Lessons Learned from Designing a Comprehensive Case-Based Reasoning (CBR) Tool for Support of Complex Thinking

Doug Richmond

Dissertation submitted to the faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
In
Curriculum and Instruction

Mary Alice Barksdale, Co-Chair
Ken Potter, Co-Chair
Lawrence Cross
Barbara Lockee
Jane Ann Falls

May 4, 2007
Blacksburg, Virginia

Keywords:
Developmental research, complex thinking skills, case-based reasoning (CBR), performance support tools, relational database, World Wide Web

Copyright 2007, Doug Richmond
Lessons Learned from Designing a Comprehensive Case-Based Reasoning (CBR) Tool for Support of Complex Thinking

Doug Richmond

ABSTRACT

This research study focused on learning lessons from the experience of designing a comprehensive case-based reasoning (CBR) tool for support of complex thinking skills. Theorists have historically identified, analyzed, and classified different thinking processes and skills. Thinking skills have been increasingly emphasized in national standards, state testing, curricula, teaching and learning resources, and research agendas. Complex thinking is the core of higher-order thinking. Complex thinking is engaged when different types of thinking and action converge to resolve a real-world, ill-structured issue such as solving a problem, designing an artifact, or making a decision. By integrating reasoning, memory, and learning in a model of cognition for learning from concrete problem-solving experience, CBR can be used to engage complex thinking. In similar and different ways, CBR theory and the related theories of constructivism and constructionism promote learning from concrete, ill-structured problem-solving experience. Seven factors or characteristics, and by extension, design requirements, that should be incorporated in a comprehensive CBR tool were extracted from theory. These requirements were consistent with five theory-, research-based facilitators of learning from concrete experience. Subsequent application of the Dick, Carey, and Carey model to these design requirements generated twenty-nine specifications for design of the tool. This research study was carried out using developmental research methodology and a standard development model. The design process included front-end analysis, creating a prototype of the tool, and evaluating the prototype.
# Table of Contents

LIST OF TABLES ........................................................................................................ V

LIST OF FIGURES ..................................................................................................... VIII

CHAPTER 1 – INTRODUCTION .................................................................................. 1

Historical Background of the Research Study ......................................................... 1

Impetus for the Research Study ............................................................................. 3

The Research Problem ......................................................................................... 4

Research Study Objectives and Methods ............................................................ 6

Importance of the Research Study ....................................................................... 7

CHAPTER 2 – LITERATURE REVIEW ..................................................................... 8

The Case-Based Reasoning (CBR) Process as a Means for Learning Thinking Skills ................................................. 8

Complex Thinking ............................................................................................. 9

Constructivism .................................................................................................. 11

Constructionism ............................................................................................... 18

The CBR Model and Process ............................................................................ 20

Characteristics of a Comprehensive CBR Tool .................................................. 30

CHAPTER 3 – THE DESIGN PROJECT .................................................................... 34

The Research Methodology .............................................................................. 34

A Generic Design Process and a Design Model .................................................. 34

Stage 1 – Analyze Needs .................................................................................. 36

Stage 2 – Identify, Analyze, and Classify Goals ............................................... 39

Stage 3 – Analyze Learners and Contexts ......................................................... 58

Who are the learners using the tool? ................................................................ 59

What contexts are involved in using the tool? ................................................. 60

Interaction of learners with performance/learning contexts .......................... 62

Stages 4-6 – Write Performance Objectives, Develop Assessment Instruments, and Develop an Instructional Strategy ......................................................................................................................... 64

Stage 7 – Create a Prototype ............................................................................ 69

Stage 8 – Design and Conduct a Formative Evaluation .................................... 95

The formative evaluation .................................................................................. 96

Formative evaluation data .............................................................................. 99
CHAPTER 4 – LESSONS LEARNED AND FUTURE DIRECTIONS 117

Lessons Learned 117
Lessons Relating to Design 117
Lessons Relating to Development 119
Lessons Relating to Evaluation 124

Future Directions 126
Implications for Production 126
Implications for Additional Research 126
Implications for Practice 127

Conclusion 127

REFERENCES 129

APPENDIX A 147
Module Menus 147

APPENDIX B 149
Formative Evaluation Instrument: Visibility of Strategies 149

APPENDIX C 156
Formative Evaluation Instrument: Usability Principles 156
List of Tables

Table 1 - Examples of Supports for Experience-Based Learning ____________________________________ 26
Table 2 - Examples of Case Libraries Serving as Learning Resources ______________________________ 28
Table 3 - Five Facilitators of Learning Effectively from Experience ______________________________ 31
Table 4 - Seven Theory-, Research-based Design Requirements _________________________________ 33
Table 5 - Stages of the Dick, Carey, and Carey Model ________________________________________ 35
Table 6 - Design Requirements and Specifications ___________________________________________ 37
Table 7 - Facilitators and Specifications ___________________________________________________ 38
Table 8 - Classification of Goal Steps/Sub-steps According to Four Learning Taxonomies ________ 57
Table 9 - Organization of a Standard Macro Instructional Strategy _____________________________ 66
Table 10- Stage 4-6 Design Factors ______________________________________________________ 68
Table 11- Stage 4-6 Specifications _______________________________________________________ 69
Table 12 - Operationalization of Specifications in the Prototype ______________________________ 71
Table 13 - Location of Operationalized Specifications in Prototype ____________________________ 93
Table 14 - Experts Serving as Formative Evaluators _________________________________________ 97
Table 15 - Format for the Two Formative Evaluation Instruments ______________________________ 99
Table 16 - Visibility of Macro Strategy: Problem-Solving – Likert Ratings _____________________ 100
Table 17 - Visibility of Macro Strategy: Problem Solving – Comments _________________________ 100
Table 18 - Visibility of Micro-Strategy: Cases – Likert Ratings ________________________________ 101
Table 19 - Visibility of Micro-Strategy: Cases – Comments __________________________________ 101
Table 20 - Visibility of Micro-Strategy: Information Resources – Likert Ratings ________________ 102
Table 21 - Visibility of Micro-Strategy: Information Resources – Comments ____________________ 102
Table 22 - Visibility of Micro-Strategy: Cognitive Tools – Likert Ratings ______________________ 103
Table 23 - Visibility of Micro-Strategy: Cognitive Tools – Comments __________________________ 103
Table 24 - Visibility of Micro-Strategy: Conversation and Collaborative Tools – Likert Ratings _____ 104
Table 25 - Visibility of Micro-Strategy: Conversation and Collaborative Tools – Comments _________ 104
Table 26 - Visibility of Micro-Strategy: Social and Contextual Support – Likert Ratings _________ 105
Table 54 - Usability Principle 10: Help and Resources 116
Table 55 - Visibility of Usability Principle 10 – Comments 116
Table 56 - Overall Visibility of Usability Principles 116
Table 57 - Overall Visibility of Usability Principles – Comments 116
List of Figures

