CHAPTER VI
TOWARDS AN OBJECT-ORIENTED INSTRUCTIONAL DESIGN MODEL

1. Defining instructional design

In general, instructional design describes a set of procedures that guides authors in the systematic process of designing instructional interventions based on principles of learning and instruction (e.g., Smith & Ragan, 1993, p. 2; Tennyson, 1995a, p. 113; 1994, p. 29; Tennyson & Foshay, 2000, p. 111). The goal of the whole process is to analyze what is to be taught and/or learned, to determine how it is to be taught and/or learned, to conduct tryouts and revisions as well as to assess whether learners did learn (Gustafson, 1996, p. 27). Reiser (2001a) states that “[t]he field of instructional design and technology encompasses the analysis of learning and performance problems, and the design development, implementation, evaluation and management of instructional and noninstructional processes and resources intended to improve learning and performance in a variety of settings” (p. 53; same citation in Reiser, 2001b, p. 57). The literature indicates that the use of systematic design procedures can help to make instruction more effective, efficient, and relevant. For instance, Gustafson & Tillman (1991) state that

    instructional design involves systematically applying a set of principles to achieve effective, efficient, and relevant instruction. .... It is based on the assumption that the quality of the learning experience can be improved by systematically applying the principles and procedures [of instructional design]. (p. 4)

As Gustafson & Tilman (1991, p. 5-6) further explain, the instructional design process is based on six core characteristics. First, rooting in General Systems Theory, instructional design is a systems approach that looks at a system as an
integrated set of elements that interact with each other. Second, in planning a system according to the systems approach, analysis is required of how the components of a system interact with each other and how the contributions of the different players in that system can be best coordinated. Third, the instructional design process represents an orderly, but flexible sequence of steps. The sequence does not imply a linear process, but rather an iterative process that allows for self-correction. Fourth, instructional design is based on research, which, fifth, calls for empirical testing and improvement of the instructional plan. And, sixth, the final version of the instruction needs to be compared with an alternative or at least with the original objectives.

In a study on the role of the instructional designer, Burkman (1987, p. 432) writes that an instructional designer is defined as someone who creates procedures to improve instruction. An instructional designer specifies learning environments for learners in order to perform learning tasks by establishing learning objectives and criteria to measure their accomplishment. The designer develops and implements documented and replicable procedures to organize the conditions for learning. And, the designer defines and measures their accomplishment in terms of learner performance. According to Burkman (p. 432-433) the product of most instructional design processes are instructional materials, which incorporate the learning objectives, instructional strategies, and assessment tools. The author distinguishes between instructional designers as materials developer and as model builder. The first primarily produce instructional materials, whereas the latter are more concerned with establishing a model about generating learning objectives,
assessment tools, and instructional procedures. Further, Burkman (p. 433-434) distinguishes instructional designers by the situation in which they are working. On the one hand, micro-instructional designers are usually concerned with the development of materials ("materials developer") working on a small scale and aiming at directly influencing the behavior of individual learners. On the other hand, macro-instructional designers develop educational products in complex settings with the goal of changing the instructional practices of complex organizations. They can be both model builders or materials developers. Both groups of instructional designers share the same goal, to convince and train instructors to utilize instructional materials properly. Reiser (2001a) concludes that the two core practices of instructional design “are (a) the use of media for instructional purposes and (b) the use of systematic instructional procedures (often simply called instructional design)” (p. 54, italics in original; same citation in Reiser 2001b, p. 57). In a more recent study on the competencies of instructional designers, Richey, Fields & Foxon (2001, chap. 5) indicate that in addition to the generalist instructional designer a number of common specialist roles have emerged in the field of instructional design, including the analyst, the evaluator, the e-learning specialist, and the project manager.

Basically, instructional design is a planning science (Seel, 1999, p. 4). It looks at the process of planning (preparation and making the plan) as well as the product of this process (the actual plan). According to this understanding, instructional design primarily deals with specifying the object of planning, organizing the
planning process, determining the planning tools, conducting the planning process and handling the problems of the implementation.

Als mit der Planung, Gestaltung, Implementierung und Evaluation von Lehr-Lern-Systemen befaßte Disziplin stellt Instruktionsdesign eine Technologie bereit, Lehren und Lernen in verschiedenartigen Lernortsystemen einer zweck- und zukunftsorientierten Gestaltung zu unterwerfen. (Seel, 1999, p. 6)

[As a discipline dealing with the planning, design, implementation, and evaluation of systems for teaching and learning, instructional design provides a technology which helps to deal with the goal-oriented and forward-looking design of teaching and learning in different learning environments. (trans. by author)]

Typically, the core elements of the instructional design process consist of analysis, design, development, implementation and evaluation (e.g., Dick & Carey, 1996, p. 4; Gustafson & Branch, 1997a, p.12, 19; 1997b, 73-74; Kemp, Morrison & Ross, 1998, p. 2; Reiser, 2001a, p. 53; 2001b, p.57)—often referred to as ADDIE (e.g., Reigeluth & Nelson, 1997, p. 30)—which corresponds with the core competencies of instructional designers that Richey, Fields & Foxon (2001, p. 48-55) have identified. In addition to the general professional foundations, the authors list planning and analysis, design and development as well as implementation and impact as the general domains of instructional design competencies. This conception of instructional design extends AECT’s (Seels & Richey, 1994) 1994 definition of the field—i.e. design, development, utilization, management, and evaluation—by adding analysis as a separate category to the process.

1.1 Analysis

As Dijkstra & Merriënboer (1997) explain, the instructional design of learning environments begins with the analysis of learning needs, or what is also called a situational evaluation “to determine if a problem or need exists and if a
suitable solution can be proposed” (Tennyson & Foshay, 2000, p. 127). This step leads to the specification of learning goals or objectives and the program requirements. The problem situations are then selected, which form the basis for the actual design of the instruction, the learning environment and the instructional material, which are finally evaluated and revised.

Analysis may include a needs assessment at the micro or macro level to determine goals in order to identify discrepancies and to establish priorities for actions for instruction (e.g., Burton & Merrill, 1991; Dick & Carey, 1977; Gentry, 1994, chap. 2; Kemp, Morison & Ross, 1998, p. 20-26; Rossett, 1991; Smith & Ragan, 1993, p. 27-30). It may also include a goal analysis, “a visual display of the specific steps the learner would do when performing the instructional goal” (Dick & Carey, 1996, p. 37). Goal analysis was introduced by Mager (1972) to help in determining the performance requirements that collectively define the goal in order to determine which performances need to be taught and to select the appropriate instructional procedures (see also Kemp, Morison & Ross, 1998, p. 28-31).

Another important activity is task analysis, which was first introduced by R. B. Miller (1963) in the 1950s and 1960s. A task analysis is conducted to identify and determine the kind of learning that a learner is expected to perform, which helps to articulate the overall mission for the design process (e.g., Jonassen, & Hannum, 1986/1991; Jonassen, Hannum, & Tessmer, 1989; Jonassen, Tessmer, & Hannum, 1999; Kemp, Morison & Ross, 1998, p. 51-61; P. F. Merrill, 1987). In addition, an instructional analysis of subordinate skills and entry behaviors can be conducted as well as an analysis of learner characteristics (e.g., Dick & Carey, 1996, chap. 4-5;
Kemp, Morrison & Ross, 1998, p. 38-46; Smith & Ragan, 1993, p. 30-55). In order to address context variables in the instructional design process that can facilitate or inhibit learning, an environmental analysis of the learning contexts is recommended (e.g., Richey & Tessmer, 1995; Seels, 1995; Tessmer, 1990; Tessmer & Harris, 1992; Tessmer & Richey, 1997).

1.2 Design

Design would typically consist of writing instructional objectives in measurable terms (e.g., Gentry, 1994, chap. 4; Gronlund, 1995; Kemp, Morrison & Ross, 1998, chap. 5; Yelon, 1991). The need to identify specific learning objectives was popularized by Mager (1984) in the early 1960s, but was actually first introduced by Tyler in the 1930s (see Reiser, 2001b, p. 59). Typically, a learning objective includes an observable behavior, the conditions under which the behavior needs to be performed, and the criteria based on which the performance will be assessed. In addition, strategies for instruction can be designed in order to determine teaching techniques and methods as well as suggestions to sequence the content (e.g., Dick & Carey, 1996, chap. 8; Gustafson & Tillman, 1991; Kemp, Morrison & Ross, 1998, chap. 6-7). Also, means for the assessment of learner performance need to be designed (e.g., Baker & O’Neil, 1986; Dick & Carey, 1996, chap. 7).

In addition, design may include message design for instruction and learning by developing a plan or blueprint for the symbols or patterns of signs used to facilitate learning (e.g., Fleming & Levie, 1993; Grabowski, 1991; Kemp, Morrison & Ross, 1998, chap. 8). This is closely related to the selection of media for
instruction, which is also part of design (e.g., Merrill & Goodman, 1972; Reiser & Gagné, 1982; 1983; Reynolds, 1993; Reynolds & Anderson, 1992; Romiszowski, 1988).

1.3 Development

During development, all the required materials for learners and instructors are produced based on previously established design specifications (e.g., Dick & Carey, 1996, chap. 9, 11; Gentry, 1994, chap. 5-6; Pett & Grabinger, 1991; Smith & Ragan, 1993, 362-368). Typically, the production process of an instructional unit is broken down into the pre-production stage during which the design specifications are acquired, and the actual breakdown of the work into subprojects, the actual production of the instructional material, and the post-production stage, which consists of the assembly and revision of the prototype and the actual delivery of the final production element masters (Gentry, 1994, chap. 5). A detailed example for this activity in an educational setting is Bergman & Moore's (1990) development model for interactive video/multimedia applications.

1.4 Implementation

Implementation refers to the installation and delivery of instruction as well as to its maintenance and monitoring (e.g., Kemp, Morrison & Ross, 1998, chap. 9; Gentry, 1994, chap. 7-8). As Gentry (p. 5, 177, 197) writes, the implementation process needs to ensure that the necessary conditions for the operation of an instructional product or process have been properly established, and the
maintenance process helps to guarantee the ongoing operation of the instructional product or procedure after its installation.

