for several simultaneous authors at both local and remote sites;

-Large data storage capacity for knowledge bases and programs;

-Sophisticated graphics capability

-Provision for incorporating special-purpose programs (for example, to support research projects) into ISD Expert on an ad hoc basis;

-Employment of interactive media.

Where hardware is concerned, a basic decision is whether to implement ISD Expert on a central mainframe or minicomputer, or on microcomputers. Simons (1985) and Harmon, Maus, and Morrissey (1988) address the expanding role of the microcomputer in AI development, citing growing hardware capacity, wider availability of sophisticated software tools and increasing user familiarity with microcomputers as the forces contributing to the growth in expert system development for microcomputers.

We are proposing that ISD Expert be implemented in a network of PC's connected to a central network and file server with one or more large-capacity (perhaps 300 megabytes) hard disk drives for program and knowledge base storage. While there are a number of physical network topologies that could be used to implement ISD Expert, Figure 4 represents the general concept. There are some tradeoffs involved in using this configuration as contrasted with a network of "dumb" terminals connected to a single, central mainframe and data storage. For example, transmitting large quantities of data to/from the central file server to the PC's does require system overhead. However, the advantages outweigh the drawbacks. Given the local processing power of PC's, the intelligence of the system will be distributed throughout the system, minimizing the demands on the central unit. There is a large and growing quantity of AI software available for microcomputers at relatively low prices in contrast to mainframes. PC graphics are superior to all but the most sophisticated and expensive mainframe graphics systems.

The software used to create ISD Expert must provide an open architecture. That is, it must be practical to write local programs for special purpose functions (e.g., as research projects) and link them into the standard software with a minimum of effort. Also, the knowledge bases must be accessible to local programs as well as to the standard software. Expert system development is done either by using expert systems shells, which are commercially-available skeleton systems that can be instantiated with the specific domain knowledge required for an application, or by writing the expert system from scratch in a general or special purpose programming language. Harmon et. al. (1988) report that of 115 expert systems surveyed by them in actual use in the United States in 1986, 92 were produced using shells while 23 were written using programming languages (chiefly LISP).

Proposed is that ISD Expert be implemented using commercially-available expert system shells. However, in view of the fact that ISD Expert must also support customization, the shells that are chosen must support what are termed "own-code exits" to facilitate the linking in of custom programs. These custom programs must be written in a high-level computer language, preferably one with extensive AI features (e.g., LISP; PROLOG).

Summary

To establish a framework for ISD Expert, it is important to clearly specify the philosophy of the system (Morgan, 1989). A well specified philosophy will help keep the system under control during development and later when doing revisions. Proposed in this section is that ISD Expert have a foundation in cognitive psychology (Newell & Card, 1985). And, that this foundation specifies for the system both the content and the author-computer interface (Norman, 1986). Specific areas of the proposed philosophy are as follows:

-An expert system that has both diagnostic and prescriptive functions

-An expert system that will serve experienced and inexperienced authors

-An intelligent ACI system with both advising and coaching capabilities

-Knowledge base content will employ the fourth generation ISD model

-Employment of interactive media

-Cognitive learning theory as the foundation of the ISD procedures

-Cognitive paradigm approach to ACI

-Entry to system based on individual author situation

-Training as a concurrent activity with ISD activities

-A computer-based network system with remote capabilities

-Software tools that provide an open architecture

-Employment of a high-level language (e.g., an AI language)

-Commercial shells that include access to own-code programs

-Data dictionaries for knowledge acquisition components

The above discussion on a proposed philosophy for ISD Expert provides the foundation for the following section on framework specifications. The next section presents a basic framework for ISD Expert.

ISD Expert: System Framework

The purpose of this section is to propose framework specifications employing the above described philosophy of an expert system for instructional systems development. Because of the range in authors knowledge of ISD, we are proposing that ISD Expert be designed according to the methodology of intelligent human-computer interface systems. That is, rather than either attempting to teach ISD to the author or to develop a system around one linear approach that restricts and narrows the richness of ISD, our proposal is the design of a system that begins with the individual author's given situation. In this proposal, the intelligent ACI method will be concerned with improving both the authors application of ISD and their own models of instruction. As such, it will employ coaching and advising methods of human-computer interaction.

Furthermore, the proposed system will encourage the growth of the authors knowledge of ISD, but with the complexity of ISD being transparent. The purpose of ISD Expert will be to diagnosis the given situation of the author and then prescribe recommendations for dealing with his/her individual situation. It is assumed that each author will present a different situation and, therefore, will require a unique prescription. To accomplish this goal, the employment of heuristics is proposed for programming ISD Expert (see Bonnet, Haton, & Truong-Ngoc, 1988; Waterman, 1986). Two important features of the heuristic method, as contrasted, for example, with production rules, are (a) the flexibility needed to implement prescriptions in conditions of uncertainty or novelty (i.e., prescriptions are established in real time by integrating best available information from the system's knowledge base) and (b) the elimination of the need for an exhaustive reduction of ISD content knowledge to production rules.

One of the serious problems in expert systems design for nonstatistical areas has been the attempt to reduce complex and abstract concepts to production rules (e.g., Merrill's ID Expert, 1986). Even though we are proposing the use of a network and file server (with large capacity disk storage) system for the operation of ISD Expert, it is the programming time involved in trying to apply the reductionist approach to an environment as complex as ISD that rules out the exclusive use of the production rules programming methodology. The software architecture of ISD Expert must be open to allow for future extensions. The production rule method is not suitable for this type of complex situation (Clancey, 1983). So much of the ISD process is context bound; therefore, the system must be adaptable, allowing for prescriptions to be finalized by the author.

Proposed for ISD Expert is an expert system with four main components: an intelligent author-computer interface component, a diagnosis function component, a prescriptive function component, and an instructional production guide component (Figure 2). The intelligent ACI component will be the means by which authors will interact with ISD Expert. Rather than use a menu driven system, I am proposing a tutorial interaction between the author and ISD Expert. The diagnostic component will function as the evaluator of each author's situation and provide an evaluation report (Guba & Lincoln, 1986). This report will serve as the guidelines in preparing the prescription. Additionally, the prescription will be based on the author's ISD model as well as the diagnostic report. The fourth component will provide the author with assistance in the production of materials from the prescription(s). The level of assistance will again be influenced by the author's ISD model.

Intelligent Author-Computer Interface Component

The intelligent ACI component for the ISD Expert is illustrated in Figure 3. The main modules are as follows: (a) an author's model of ISD; (b) the ISD tutor; (c) the ISD knowledge base model; and (d) the content knowledge base. I will now discuss the role of each component of the ISD Expert tutor.