Figure 1 - Complex Thinking _____________________________________________________________ 10
Figure 2 - Levels of Knowledge: Five Interpretations _________________________________ 14
Figure 3 - Relationship: Levels of Education/Experience and Levels of Knowledge/Proficiency ______ 14
Figure 4 - The Constructivist Continuum _____________________________________________ 18
Figure 5 - CBR: The Cycle ___________________________________________________________ 23
Figure 6 - CBR: The Hierarchy of Tasks and Methods _________________________________ 23
Figure 7 - A Flowchart Analysis: The General Problem-solving Process ____________________ 42
Figure 8 - A Flowchart Analysis: The Case-Based Reasoning (CBR) Process ______________ 42
Figure 9 - A Flowchart Analysis: Representative Subordinate Skills for Problem-solving Steps ______ 54
Figure 10 - A Flowchart Analysis: Representative Subordinate Skills for a Problem-solving Decision __ 55
Figure 11 - Learning Contexts Ranked According to Degree of Authenticity _____________ 62
Figure 12 - Relationship Between Stage 1-3 Outputs (Front-end Analysis) and Stage 4-6 Outputs _____ 64
Figure 13 - An Option for Strategy: A Macro-strategy and Eight Supportive Micro-strategies ______ 67
Figure 14 - Background Component: Screen Capture and Purpose ________________________ 87
Figure 15 - Operations Component: Screen Capture and Purpose _________________________ 88
Figure 16 - Help Component: Screen Capture and Purpose ______________________________ 89
Figure 17 - Evaluation Component: Screen Capture and Purpose _________________________ 90
Figure 18 - Exploration Component: Screen Capture and Purpose _________________________ 91
Figure 19 - Administration Component: Screen Capture and Purpose ______________________ 92
Figure 20 - Associating Evaluation Elements and Design Specifications Using a Relational Database __ 98
Figure 21 - The Instantiation Process Implemented in This Research Study ________________ 119
CHAPTER 1 – Introduction

Theorists have historically identified, analyzed, and classified thinking processes (Marzano, Brandt, Hughes, Jones, Presseisen, Rankin, & Suhor, 1988). Over the years, thinking skills have received increasing emphasis in national standards, state testing, curricula, learning resources, and research agendas (Kallick, 2001; Marzano et al., 1988; Resnick & Klopfer, 1989). Complex thinking is the core of higher-order thinking skills (Iowa Department of Education, 1989; Iowa State University, 1999; Jakovljevic, Ankiewicz, de & Swardt, 2005; Jonassen, 1996a, 2000; Jonassen & Carr, 2000; Jonassen, Carr, & Yueh, 1998; Slangen & Sloep, 2005). Complex thinking is engaged when different types of thinking and action converge to resolve real-world issues. Powerful, advanced knowledge formations result from applying complex thinking skills to resolution of these issues. Case-based reasoning (CBR) theory (and the related theories of constructivism and constructionism) promote and inform learning from resolving real-world issues (Aamodt & Plaza, 1994; Kolodner & Guzdial, 2000; Kolodner, Owensby, & Guzdial, 2004). This research study seeks to answer two questions instrumental to further exploring CBR theory’s potential for support of complex thinking, and includes lessons learned from the experience of conducting the study.

Historical Background of the Research Study

I believe that I have almost always been interested in – intensely interested in – thinking and the nature of knowledge. Whether or not I have always known the technical terms, I have been interested in complex thinking and the advanced learning and knowledge that result from complex thinking.

At some point, I learned that the philosophy and science of knowledge is called epistemology. Epistemology deals with the theory of knowledge – its nature, origins, and limits (Ormrod, 1999; Papert, 1999). The philosophy of epistemology can become intellectually and emotionally deep, and wonderfully fascinating. On the other hand, the science of epistemology is grounded in the hard realities of the everyday world, and is concerned with the pragmatics of what actually works and what does not work in teaching and learning (Papert, 1999). I am ultimately interested in exploring both the philosophical and scientific dimensions of epistemology, but professionally and in this paper I focus on the scientific.

The learner’s conception of learning and knowledge – learning outcomes – can be a deciding factor in determining that learner’s choice of learning strategies and success in learning (Ormrod, 1999). If the learner conceives of learning outcomes as acquisition of isolated facts, that learner may choose memorization as a strategy.
On the other hand, if the learner conceives of learning outcomes as in-depth understanding and ability to apply and extend knowledge, that learner may instead seek hands-on, real-world experience as a strategy.

Instructional design theory recognizes this relationship between outcomes and methods. It mandates a critical matching of learning outcomes with methods appropriate for effectively achieving them (Dick, Carey, & Carey, 2005; Ertmer & Newby, 1993; Spiro, Coulson, Feltovich, & Anderson, 1988; Spiro, Feltovich, Jacobson, & Coulson, 1992a; Spiro, Feltovich, Jacobson, & Coulson, 1992b; Spiro, Vispoel, Schmitz, Samarapungavan, & Boerger, 1987). Methods for acquiring introductory knowledge and outcomes differ significantly from methods for acquiring advanced knowledge and outcomes. Introductory knowledge in any domain tends to be more disconnected and inert (Ormrod, 1999; Spiro et al. 1988). On the other hand, advanced knowledge tends to be more interconnected (web-like) and spontaneously recalled in settings beyond the classroom, settings such as the workplace (Ormrod, 1999; Spiro et al., 1988). It is said to “transfer” from the learning context to performance contexts (Perkins & Salomon, 1992; Salomon & Perkins, 1989). Introductory knowledge about computer programming can be effectively acquired from hearing a lecture and/or reading a textbook about this topic. Advanced knowledge about programming must be acquired by dealing with the real, ill-structured, challenging issues of computer programming in a problem- or project-based learning environment. This is the type environment that guides and supports engagement and application of complex thinking skills to resolving real, ill-structured, challenging issues. The outcome of interest in this research study is an advanced outcome – complex thinking. CBR theory will be more fully explored here as one means for supporting performance of complex thinking skills by designing a comprehensive CBR-inspired tool intended to support those skills.

In addition to this interest in epistemology – complex thinking and advanced learning and knowledge – I have also held a lifelong interest in technology – from childhood to this day. I formerly administered a library media/information technology program at a local PK-12 school. My role was primarily administrative but, with the Board of Education’s approval, I elected to develop and implement an advanced course on emergent Internet technologies for primarily ninth-grade students. In this course, I expected and rewarded complex thinking and advanced learning. My longtime interests in complex thinking, advanced learning, and technology thus converged in this course.
I found that I faced the rewarding challenge of “de-conditioning” my students. Course admission requirements were stringent and the students overall were true academic stars. However, they were seemingly conditioned to very conscientiously seeking the “correct” answers. The new ambiguity of ill-structured issues and need for complex thinking in this technology course perhaps made them uneasy at first. A supportive, well-meaning, and sincere parent perhaps also viewed the course to be about finding and knowing the correct answers. He contacted the school principal to say: “he does not give my daughter the answers to her questions ... he asks her more questions.” Perhaps the father’s intent was to bring about a change in outcomes and the teaching methodology. However, with the full support of the school principal, I respectfully and diplomatically presented to him the case for the outcomes and methodology. In the end, the highly gifted, academically successful students of the class advanced from simple answer-seeking to grappling with ill-structured technological issues, applying complex thinking to resolving the issues, and achieving an advanced understanding of the emergent Internet technologies.

My students and I capped off the course in May of that spring semester with a field trip to Virginia Tech, where I would soon begin doctoral studies in instructional technology (IT). My future studies in IT would allow me to further explore the educational potential of complex thinking, advanced learning, and emergent Internet technologies.

*Impetus for the Research Study*

When I began my doctoral studies in IT at Virginia Tech, I was excited to find much in the IT literature on complex thinking, advanced learning, and Web-based technologies to stimulate and to support complex thinking and advanced learning. I took the “Trends and Issues in Instructional Technology” class during my first semester at Virginia Tech. I was assigned to read journal articles on topics of tremendous professional interest to me: applying instructional technologies to support problem-driven, experience-based learning; realistic learning environments; and many others.

In my IT classes and my informal interactions with fellow doctoral students in IT, I soon began hearing about a “famous” book which is sometimes said to be the instructional technologist’s bible. It was the *Handbook of Research for Educational Communications and Technologies*. During my perusal of this book in the “Research in Instructional Technology” class, I found a chapter in which D. H. Jonassen and T. C. Reeves (1996), prominent scholars in my new field of IT, proposed application of problem-based technologies to engaging complex thinking
and producing advanced learning. In this application of technology, the technology functions as a cognitive tool for stimulation of thinking and learning (Jonassen, Peck, & Wilson, 1999; Jonassen & Reeves, 1996; Kommers, Jonassen, & Mayes, 1992; Lajoie & Derry, 1993; Reeves, 1999).