1.5 Evaluation

In general, evaluation in education is the “process of collecting and analyzing data about and analyzing values about and assigning values to an ongoing instructional unit, for enabling decisions on maintenance, revision, and/or elimination of its elements” (Gentry, 1994, p. 5; 215). Evaluation includes both formative evaluation and summative evaluation (e.g., Dick & Carey, 1996, chap. 10, 12). The goal of formative evaluation is to identify information with regards to the revision of the instruction (e.g., Dick & Carey, 1991; Tessmer, 1993; 1996). For instance, Nichols (1997) lists e-mail, chat, and on-line forms as techniques that can effectively be used in conducting a formative evaluation at a distance. Summative evaluation is directed at assessing the degree to which the objectives of the instructional event have been achieved (e.g., Carey & Dick, 1991). “[It] is defined as the design of evaluation studies and the collection of data to verify the effectiveness of instructional material with target learners” (Dick & Carey, 1996, p. 323). The resulting suggestions for the revision of instruction can lead either to a change in the accuracy or effectiveness of instructional materials or to a change in the procedures of using the instructional materials. The goal of a summative evaluation is to make decisions about maintaining the instructional materials.
currently being used or adopting new materials that have a greater potential to meet an organization’s defined instructional needs. (Dick & Carey, 1996, chap. 11-12).

2. Models of instructional design

As mentioned earlier, instructional design models capture the different core elements and tasks involved in this process. They are seen as tools to conceptualize and communicate the associated core elements and procedures involved in the process (Gustafson & Branch, 1997a, p. 12-13; 1997b, p. 77-78). In general, instructional design models help to improve learning and instruction, the management of instructional design and development, the evaluation processes as well as the testing or building of learning or instructional theory (Andrews & Goodson, 1980/1991, p. 136).

2.1 Four generations of instructional design

In general, the literature suggests that the origins of instructional design in the United States date back to World War II, when there was an increased demand in the military to improve training programs (e.g., Dick, 1987, p. 183-184; Reiser, 2001b, p. 58; 1987, p. 12; Tennyson & Foshay, 2000, p. 114). In the years following the war, various innovations have been made in analysis, design and evaluation procedures (e.g., task analysis, formative and summative evaluation, learning objectives, and criterion-referenced testing), which were later in the early and mid-1960s “linked together to form processes, or models, for systematically designing instructional materials” (Reiser, 2001b, p. 61). During this period, terms such as instructional design, system development, systematic instruction, and instructional systems have been introduced in early models, for instance, by Barson,
Tennyson (1995a, p. 117-122; 1992, p. 38-42; Tennyson & Foshay, 2000, p. 114-127) has identified four generations of instructional design models that have been developed since World War II. The first generation of instructional design models of the 1960s, ISD$^1$, is characterized by a linear, step-by-step approach for subject matter experts to develop instruction based on the behavioral paradigm of learning. ISD$^1$ models were primarily derived from Glaser’s action research methodology of the early 1960s.

With the advancement of instructional technology the ISD process became more complex. Second generation models of the 1970s, ISD$^2$, addressed this issue by applying systems theory based on Bertalanffy’s general systems theory (e.g., Banathy, 1987) according to which the different tasks involved in the instructional design process have been integrated (e.g., Dick, 1987, p. 189-192; Gustafson, 1996, p. 28). For instance, Banathy—an early proponent of this approach—defines instructional systems design (ISD) as

a creative, disciplined, and decision-oriented inquiry that aims to: (a) formulate and clarify ideas and images of alternative desired states of a system; (b) prepare descriptions, representations or “models” of the system; and (c) devise a plan for the development and implementation of the selected (most promising) model. (p. 89)

Dick (1995) writes that ISD helps to determine what to teach and how to teach it. He states that “[t]he entire ISD process is systematic in that each step flows from the preceding one, and evaluation and associated revisions are used to determine when the instruction is acceptable to the client” (p. 13). According to
Banathy (1987, p. 90-92) ISD is based on the interplay of several conceptual spaces. The main inquiry of ISD is centered around the design-solution space. It begins with establishing the design purposes and goals, and is completed with the creation of the model of the system. Linked to this space are the contextual space, the experience space, and the organized-knowledge space. The contextual space (also called the environment of the space or problem space) represents the foundation of the definition, study, and characterization of design-relevant systems as well as systems of problems relevant to the design. The experience space is a creation of the instructional designer in order to test design alternatives and the emerging model. Finally, the organized-knowledge space contains the knowledge areas to be explored in order to establish and enrich the knowledge base for the design.

These second generation models still apply a linear, step-by-step approach, where the output of one authoring activity becomes the input for the next in a closed loop, but now more activities have been added, e.g., establishing goals as well as objectives, analyzing the target audience, reviewing and selecting existing materials, selecting an instructional delivery system, and issues associated with implementation. In addition, these models considered the instructional design process as a team process involving an instructional development technician alongside with a subject matter expert. A well-known example of this generation is Dick & Carey’s (1996) model of the systematic design of instruction, which was first introduced in 1978 and has just recently been published in its fifth edition in 2001; actually, Gagné introduced Dick & Carey’s model in the third edition of his
most influential work on the principles of instructional design (Gagné, Briggs & Wager, 1988, p. 22).

The third generation of instructional design of the 1980s and 1990s, ISD\textsuperscript{3}, opened the closed-loop system design of ISD\textsuperscript{2} models by introducing iterative procedures in each of the core phases of analysis, design, development, implementation, and evaluation, allowing to take account for situational differences among applications especially with regards to the design of technology-based, multimedia systems with increased learner control. However, ISD\textsuperscript{3} models still maintained a basically linear process between the individual phases. The instructional development technician gave way to instructional design experts who were able to manipulate the individual phases based on specific application needs and constraints. ISD\textsuperscript{3} models paid greater attention to the analysis (or assessment) phase and introduced new forms of evaluation at each phase of the process, i.e. feasibility and maintenance, increasing the function of project management in a more flexible work flow compared to the linear, top-down structure of ISD\textsuperscript{2} models. An example of ISD\textsuperscript{3} is Bergman & Moore’s (1990) development model for interactive video / multimedia projects.

As indicated earlier, the core elements of the instructional design process have mostly been modeled in a linear fashion beginning with analysis and ending with evaluation/revision (Gustafson & Branch, 1997a, p. 20-23; 1997b, p. 74). A more recent example of this notion is presented in Braden’s (1996) 15-steps top-to-bottom formative evaluation instructional design model, which follows much of Dick & Carey’s (1996) ISD model—for a positive reassessment of their own model
Another example is provided by Clark (1999, p. 11), which introduces Horn’s (1982; 1989) structured writing approach as a minor revision of the basically linear instructional design process. Reigeluth & Nelson (1997), in turn, propose a new paradigm to overcome the notion of a linear process in instructional design. They suggest an iterative series of design decisions, each of which would require a cycle of analysis, synthesis, evaluation and change, thus collapsing design and development into a single step in the process. This user-designer approach integrates elements of rapid prototyping, which allows for greater participation of learners in the overall design process.

The notion of an iterative, incremental design process is at the core of instructional design models of the fourth generation, ISD⁴, which began to emerge in the 2000s. Advancements in disciplines such as cognitive psychology, organizational theory, complexity theory, and software engineering have increased the complexity of instructional design by altering and expanding the variables and conditions of ISD. ISD⁴ models try to address this complexity and situational application differences by introducing a dynamic, interactive diagnostic/prescriptive systems design approach using a rule-based knowledge system, which results in "a nonlinear system design that can continuously interact between the problem, the solution, and the authoring activities" (Tennyson & Foshay, 2000, p. 124). An early example of an ISD⁴ model is Merrill’s ID₂ system created for automated instructional design (e.g., Merrill, Li & Jones, 1990a; 1990b).

Recent developments indicate that instructional systems design needs to be revised and enhanced by integrating, for instance, rapid prototyping or automated
systems design (e.g., Gustafson & Branch, 1997b, p. 83-86; Reiser, 2001b, p. 63; Tennyson & Foshay, 2000, p. 123-124; Wilson, Jonassen & Cole, 1993). For instance, electronic performance support systems (EPSS) provide new mechanisms to enhance the design process itself (Dick, 1995, p. 16-19; Reiser, 2001b, p. 63; Wilson, 1999). These systems not only support the automation of the instructional design process itself, but also help participants in this process to engage in corporate problem-solving, thus enhancing the notion of instructional design as “a systematic process of problem-solving” (Wager, 1995, p. 10; italics in original). Wager also explains with regards to EPSS that instructional design needs to make greater use of technology like computers, expert systems, information databases, and computer networks in order to increase productivity and to better meet the needs of learners (p. 11). More recently, knowledge-based information systems for instructional design have been proposed to enhance human performance and to help in locating and improving access to useful knowledge within organizations (e.g., Reiser, 2001, p. 64; Schott, 1991, p. 210-213).

Similar to Reigeluth’s & Nelson’s (1997) notion of an iterative, incremental design process, Wilson, Teslow & Osman-Jouchoux (1995) argue for a greater participation of all stakeholders in the instructional design process. The authors suggest a set of guidelines for revising instructional design based on constructivism. They criticize that in conventional instructional design usually a single designer and subject matter expert are working in isolation. Instead, the authors suggest that all major stakeholders should be represented in the design team of a constructivist instructional design project. This participatory design approach allows for the
incorporation of multiple perspectives in the design process. As such, the final product may be able to accommodate multiple goals, styles, and perspectives in instruction. Their guidelines are organized according to the core elements of instructional design, but provide a new understanding of the individual tasks (p. 148-154).

Basically, the constructivist instructional design process is founded on a holistic or systematic model that considers all instructional factors that may have an impact. The authors suggest to apply a use fast-track or layers-of-need model—in correspondence to Tessmer’s & Wedman’s (1990) layers-of-necessity instructional development model—in order to better accommodate the learner’s needs in a given situation. The participatory approach includes instructors and learners in the process, among others, while implementing rapid prototyping techniques. Thus, the design model can be developed “in the field” and be tested early in the development process. During needs assessment greater emphasis is to be put on the performance context (e.g., EPSS). Gap-oriented strategies need to be enhanced by consensus- and market-oriented needs assessment. During goal/task analysis designers need to distinguish between educational and training situations; both may arise in an individual organization. The objectives derived from this process serve as heuristics to guide the design process. The authors suggest to develop multiple layers of objectives that cluster around rich learning experiences. They argue that the content for instruction as well as the goals cannot be fully captured. Thus, during instruction room needs to be provided to let these elements emerge and to be
integrated while the instruction is in progress—e.g., by designing flexible shells in a technology-mediated learning environment (Winn, 1991, p. 205-208).

The primary goal of instruction is problem-solving and meaning construction. This can best be achieved by providing authentic, information-rich methods for representing content and assessing performance. In order to allow for different perspectives in dealing with a topic, multiple ways in approaching and representing a problem should be provided. Supporting learners in personal knowledge-building should be the primary goal of any instructional strategy. Thus, designers should rather develop learning environments instead of selecting strategies in order to allow for multiple goals for different learners by implementing multiple perspectives. This may include, among others, the use of multiple representations in the context of ill-structured domains, which help to develop cognitive flexibility (see for instance Jones & Spiro, 1995; Spiro et al., 1991; Spiro & Jehng, 1990), or the use of cognitive apprenticeships, which apply the modeling, coaching, and fading paradigm of the traditional apprenticeship model, but focus on cognitive rather than physical skills, and allow for situated learning, i.e. the learning of knowledge and skills in a relevant context that will be useful in real life (see for instance A. Collins, 1991; Collins, Brown & Newman, 1989; LeGrand, Farmer & Buckmaster, 1993; Teles, 1993).