Author's model of ISD. The purpose of this module is twofold: (a) to establish the level of ISD expertise of the author and (b) to help the author improve his/her own model of instruction. This is necessary because no formal attempt is to be made to directly train the authors in ISD. The individual author's model will be updated with each use of the system. This profile of the author will help the system in its prescriptive recommendations. For example, experienced authors will have a narrow and limited knowledge of ISD and, also, of the ways in which their instruction could be improved; thus, prescriptions would be at their level of understanding. On the other hand, more experienced authors would be able to use more advanced prescriptions. It is important to keep the ISD prescriptions at the level of the author's experience and also to provide an opportunity for creativity and the possible use of different ideas generating from the author (Russell, Moran, & Jordan, 1988). A key feature of the proposed ISD Expert is the power of the author to disagree with a given prescription and still to be able to continue with the ISD process.

ISD tutor. Intelligent HCI systems work on the premise that a meaningful dialogue must be established between the user and the system. An important feature of the dialogue is the mixed initiative, where the user has an opportunity to query the system as well as being controlled by the direction of the system. The ISD Expert tutor will approach the diagnostic function from the context (situation) of the author. Personalizing the diagnostic activity will provide the opportunity for the tutor to search the content knowledge base to include specific references in the prescription to available existing materials and resources.

Because of the range of knowledge and experience in ISD of potential authors, two basic modes of interface are proposed. At one extreme will be authors who are completely inexperienced in ISD. For these individuals, a coaching mode is proposed. The coaching mode is a well established method of instruction used in intelligent computer-assisted instruction (ICAI). This

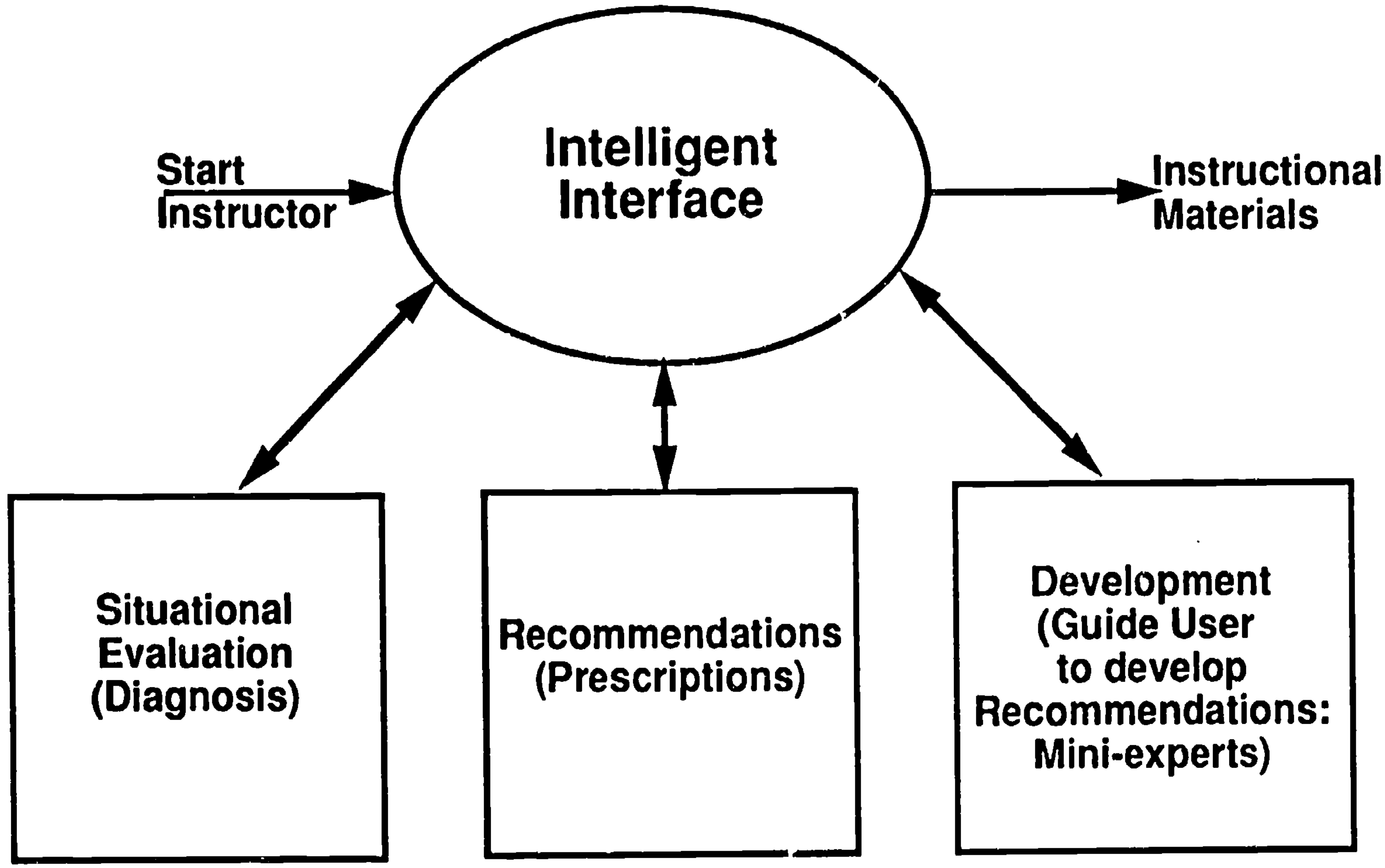


Figure 2. ISD Expert Components.