The convergence of my longtime interests in complex thinking, advanced learning, and technology with the opportunity of my doctoral studies was the impetus for this research study. This study sought to answer two questions instrumental to more fully exploring CBR theory as one means for support of complex thinking, and also included lessons learned from the experience of conducting the study.

*The Research Problem*

This research study seeks to answer two questions instrumental to more fully exploring CBR theory as one means for supporting complex thinking:

(1) Is it possible to identify a relevant set of theory- and/or research-based, operationalizable factors that should be incorporated in a comprehensive CBR tool for engaging complex thinking? (Chapter 2)

(2) Given these factors, is it possible to create a comprehensive CBR tool that is both consistent with the factors and has the potential to support complex thinking? (Chapter 3)

This research study also includes lessons that were learned from the process of answering these questions.

CBR’s support of complex thinking (and ultimately advanced learning) lies in CBR’s role as a model of cognition for learning from concrete problem-solving. Complex thinking is the core of higher-order thinking (Iowa Department of Education, 1989; Iowa State University, 1999; Jakovljevic, Ankiewicz, & de Swardt, 2005; Jonassen, 1996a, 2000; Jonassen & Carr, 2000; Jonassen, Carr, & Yueh, 1998; Slangen & Sloep, 2005). Over the years, theorists have identified, described, and classified thinking processes (Iowa Department of Education, 1989; Iowa State University, 1999; Jakovljevic, Ankiewicz, & de Swardt, 2005; Jonassen, 1996a, 2000; Jonassen & Carr, 2000; Jonassen, Carr, & Yueh, 1998; Marzano et al., 1988; Slangen & Sloep, 2005). Thinking skills have been increasingly emphasized in national standards, state testing, curricula, teaching and learning resources, and research
agendas (Iowa Department of Education, 1989; Iowa State University, 1999; Jakovljevic, Ankiewicz, & de Swardt, 2005; Jonassen, 1996a, 2000; Jonassen & Carr, 2000; Jonassen, Carr, & Yueh, 1998; Kallick, 2001; Marzano et al., 1988; Resnick & Klopfer, 1989; Slangen & Sloep, 2005). As the core of higher-order thinking, complex thinking is engaged when action and three types of thinking converge to resolve a real, ill-structured, challenging issue, such as solving a problem, designing an artifact, or making a decision (Iowa Department of Education, 1989; Iowa State University, 1999; Jakovljevic, Ankiewicz, & de Swardt, 2005; Jonassen, 1996a, 2000; Jonassen & Carr, 2000; Jonassen, Carr, & Yueh, 1998; Slangen & Sloep, 2005). The three types of thinking are content/basic thinking (accepted knowledge, including metacognition), critical thinking (a convergent thinking process of reorganizing knowledge), and creative thinking (a divergent thinking process of generating new knowledge). The outcome of engaging and applying complex thinking to resolving challenging issues is advanced learning and knowledge.

In similar and different ways, CBR – and the related theories of constructivism and constructionism – promote learning from the experience of solving concrete, challenging problems. Historically, significant research has been conducted on CBR applications in the computer science field of Artificial Intelligence (American Association of Artificial Intelligence, 2004; Kolodner, 1993; Schank, 1982, 1990; Turing, 2004). Results of this research include not only many useful computer-based reasoners or problem-solving applications but also further clarification of CBR theory (Kolodner, 1993). Relatively less and later research has been conducted on educational applications of CBR (Kolodner & Guzdial, 2000; Wang, Moore, Wedman, & Shyu, 2003). Wang and others (2003, p. 46) noted a lack of research on educational applications of CBR and called for more research in this area:

... CBR research and development in the education domain is conspicuously lacking. It becomes apparent that there is a need to explore CBR applications in education. While there is a plethora of projects and practical pursuits for CBR applications in business, literature relating to designing and developing practical CBR educational applications is particularly needed.

This research study’s exploration of CBR as one means for engaging complex thinking will add to needed research on educational applications of CBR.
Research Study Objectives and Methods

The purpose of this research study is to more fully explore CBR theory as one means for supporting complex thinking, the core of higher-order thinking that is engaged during solving problems, designing artifacts, and/or making decisions (Iowa Department of Education, 1989; Iowa State University, 1999; Jakovljevic, Ankiewicz, & de Swardt, 2005; Jonassen, 1996a, 2000; Jonassen & Carr, 2000; Jonassen, Carr, & Yueh, 1998; Resnick & Klopfer, 1989; Slangen & Sloep, 2005). Developmental research methodology was used for conducting this research study.

Development-focused domains like engineering and instructional technology invite developmental research where lessons can be learned from the development of a domain-significant artifact (Richey & Nelson, 1996; Richey, Klein, & Nelson, 2004). As a type of applied research (Pedhazur & Pedhazur-Schmelkin, 1991), developmental research produces findings that can be immediately applied to resolving domain issues. According to the IT literature, a developmental research study consists of two major components: (a) a development component and (b) a traditional research component (Richey & Nelson, 1996; Richey, Klein, & Nelson, 2004). In the development component, a standard instructional systems development (ISD) model is used to guide the systematic development of an educational program or product. Walter Dick’s and James and Lou Carey’s (2005) model published in Systematic Design of Instruction was used here. In the traditional research component of a developmental research study, traditional research methodologies are applied to studying the development of the program or product and learning lessons from the development experience (Richey & Nelson, 1996; Richey, Klein, & Nelson, 2004). These traditional research methodologies may include qualitative, quantitative, evaluation, survey, and/or case study methods.

In this research study, a comprehensive CBR tool was designed to more fully explore CBR as one means for supporting complex thinking. The tool is based on seven theory- and/or research-based factors or characteristics – and by extension, design requirements – that should be included in a comprehensive CBR tool. These requirements are consistent with five theory-, research-based facilitators of effectively learning from experience. Application of the Dick, Carey, and Carey model to the requirements generated twenty-nine design specifications. This study includes lessons learned from the experience of conducting the study. Lessons were classified according to the three major stages of developmental research: Lessons Relating to Design, Lessons Relating to Development,

**Importance of the Research Study**

This research study adds to the relatively new research agenda for educational applications of CBR theory. Specifically, the potential of CBR for engaging complex thinking is explored. Complex thinking is essential to effectively resolving real-world issues such as solving a problem, designing an artifact, or making a decision. Jonassen and Tessmer (1996, p. 32) believe the capability of effectively addressing real-world (situated) issues is “one of the most important learning outcomes that can ever be supported by instruction” because workers receive compensation for this type of activity in the real world. Instructional designers have ranked real-world problem-solving as a highly effective learning strategy (Merrill, 2002). A task analysis of the CBR process was performed to reveal seven factors or characteristics – design requirements – that should be incorporated in a comprehensive CBR tool. These characteristics are consistent with five theory-, research-based facilitators of effectively learning from experience. These requirements formed the basis for generating twenty-nine design specifications.