Finally, media and assessment also play an important role in supporting the design of an constructivist learning environment. It is important to select media early in the design cycle and to base the selection on media literacy and biases of the learner towards particular kinds of media. Assessment should be fully integrated
into the learning experience by evaluating not only the process but also the product of instruction—i.e. formative and summative evaluation. Therefore, formal assessments tools should be combined with informal assessment, e.g., observations of eye contact, body language, facial expressions, and work performance.

According to Jonassen (1999; 1997; see also Duffy & Jonassen, 1992b), a technology-based constructivist learning environment is centered around a problem, question, or project. The goal of the learner is to arrive at an interpretation and a solution of the problem or to complete the project. The problem is represented in a context which is authentic and allows the learner to manipulate something and affect the environment in some way. In order to understand a problem, a learner needs to experience it and construct a mental model. When these experiences are missing, the learning environment provides a set of related experiences or cases that help in scaffolding the learner’s memory and enhance cognitive flexibility. In addition, the learner needs information resources that help him/her to construct a mental model and express hypotheses that lead to the manipulation of the problem space. In order to help the student construct knowledge, a set of cognitive tools is provided, which scaffold the learner’s abilities to perform a specific task. These tools are designed to represent the problem and/or task, to model static and dynamic knowledge, to support student performance and to gather information. Tools are also needed to allow for conversation and collaboration. Finally, the constructivist learning environment must make opportunities available to the learners in order to explore knowledge, to articulate their hypotheses and to reflect about one’s own performance. In turn, the system needs to provide the necessary
instructional support through, for instance, modeling, coaching, and scaffolding (see the earlier discussion in the context of learner-centered design in Chapter IV).

More recently, Willis (2000; Willis & Wright, 2000) has developed a constructivist instructional design approach called the R2D2 model: Recursive, Reflective Design and Development—a revision and refinement of an earlier proposal (Willis, 1995)—which confirms much of the elements of designing a constructivist learning environment as discussed in the previous section. To summarize, Willis’ model is based on three flexible guidelines, i.e. reflection, recursion (iteration), and participation. First, instructional design is recursive (i.e. iterative) and non-linear, an approach that has been introduced earlier in the context of the object-oriented paradigm. This approach supports the development of instructional materials in increments, which allows both the users and the designers to fully participate in the process of revising and reformulating the product. Second, reflective design indicates that cycles of problem framing, implementation, and improvisation need to be implemented as many problems cannot be well-formed and solved with preformulated solutions. Finally, participatory design refers to the importance of allowing all stakeholders to fully participate in the design process based on scenarios (or use cases) as well as in conducting a contextual analysis. To conclude, Willis & Wright (2000) write that the R2D2 model views instructional design as a process of progressively solving multiple problems in context. That is, “solutions”, such as the set of objectives for a project, progressively emerge across the entire design process instead of being completed early and then used to guide design and development work. The fuzzy objective you begin with will influence your design and development work, but conversely, design and development will also influence the objectives. R2D2 views
design work as a richly interactive process, in which solutions emerge across a process, and in which work on many different parts of the whole influence each other (and the whole). (p. 7-8)

The R2D2 model expands the notion of user-centered design towards user-involved design. It not only complements instructional design from a constructivist perspective, but also incorporates successfully methods and principles from other design disciplines, i.e. user interface design.

The above discussion of developing constructivist learning environments is reflected in Tennyson’s (1997; 1995a; 1995b; 1994; Tennyson & Foshay, 2000) concept of ISD⁴, which is not based on a particular learning theory or instructional theory and allows for moment-to-moment adjustments during instructional development. Tennyson’s system dynamics approach to instructional systems development (ISD) is based on the assumption that each learning problem needs to have a different instructional design solution. Instead of a linear solution system, the system dynamics model presents a model- and simulation based methodology. It is designed in such a way that it can dynamically adapt to the problem conditions of a given situation. Tennyson (1997) describes the system dynamics approach to ISD as a complex entity (or phenomenon) that prescribes a solution to a given learning based entirely on the conditions of the situation at hand. The dynamic feature of the nonlinear systems is the possible continuous interaction between the problem diagnosis (i.e. situational evaluation) and the instructional design prescription. (p. 415)

Tennyson system dynamics approach to ISD borrows from disciplines such as complexity theory (Tennyson, 1995b, p. 34; 1997, p. 413-414). Using concepts from this theory, ISD⁴ models are able to provide a more flexible approach to complex problem situations. Tennyson states that because of the differences in each
learning problem, a different instructional design solution is required each time. Linear instructional design models were not able to provide this flexibility and dynamic in responding to a given problem situation. Instead, Tennyson suggests a system dynamics model, which overcomes the rigidness of linear process models and is able to present different treatments to a problem situation.

This notion of handling complexity in an learning environment corresponds to Reeves’ (1999) discussion of decomposing complexity using a learner-centered design approach. Similar to Tennyson, Reeves draws from other disciplines such as cognitive and information science. He argues that cognitive complexity should be the focus as a specific design problem, where the goal of learner-centered design is “to manage the complexity of learning at the interface between the learner and the information source such that the user and object can combine to scaffold efficient and effective interaction” (p. 163). In his toolkit for learner-centered design Reeves combines the different notions of a learner as perceiver, model builder, categorizer, searcher, expert and student in a single framework which provides design prescriptions to develop an interactive system for content and usability based on principles from information design, mental models/hypertext, categorization/classification, visualization, knowledge engineering, and schema-based instruction. Reeves combination and integration of different disciplines and approaches for learner-centered design complements and extends Tennyson system dynamics model.
Figure 17. Concepts of fourth generation ISD models, ISD$^4$ (redrawn and revised from Tennyson & Foshay, 2000, p. 122)
As illustrated in Figure 17 on page 363, the dynamic systems model has replaced the sequential structure of the instructional design models from previous generations with an interactive set of six domains or clusters of authoring activities. Based on a situational evaluation (also called feasibility evaluation or diagnosis) of the learning situation, a set of prescriptions is suggested to help address the problem. Thus, the situational evaluation determines which kinds of authoring activities need to be performed within the set of interactive instructional development (ID) domains, i.e., foundation, design, production, implementation, and maintenance. Taken together, these domains form a knowledge base. This concept helps to better address the complexity and interdependency of instructional decision-making. The activities are connected by a given context, rather than their specific position in a sequential model. This allows to focus more on one set of activities in a given situation than another that might call for a different set of activities. As Tennyson & Foshay (2000, p. 128) point out, the situational evaluation may lead to the conclusion that the current learning situation does not require any changes, or that only existing instructional materials have to be adopted, which may only require the implementation domain, or that existing instruction has to be adapted or a new learning environment has to be developed, which may require a more complex system designs.

The foundation domain consists of authoring activities, which establish the theoretical basis (i.e., educational philosophy, educational learning theory, and instructional theory) on which the instructional development decisions are founded. The design domain is concerned with preparing detailed specifications (e.g.,
prerequisites, instructional strategies, message design) of the learning environment that are needed for the actual development. During production, all the instructional materials for the learning environment are developed. The implementation domain includes activities that lead to the integration and delivery of the newly learning environment. Finally, the maintenance domain has been included to ensure a level of effectiveness of the learning environment when it was first implemented.

A critical aspect of Tennyson’s aspect is the fact that evaluation is not listed as a separate domain, but is fully integrated at various places in the process. First, a evaluation plan is prepared (see the foundation-design subdomain), which helps to establish the quality management of the development effort. The design specifications are first revised and refined during a formative evaluation, which is part of the design domain. Another formative evaluation follows when the rapid prototype of the learning environment is assessed and revised (see design-production-implementation subdomain). Yet, another formative evaluation is conducted to refine the rapid prototype (see production-implementation subdomain). Finally, a summative evaluation is conducted and reported to continuously assess the effectiveness of the new learning environment (see implementation-maintenance subdomain).

In a discussion of Tennyson’s system dynamics model, Kerres (1998, p. 324-325) falsely attributes Tennyson’s situational evaluation as a separate authoring activity instead of both identifying it as a diagnostical tool for front-end analysis, which leads to various authoring activities in the other instructional development domains, and noticing that formative as well as summative evaluation are integral
parts at different stages of the overall development process. In addition, Kerres states that Tennyson’s model does not serve well for a structured approach of instructional planning activities because it does not provide enough input for the actual instructional designer, especially with regards to a particular time-line. The author thinks that Kerres’ critique does not fully capture the main point of Tennyson’s model, in that it actually wants to abolish the linear or sequential approach of previous instructional design generations. Rather, Tennyson’s approach allows an instructional designer to conduct a situational evaluation that would allow him/her to make decisions on how to further proceed in the process. This decision can then affect the selection of a set of authoring activities from certain clusters of domains based on the situational evaluation. Thus, depending on the demands of the situation, an appropriate instructional design model can be created that is flexible and dynamic. As Tennyson & Foshay (2000) explain— in correspondence to Boehm’s (1988) spiral model of software engineering—the instructional systems development process “progresses module by module in a ‘spiral’ from the foundation-design subdomain through prototyping, evaluations, production, and finally through implementation and maintenance” (p. 141). For instance, this flexibility is very important for instructional Web sites design, as it allows to make changes according to the scope of the various possible manifestations of Web use for learning and instruction. However, taking into consideration the knowledge base aspect of Tennyson’s model that is very suitable for an automated instructional design model, it would be interesting to further
refine the model with approaches from knowledge engineering and management, e.g., CommonKADS (Schreiber et al., 2000).

2.2 Classifying instructional design models

The literature lists hundreds of instructional design models that have been developed over the decades for different purposes and contexts. Based on their comparative analysis of forty models of instructional design, Andrews & Goodson (1980/1991) state that the basic goals of these models are to improve learning and instruction, the management of instructional design and development, the evaluation processes as well as to test or build learning or instructional theory. The authors write that the instructional design model is often used as a kind of game plan intended for development efforts. “This plan assures the educator that every piece of instruction that is used will, regardless of content, have recognizable elements” (p. 137). A revised version of Andrews & Goodson’s (1980/1991) framework for the comparative analysis of instructional design models has been proposed by Edmonds, Branch & Mukherjee (1994), which is based on an analysis of fifteen models. The authors indicate that instructional design is both systematic because of its inherent input-process-output paradigm and systemic because each of the outcomes of each component influences the other components of the instructional design process. They conclude that “instructional design is intended to be responsive to the educational environment, learner-centered, goal-driven,
procedural and sensitive to each relationship that occurs during the instructional episode” (p. 57).