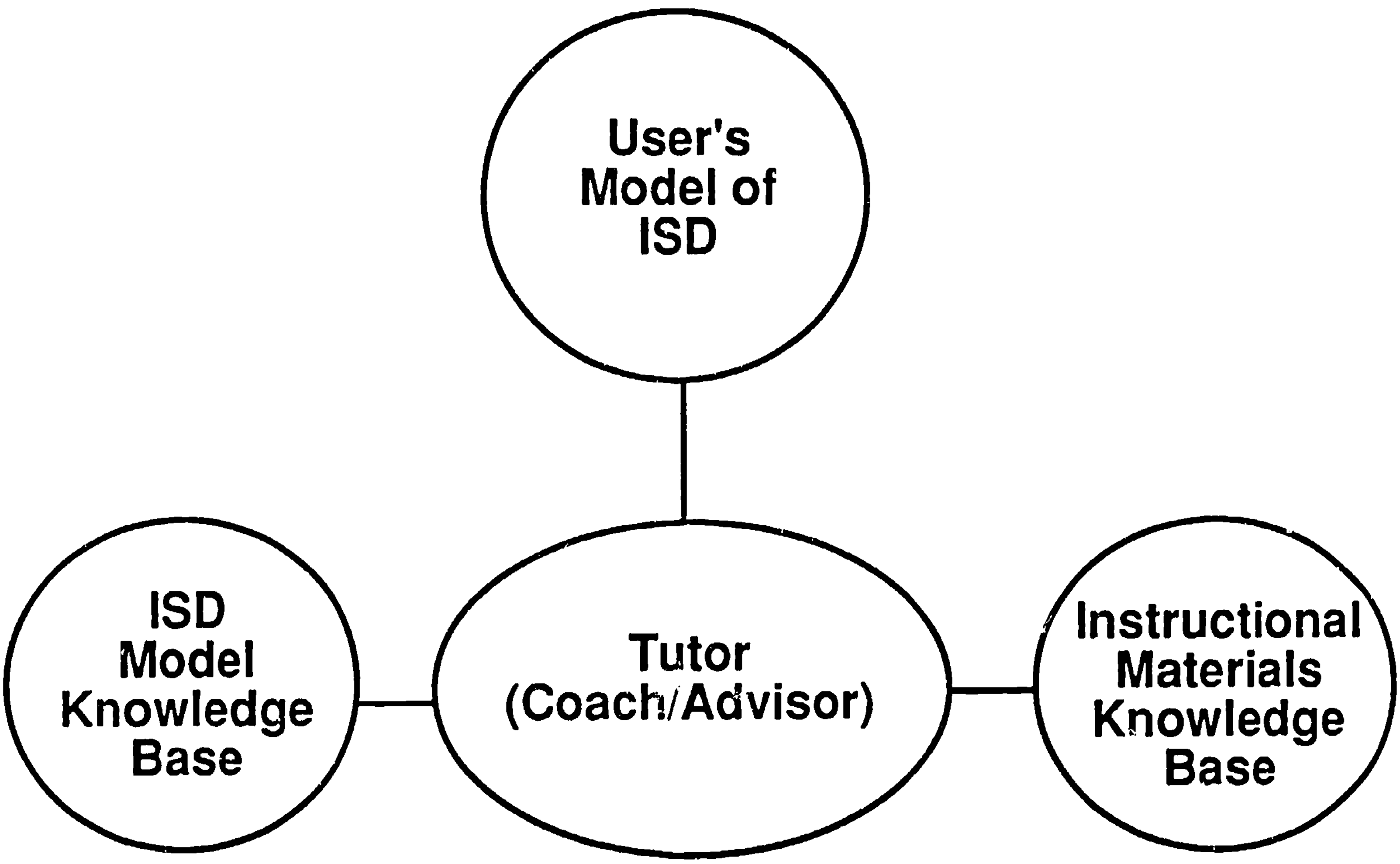


Figure 3. AIDA Intelligient Interface Component.

mode assumes that the author will need direct and controlled assistance in dealing with his/her given situation. The function of the tutor as coach is to approach the ISD activity in a disciplined way while helping the author develop ISD skills. Prescriptions for the situation are specific and the coach is responsible for the decision making. In contrast is the advising interface, For the experienced author, the tutor as advisor would offer alternative prescriptions, with the final decision(s) in the hands of the author.

The tutor, as part of the intelligent ACI component, is the point of contact between the author and the other ISD Expert components (see Figure 2). In the proposed design, the tutor gathers information about the author's specific situation and, by interaction with the Situational Evaluation component, prepares a report of the given problem/need. This evaluation report is sent to the Recommendations component where a prescription(s) is prepared. When the prescription(s) is prepared, the tutor presents it to the author; at that point, depending on the mode of the tutorial interaction (i.e., coaching or advising), there may occur a dialogue between the author and the tutor to finalize the prescription. Once a final prescription is prepared, the tutor interacts with the ISD model knowledge base to set up the authoring activities. The tutor also assists the author in certain aspects of materials production through the fourth component of the ISD Expert system. Updating of the author's model will be the continuing role of the tutor in ISD Expert.

ISD model knowledge base. The content knowledge of ISD Expert will reside in the ISD model knowledge base (KB) (see Figure 1). Once the prescription(s) is decided upon, the necessary authoring activities are compiled by the tutor from the ISD model knowledge base and presented to the author. (Authoring activities of the knowledge base are presented in Appendix B.) Information within this KB will be stored as structured data files, organized as an associative network. The purpose here is to efficiently locate information without the restrictions of rigid production rules. That is, the ISD model knowledge base should exhibit the heuristic search characteristics of an information retrieval system.

Content knowlege base. The fourth module of the proposed intelligent ACI system for ISD Expert, the content knowledge base, is a source from which curricular and instructional materials resources may be obtained. These materials may be included in the implementation of prescriptions developed by ISD Expert or they may stand alone. For example, if an author wants a simulation for a given lecture, he/she could query the content knowledge base to see what might be available. In another situation, ISD Expert may develop a prescription and obtain the necessary materials from the content knowledge base without the author explicitly requesting the action. Access to the content knowledge base may be either by direct author query via the tutor or indirectly as a result of the implementation of prescriptions.

The content knowledge base will help eliminate duplication of effort in instructional development by providing a catalog of available materials.

Information in the content knowledge base would come from two sources. Material that is developed on ISD Expert as a result of instructional development can be added to the content knowledge base. Material may also be input from sources external to ISD Expert. For example, many materials and resources that are developed in R & D efforts independently of ISD would be useful in course applications if authors had access to them. General information manuals and other media-based resources (e.g., video disk materials) are another example of materials from external.

Situational Evaluation Component

The first activity in the proposed ISD Expert system is the evaluation of the given author's situation. The assumption is that each author will have a different need or problem, depending on his/her given situation. As the ISD Expert tutor establishes the author's model of instruction (see Figure 3), the Situational Evaluation Component will diagnosis the situation employing AI techniques. Again, it is assumed that the tutor will determine the experience level of the author and in turn adjust the report of the evaluation. For example, if the tutor determines that the author is experienced in ISD, and the situation is to develop a lesson on trouble shooting, the report would indicate those two conditions, which would influence the type of prescription(s) recommended. By focusing on the given situation, ISD Expert can employ the complexity and richness of the fourth generation ISD model without directly training the author about the entire model.

Recommendations Component

The purpose of ISD Expert is to help authors improve their instructional product development by applying the most advanced variables and conditions of instructional design theory. This is made possible by the recommendations component, which interacts with the ISD knowledge base to interpret the situational evaluation diagnosis and recommend a prescription to deal with the given instructional situation. Also, the prescription is adjusted to the author's level of experience. This is an important feature of the proposed ISD Expert because it prescribes an effort of development that can be efficiently accomplished by the author. For example, if an inexperienced author is presented with a prescription that would fit an experienced author's profile, the novice author would not be able to adequately follow the production activities (Component 4). The result would be that the prescription is implemented inefficiently or not at all. Presentation of the prescription will likewise be based on the experience of the author. The experience level of the author will determine the program control (i.e., coaching or advisement) employed in the production component.