This developmental research study suggests significant implications for further exploring the creation of Web-based, database-enabled learning supports. Tools built on the foundation of a Web-based, database-enabled platform consist of two basic parts: (a) an interface consisting of familiar, ubiquitous Web pages and (b) a robust content base held in a relational database. A relational database is an advanced computer-based system for collecting, organizing, holding, revising, and retrieving significant amounts of data and content. A Web-based, database-enabled platform innovatively integrates Web and database technologies to form a flexible, high-capacity, user-friendly, worldwide workspace for practice CBR and engagement and application of complex thinking to real, ill-structured, challenging issues and achieving advanced levels of learning.
CHAPTER 2 – Literature Review

*The Case-Based Reasoning (CBR) Process as a Means for Learning Thinking Skills*

Thinking skills have found increasing emphasis in state and national standards and state tests (Costa, 2001; Galbreath, 1999; Hallett, 1987; Jones & Idol, 1990; Kallick, 2001; Tapscott, 1998; Thornburg, 1998; Tyson, 1997). Verbs such as *explore, contrast, elaborate,* and *represent* are pervasive in standards and tests, and various educational programs have been developed to promote acquisition and refinement of thinking skills. Marzano and others (1988) proposed making thinking the foundation of schooling by infusing the teaching of thinking throughout the regular curriculum. They created the “Dimensions of Thinking” framework to ensure consistent and orderly development of thinking skills curricula, instruction, and in-service training programs for educators. Their framework addresses the following dimensions: (a) metacognition, (b) critical and creative thinking, (c) thinking processes (for example, problem solving), (d) core thinking skills (for example, organizing skills), and (e) the relationship of content-area knowledge to thinking.

Resnick and others proposed advancing beyond a curriculum that focuses on coverage of disciplinary knowledge and cultural literacy to a “thinking curriculum:” “… what is needed is to emphasize the teaching of thinking processes and skills” (Resnick & Klopfer, 1989, p. vi). These processes and skills comprise the tools of inquiry for discovery and validation of knowledge (Resnick & Klopfer, 1989, p. vi). The “thinking curriculum” is not restricted to higher-order thinking processes and skills. Instead, it includes all thinking processes and skills, including, for example, those involved in learning the most elementary levels of mathematics, reading, and other subjects.

In short, thinking skills can be taught; students’ thinking can be improved and changes in their thinking skills can be assessed. Although there are multiple forms of thinking, this research study focuses on complex thinking. Complex thinking is an advanced form of thinking that is goal-directed, strategic, and multi-session (Clifford, 1990; Iowa Department of Education, 1989; Iowa State University, 1999; Jakovljevic, Ankiewicz, & de Swardt, 2005; Jonassen, 1996a, 2000; Jonassen & Carr, 2000; Jonassen, Carr, & Yueh, 1998; Means & Olson, 1995; Slangen & Sloep, 2005).

This study also examines three interrelated topics that play an important role in the development of complex thinking skills: constructivism, constructionism, and case-based reasoning. Constructivism addresses the
issue of how mental constructions are formed (Spivey, 1997). Constructionism is a particular approach to forming
the mental constructions through the creation of artifacts (Kafai, 1996). Case-based reasoning is a specific process
that addresses practical, real world needs while forming mental constructions (Kolodner, 1993). Tools are needed to
support the various processes promoting the development of complex thinking skills. This chapter presents a
synthesis of the salient factors influencing the development of such tools. The synthesis is based on constructivism
and extended by constructionism and CBR.

Complex Thinking

As used here, complex thinking is the core of higher-order or advanced thinking (Iowa Department of
Education, 1989; Iowa State University, 1999; Jakovljevic, Ankiewicz, & de Swardt, 2005; Jonassen, 1996a, 2000;
Jonassen & Carr, 2000; Jonassen, Carr, & Yueh, 1998; Slangen & Sloep, 2005). Complex thinking is a goal-
oriented, strategic, and multi-step process that is required for resolving real-world issues. Real-world issues include
designing artifacts, making decisions, and solving problems. Designing artifacts consists of planning, inventing,
assessing, and revising a product. Making a decision consists of identifying the issue, generating alternatives,
assessing consequences, making a choice, and evaluating the choice. Solving a problem consists of defining the
problem, finding alternatives, choosing a solution, and building acceptance.

Real-world issues such as those listed and described above tend to be ill-structured and therefore
intellectually challenging (Ertmer & Newby, 1993; Gagne, 1985; Jonassen & Tessmer, 1996; Merrill, 2002). They
naturally demand and engage complex thinking as the most advanced order of thinking (Iowa Department of
Jonassen, Carr, & Yueh, 1998; Slangen & Sloep, 2005). Specifically, they demand an integration of content/basic
thinking, creative thinking, and critical thinking. In fact, the model of complex thinking proposed here has been
described as a holistic, interactive, or integrated model of thinking.

Content/basic thinking, creative thinking, and critical thinking are different but complementary (Iowa
Department of Education, 1989; Iowa State University, 1999; Jakovljevic, Ankiewicz, & de Swardt, 2005; Jonassen,
thinking represents accepted knowledge and includes metacognition. It consists of problem solving, designing, and
decision making. Creative thinking is a divergent process and represents generated knowledge (Csikszentmihalyi,
Divergent thinking explores alternatives. Creative thinking consists of synthesizing, imagining, and elaborating. Critical thinking is a convergent process and represents reorganized knowledge. It consists of analyzing, evaluating, and connecting. A convergent process seeks to establish facts and to understand phenomena. This integration of thinking processes as applied to resolving real-world issues is depicted in Figure 1:

![Complex thinking diagram](basic_design.png)

Figure 1. Complex thinking.

(Basic design inspired by and data obtained from: Iowa Department of Education, 1989; Jonassen, 2000)

This research study focuses on designing a tool for engaging complex thinking. As discussed above, complex thinking is intimately connected with concrete, real-world experience: designing artifacts, making decisions, and/or solving problems (Iowa Department of Education, 1989; Iowa State University, 1999; Jakovljevic, Ankiewicz, & de Swardt, 2005; Jonassen, 1996a, 2000; Jonassen & Carr, 2000; Jonassen, Carr, & Yueh, 1998; Slangen & Sloep, 2005). Therefore, design of this tool is appropriately informed by three bodies of theory that are likewise connected to concrete, real-world experience: constructivism, constructionism, and case-based reasoning. All three address how to effectively learn from concrete, real-world experience but do so in different ways (Kolodner & Guzdial, 2000; Kolodner, Owensby, & Guzdial, 2004; Riesbeck, 1996). As a learning theory, constructivism addresses use of experience for forming powerful, advanced knowledge constructions. As a type of pedagogy, constructionism offers an experience-based, hands-on method for forming the powerful, advanced knowledge constructions of constructivism. Case-based reasoning offers a model with structures and processes to
advise and predict learning effectively from concrete, real-world experience. In the next section, constructivism is explored for advice on designing a complex thinking tool that can be used to extract maximum learning benefit from concrete, real-world experience.

**Constructivism**

Learning is a complex phenomenon with multiple dimensions (Ertmer & Newby, 1993). In an eclectic approach to understanding this complexity, behaviorist, cognitivist, and constructivist theories have been proposed to explain the different dimensions (Ertmer & Newby, 1993). The overarching tenet of constructivism is that learning is a process of internally, actively constructing knowledge from experience and existing knowledge (Dick, Carey, & Carey, 2005; Doolittle & Camp, 1999; Spivey, 1997). Powerful knowledge constructions are thought to be constructed as a result of responding to a challenging, perturbing environmental stimulus, such as the task of designing an artifact, making a decision, or solving a problem. These are the types of real-world issues that engage complex thinking.

The fundamental character of constructivism can be better understood by examining the different metaphors for the learning process that have been productively used over the years to clarify and promote learning (Wilson, 1995; Joyce & Weil, 1996; Papert, 1993; Spivey, 1997). For example, comparing the constructivist metaphor and the transmissionist metaphor can help to clarify the meaning of constructivism.

A transmissionist interpretation of the learning process has historically dominated American education (Erickson & Lehrer, 1998; Harel, 1991; Papert, 1980, 1993, 1996, 1998, 2000). As the name indicates, this metaphor denotes basically a transmission process, one of transferring an object – a preserved, intact entity – from one point to another, much as a telegraph message moves from sender of message to receiver or a Frisbee passes back and forth between college students (Jonassen, 1991; Joyce & Weil, 1996; Oliver, 2000; Papert, 1996, 1998, 2000). From this perspective, knowledge and meaning are objective. They originate and exist outside the learner. Knowledge domain experts (for example, subject matter experts) encode this knowledge (content) using various symbol systems such as written and spoken language. Educators then in mediated instruction attempt to effectively and efficiently deliver it to students. Educators “[pour] an ocean of facts through a straw into the child's memory” (Pea, 1985, p. 176). Students are expected to listen, observe, internalize knowledge, practice, reproduce, and show a “... complete and
correct understanding” (mastery) of knowledge (Duffy & Jonassen, 1991, p. 8). The learner may interact with the environment so as improve the knowledge transference and internalization process.