Recently, a more comprehensive classification system for instructional design models has been suggested that integrates the activities into three classes of classroom orientation, product orientation and system orientation (Gustafson, 1996, p. 29-30; Gustafson & Branch, 1997a, p. 27-32; 1997b, p. 78-81). The classroom-oriented instructional design models are based on the assumption that a single instructor is responsible for the preparation and organization of instruction. An example is the model suggested by Kemp, Morrison & Ross (1998), which focuses on curriculum planning. Another example for this group is the instructional design template (IDT) for instructional design proposed by West, Farmer & Wolff (1991), which introduces principles and strategies from cognitive science (e.g., chunking, framing, concept mapping, advance organizer, metaphor, rehearsal, imagery, and mnemonics). The product-development instructional design model is oriented towards the production of specific instructional materials for a limited scale. A well documented example of this group is Bergman & Moore’s (1990) model for the design of interactive video and/or multimedia projects. Finally, the system-oriented model focus more on large-scale projects such as a course or an entire curriculum. The most commonly known model of this group is the one developed by Dick & Carey (1996; see also Dick, 1996; 1997), which is widely used as an introductory textbook in the teaching of instructional systems development. Other models in this category include the one from Smith & Ragan (1993), which also introduces principles from cognitive psychology, and Gentry’s Instructional
Project Development and Management (IPDM) model, which also addresses supporting components for the instructional design process.

2.3 Instructional Web site design

As alluded to earlier, it is important to integrate instructional design when developing Web-based learning environments. However, because of the different possibilities and the varying scope of incorporating the Web as an instructional technology in the teaching and learning process ranging from a supplement for a single instructional event to an educational system that is fully embedded in the Web, a classroom-orientation, a product-oriented, and/or a system-oriented model could be applied. In addition, not only the specific structure of the Web site has to be developed to deliver the learning environment, but also the various Internet media types that can be embedded in this distributed hypermedia environment, which would require knowledge and expertise from specialists in graphic design, interaction design, interface design, audio, video, illustration, animation, programming, etc.

In fact, most of these specialists use process models that consist of core elements similar to that of instructional design (e.g., analysis, design, development, implementation, evaluation, and maintenance); see for instance, designing for electronic publishing (e.g., Assadi, 1998, chap. 2) and print publishing (e.g., Cottrell, 1998, chap. 4), educational software engineering (e.g., Unesco, 1990, chap. 1.3), information design (e.g., Jansen & Scharfe, 1999, p. 58-59), information architecture for the Web (e.g., Reiss, 2000; Rosenfeld & Morville, 1998), multimedia authoring (e.g., Clarke & Swearingen, 1994, chap. 1), multimedia/
hypermedia design (e.g., Apple Computer, 1994, p. 59-257; Cotton & Oliver, 1997, p. 94-105; England & Finney, 1999; Mok, 1996, p. 54-59; Schifman, Heinrich & Heinrich, 1997, p. 83-156), typography (e.g., Rüegg, 1989, p. 121-124), and Web design (e.g., Black, 1997, p. 172-181; Conger & Mason, 1998; Donnelly, 2001, chap. 2).

For instance, Ruffini (2000a; 2000b) suggests a systems approach for the design of an educational Web site for classroom use based on Kemp, Morrison & Ross’ (1998) model. Similar to R. Hall’s (1999) instructional Web site design principles, Ruffini integrates instructional design decisions, e.g., determining learner characteristics, goal analysis, and specifying learning objectives, with basic hypermedia design principles, e.g., site structure, navigation system, page design, test design, and visual design. Authors like S. Horton (2000) or Keating & Hargitai (1999) stress more the importance of Web design principles in preparing for Web teaching, but do not discuss in detail the need for instructional design in this process. Others, like M. Driscoll (1998) or W. Horton (2000)—he calls this structure a course framework (chap. 4) in correspondence with Gagné’s (Gagné, Briggs & Wager, 1992, p. 185; 1988, p. 177) notion of a course architecture (see also Clark, 2000)—explain in detail how to develop a Web-based learning environment based on instructional design principles and how to be involved as an instructor in such a system, but at the expense of the input from other design disciplines that should be part of this process. Finally, Lee & Owen’s (2000) instructional design prescriptions for technology-based instruction (i.e. multimedia, CBT, WBT, and distance learning) does not really integrate instructional design with
the design tasks of the individual domains or technologies as the title of multimedia-based instructional design might implicate. They simply use a sequential multimedia production model which focuses on the creation of a learning environment without indicating the decisions that have to be made with regards to the types of media to be selected as well as the philosophical or theoretical foundation based on which the learning environment is constructed. Their suggested strategies very much resemble those used in CBT including items for presentation, demonstration, practice, and assessment (see for instance Gibbons & Fairweather, 1998, chap. 8-9), which are based, in turn, on Gagné’s (Gagné, Briggs & Wager, 1992, chap. 10) events of instruction.

Most of the authors fail to base the instructional Web design process into a broader framework that not only address instructional design principles, but also methods and approaches from other disciplines such as user interface design and software engineering that play an important role in the design of interactive hypermedia systems. What is needed is a specific instructional design model for Web site development that would integrate all these different aspects from other disciplines in a single framework. As discussed earlier, the literature on the design of computer-assisted instruction (CAI) has illustrated how the instructional design process can be integrated with systems design in educational software engineering (e.g., Venezky & Osin, 1991) or for interactive multimedia instruction (e.g., Schwier & Misanchuk, 1993, chap. III). However, these approaches are too limited in their scope as they do not include the entire physical and social infrastructure of
interactive hypermedia and their potentials to motivate learners for independent or collective learning process as suggested by Kerres (1998, p. 17).

More recently, authors like the Joint Advanced Distributed Learning (ADL) Co-Laboratory (Joint ADL, 2001), Bruns & Gajewski (2000, chap. 2-3), Conrad & TrainingLinks (2000), or McCormack & Jones (1998, chap. 3) provide more integrated approaches that address instructional design and media design as elements of an overall process for instructional Web site development. In their guidelines for the design of Web-based instruction, ADL suggests a collaborative and iterative systems design process which fully integrates Web site design and instructional systems development. Following the conventional ADDIE phases, the authors list the different authoring activities of the two combined processes parallel in time. Bruns & Gajewski’s notion of the design of a Web-based learning environment integrates decisions about the selection of different media assets and their integration into a hypermedia system as well as the composition of the user interface and its elements with instructional design. In their training development cycle, Conrad & TrainingLinks (p. 14-15) integrate these processes by adapting decisions about multimedia / hypermedia design—in particular user interface design—to the activities of creating an instructional Web site. According to their model, the instructional design process generates a blueprint that is used as a specification for Web user interface design. Most helpful is their notion of a learning path in the system (chap. 5) that clearly integrates the instructional prescriptions with the information architecture and navigation system of the Web site, which basically constitutes the user interface of the system (see also the discussion of a
McCormack & Jones suggest a five-step process for instructional Web site design. Using rapid prototyping with paper-mockups the authors suggest to first develop a list of educational goals based on which, then, the implementation methods are identified. Third, different approaches or instructional strategies are specified, which take into consideration the capabilities and limitations of the system. Fourth, the overall content is structured and organized into an information architecture along with the appropriate navigation system. Finally, the actual presentation design of the user interface is conducted. The resulting paper-based prototype is used as a blueprint or specification for the subsequent content production and distribution.

Although the authors introduced above do integrate the different disciplines that converge in the instructional Web design process, they do not establish a particular instructional design model that could be applied as a general framework, especially with respect to the use of learning objects—as in the case of the Joint ADL Co-Laboratory who discuss the implementation of reusable learning objects. A more integrated framework for instructional design is presented by educators in Germany in recent years based on the reception and intensive discussion of U.S.-based instructional design theories called multimedia didactics, which integrates general instructional design (called didactics) with message design or media selection and development of instructional technology (called media didactics) into a design-oriented development model for instructional multimedia / hypermedia (e.g., Issing, 1994a; 1994b; 1997; Kerres, 1998, chap. 1; Strittmatter & Mauel,
Based on the five views of instructional systems design presented by Schiffman (1986/1991), Schott (1991) also comes to the conclusion that instructional design is often limited to a process model for media selection and development. Separating the media from the instructional design process leads to their isolation, which focuses more on the physical qualities of the media instead of the conditions under which they can be applied in instruction (see also Kerres, p. 11). Rather, Schott suggests a model for knowledge systems design, which integrates instructional design along with what he calls knowledge-meta-media that represent a combination of different media assets, into a comprehensive knowledge delivery and usage system. More specifically, a design-oriented multimedia didactics is needed, as suggested by Kerres (p. 12), which looks at both the decisions about selecting media for instruction as well as how to produce them in a single process. Based on this concept, multimedia learning environments represent a total social and physical system which emphasizes technology-mediated learning by integrating media and other support tools (p. 16). Design-oriented multimedia didactics resembles a product-oriented or production-oriented design model, which is not primarily interested in the analysis and reflection of learning with media nor in the basic decisions about the use of media and their impact, but rather in the scientific comprehension of the conceptualization and development of multimedia learning environments (p. 23-24). As Kerres (1998) explains:

Es kann also festgehalten werden, daß die gestaltungsorientierte Medienproduktion auf dem Hintergrund didaktisch-konzeptueller, also den Prozeß des Lehr-Lerngefahrens übergreifender, Erwägungen betrachtet. .... Die hier vorgestellte Mediendidaktik versteht die Entwicklung und Gestaltung medialer Lernumgebungen jedoch als eigenständige Aufgabe.
unabhängig von Schule und Unterricht und zielt so auf die Professionalisierung der Produktion didaktischer Medien ab. (p. 29)

To conclude, design-oriented media didactics views all planning and development steps of media production based on didactic-conceptual decisions that encompass the process of learning-teaching events. The proposed media didactics views the development and design of media-based learning environments as an independent task that is not limited to teaching and instruction in schools. Instead it is aiming at professionalizing the production of instructional media. (trans. by author)

The core element of this process is the so-called interactive script (or storyboard) for the instructional event (Issing, 1997, p. 214; Kerres, 1998, chap. 4.3). It is based on the conceptual model of the interactive learning environment, which was developed during the analysis phase. It serves as the blueprint or specification for the design of the individual media assets (e.g., photographs, video, and sound, text and screen design, as well as computer graphics and animation) which are embedded in the overall information architecture of the interactive system along with the navigation system. Finally, this system is developed, duplicated, distributed, and implemented. Formative evaluation during the design and development process, and summative evaluation after the implementation produce the input needed for revisions.