Production Component

The term production is used here to reflect a variety of different types of instructional situations that might occur. ISD includes, in addition to instructional development, test development, computer-based management development, print materials development, instructional aids, visual aids, etc. The function of this component is to guide the author in the production process. As such, this part of the expert system directly interacts with the tutor. Because of the range of ISD activities, this component would be composed of mini-experts, each reflecting a different authoring activity (see Appendix B). That is, the mini-experts would be the various activities within the ISD model. For example, if the situation is to develop a test for trouble shooting, the author's model indicates an experienced author, and the prescription recommends a simulation, a mini-expert on design of simulations within component 4 would guide the author in the production of an appropriate simulation. An important feature of ISD Expert will be to facilitate the employment of avdvanced interative technology for instructional delivery. For example, for computer-based instruction, this component would directly produce the courseware (Tennyson, 1990c).

Once the production effort is completed, component 4 would send a report back to the tutor to update the author's model and to reference the effort in the content knowledge base. To further improve the efficiency of ISD Expert, instructional strategy (IS) shells will be accessible by the mini-experts to do the actual product development: IS shells would only require that the author enter into the system content information and the system would develop the product.

The above four components of the proposed ISD Expert system would be designed and programmed as independent expert systems so as to allow for future additions and elaborations. This is necessary because of the continuing growth in the instructional design theory field. That is, most expert systems are designed for specific, contemporary applications; when changes occur, a new expert system is designed and implemented.

Central Network System

In this section, we propose for the computer-based environment a configuration for a centrally-based network and file server for both local and remote PC workstations (Figure 4). Because of the proposal for a content knowledge base (see Figure 3) with acquisition capabilities and an author's instructional model within ISD Expert, a large capacity disk storage should be an integral part of the system. Also, given the computing power of PCs, much of the intelligent interfacing would take place at the workstation.

Although there are a large number of commercially-available shells for program development (e.g., HYPERcard), most do not allow for "own-code" exits. With such software, the development effort becomes constrained by the closed architecture of the given shell; the shell becomes a methodology in itself rather than a tool to be used in implementing multiple methodologies. I am proposing that ISD Expert be programmed in a high-level language with artificial intelligence features, but that commercially-available shells be used when feasible to augment the system features.

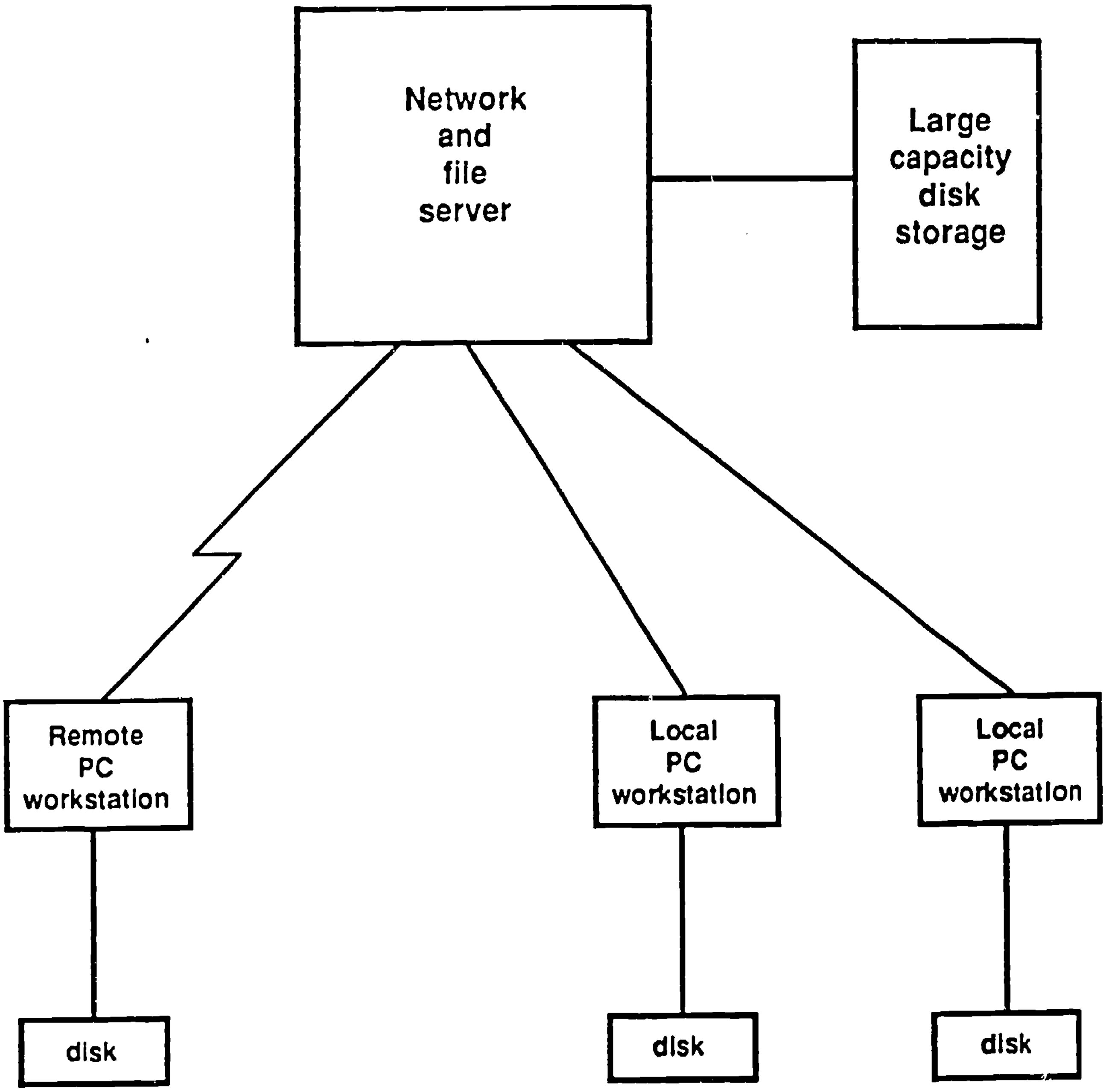


Figure 4. Distributed processing with PC workstations in local and remote locations connected to central file and network server.

Development Plan

The framework specifications presented in this paper offer a complete expert system for automating instructional development. To produce such a system, there are two possible approaches. The first would be to develop the ISD Expert system as presented. The second would be to follow an incremental approach in which an initial prototype is developed that only has a minimal set of features and is aimed at an experienced author. That is, an ISD Expert that would only have an advisor level tutor and the situational evaluation and recommendations components. The content knowledge base and acquisition features of the intelligent interface tutor and the production component would be added in subsequent elaborations.