Constructivist learning consists not of a transmission process as seen in the transmissionist metaphor above but rather a construction process, much as carpenters follow a plan and construct a house piece by piece. According to constructivism, information exists outside the learner, but knowledge originates and exists inside the learner. Information is “out there,” passive, disconnected, “dead,” “inert,” “data devoid of purpose,” stored and retrieved (Jones & Idol, 1990; Spivey, 1997). Learners are manipulators of data and information, “information workers,” constructors of knowledge, “authors of knowledge,” and “active knowledge workers” (Lehrer, 1993; Lehrer, Erickson, Connell, 1994; Perkins, 1986; Pohl & Purghatohfer, 2000, p. 813; Stager, 2001). “Children don't get ideas; they make ideas” (Kafai, 1996, p. 1). The learner constructs knowledge using his existing knowledge and experience. The learner actively constructs new knowledge as a response to environmental stimuli such as designing an artifact, making a decision, or solving a problem. These are the types of real-world issues that engage complex thinking processes.

Piaget attempted to explain the knowledge construction process, as it is theorized to occur in childhood, by proposing cognitive structures, processes, and states of knowledge construction, including the structures of schemata, the processes of assimilation, accommodation, and tuning, and the states of disequilibrium and equilibrium (Norman, Gentner, & Stevens, 1976; Piaget, 1972, 1990; Gruber & Voneche, 1977). The structures of schemata are networked theme-based nodes of knowledge – concepts. As complex wholes, schemata/concepts amount to more than a sum of their parts. With the assimilation process, familiar environmental experiences fit into existing schemata. For example, completing a third set of whole-number addition problems does not involve learning new knowledge (new schemata) but rather reinforcing existing knowledge (existing schemata). With the accommodation process, unfamiliar environmental experiences – adding fractions after learning addition of whole numbers – do not fit into existing schemata. This misfit of new, different experiences to existing, inadequate schemata produces cognitive dissonance in the learner and a state of disequilibrium. Existing schemata must be restructured to fit or accommodate the unfamiliar experience and to return to equilibrium. In this example, the schemata/concept of addition must be extended to include fractions. Finally, the clarity of schemata is improved in
the refinement process of tuning. Increasing quantity and quality of experience with adding whole numbers and fractions produces refinement and automaticity.

The knowledge constructed within schemata through assimilation and accommodation can vary in quantity and quality. Knowledge constructions consisting of less quantity and quality of knowledge represents introductory learning. Constructions consisting of greater quantity and quality of knowledge represent advanced learning. Five interpretations of levels of knowledge acquisition from lower to higher levels are depicted in Figure 2:
Multiple factors influence the acquisition of the different levels of knowledge described in Figure 2. For example, Ormrod (1999) suggests different degrees of education and experience are responsible for producing different levels of knowledge as depicted in Figure 3:

![Figure 3. Levels of knowledge: Five interpretations.](image)

1Jonassen, Mayes, & McAleese, 1993
2Alexander, 1997
3Spiro, Coulson, Feltovich, & Anderson, 1988
4Benner, 1984
5Ertmer & Newby, 1993

(Data obtained from: Ormrod, 1999)
The pre-novice level of knowledge is fragmented, atomistic, discrete, and isolated, with reflexive performance, whereas expert knowledge is more holistic, tightly integrated, refined, and generative knowledge, with flexible, automatic, creative performance (Alexander, 1997; Bruning, Schraw, & Ronning, 2004; Jonassen, 1992; Jones & Idol, 1990; Kohler, 1971; Newell, 1993; Perkins, 1986; Perkins & Salomon, 1992; Salomon & Perkins, 1988; Spiro et al., 1988; Spiro et al., 1992a). Overall, knowledge becomes increasingly complex and/or ill structured toward the advanced end of the continuum (Spiro et al., 1988; Spiro et al. 1992a; Spiro et al., 1992b). In addition to possessing greater quantity and quality of knowledge, a learner at the high end of the continuum is also more likely to practice effective learning strategies and to demonstrate a personal interest in the domain (Alexander, 1997).

There is overall agreement among theorists on the overarching tenet of constructivism, that is, learning is a process of internally, actively constructing knowledge from experience and existing knowledge. However, theorists have not yet reached much consensus on other issues with regard to describing constructivism. In fact, constructivism is not a simple, unitary construct but rather a complex, differentiated construct with different interpretations or types of constructivism (Carey, 1998; Cognition and Technology Group at Vanderbilt, 1991; Cunningham, 1992; Dalgarno, 2001; Dick, 1992; Dick, Carey, & Carey, 2005; Doolittle, 2000; Doolittle & Camp, 1999; Garrison, 1997; Jonassen, 1992; Karagiorgi & Symeou, 2005; Merrill, 1992; Moshman, 1982; Perkins, 1991, 1992; 1995, 1999; Phillips, 1995; Spiro et al., 1992b; von Glasersfeld, 1984, 1998). Understanding this typology is essential to fully grasping the theoretical underpinnings of this research study and, ultimately, the design features of the complex thinking tool created in this study (Bruning et al., 2004; Carey, 1998; Dalgamo, 2001; Dick, Carey, & Carey, 2005; Doolittle & Camp, 1999; Moshman, 1982; Perkins, 1991, 1999; Phillips, 1995; Reigeluth, 1992, 1999; Roblyer, 1996; Spivey, 1997; Wilson, 1997).

Given the lack of consensus, it is not surprising that different interpretations of constructivism have been advanced. These different interpretations have been categorized into three basic types of constructivism, ranging from more "conservative" types, to more "moderate" and "radical" types on the constructivist continuum presented in Figure 4 (from left to right on the continuum). Since the descriptors for the three major types of constructivism in Figure 4 are basically equivalent, descriptors of only the first set of descriptors – exogenous, dialectical, and endogenous constructivism – will be described and distinguished.
The word exogenous basically means "from outside the organism" (Merriam-Webster, 2006a). With exogenous constructivism (Dalgarno, 2001; Moshman, 1982), the environment heavily influences the learner's knowledge construction process. Emphasis is on learner control and active construction of knowledge. However, while direct instruction is recognized to have value, it is viewed to play a different role. Knowledge construction is considered to be internalization and reconstruction of external structures according to experience. Knowledge constructions should truly mirror an external objective reality. Different learners can construct similar or identical mental models based on similar or identical experiences. An example is a hypermedia environment that includes context sensitive pedagogical guidance.

Endogenous means "within the organism" (Merriam-Webster, 2006a). In endogenous constructivism (Dalgarno, 2001; Moshman, 1982), the emphasis is on learner-directed exploration and discovery, and the construction of new knowledge from earlier personal knowledge structures. Knowledge constructions result from internal cognitive processes. Piaget's developmental stages are an example of endogenous constructivism.

Dialectical constructivism (Dalgarno, 2001; Moshman, 1982) is midway between these two extremes of the continuum. Learning comes from the interaction (dialog) between the learner (internal factors) and the environment (external factors). Vygotsky (1962, 1978) is perhaps the best known proponent of dialectical constructivism. With social interaction, a learner (novice) can advance through his Zone of Proximal Development (ZPD) from his current performance level to a potential level that can be achieved with an expert's assistance. Moshman (1982, p. 196) concludes that dialectical constructivism “provides the most general perspective and has become increasingly important in cognitive psychology.”