3. Object-oriented instructional systems development

As indicated in previous chapters, the object-oriented paradigm is not only being used in software engineering and user-interface design, but has also entered the realm of designing multimedia for instruction. For instance, Allen, Chiero & Hoffman (1996) explain that multimedia developers tend to organize their software applications around media objects (e.g., sound, pictures, text, and graphics) that are stored in media databases. The authors argue that object-oriented thinking can help
to reduce the complexity of a knowledge domain, an approach that also has implications for the design of constructivist learning environments. Gibbons & Fairweather (1998, chap. 3, 6; 2000, p. 419-420), for example, discuss the use of (graphical and logical) objects in the context of creating frame-like structures in computer-based instruction (CBI). According to the authors (chap. 4), a frame is considered an author object, which is used for data entry in order to create dynamic experiences for the learner. They are the building blocks of the instructional strategies. “A frame represents a grouping of smaller objects which are assembled into a kind of pre-fabricated unit” (p. 58). Typically, a frame consists of an expressive or visual display element and an unseen computer branching logic. The basic frame types include presentation frame, menu frame, question frame, and calculation frame. Objects in a frame can be manipulated by changing their attribute values.

Another example of using the object-oriented approach in the design of instructional multimedia is presented by Boles, Boles, Dawabi, Schlattmann, Trunk & Wigger (1998). The authors demonstrate how an object-oriented approach in combination with UML can be used to develop a multimedia application about an experiment as part of a virtual genetics laboratory system with a specific authoring tool (i.e. Macromedia Director). The object-metaphor underlying this approach

1. The use of frame in this context differs from frames applied in artificial intelligence or cognitive science (see for instance West, Farmer & Wolff, 1991, chap. 3-4). According to their understanding, a frame represents a grid or matrix with slots for concepts, categories and/or relationships, whose attributes are filled with values. For instance, Protégé-2000 is a so-called frame-based knowledge engineering system that follows this understanding (Stanford Medical Informatics, 2000).
implies that the objects and their properties are represented visually and can be manipulated by graphical means.

3.1 Introduction

According to the object-oriented paradigm, the object-oriented system architecture consists of object views, which combine process instances and files into objects. “In the object-based view, a system is a network of objects that communicate by message passing” (Collins, 1995, p. 306). This understanding leads to the notion of interactive systems and applications that constitute networks of communicating objects, which collaborate to provide visible interfaces to end users (p. 335).

The diagram shown in Figure 18 on page 378—which was inspired by Mok’s (1996, p. 123) system architecture of a Web site—provides a view of how various notions of objects within a computing environment relate to each other. It not only illustrates a learning technology systems architecture, but also provides an integrated view of the elements required to create that architecture by following the notion of the object-oriented system architecture. Developing such a learning technology systems architecture to implement and deliver learning objects requires the collaboration of different specialists. The lower levels of the diagram demand the skills of computer programmers, indicated by the shades in dark grey. In contrast, the top layers require the more abstract and conceptual design skills of specialists like interface designers, information architects, and instructional
designers, as indicated by shades in light grey, who are able to mediate between these two worlds.

Figure 18. The layers of a learning technology systems architecture

The conceptual framework of the object-oriented instructional design
(OOID) model, as discussed in this section with the use of UML, follows this notion of layers and components embedded in a systems architecture. It is based on Tennyson’s & Foshay’s ISD model and incorporates elements from Oesterreich’s (1997/1999, chap. 3) business-oriented software engineering process with the following features: use case-driven, architecture- and component-centered, iterative, and incremental.

First, use cases are identified as the basis for further authoring activities. As suggested by Douglas (2001, p. 3-4), use cases are introduced in order to extend common analysis methods in ISD such as task analysis by utilizing established object-oriented analysis methods. Use cases are employed to determine learner and system requirements of the learning environment.

Second, the OOID model is architecture- and component-centered. OOID is aimed at supporting instructional designers in developing learning objects, which must be adaptable to different learning technology systems architectures. As indicated earlier, such an architecture consists of layer models and is further subdivided into components and subsystems, and could be applied to a broad range of learning scenarios. For instance, according to the IEEE Standard for Learning Technology Systems Architecture (LTSA) (Learning Technology Standards Committee, 2001; see also Sonwalkar, 2002), such an architecture consists of five refinement layers (i.e. learner and environment interactions, learner-related design features, system components/conceptual model, implementation perspectives and priorities, and operational components and interoperability). This standard describes the requirements, functionality, conceptual model, and the semantics of a
learning technology systems architecture. However, only the third, the component layer, is normative according to this standard. This layer represents the component-based architecture. It consists of four teaching/learning processes (learner entity, evaluation, coach, and delivery process); two data stores (learner records and learning resources); and thirteen information flows among these components (behavioral observations, assessment information, learner information, query, catalog info, locator, learning content, multimedia, interaction context, and learning preferences). LTSA provides interoperability interfaces for learning technology systems, similar to what Tennyson & Foshay (2000) refer to as management system and delivery system. However, it does not identify interoperability interfaces for content development (i.e. instructional design) or administrative systems (e.g., business operation services). Learning objects are a critical element in the LTSA standard. They describe learning content with a focus on reusability that is integrated with the structure and sequencing of units of knowledge and information, organized into learning resources. In the shape of catalog info, learning object metadata describe the learning resources in order to enhance the facilitation of searches.

Finally, in correspondence to Tennyson’s & Foshay’s (2000, p. 139-140) assertion to employ the rapid prototyping design methodology, the development process is iterative and incremental. The use-case driven approach allows to develop rough requirement specifications, based on which a system can be structured into subsystems or components. Subsequently, these subsystems or components can be developed by separate teams and later integrate into the learning environment.
Thus, this approach is more flexible than the conventional linear process with several separate phases—which ISD\(^4\) replaces by a dynamic approach—in that the intermediate results (or prototypes) are regularly reviewed (i.e. formative evaluation of the prototype for revisions and refinement). Oesterreich (1997/1999, p. 52) and Tennyson & Foshay (2000, p. 141) indicate that each cycle or iterable step of the development process spiral consists of detailed planning and an analysis-design-implementation sequence. Thus, iterative means that the development process is decomposed into several similar steps. And, incremental refers to the growth of the total functionality of the system with each step.

The OOID process model is characterized by the subdivision of authoring activities into small units and the junction of these activities with different levels of detail. Similar to Oesterreich’s business-oriented software engineering process, Tennyson’s & Foshay’s (2000) ISD\(^4\) model represents a highly dynamic, interactive diagnostic/prescriptive approach, which is “a nonlinear system design that can continuously interact between the problem, the solution, and the authoring activities” (p. 124). Thus, the openness and flexibility of this process model allows to better address problem situations and concrete application areas by formulating prescriptions within the conditions and variables of a given situation.

3.2 The OOID development phases

As illustrated in Figure19 on page382, the development process can be more or less divided into the key phases of situational evaluation and requirement
analysis, problem domain analysis and design, rapid prototyping, and implementation and maintenance,

![Diagram of OOID development phases](image)

Figure 19. Overview of the OOID development phases

Each phase is structured into a number of individual authoring activities. As indicated by Oesterreich (1997/1999, p. 52-53), the phases of situational evaluation and requirement analysis, problem domain analysis and design, and implementation and maintenance usually proceed consecutively, whereas the rapid prototyping phase represents the “development process proper”, proceeding iteratively and incrementally. The following section provides a more detailed description of the individual phases and their authoring activities with the help of activity diagrams (which are inspired by Oesterreich, 1997/1999, p. 56, 69, 72, 74), such as the requirements analysis diagram in Figure 20 on page 383. Given the scope of this study, we can only provide general descriptions of the object-oriented instructional systems development process and the corresponding authoring activities.

3.2.1 Situational evaluation and requirement analysis

In order to develop a successful learning environment, information about the underlying problem or needs have to be collected. The front-end analysis is not only important for ISD, but is of equal importance in, for instance, software engineering and user interface design, thus they can share many of the same methods. In the context of ISD, the gathering of (learner and system) requirements includes the
conditions and parameters of the problem or need of a given learning situation, from which a solution strategy or methodology for the ISD project can be derived. The result is not a description of the actual learning environment, but the instructional prescriptions to develop the system by answering the question “‘What kind of instructional development effort, if any, would be feasible and desirable?’” (Tennyson & Foshay, 2000, p. 127). The activity diagram in Figure 20 on page 383
shows the main authoring activities that typically occur at the beginning of an ISD project.

3.2.1.1 Situational evaluation

Before a solution plan to solve a learning situation can be made, the scope of the actual learning problem or need has to be determined through a situational evaluation. This may show various main solution variations, based on which a concrete solution variation or alternative solutions are derived. Generally, a needs assessment analyzes the learning problem or need and defines it in terms of curricular needs and goals as well as target audience characteristics. Further, the constraints, resources, and risks are determined in order to extend the assessment of the learning problem or need. The learning situation may ask for one or more solutions of learning activities at different levels in the system. In addition, the assessment of the target audience helps to determine the learner characteristics (user requirements). Also, an environmental analysis is needed to specify the context in which the learning environment is embedded (system requirements). Further, it is recommended to develop an initial named list of use cases with brief descriptions to provide a first orientation about the requirements. Finally, it is helpful to determine the instructional development (ID) competence of the ID author to better organize the project team and manage the overall project.

As Tennyson & Foshay (2000, p. 129) write, this process results in a proposal that specifies a solution or alternative solutions. The results from the situational evaluation may indicate either that the current situation does not need any changes or intervention or that existing materials have to be adopted, adapted,
or developed for the (new) learning environment. If the solution plan proposes an intervention, the situational evaluation document, in which the solution plan is described, should state the theme or vision of the instructional prescription, document the feasibility of the effort, specify the system design(s), and define the authoring activities.

3.2.1.2 Rapid analysis and design

In order to introduce the concept of rapid prototyping right from the beginning, a rapid analysis and design workshop may be conducted. This rapid analysis and a first prototype is carried out to determine the fundamental requirements for the general framework. Oesterreich (1997/1999) indicates that the most important business processes and use cases can be “cursorily analyzed and immediately implemented in an exemplary way, that is, first application fragments are realized” (p. 59).