Although the first approach seems possible, there are a number of problems that need to be considered that might favor the second approach. An initial problem is the cost factor. As stated earlier the majority of expert systems are developed using commercial shells. Cost in terms of software is the time required to produce a product that will be timely and profitable. That is, the proposed ISD Expert would most likely be a software product that would need to generate income within a reasonable timeframe. Rapid prototyping is a procedure to develop software employing shells that are linked by some general language (Hewett, 1989). Thus, instead of five years to produce a complete version of ISD Expert, an initial prototype could be developed in much less time.

A second major problem in producing a complete ISD Expert in the first approach is the necessary research needed for the new system. There has been minimal empirical research to date on instructional variables and conditions associated with the extension of cognitive learning theory to instructional design theory. Even though it is possible to develop an initial prototype, research in cognitive instructional design theory needs to be done as well as the interaction of media within this theoretical framework. A third problem area relates to the specification of the human-computer interaction variables and conditions necessary to run and manage the complex environment of ISD Expert.

Within the constraints defined above for approach one, we recommend the following incremental approach to ISD Expert development.

1. Framework specifications. This step conceptualizes the idea or vision of the expert system. This chapter serves as an example of the first activity in producing an automated instructional development system.

2. Functional specifications. From the initial outline of the basic system, the specific functions provided by the system need to be defined. From this step a rapid prototype can be developed as follows:

-Write functional specifications;

-Summarize what is known/not known about the functions;

-Estimate the complexity of the functions;

-Based on the summary and estimates, group the functions into ISD Expert 1 (i.e., a prototype), and then prioritize the functions for successive implementations for versions ISD Expert 2, ISD Expert 3, etc. Each version would add layers of functions and increased use of a high-level computer language.

3. Logical design. Starting with the prototype, define the logical components that provide the specified functions.

4. Physical design. Define the software modules which implement the logical design of the system.

5. Programming. With the prototype, rapid software development is recommended while with the successive versions, the software procedures defined in this paper would be followed.

6. Testing. Once the prototype is developed, it should be tested following standard computer software benchmark criteria.

7. Implementation. Complete the remaining tasks to implement ISD Expert1 while simultaneously accumulating the experience and research findings needed to produce ISD Expert 2.

8. Incremental development. Basically starting with number 3 above, iteratively build ISD Expert towards a system that includes all of the functions defined in numbers one and two.

Cycling through an incremental approach to development of ISD Expert would produce an initial product for employment and research within the constraints of costs and system knowledge. The initial prototype (or ISD Expert1) would exhibit many standard characteristics of the third generation ISD model (see Figure 3 Appendix A) with successive versions taking on more of the ideas associated with the fourth generation ISD model (see Figure 1).

Conclusion

The purpose of ISD Expert is to improve learning by aiding authors in the employment of contemporary instructional design theory. This paper presents framework specifications for an expert system to implement the concept of ISD Expert. Because of the experience range of authors for ISD Expert in terms of instructional design theory and practice, we are proposing an expert system that employs an intelligent human-computer interface system. That is, ISD Expert will interact with authors on an individual basis according to their respective experience with principles and variables of instructional design theory. ISD Expert will dialogue with authors along a continum of decision control ranging from system control (coaching function) to complete author control (advising function). Inexperienced authors will be coached to develop basic ISD skills while the more experienced authors will be advised on the employment of advanced instructional design variables and conditions.

An author's instructional design model is a necessary module for an intelligent human-computer interface component because it replaces the need for a separate training program for authors. The sophistication of ISD Expert's prescriptions will be directly influenced by the author's instructional design model. Therefore, both ISD novices and experts will be able immediately to use ISD Expert. That is, the proposed system would take into account experience in instructional design theory and practice.

Proposed for the ISD knowledge base is the fourth generation ISD model (see Figure 1). The initial set of authoring activities for the ISD knowledge base is presented in Appendix B. The content knowledge base is proposed as a data base for instructional materials within subject matter areas. Both knowledge bases will have acquisition capabilities.

The basic proposed ISD Expert system will have four interactive components: (a) an intelligent author-computer interface component; (b) a situational evaluation component (diagnosis), (c) a recommendations component (prescriptions), and (d) a production component. Proposed is that ISD Expert be designed for a computer-based network system using a high level AI type language and expert system shells. The system will also employ a large capacity disk storage for the two knowledge bases (i.e., ISD model and instructional content and materials); both knowledge bases will employ acquisition capabilities.

To implement the concept of the fourth generation ISD model, the situational evaluation component of ISD Expert will diagnose each author's given problem and/or need. This will make the system application orientated rather than the conventional lock-step system that is most suited for the teaching of ISD. From the diagnosis, ISD Expert will generate a prescription. For those authors who seek assistance in implementing the prescription, especially those requiring the development of instructional materials, the fourth component will guide the production effort. This production component will be composed of mini-expert systems that have specific functions (e.g., instructional strategy shells).

Because of both development costs and gaps in instructional design theory, we recommend an incremental approach to the development of ISD Expert. Initially, the project should begin with rapid prototyping techniques to produce a version one of ISD Expert. Subsequent versions would be elaborated according to the functional specifications as outlined in this chapter and from on-going research findings.

In conclusion, we are proposing an expert system that will bring the power of instructional design theory and practice to educators who would not normally have the opportunity to employ such knowledge in their instructional efforts. The proposed ISD Expert will improve learning by making instructional development both effective (i.e., by employing the most advanced principles and variables of learning and instructional theories) and efficient (i.e., by reducing the time and cost of conventional methods of instructional development).

References

Allen, M. W. (1986). Authoring aids: The means to improved authoring procedures. Proceedings of the SALT Conference. Orlando, FL: Society for Applied Learning Technology.

Anderson, J. R. (1988). The expert module. In M. C. Polson & J. J. Richardson (Eds.), Intelligent tutoring systems (pp. 21-53). Hillsdale, NJ: Erlbaum.

Bonner, J. (1988). Implications of cognitive theory for instructional design: Revisited. Educational Communication and Technology Journal, 36, 3-14.

Bonnet, A., Haton, J-P., and Truong-Ngoc, J-M. (1988). Expert systems: Principles and practice. Englewood Cliffs, NJ: Prentice Hall.

Branson, R., Rayner, G. T., Cox, J. L., Furman, J. P., King, FJ, & Hannum, W. H. (1976). Interservice procedures for instructional systems development (5 Vols.) (TRADOC Pam 350-30 and NAVEDTRA 106A). Fort Monroe, VA: U. S. Army Traning and Doctrine Command. (NTIS No. ADA-019 486 ADA-019 490)

Brown, J. S., Collins, A., & Duguid, P. (1989). Situated cognition and the culture of learning. Educational Researcher, 18(10, 32-42.

Clancey, W. J. (1979). Tutoring rules for guiding a case method dialogue. International Journal of Man-Machine Studies, 11, 25-49.