The three basic types of constructivism distinguished among above have been rated as "weak" or "strong," not to indicate relative educational value but rather to indicate each type's degree of adherence to four foundational tenets of constructivism which have been proposed in the literature (Dewey, 1916, 1980; Doolittle & Camp, 1999; Garrison, 1997; Larochelle, Bednarz, & Garrison, 1998; Gergen, 1995; von Glasenfeld, 1984, 1998). These four fundamental tenets of constructivism which have been proposed in the literature are:
• Constructivist learning involves active cognizing: Learning is not passive but active
• Constructivist learning involves adaptive cognizing: Learning involves adaptation to the environment
• Constructivist learning involves sense-making: Learning does not involve simply creating accurate reconstructions of external structures but interpretation experience
• Constructivist learning involves social/cultural cognizing: Learning has a social/cultural basis as well as a biological/neurological basis (the brain and the mind)

The three types of constructivism are arranged on the constructivist continuum depicted in Figure 4 (Maddux & Cummings, 1999; Spivey, 1997; Winn, 1991). These types range from more “conservative” to more “radical” types depending on the degree of environmental influence exerted on the knowledge construction process. Each type also is rated as “weak” or “strong” according to how closely that type adheres to basic tenets of constructivism which have been proposed in the literature. The “weak”-“strong” ratings of the three basic types of constructivism also can be used to help clarify the differences between the types. Three different sets of descriptors for these three basic types of constructivism are presented in Figure 4:
The Constructivist Continuum

Type Descriptor Set 1
- Exogenous Constructivism
- Dialectical Constructivism
- Endogenous Constructivism

Type Descriptor Set 2
- Cognitive Constructivism
- Social Constructivism
- Radical Constructivism

Type Descriptor Set 3
- BIG Constructivism (Beyond Information Given)
- WIG Constructivism (Without Information Given)

Rated as “weak” constructivism on basis of adherence to tenets of Active Cognizing and Adaptive Cognizing
Rated as “strong” constructivism on basis of adherence to tenets of Active Cognizing, Adaptive Cognizing, Sense-Making, and Social/Cultural Cognizing
Rated as “strong” constructivism on basis of adherence to tenets of Active Cognizing, Adaptive Cognizing, Sense-Making, and partial adherence to Social/Cultural Cognizing

Figure 4. The constructivist continuum.

1 Dalgarno, 2001; Moshman, 1982
3 Perkins, 1991
4 von Glasersfeld, 1984

Constructionism

Constructivism, as a theory of learning, explains how students learn by constructing knowledge formations from concrete, real-world experience (Spivey, 1997). However, constructivism does not identify specific methods for stimulating the creation of these knowledge constructions (Doolittle, 2000). Fortunately, methods for facilitating constructivism can be extrapolated from constructivism (Cunningham, Duffy, & Knuth, 1993; Duffy & Cunningham, 1996; Dunlap & Grabinger, 1996; Honebein, 1996; Honebein, Duffy, & Fishman, 1991; Jonassen,
Constructionism, a derivative of and complement to constructivism, provides such a method that is centered on concrete, hands-on activities and experience (Ackerman, 2002; Kafai, 1996. Papert, 1990; Resnick, M., 1996).

Papert developed the idea of construction-based learning activities to serve as a method of constructivism and coined the similar term constructionism to denote it. (Papert, 1993). Constructionism can therefore be said to facilitate constructivism. Constructionism is essentially a method of constructivism that is centered on student construction activities (Ackerman, 2002). Knowledge construction "happens felicitously when the learner is engaged in the construction of something external or at least shareable [inspectable by a potentially critical audience]...a sand castle, a machine, a computer program, a book." (Papert, 1990).

The hands-on, thinking-in-action construction of an external artifact – a "product" – can facilitate and represent the construction of an internal artifact – a cognitive artifact or a better understanding of a topic of study (Harel, 1991, 2000; Harel & Papert, 1990; Kafai, 1996; Papert, 1990). Constructionism “leads us to a model using a cycle of internalization of what is outside, then externalization of what is inside and so on” (Papert, 1990). The external artifact is "physical" and overt while the internal artifact is cognitive and covert. The external and internal artifacts co-evolve in their design and construction. They represent mirror images of each other. The external artifact becomes a learning device or object-to-think-with. The making of the object both facilitates learning (the process dimension of learning) and represents learning outcomes (the product dimension of learning). Constructed learning devices or products can take many different forms, including: (a) a sand castle, (b) a LEGO machine, (c) a computer program, (d) a multi-user domain computer program, (e) a poem, (f) a theory, and (g) a research project (Harel, 1991; Kafai, 1996; Papert, 1990; Resnick, M., 1994).

A particularly illustrative example of a constructed artifact is the hypermedia-based product (Burton, Moore, & Holmes, 1995), which is the learning device constructed in the learner-as-designer teaching model (Ayersman, 1996; Carver, Lehrer, Connell, & Erickson, 1992; Cunningham, Duffy, & Knuth, 1993; Erickson, & Lehrer, 1998; Lehrer, 1993; Liu, 2001; Liu & Pedersen 1998; Liu & Rutledge 1997; Pohl, 1998; Pohl & Purgathofer, 2000; Reed & Rosenbluth, 1995; Wilhelm, 1995; Wilhelm, Friedemann, & Erickson, 1998). Hypermedia construction has been called "hyperlearning" (Wilhelm, 1995; Wilhelm et al., 1998). In this process, a Web page or
a HyperCard program is constructed as the learning device. Hypermedia construction is a particularly illustrative of constructionist learning mechanisms because the external construction mechanisms (hypermedia) closely reflect the internal mechanisms (the mind). The non-linear linking (meta-relational) capabilities offered by hypertext are at least crudely analogous to associative mechanisms at work in the human mind (Bush, 1945). The efficacy of hypermedia construction was investigated by Reed & Rosenbluth (1995). Thirteen upcoming high school seniors in the 1991 West Virginia Governor's Honors Academy served as hypermedia authors and created programs about factors that affected the values of four different decades - 1920s-30s, 1945-59, and 1960s. In general, findings showed that students' awareness of interrelationships among factors increased.

The activities associated with the construction of learning devices can occur locally or they can be distributed according to the relative geographic location of collaborating learners (Jonassen, Howland, Moore, Marra, 2003; Resnick, M., 1996). Local activities are conducted in one physical place. Distributed constructionist activities involve multiple students in dispersed physical locations who often collaborate by means of computer network technology, such as local area networks (LANs) and the Internet.

Although constructivism explains how learners form either very basic or powerful and advanced, knowledge constructions from concrete, real-world experience and constructionism explains how concrete, hands-on activities and experience facilitate the formation of these knowledge constructions, a model and defined process for extracting maximum educational value benefit from concrete, real-world experience has not been discussed. Case-based reasoning theory offers that model and process and is discussed in the next section.

**The CBR Model and Process**

The case-based reasoning (CBR) method for enhancing complex thinking skills is based upon a model of cognition that (a) treats concrete experience (concrete, specific knowledge as opposed to abstract, general knowledge) as primary to learning, (b) exposes learners to new experiences, and (c) integrates the factors of dynamic memory, reasoning, and learning (Aamodt & Plaza, 1994; Jonassen, Tessmer, & Hannum, 1999; Kolodner, 1993; Kolodner & Guzdial, 2000; Kolodner & Leake, 1996; Kolodner, Owensby, & Guzdial, 2004; Leake, 1996a, b; Riesbeck & Schank, 1989; Schank, 1982, 1990, 1999). Through a natural, analogical reasoning cycle, stories of concrete problem-solving experience retrieved from dynamic memory (analsgs) are adapted to interpreting and solving a new problem. The problem solving experience affects dynamic memory by generating a complete, new
story which is saved for future use. As a result, dynamic memory evolves with the integration of each new story. This evolution represents learning.