3.2.1.3 Business process analysis

As noted by Oesterreich (1997/1999, p. 56, 59-60), business process modeling clarifies how the system to be developed can be best integrated into existing business processes. Thus, these processes have to be analyzed in the context of the new system. Similar, the processes within an educational setting (e.g., teaching and learning, management and administration) need to be analyzed and modeled as well, including organizational aspects, in order to optimally integrate the system to be developed into the existing environment. For instance, according to Gentry’s (1994) Instructional Project Development and Management (IPDM) model the development or authoring activities are assisted by a set of supporting components, which include the communication of essential information, the
management of resources, the handling of necessary information, the acquisition and allocation of resources/budgets, personnel decisions, and the organization and renovation of facilities. In order to integrate the new learning environment, the points of contact with these surrounding processes need to be specified, which in the literature is called business process modeling.

In fact, UML contains a profile for business modeling. It describes how UML can be customized for business modeling. It contains common stereotypes and some useful terminology (e.g., UseCaseModel, ObjectModel, OrganizationUnit, Worker, CaseWorker, etc.). They can be employed to describe business procedures and documents as well as organization units and resources of the business. Thus, the results of this activity can be expanded in more detail for an instructional setting using an adaptation of this UML profile.

3.2.1.4 Psychological foundations definition

In order to achieve a theoretical framework for the development activities, the underlying psychological foundations need to be established. This includes the definition of the educational philosophy and the educational learning theory, which influence each domain in the instructional development process. Directly linked with this activity is the definition of the instructional theory.

3.2.1.5 Use case analysis

The processes relevant for the learning environment are analyzed and described in so-called use cases. Use cases are not only employed for describing the desired performance requirements for learners, but also for introducing the ID author into the special subject matter of the learning environment. As such, use
cases extend the commonly employed analysis of a domain of information (i.e. content, task, and jobs).

[Image: Diagram showing use case analysis and activity diagram]

Figure 21. Simplified activity modeling of the situational evaluation process

The diagram in Figure 21 on page 387 (based on an example for software engineering in Oesterreich, 1997/1999, p. 129), presents a simplified use case analysis of situational evaluation in a hypothetical automated instructional design tool. The use case diagram combines a set of use cases that have been identified for this activity (actors have not been included in this case). They are translated into a use case description. This description forms the basis of an activity diagram, which
can later be employed to determine the class model as well as operations (sequence) of actions. Additional methods that can be applied in this setting include CRC cards, concept maps, task/user tables, action/object tables, etc.

A use case diagram describes the relationship among use cases within a system and the actors involved. In general, a use case shows the current state or is-state of a (learning) situation. In addition, it needs to be followed up by the development of projections of the future learning environment (i.e. desired case), which formulate and illustrate what information to be learned and the learning processes required to be supported by the learning environment to be developed will look like.

The description of use cases is interlocked with the specification of goals and learning objectives, which form the basis for the development of learning objects and information objects respectively. The goals of the learning environment are determined with regards to those that deal with the acquisition of knowledge (e.g., declarative, procedural, and contextual) and those which deal with the employment and improvement of knowledge (e.g., cognitive complexity). The definition of learning objectives relates to the goals of the learning environment. Cognitive learning objectives—as opposed to behavioral objectives which define end-of-instruction learning outcomes—describe the type of knowledge or knowledge structure and cognitive abilities that a learner needs to acquire. Tennyson & Foshay (2000, p. 131-132) propose a set of five categories of learning objectives that extend Gagné’s (1985) conditions of learning and Bloom’s (1956) taxonomy of educational objectives to include the following: declarative knowledge (verbal information),
procedural knowledge (intellectual skills), contextual skills, cognitive complexity, and constructivism; we may also want to add structural knowledge as proposed by Jonassen, Beissner & Yacci (1993).

Finally, as use cases identify the actor(s) involved in a scenario, the learner model can also be specified. This would include the definition of prerequisite skills as well as the entry knowledge that is needed for the acquisition of information, i.e. background knowledge, associative knowledge, and prerequisite knowledge.

3.2.1.6 Architecture and authoring tools evaluation

In many ISD projects, new technological ground is entered. Perhaps a new learning management system is employed, or a new authoring tool, a different learning technology systems architecture, particular learning metadata standards, or perhaps special existing learning object repositories are to be integrated. One strategy to minimize the risks is to analyze them as early as possible. For instance, by using a prototype in the process that focuses on these risks, they can be studied and possible other weaknesses can be eliminated. The rapid prototyping results in model solutions, authoring guidelines, and an architecture model which describes the fundamental structure of the learning environment to be developed.

This activity helps to specify the learning technology systems architecture (LTSA), which includes the management and delivery system of the learning environment. The ID author needs to establish the means for managing the learning environment, which establishes the conditions of control and responsibility of learning with the environment, ranging from program control to learner control. With regards to the specification of the delivery system, Tennyson & Foshay (2000)
write that the ID author needs “to be aware of the effect of transmediation on learning and to consider delivery systems that improve the representation of information” (p. 133).

The architecture model of the learning environment influences the instructional development plan. In addition, it must be supported by the authoring tools and therefore should be integrated in the analysis.

Finally, the ID author reviews possible curricular and instructional materials. Based on this review a decision can be made whether to develop instructional materials or employ existing materials. This activity can also be extended by the notion of learning objects, as ID authors in the course of developing an instructional intervention can search repositories to verify if learning objects exist and can be integrated, or if new learning objects have to be developed and later added to the repository.

3.2.1.7 Learning environment and instructional development (ID) planning

The results and assessments of the preliminary steps provide enough information at the end of the front-end analysis to define the variables and conditions of the learning environment and to specify the project team structure and plan of work (authoring activities/schedule) and budget which are required to build the learning environment. A key aspect in establishing the project team structure is the identification of sources of risk for the project and the creation of activities that
control or help to minimize those risks, e.g., when organizational changes occur during transition from one domain/phase to another.

The specification of the end product of the development effort updates the solution plan developed in the situational evaluation. According to Tennyson & Foshay (2000, p. 133) the information gathered so far documents the length of the instructional intervention, the proportion of the instruction presented by available media and learning objects, the description of the target audience, the definition of the learning environment constraints, the specification of the learning goals and objectives, the selection of management and delivery systems, and the specification of situational variables.

In addition, the evaluation plan for the ISD project is defined (Tennyson & Foshay, 2000, 134). This plan contains prescriptions for formative evaluation in order to gather the data required for refining and revising the learning environment during design and development. The plan also includes details to prepare and conduct summative evaluation during implementation as well as maintenance evaluation during the actual operation of the learning environment.

3.2.2 Problem domain analysis and design

As shown in Figure 22 on page 392, the problem domain analysis and design includes those authoring activities that consider the outcomes from the requirement analysis phase and turn them into more specific solution approaches. However, they cannot yet be employed for reasonable rapid prototyping. The authoring activities in this phase primarily serve to organize and structure the requirements, and to
identify and describe the components (learning objects) required for the learning environment to be developed.

**Problem domain analysis and design**

[Diagram of activity diagram of the problem domain analysis and design phase]

Figure 22. Activity diagram of the problem domain analysis and design phase

3.2.2.1 Learning activity modeling

At this stage the previously defined use cases are reassessed and further subdivided into a number of individual activities by building the corresponding activity diagrams for each use case. Based on this activity, the appropriate sequence and organization for the presentation of the information is specified, i.e. embedding information objects and learning objects in a larger hierarchy of modules, units, courses, and curricula. In addition, organizing and sequencing information of
instruction can also employ established principles and approaches from information design, as described earlier.

The organization and sequencing of the information to be learned leads to what can be referred to as an instructional architecture (see for instance Clark, 2000; see also Gagné, Briggs & Wager, 1992, p. 185; 1988, p. 177)). This architecture provides the framework for the appropriate instructional strategies, which are based on the psychological foundations chosen during the requirement analysis. Clark suggests four such architectures, i.e. receptive, behavioral, situated guided discovery, and exploratory. Based on Mok’s (1996, p. 141) diagram of the functions in an interface of an on-line information service, the diagram in Figure 23 on page 394 illustrates the learner-directedness and the sequence of instruction within the four main instructional architectures in order to depict the specific behaviors of the user interface. As Clark indicates, these architectural types are oversimplifications and may overlap in a given situation. For instance, an instructional event may employ a situated guided discovery, but may also choose to offer opportunities for exploratory learning. The main differences are how learner-structured and how sequential the learning activities are conceived. The exploratory architecture is the most learner-structured and least sequential mode. All four instructional strategies involve browsing.

These architectures correspond to Tennyson’s & Foshay’s (2000, p. 136-138) five instructional prescription categories that can be implemented in a learning environment. These categories include expository strategies, practice strategies, problem-oriented strategies, complex-dynamic strategies, and self-directed
experiences. Similar to the previous authoring activity, the specification of instructional strategies and architectures can also be informed by principles and approaches from other disciplines, such as information design and information architecture.

Finally, a system for learner evaluation and/or assessment is established (Tennyson & Foshay, 2000, p. 140). The ID author needs to determine the use of pretests, progress checks, post-tests, and retention tests. In addition, the way how the assessments are administered (e.g., written, oral, via computer) has also to be determined.

3.2.2.2 Business class modeling

Similar to the learning activity modeling, the use cases from the requirement analysis are employed for business class modeling or class analysis. At this stage, business classes are created that describe an object, a concept, a place, or a person from the real (teaching and learning) setting (Oesterreich, 1997/1999, p. 70). The result is the so-called analysis model (or the designer's model in object-oriented user
interface design terms). It captures a rough outline of the first interface design of these business objects and the associative relationships among them.

Part of this activity is the specification of the learning management system for the instruction. Here, the ID author determines the level of control the learner has in managing the learning events, and the corresponding support provided by the instructional technology. As Tennyson & Foshay (2000) write, “[a]n important variable to consider when increasing learner control of instruction is feedback to the student on cumulative progress in learning” (p. 139).

The specification of the instructional message design in technology-mediated instruction (e.g., Fleming & Levie, 1993) is also part of this step. Here, the display characteristics in relation to the information to be presented are determined. In addition, the human factors are specified in terms of means for interacting with the learning environment (e.g., menus, prompts, function keys, etc.).

3.2.2.3 Domain class modeling

The domain class modeling processes subdivides the business class model into more detailed units in order to represent the technical structures of the problem domain. This is also known as the design model (or the programmer’s model in object-oriented user interface design terms).

3.2.2.4 Instructional component building

The previous activity diagrams and business class model are now further tailored to components or subsystems of the learning environment. For instance, the learning objects that have been identified based on use cases can now be formally united with information objects, or integrated into even larger hierarchies. Likewise, the specification of the instructional message design and the human
factors in technology-mediated instruction can be continued by means of object-oriented interface design and interactivity design. At this stage, the object views and the composition of these views are designed along with the basic interaction tasks, object interactions, and task interaction sequencing.