Clancey, W. J. (1983). The epistemology of a rule-based expert system--a framework for explanation. Artificial Intelligence, 20, 215-251.

Clancey, W. J. (1989). The knowledge level reinterpreted: Modeling how systems interact. Machine Learning, 4, 287-293.

Glaser, R. (1966). The design of instruction. In J. I. Goodlad (Ed.), The changing American school: NSSE 65th Yearbook (pp. 215-242).

Glaser, R., & Bassok, M. (1989). Learning theory and study of instruction. Annual Review of Psychology, 40, 631-666.

Guba, E. G., & Lincoln, Y. S. (1986). The countenances of fourth-generation evaluation: Description, judgment, and negotiation. Evaluation studies review annual, 11, 70-88.

Harmon, P., Maus, R., & Morrissey, W. (1988). Expert system tools and applications. New York: Wiley.

Hewett, T. T. (1989). Towards a rapid prototyping environment for interface design: Desirable features suggested by the electronic spreadsheet. In

A. Sutcliffe & L. Macaulay (Eds.), People and computers V (pp. 305-314). Cambridge, UK: Cambridge University Press.

Hermans, J. (1990). Computer-aided instructional system development. Educational Technology, 30(3), 42-45.

Merrill, M. D. (1986). Prescriptions for an authoring system. Journal of Computer-based Instruction, 12, 90-96.

Merrill, M. D. (1990). Introduction to special issue: Computer-based tools for instruction design. Educational Technology, 30(3), 5-7.

Merrill, M. D., & Li, Z. (1990). An instructional design expert system. In S. Dijkstra, B. van Hout Wolters, & P. C. van der Sijde (Eds.), Research on instruction: Design and effects (pp. 21-44). Englewood Cliffs, NJ: Educational Technology.

Morgan, C.C. (1989). Programming from specifications. New York: Prentice-Hall.

Newell, A., & Card, S. (1985). Prospects for psychological science in human computer interaction. Human Computer Interaction, 1, 209-242.

Norman, D.A. (1986). Cognitive engineering. In User-centered system design (pp.31-62). Hillsdale, NJ: Erlbaum.

Perez, R. S., & Seidel, R. J. (1990). Using artificial intelligence in education: Computer-based tools for instructional development. Educational Technology, 30(3), 51-58.

Ranker, R. A., & Doucet, R. M. (1990). SOCRATES: A computer-based lesson development advisor. Computers in Human Behavior, 6, 162-171.

Russell, D. Moran, T., & Jordan, D. (1988). The instructional design environment. In J. Psotka, L. Massey, & S. Mutter (Eds.), Intelligent tutoring systems: Lessons Learned (pp. 96-137). Hillsdale, NJ: Erlbaum.

Schiele, F., & Green, T. (1990). HCI formalisms and cognitive psychology: the case of Task-Action Grammar. In M. Harrison & H. Thimbleby (Eds.), Formal methods in human-computer interaction (pp. 9-62). Cambridge, UK: Cambridge University Press.

Simons, G. L. (1985). Expert systems and micros. Manchester, UK: NCC Publications.

Smith, S. F. (1984). Adaptive learning systems. In R. Forsyth (Ed.). Expert systems: Principles and case studies. New York: Chapman and Hall.

Swartout, W. R. (1983). XPLAIN: A system for creating and explaining expert consulting programs. Artifical Intelligence, 21, 285-325.

Tennyson, R. D. (1977). Instructional systems: Development, evaluation, and management of learning environments. In D. J. Treffinger, J. K. Davis, & R. E. Ripple (Eds.), Handbook on teaching educational psychology (pp. 215-242). New York: Academic Press.

Tennyson, R. D. (1978). Evaluation technology instructional development. Journal of Instructional Development, 2(1), 19-26.

Tennyson, R. D. (1989). Cognitive science and instructional technology updates of ISD. Technical Report. U.S. Air Force Human Resources Laboratory/IDC. Brooks Air Force Base, San Antonio, TX.

Tennyson, R. D. (1990a). A proposed cognitive paradigm of learning for educational technology. Educational Technology, 26(6), 12-17.

Tennyson, R. D. (1990b). Cognitive learning theory linked to instructional theory. Journal of Structural Learning, 13, 362- 371.

Tennyson, R. D. (1990c). Computer-based enhancements for the improvement of learning. In S. Dijkstra, B.H.A.M. van Hout Wolters, & P. C. van der Sijde (Eds.), Research on instruction: Design and effects. Englewood Cliffs, NJ: Educational Technology.

Tennyson, R. D. (1990d). Integrated instructional design theory: Advancements from cognitive science and instructional technology. Educational Technology, 26(6), 18-28.

Tennyson, R. D., & Rasch, M. (1988). Linking cognitive learning theory to instructional prescriptions. Instructional Science, 17, 369-385.

Tennyson, R. D., & Rasch, M. (1990). Instructional design for the improvement of learning and cognition. In H. Feger (Ed.), Wissenschaft und Verantwortung (pp. 71-90). Goettingen, Fed. Rep. of Germany: Hogrefe-Verlag.

Waterman, D. A. (1986). A guide to expert systems. Reading, MA: Addison-Wesley.

Appendix A

Figure 1. 1st Generation ISD Model (1960s).

Figure 2. 2nd Generation ISD Model (1970s).

Figure 3. 3rd Generation ISD Model (1980s).