Medicine (Jonassen, 1996b) and everyday grocery shopping (Kolodner, 1993) offer examples of CBR applications. A physician can utilize CBR to decide how best to treat a patient's condition. After establishing similarity and relevance of past patients' situations to the present patient's (a problem interpretation process), the physician can adapt treatment regimens from those situations to the present situation (a problem solving process). A grocery shopping trip offers another example of the CBR process. A shopper visits an unfamiliar grocery store and realizes there is a problem: the dairy section must be found. The shopper remembers navigating other grocery stores on past shopping trips. The dairy section was usually found along the perimeter of the store. The shopper adapts this past solution to his present situation in the unfamiliar store, and finds the dairy section in the expected location. Finding it in the expected location confirmed the past solution. Not finding it there would have required modifying the past solution.

These examples of CBR applications, particularly the medical example, imply engagement of complex thinking skills. In fact, this literature review focuses on the use of CBR as a means for promoting complex thinking skills. When used in this manner, CBR involves the interaction of processes and components. Basic components of the CBR method include: (a) historical cases, (b) the indexes of cases, and (c) a means of processing the cases (Aamodt & Plaza, 1994; Kolodner, 1993; Kolodner & Guzdial, 2000; Kolodner & Leake, 1996; Kolodner, Owensby, & Guzdial, 2004). Historical cases are basically complete stories (interpretations) of past problem-solving episodes, which can be adapted to solving similar, new problems (Hernandez-Serrano & Jonassen, 2003; Schank, 1990).

As indicated above, cases are primarily made up of concrete, specific knowledge (Aamodt & Plaza, 1994; Kolodner, 1993). Indexing is a process that produces classifying descriptors of the concrete experiences represented in the cases (Aamodt & Plaza, 1994; Jonassen, Tessmer, & Hannum, 1999; Kolodner, 1993; Kolodner & Guzdial, 2000; Schank, 1990). Indexes point to the usefulness of cases – their similarity (Ross, 1986) and adaptability to new situations – and allow retrieval of the cases from a case library. So as to fully describe the case, indexes are assigned to all aspects of the case; that is, indexes for problem, for solution, and for outcome. Indexes are revised as continuing experiences produce new insights about the value and applicability of the case. Schank (1990, pp. 84-85)
pointed to the central role of indexes in his assertion that "the bulk of what passes for intelligence is no more than a massive indexing and retrieval scheme that allows an intelligent entity to determine what information it has in memory that is relevant to the situation at hand, to search for and find that information."

The last of the components listed above, the case processor, performs CBR operations – creating, indexing, and adapting cases (Kolodner & Guzdial, 2000). When used to promote complex thinking skills, these operations will be performed by a human acting alone or with the assistance of technology.

In general, the CBR process begins with the introduction of a new problem to be solved and ends with that problem being solved (Aamodt & Plaza, 1994). The concrete experience of each new problem-solving episode is captured in the format of a new case and saved to memory where it joins a case library for adaptation to future problem-solving needs. The CBR process restarts with the introduction of yet another problem to be solved.

Aamodt and Plaza (1994) presented CBR as a problem-based learning process consisting of four major, integrated steps, stages, or tasks (the four REs): (a) REtrieve cases from a case library that are similar to a new, unsolved problem, and therefore relevant to solving it; (b) REuse the information and knowledge offered by the retrieved cases; (c) REvise the suggested solution to the new problem; and (d) RETain useful parts of this most recent problem-solving episode in a new case for application to future problem-solving needs. Each of these major tasks is comprised of a hierarchy of sub-tasks.

Two different views of the CBR process are depicted in Figures 5 and 6. The recurring cycle of four major tasks – REtrieve, REuse, REvise, and RETain – is emphasized in Figure 5. The hierarchy of task/subtasks and alternative methods that comprise each of these four major tasks is emphasized in Figure 6. In Figure 6, the top-most task is problem solving and learning from experience. The method for performing that task is case-based reasoning. Succeeding levels of tasks, beginning at the level of REtrieve ↔ REuse ↔ REvise ↔ RETain, are indicated by solid lines; relevant alternative methods, by dotted lines. General sequencing of tasks and methods is indicated by their vertical orientation.
Figure 5. CBR: The cycle.

(Used with permission of Aamodt & Plaza, 1994, p. 8)

Figure 6. CBR: The hierarchy of tasks and methods.

(Used with permission of Aamodt & Plaza, 1994, p. 10)
Types of knowledge include *abstract/general* knowledge and *concrete/specific* knowledge (Aamodt & Plaza, 1994; Didierjean & Cauzinille-Marmeche, 1998; Kolodner, 1993). Complex thinking involves both types of knowledge (Iowa Department of Education, 1989; Iowa State University, 1999; Jakovljevic, Ankiewicz, & de Swardt, 2005; Jonassen, 1996a, 2000; Jonassen & Carr, 2000; Jonassen, Carr, & Yueh, 1998; Slangen & Sloep, 2005). Similarly, the CBR process is centered on concrete/specific knowledge, but execution and outcomes of the process often include abstract/general knowledge. This abstract/general knowledge is found in (a) established procedures applied to problem solving, (b) indexes of cases, and (b) lessons learned from a problem-solving episode and included in a case. Case-based problem-solving and learning can therefore be said to be both case-based (involving concrete/specific knowledge) and schema-mediated (involving abstract/general knowledge; Aamodt & Plaza, 1994; Didierjean & Cauzinille-Marmeche, 1998).

CBR is a relatively new model of human cognition that was originally applied to humans and later extended to machines (Jonassen, Tessmer, & Hannum, 1999; Kolodner, 1993; Kolodner & Guzdial, 2000). The model was inspired by Schank's dynamic memory theory, a theory of reminding and learning (Schank, 1982, 1990, 1999) that was introduced about twenty-five years ago. Researchers created the model on the basis of observing the case-based problem-solving behavior of human experts, for example, tank commanders, physicians, lawyers, and apprentice mechanics (Jonassen, Tessmer, & Hannum, 1999; Klein & Calderwood, 1988; Lancaster & Kolodner, 1988; Kolodner, 1993; Kopeikina, Brandau, & Lemmon, 1988). When confronted by a problem to be solved, these experts would think back to similar problems they faced in the past and how they dealt with them, and then adapt old solutions to solving the new problem.

Although this review focuses on human uses of CBR, research has been conducted on both human and technological applications of CBR (Kolodner, 1993). In fact, technological applications of CBR often deal with Artificial Intelligence (AI), and research on AI applications has historically dominated and defined the CBR literature (Kolodner, 1993; Kolodner & Guzdial, 2000; Leake, 1996a, b; Riesbeck & Schank, 1989; Watson, 1997). However, research on AI applications benefits research on human applications. Specifically, the computational modeling of the CBR model as found in PERSUADER, CELIA, and similar machine-based applications (or reasoners) has not only resulted in the creation of useful problem-solving applications, but has also intentionally
served to clarify this model of human cognition (Kolodner, 1993; Kolodner & Guzdial, 2000; Kolodner, Owensby, & Guzdial, 2004).

Observing a lack of research on educational applications of CBR, some researchers have called for more research in this area (Kolodner, 1997; Kolodner & Guzdial, 2000; Schank, 1999; Schank, Berman, & Macpherson, 1999; Wang, Wedman, & Shyu, 2003). According to Wang (Wang et al., 2003, p. 46): “…. CBR research and development in the education domain is conspicuously lacking …. there is a need to explore CBR applications in education … literature relating to designing and developing practical CBR educational applications is particularly needed.” Most early research relating to educational applications of CBR deals with developing the learning supports, and includes relatively little empirical research on the efficacy of these supports (Hernandez-Serrano & Jonassen, 2003).