3.2.2.5 Learning environment plan

In order to ensure both the quality of the content and its proper representation in the component model, a formative evaluation is conducted to revise the design of the component model. In addition, the implementation of the behavior-driven component in the more static structures of the domain class model is also reviewed. For instance, the specifications for the user interface can now be further tailored into the so-called model-view-controller (MVC) (e.g., Collins, 1995, p. 304-305; Oesterreich, 1997/1999, p. 114). MVC is a design pattern to describe the user’s interaction with a computer system in terms of the representation of the user’s conceptual model in the system’s information model, the representation of the interactive system in terms of views, and the implementation of controllers that provide the means for the user to interact with the model.

The findings of the problem domain analysis and design phase lead to an update of the solution plan. The resulting design document contains a complete description of the learning environment, based on which the actual iterative and incremental development of the learning environment using rapid prototyping can commence. This shows that the transition from analysis and design to rapid
prototyping in ISD corresponds to the same transition in object-oriented user interface design as well as in object-oriented software engineering.

3.2.3 Rapid prototyping

The goal of the actual development phase of the ISD process is to build a prototype of the proposed instruction within the learning environment and to produce and integrate the prescribed instructional materials into the learning environment. This phase consists of authoring activities combined with a number of evaluations at different stages during the process to ensure that the necessary revisions and refinements are performed before the product is actually implemented.

3.2.3.1 Instructional component development

The development of instructional components within the learning environment marks the beginning of the iterative and incremental rapid prototyping process. As Tennyson & Foshay (2000) write “the goal is the development of a prototype that exhibits the main features of the instruction” (p. 139). Having collected the detailed specifications in the previous phases, rapid prototyping can now proceed in a more organized and planned manner. Rapid prototyping is concluded with an initial evaluation prior to the production and integration of the instructional materials into the learning environment.

Oesterreich (1997/1999, p. 72-76) explains that rapid prototyping also requires planning (see Figure24 on page398). He distinguishes between release planning and iteration planning. The first determines which requirements have to be implemented and when (e.g., coarse implementation, detailed implementation, and complete implementation). The latter specifies the measurable and assessable
goals to be achieved at the end of an iteration as well as the planning of the steps for the subsequent iteration. This process also extends into component-specific planning during which the concrete requirements for each component are transferred.

**Planning of rapid prototyping**

![Activity diagram of planning the rapid prototyping process](image)

Figure 24. Activity diagram of planning the rapid prototyping process

Once these steps have been completed, the instructional component development can proceed (see Figure 25 on page 399). First, the necessary subject matter expert (SME) content materials have to be acquired and documented in order to prepare the narratives for the learning activities. This step can represent an extension of the earlier process of reviewing and selecting instructional materials. Then, the content narratives are structured and written. Each iteration consists of a set of internal and external reviews. This formative evaluation requires that all components are integrated before the end of an iteration, so that the necessary
corrections and modifications can be specified, which may lead to a new set of requirements. These reviews are often called alpha evaluation or beta evaluation respectively. The first represents internal or in-house tryouts in a laboratory setting (i.e. usability testing) to ensure that the instructional component functions as a whole. The latter refers to external reviews, usually conducted in the actual learning environment.
environment of the potential user (i.e., simulation), to make sure that the product can function independently from the developer.

3.2.3.2 Integration

Once the prototype has been completed into a functional learning systems architecture, including reviews and revisions, the necessary instructional materials for the learning environment are integrated into the learning environment for implementation. This authoring activity is based on the delivery mode as outlined in the design plan, e.g., print, audio, video, computer, and multimedia. This step incorporates what Kerres (1998) and Schott (1991) call multimedia didactics, which, as discussed earlier, calls for a stronger integration of media selection and production processes into the overall ISD process. Using this integrative approach, the ID author is able to incorporate instructional materials that have most probably been developed by other production teams. Finally, the management system for the learning environment is developed, also as specified in the design plan. This is performed in the context of the initial instructional prototype. Based on the results from the rapid prototyping process, the outcome of this phase is the completion of the learning environment as it was first proposed in the situational evaluation.

As indicated in Figure 26 on page 401, a formative evaluation of the complete learning environment is conducted with subject matter experts (SME) and external evaluators before the actual implementation. The first verify that the instructional materials and learning activities are technically accurate and complete (also called integration testing; see Schacht, 1997, p. 40). The latter assess the functionality of the learning environment to verify that it meets the requirements
Figure 26. Activity diagram of integration process
specified in the design plan (also called product testing; see Schacht, 1997, p. 40).
Finally, a field test (beta test) is conducted, focusing on mechanical and technical aspects of the implementation, not on the content. At this point, only refinements should be made, if necessary. A final aspect of this process is acceptance testing (Schacht, 1997, p. 40). In this case the software is delivered to the client, who evaluates it on the actual hardware using actual data, as opposed to test data.

Once all the refinements have been completed, the authoring activities of the project should be documented in its entirety. As Schacht (1997, p. 42) points out, documentation is of great importance for maintenance, which needs the product’s
specifications and design documents in order to perform the appropriate maintenance evaluation of the product.

Finally, a plan for the dissemination of the learning environment is specified. It addresses issues associated with reaching the acceptance of a (new) learning environment.

3.2.4 Implementation and maintenance

Implementation refers to the employment of a newly developed learning environment based on the previously established dissemination plan. As shown in Figure 27 on page 403, this phase begins with the reproduction of the curricular/instructional program. Then, the necessary support services are established and/or modified to ensure the proper management of the learning environment. Finally, the program is actually distributed and operated.

In order to ensure that the learning environment operates effectively in relation to the problem or need as identified in the situational evaluation, a summative evaluation is conducted. The findings documented in the summative evaluation report are used to indicate the need for and the scope of maintenance during the operation of the learning environment.

By definition, any changes to a (software) product after it has been released and is in operation constitute maintenance. Maintenance was introduced as an integral authoring domain into ISD\(^4\) by Tennyson & Foshay (2000, p. 142-143) because it has not been properly incorporated into previous generations of ISD models. In correspondence to the literature on software engineering, the authors acknowledge the importance of maintenance for the overall development process.
For instance, Schacht (1997, p. 12) points out that maintenance accounts for almost 70% of the relative costs of the phases of the software life cycle. He underscores the importance of maintenance when he writes that

> maintenance is not an activity that is grudgingly carried out after the product has gone into operation mode. On the contrary, it is an integral part of the software process that must be planned for from the beginning. (p. 41)

Schacht (1997, p. 473-476) also points out that the use of the object-oriented paradigm helps to promote maintenance. Because of the fact that objects are independent units of a program and, thus, exhibit conceptual independence, it is easier to determine which part of the program needs to be changed in order to achieve a specific maintenance goal. In addition, because of the fact that objects use information hiding to ensure that implementation details are not visible outside of

Figure 27. Activity diagram of implementation and maintenance phase
these objects, changes that are made to an object are less likely to have an impact outside of that object.

In the context of ISD, maintenance helps to ensure that the effectiveness and efficiency of instruction is maintained at the level when the learning environment was first implemented. In order to achieve this, a maintenance plan needs to be developed and implemented according to which the learning environment is continuously evaluated.

Tennyson & Foshay (2000, p. 142-143) list five key areas that require special attention during maintenance evaluation of a learning environment. First, the question concerning whether the instructional materials are still worth using in the learning environment has to be addressed in terms of a cost-benefit analysis of the product. Second, the update of the learning environment is an important consideration in order to keep the instructional products and materials current. Third, the learner attitudes toward the instruction and the materials should be assessed together with performance measures, because both may be fluctuating. Fourth, changes in the characteristics of the learner, learning goals, learner prerequisites, societal policies, etc. need to be evaluated in order to make the appropriate adjustments. For instance, if the learning environment is planned to be used in an international setting, internationalization and localization have to be addressed accordingly, as discussed earlier in chapter V. Finally, special media types used in the learning environment have to be evaluated and maintained as well. For
instance, new media sources can be integrated to improve the efficiency of the learning environment.

Based on the findings gained in the maintenance evaluation, the learning environment can be updated and new advancements in instructional technology integrated. As mentioned earlier in the context of software engineering, maintenance can be performed for three main reasons. First, if any faults or errors have to be repaired, corrective maintenance is needed. In order to improve the effectiveness of the product, perfective maintenance is conducted. And third, adaptive maintenance is applied when changes are made to the product in order to react to changes in the environment.

Finally, it has also to be ensured that the required changes made to a product during maintenance have been properly implemented and that the functionality of the rest of the product has not been compromised because of these changes. Therefore, Schacht (1997, p. 42, 477-478) suggests to use regression testing during which the product is tested against previous test cases.

3.3 Summary

The above discussion of the instructional design model in object-oriented terms has shown that ISD can benefit from the incorporation of object-oriented analysis and design methods. It allows for the integration of different disciplines in the development process, such as user interface design and software engineering, using shared approaches (i.e. the object-oriented methods) and a common modeling notation language (i.e. UML). For instance, use cases can help to capture the different activities and scenarios the learning environment needs to address. The
The object-oriented instructional design model, in UML terms, can be represented as an abstraction of a system ("system model") that consists of models for analysis, design, production, implementation, and maintenance. The different models of the same system show different aspects of the system, from different viewpoints and/or levels of abstraction. Combined, these different models represent a package. For instance, the package for the authoring activities in the (instructional) design phase could contain the following elements: objective editor, instructional strategy selector, media selection tool, etc. These packages can be extended in their functionality to support behaviors within the system, thus representing a subsystem. Such a subcomponent or subsystem of (instructional)
design is illustrated in Figure 28 on page 406 (based on an example for software engineering in Oesterreich, 1997/1999, p. 134), which depicts the steps in selecting learning objects in an activity diagram and translates those into components which constitute the subsystem. The component diagram itself can be further extended by indicating the different component interfaces that are needed as parts of that subsystem.

In fact, UML includes a profile for software development processes. It describes how UML can be customized for specific domains, in this case software engineering. The profile contains common stereotypes and some useful terminology (e.g., Analysis Model, Design Model, Analysis System, Design System, Implementation Subsystem, etc.). They are characterized by the life cycle stages they represent. Similar to the UML profile for business modeling, which was discussed earlier, the overall ISD model could then be expressed in more detail using an adaptation of the UML profile for software development processes.

4. An example of the object-oriented instructional design process

As indicated by Tennyson & Foshay (2000, p. 125), ISD is still an emerging concept. Similar, applying the object-oriented paradigm in ISD is also a very recent phenomenon. Therefore, completely established examples are still not very common in practice. In addition, as both ISD and the object-oriented paradigm are dynamic processes that adapt to specific problem situations, no single project is completely exemplary. However, the aspects of the object-oriented instructional design (O OID) model illustrated in this section present an example of applying the object-oriented
paradigm in ISD. As the aspects of the OOID model are introduced, their relationship with and possibilities to extend the ISD\textsuperscript{4} model will be indicated.