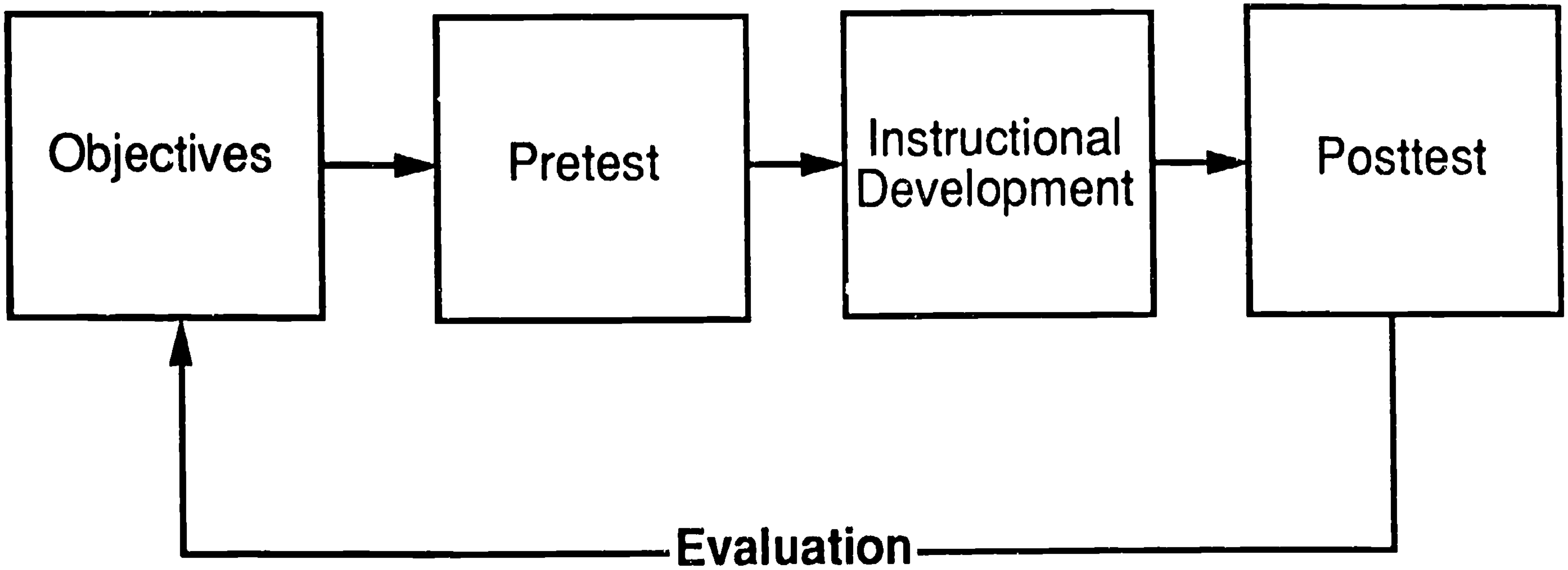


Figure 1. 1st Generation ISD Model (1960s).

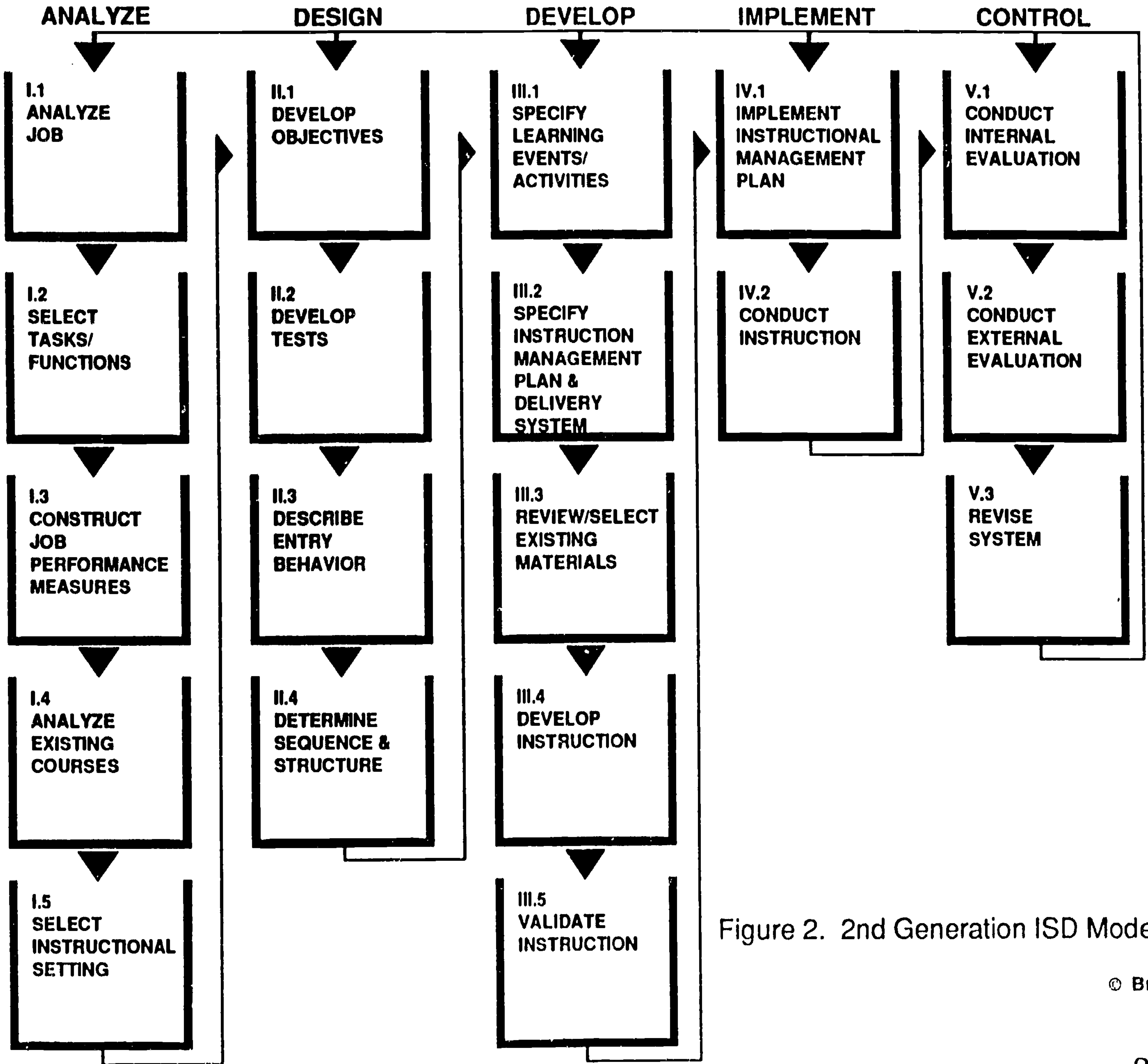


Figure 2. 2nd Generation ISD Model (1970s).

© Branson (1976)

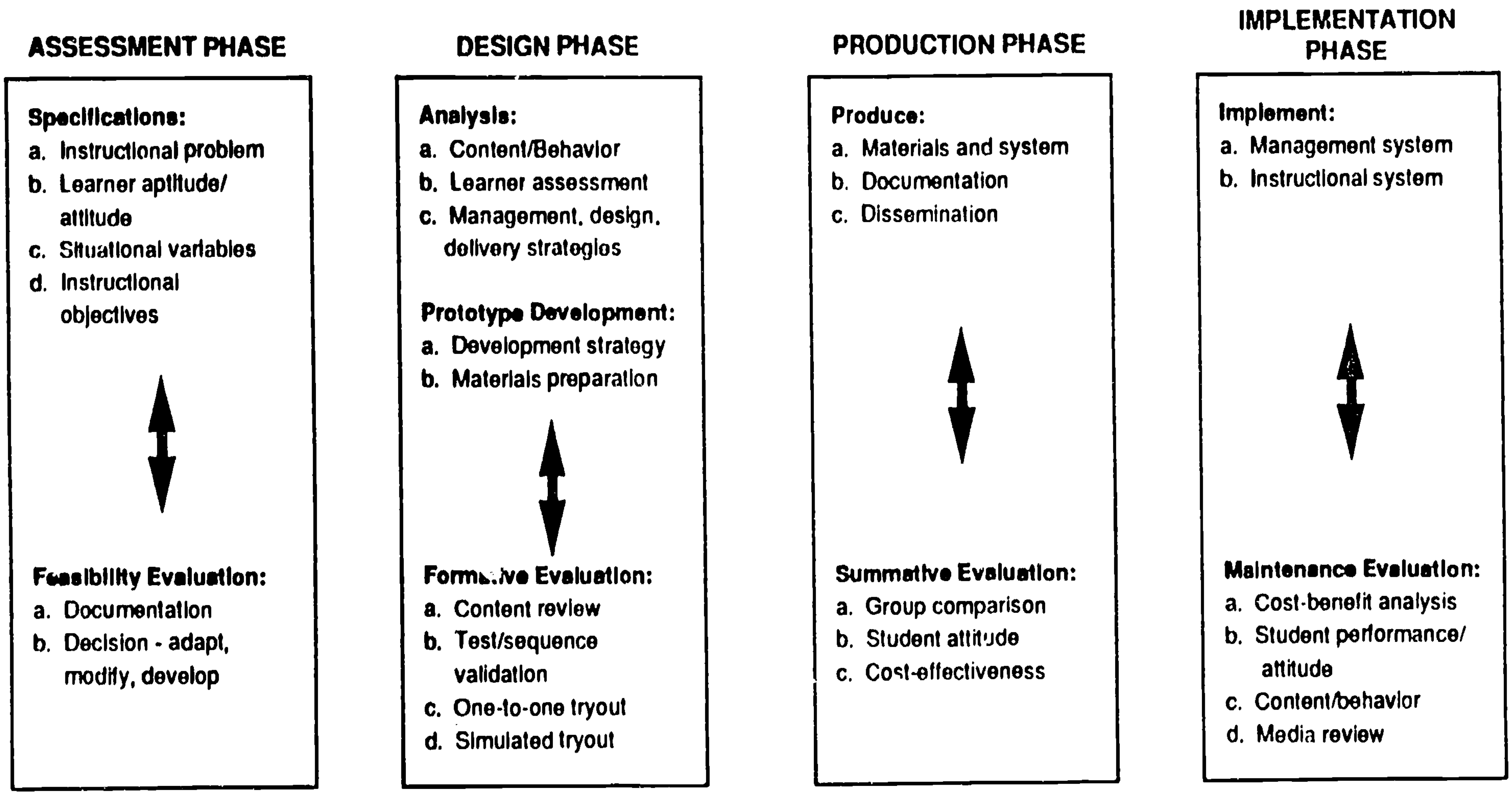


Figure 3. 3rd Generation ISD Model (1980s).

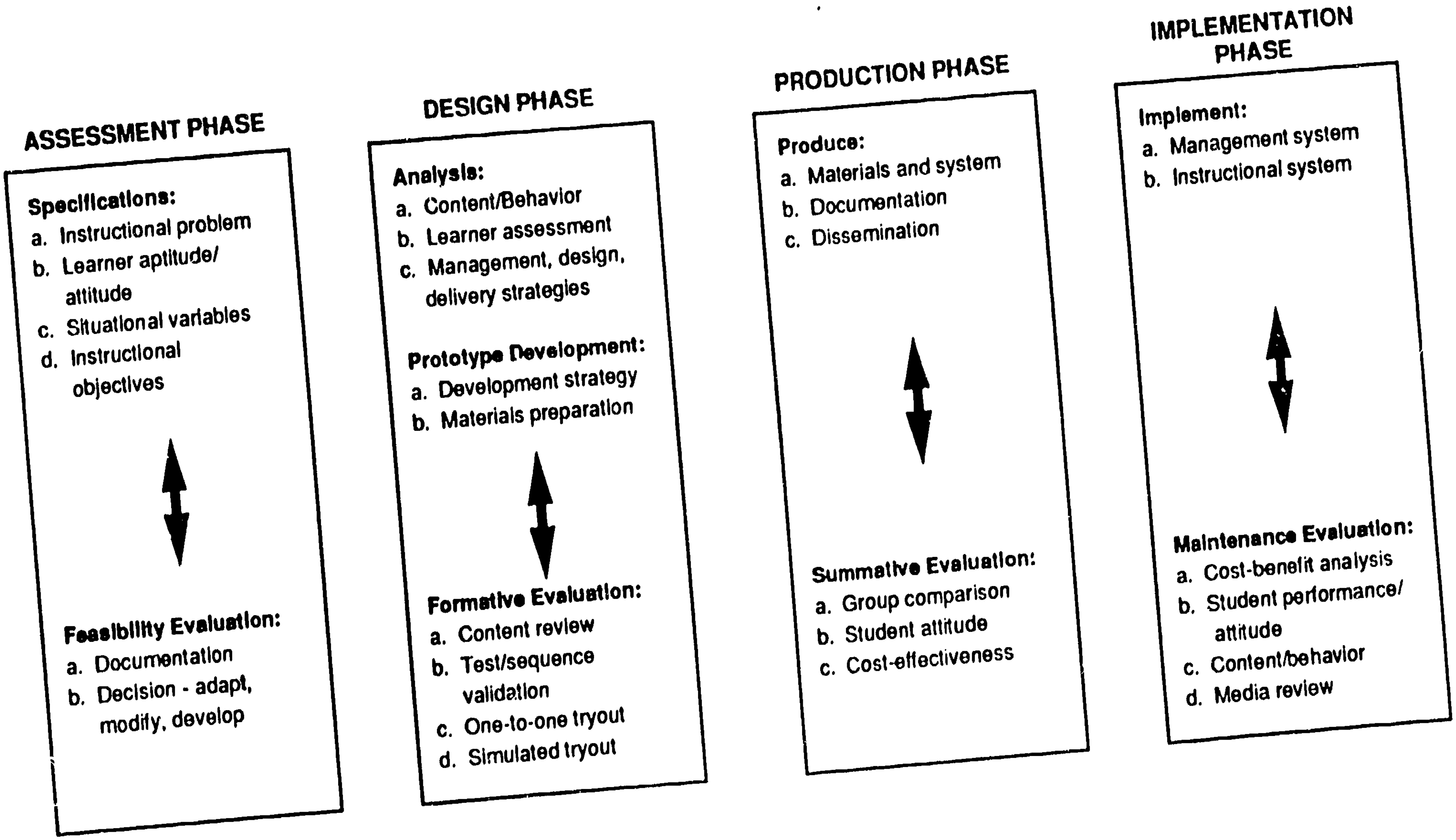


Figure 3. 3rd Generation ISD Model (1980s).