As an instructional designer with the information technology department of a public college in Massachusetts since 1999, the author has been in charge of supporting faculty member in the use of computers in education. The position of an instructional designer was created to address the increasing need for technical assistance for faculty in successfully implementing the Internet in education. As many faculty members gradually move to integrate the Web as an instructional medium in their courses, many are struggling to find appropriate ways to apply their years of experience in lesson planning for a traditional learning-teaching situation with lectures and only occasional applications of computers into the new environment of technology-mediated instruction using a variety of hypermedia in an interactive networked learning environment. At the same time, academic computing is confronted with the task to respond to a growing number of requests to develop various kinds of applications for e-learning. Thus, it seemed helpful to introduce an instructional design process that would be flexible enough to be applied to different situations as well as to allow the creation of an evolving repository of learning objects that could be applied in different contexts or could serve at least as showcases for other faculty members. At the same time, the workflow between the design, production, implementation, and maintenance would benefit from specifications written in a standard modeling notation (i.e. UML), which could be passed from one team member to another who would generally work in parallel. In addition, these specifications could also be used to request
assistance from specialists outside the academic computing division, where the author was assigned to work, for very specific technical tasks.

For instance, over the course of an academic year the author worked together with a faculty member, whose task was to convert a traditional classroom-based course into an on-line course delivered through the college's learning management system (LMS). The goal was to adapt the existing instructional material to a Web-based learning environment. The faculty member had to realize that a straight translation of the traditional teaching-learning situation into the virtual classroom setting was not possible. This situation presented an opportunity to reassess the existing course and help to redefine its goals and objectives. Because the faculty member taught a graduate course, which was attended by many students who were working professionally while attending college, it seemed appropriate to move major sections of the course to the Web and to reduce the time students had to be physically present in the classroom, which would better meet the needs of their working schedule. Although not directly expressed as such to the faculty member, use case analysis was applied to focus on scenarios within the topic domain that relate to the actual work situation of the students. The author assisted the faculty in rephrasing the learning objectives according to this analysis. Based on the newly defined objectives, core modules (i.e. learning objects) for the course were identified. This was done in the attempt to clarify the relationship among the learning objects and the particular instructional architecture and learning activities that were to be implemented. The author assisted the faculty member in creating simple prototypes to assess the structure of the new learning environment.
Unfortunately, this extended learning process was not fully concluded as the faculty member decided to just directly apply the existing instructional materials in the new on-line environment without the necessary adjustments. In practice, the material from the textbook that had traditionally been used in this course remained the sole source even for the on-line course, although it only contained minor suggestions for teaching the content in an on-line environment. In order to fully implement the proposed plan, it would have required more time and resources, which the faculty was not able to commit.

Recently, a number of faculty members at the college has received a technology grant from the State of Massachusetts to enhance the curriculum for teacher education in different disciplines. The goal of this on-going project was to revise or add new courses which would address information technology skills and concepts within disciplines which are outside the traditional computer science or engineering disciplines. The primary authors of the project were the faculty members, who generally had many years of teaching experience, but only limited instructional design experience at the basic level of step-by-step lesson planning, and the instructional designer and other courseware development specialists, who had experience in instructional development as well as in graphics and programming assistance in production. The grant recipients were required to make intensive use of the college’s LMS. In addition, the academic computing office of the author was approached to provide the necessary technical support to the faculty members in the development of various interactive modules for these courses. One of these courses was about teaching mathematics to elementary school teachers.
The goal of this course was not only to enhance the teaching of mathematics with interactive visualizations, but also to introduce the students to new media, which in turn could showcase how to implement them in their own teaching.

The project began with a situational evaluation. Because of the complexity of the content, the lesson planning had to be done by subject matter experts (i.e. faculty members), with the support from the instructional designer and other members of the information technology department. While classroom teaching was still to be continued, current and future students of elementary school education would benefit from the enhanced instruction using interactive multimedia. Since this instruction would use mathematical visualizations, WBI was a viable option for a long-time solution to deliver these interactive tools. One of the requirements specified by the faculty members was to develop drill and practice exercises that could be used as a supplement to the course. The faculty members outlined a small set of behavioral objectives, which placed the individual skills to be learned into a hierarchy and indicated the final learning outcome. Using a learning hierarchy (prerequisites) analysis, the initial outline was extended in order to verify the sequence of the skills to be taught. Based on these findings, clearer assumptions of the role of the interactive modules to be delivered on-line could be made. The learning hierarchy was further elaborated by employing use cases as the basis for the specifications of learning objects and for the development of the interactive modules. These modules were not to be integrated with the assessment tools of the
LSM. However, they would need to be deliverable on the Web and provide means to update and expand the textual matter.

Instead of using a sequential procedure, it was suggested to structure the project in such a way that analysis and development teams could work in parallel, especially as the faculty members requested to develop interactive modules in different media types in addition to those that had already been located on the Web and were shared with the students as bookmark references. The iterative structure of the development process was suggested to better control the size and technology risks of the interactive module. The workflow among the different project team members was planned to be more efficient and productive by the use of standard UML diagrams and object-oriented methods. However, because of the different levels of expertise in instructional design and the object-oriented methodology, additional training measures for the team members would have been required. Initially, it was suggested to produce prototypes that could be developed and tested in several iterations before the actual implementation. This seemed reasonable as new technologies were to be tried, about which the team members had only just begun to develop a more in-depth knowledge. At the same time, the students would be given the opportunity to participate more directly as stakeholders in the development process as well as to assess the acceptance and suitability of the different multimedia treatments. Their input to the formative evaluation of different prototyping iterations would constitute an additional source for further refinements. Summative evaluation was only accounted for in the context of the overall course, in which the interactive modules only played a supporting role. In
general, the summative evaluation would only be applied to the usual student assessments using the basic testing tools provided by the college’s LMS.

In suggesting rapid prototyping the opportunity for formative evaluations was planned to be incorporated into the project from the beginning. The final release of the product would be deferred until the instructional treatment had been fully validated through field tests of the prototype. The proposed rapid prototyping approach seemed also very suitable to elaborate more about the use of multiple representations to support the learning of mathematical concepts and rules. Although, different team members were asked to develop separate multimedia solutions, these applications were primarily based on a single simulation that would present the material and provide drill and practice exercises.

As indicated earlier, it is important to establish the theoretical foundations to be used in the instructional treatment. The learning theory for this project suggested by the faculty members was behavioral. The major task was to determine those tasks that would have to be mastered before the final task could be accomplished using the interactive module. The analysis led to a decision about which subordinate skills and information would have to be delivered in class before the students could actually use the supplemental multimedia application. Each analysis included a hierarchy of behavioral objectives to show their interdependence and the order in which they should be learned. The relevant subset of intellectual skills (procedural knowledge) was indicated, which the students needed to accomplish by rehearsing defined mathematical concepts and rules and developing problem-solving strategies based on higher-order or complex rules. Defined concepts have
specific object properties or attributes. The students would demonstrate the meaning of some particular class of objects, events, or relations and identify instances of concepts that are components of the definition as well as show an instance of their relation to one another. In addition, these concepts are governed by rules which determine the relationships among classes of objects and events. In applying the rules to one or more concrete instances students would demonstrate their knowledge. The higher-order rules would require the student to invent and use a complex rule to achieve the solution of a problem new to the learner.

Thus, the findings of the learning hierarchy analysis helped to identify nouns, verbs, and attributes to establish a class model along with its properties, behaviors, and relationships in the domain. Furthermore, these findings could also be expanded with uses cases and activity diagrams to determine what kinds of performances the students need to accomplish to demonstrate their successful understanding of the topic. Thus, the results of this activity would not only produce the objectives, structure, and strategy of the instructional intervention, but also specifications based on UML diagrams that could form the basis for the actual development of the multimedia application. Each diagram could be passed for verification to the individual project team members. For instance, the faculty members indicated a particular sequence according to which the content and exercises would be presented with which the students would interact. This was depicted in activity diagrams. Because the interaction was based on particular conditions (e.g., true or false actions and the appropriate feedback messages) sequence diagrams were created that traced the events that would take place. The
instructional designer would write specific templates and tools, which would be used by the faculty members to build the instructional treatment and check the validity of the content. Further, the diagrams would be used to determine the system structure and function to be develop for Web delivery and how it could be integrated into the existing LMS. This resulted in the design of the information presentation and the interaction and control mechanisms that would constitute the graphical user interface. By applying the object-oriented paradigm throughout the entire process, a learning object model could be identified along with its implementation in the corresponding Object-Action-Interface (OAI) model of the Web site. Thus, this planned approach would result in the development of basic prototypes that could be tried out and revised by the project team members. This process was to be continued until a Web-based instructional system was produced that could be operated successfully with the college’s LMS.

The emerging tools and techniques for fourth-generation ISD (e.g., automated instructional design) were not available to the project team members at the college. For instance, the existing LMS at the college only provides basic tools for learner management and course delivery. The college’s product does not constitute what now is called a learning content management system (LCMS), which would provide the necessary authoring, sequencing, and aggregation tools in a single environment to structure the content for the facilitation of the learning process. So the project team had to rely on prototyping using a multimedia authoring tool (e.g., Macromedia Flash), which had to be supplemented by word processing and graphics tools in creating the necessary templates and procedures.
and which was built by the team’s instructional designer and other technical support personnel. The choice was made for Macromedia Flash as the primary multimedia authoring tool to produce Flash movies because of their wide support in common Web browsers with the appropriate plug-in and the familiarity of users with this kind of media type. Flash movies can be used to deliver highly interactive environments with appealing graphical user interfaces. Because of their high compression rate, Flash movies usually have a very low download rate, which makes them very suitable for fast delivery even over slower modem connections. In addition, Flash movies can be integrated with a database-driven backend system using the ColdFusion server, which is supported on the college, to dynamically generate content in the Flash movies. However, this would require a more thorough investigation of how to implement an object-oriented database management system (e.g., Bertino & Martino, 1991; Blaha, Premerlani & Rumbaugh, 1988; Hurson, Pakzad & Cheng, 1993; Keller, 1997). In addition, the ActionScript scripting language of Macromedia Flash supports object-oriented programming. Therefore, the object-oriented paradigm could also be followed through in the actual production phase of the project. Finally, Flash and ColdFusion now also support the use of XML. This new Web technology would help to further establish the content structure and better incorporate standardized sets of educational metadata based on task ontologies (e.g., Quin & Paling, 2001). On the one hand, these materials could be more easily reused in different settings in terms of learning objects. On the other hand, the materials could then be developed for different delivery systems using different types of media.