Research dealing with CBR-inspired applications in education often concentrates on the development and/or efficacy of CBR products and may be classified into four general types: (a) CBR-inspired learning environments, (b) supports for reflection and interpretation of personal experience, (c) case libraries, and (d) hybrids combining supports for reflection and interpretation with libraries (Kolodner & Guzdial, 2000; Kolodner, Owensby, & Guzdial, 2004). Much of the research on educational applications of CBR has been carried out in two centers (Kolodner, 1997): (a) Institute for the Learning Sciences (ILS), Northwestern University, by Roger Schank; and (b) the EduTech Institute, Georgia Institute of Technology, by Janet Kolodner (Bell, Bareiss, & Beckwith, 1993; Narayanan & Kolodner, 1995; Schank, Fano, Jona, & Bell, 1993; Zimring, Do, Domeshek, & Kolodner, 1995).

Multiple CBR-inspired learning environments have been developed, including goal-based scenarios (GBS) and the learning-by-design (LBD) curricula (Kolodner, 1997; Kolodner, Camp, Crismond, Fasse, Gray, Holbrook, & Puntembakar, 2003). A GBS is an example of simulation-based learning. Playing a role in a scenario, the learner must accomplish a mission, perform a task, or achieve a goal. Learning outcomes are embedded in the scenario. Achieving the goal of the scenario involves simultaneously achieving learning outcomes. In GBS, cases are used as a resource for addressing the achievement of a goal. The basic goal of the LBD project has been to develop and publish design-based science curriculum units (including software) for middle school students. In LBD, students are solving design problems, in part by using cases from an existing library and also by creating new cases that can be added to a case library. Consistent with the CBR model, both GBS and LBD are centered on concrete experience –
reflecting on and learning lessons from experience, and anticipating how those lessons might be applied in future situations.

With regard to efficacy, GBS supports effective recall of relevant contextualized knowledge and skills, and anticipation of potential application contexts (Bell, 1996; Bell, Bareiss, & Beckwith, 1993; Bell, Davis, & Linn, 1995). LBD equaled or exceeded comparison environments with regard to recall of science content. In the areas of capability to design experiments, planning for data collection, and collaboration, students in LBD classes outperformed those in comparison environments (Kolodner, Owensby, & Guzdial, 2004). In addition, LBD environments have found to support transfer of learning (Kolodner et al., 2003).

Some approaches to experience-based learning utilize case-based reasoning (Kolodner & Guzdial, 2000; Kolodner, Owensby, & Guzdial, 2004). Supports for experience-based learning have been developed that involve reflecting on and interpreting experience, both of which are important parts of the CBR process (Kolodner & Guzdial, 2000; Kolodner, Owensby, & Guzdial, 2004). Some relevant examples are listed in Table 1:

<table>
<thead>
<tr>
<th>Learning Support</th>
<th>Concept</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case Authoring Tool (CAT)</td>
<td>Support for reading, learning from authentic cases</td>
</tr>
<tr>
<td>Reflective Learner</td>
<td>Reflective essays in project-based engineering</td>
</tr>
<tr>
<td>JavaCap</td>
<td>Reflecting on, capturing project experiences for others</td>
</tr>
<tr>
<td>Storyboard Author</td>
<td>A descendant of JavaCap with improvements</td>
</tr>
<tr>
<td>Lessons Learned</td>
<td>A descendant of JavaCap with focus on technical writing</td>
</tr>
<tr>
<td>Design Discussion Area</td>
<td>Presentation, vocabulary skills in experienced-based learning</td>
</tr>
<tr>
<td>Tools in Smile</td>
<td>Descendent of Design Discussion Area</td>
</tr>
</tbody>
</table>

(Data obtained from: Kolodner & Guzdial, 2000; Kolodner, Owensby, & Guzdial, 2004)

Case Authoring Tool (CAT) supports small groups of students in reading an expert case, abstracting what can be learned from the case, and publishing a presentation of findings that can serve as a resource for other learners (Kolodner, Owensby, & Guzdial, 2004; Nagel & Kolodner, 1999). Reflective Learner is used in undergraduate project-based engineering design courses to prompt for reflection on and reuse of experience from learning activities of the course, and the writing up what has been learned in "learning essays," a standard requirement in such courses (Kolodner, Owensby, & Guzdial, 2004; Turns, Newstetter, Allen, & Mistree, 1997).
JavaCap (Shabo, Nagel, Guzdial, & Kolodner, 1997) Storyboard Author (Nagel & Kolodner, 1999) and Lessons Learned (Kolodner, Owenbys, & Guzdial, 2004) are related applications. JavaCap was the original, and the other two descended from it and extended it. They were developed to support reflecting on and learning from activities which occur in the Learning-by-Design (LBD) learning environment. JavaCap supports reflection on a design project for what has been learned from solving a design problem in the LBD environment. It prompts for design problem analysis, alternative solutions to the problem, rationale for the selected solution, lessons learned from the experience, and creation of a resource for the benefit of other students. Storyboard Author supports articulation of experience related to addressing a project challenge. It prompts the student to analyze the project challenge, solution to it and route to the solution, application of science knowledge to developing the solution, an evaluation of how well the solution worked, and what was learned. The set of responses is published as a presentation for other students to learn from. Lessons Learned adds support for technical writing - needed specificity in describing what has been learned, and use of scientific terminology and phraseology.

Design Discussion Area (DDA) and its successor Tools in Smile (Kolodner & Nagel, 1999) supports sharing of design ideas and results as vicarious experience for others. SMILE adds tools that further facilitate this process, such as the Experiment Result Tool for reflexive analysis of results.

Although the above applications utilize CBR principles and components, research regarding the efficacy of these applications to promote complex thinking is limited. In one study, users of Reflective Learner wrote longer, more structured essays, and earned significantly higher grades than non-users (Turns, Newstetter, Allen, & Mistree, 1997). In addition, students retained more of what they learned, and were more responsible, active learners (Turns et al., 1997). Use of Design Discussion Area and its successor Tools in Smile resulted in more complete and articulate time write-ups, and a higher level of classroom discussion.

CBR programs rely on the availability of multiple cases and this need has led to the development of case libraries as learning resources (Colaric & Jonassen, 2000; Colaric, Turgeon, & Jonassen, 2003; Jonassen, Tessmer, & Hannnum, 1999; Kolodner, Owensby, & Guzdial, 2004). As an integral component of a library, the library's indexing scheme enables transfer and application of knowledge stored in the library to resolving new problem situations (Schank, 1990). Some examples (Colaric & Jonassen, 2000; Colaric, Turgeon, & Jonassen, 2003;
Jonassen, 1996b; Jonassen, Tessmer, & Hannum, 1999; Kolodner & Guzdial, 2000; Kolodner, Owensby, & Guzdial, 2004; Narayanan & Kolodner, 1995; Wang et al., 2003) are presented in Table 2:

Table 2

<table>
<thead>
<tr>
<th>Description</th>
<th>Type of Problems Solved</th>
</tr>
</thead>
<tbody>
<tr>
<td>Archie-2</td>
<td>Architectural design issues</td>
</tr>
<tr>
<td>STABLE</td>
<td>Computer programming skills</td>
</tr>
<tr>
<td>Case Application Suite</td>
<td>Application of cases</td>
</tr>
<tr>
<td>Parent-teacher conferences</td>
<td>Analysis of what happens in a conference</td>
</tr>
<tr>
<td>Technology integration</td>
<td>Analysis of technology integration</td>
</tr>
<tr>
<td>USDOE PT3/KITE database</td>
<td>Increase technology integration</td>
</tr>
<tr>
<td>Turfgrass management</td>
<td>Learn domain skills and knowledge</td>
</tr>
<tr>
<td>Medical case library</td>
<td>Diagnostic and causal reasoning skills</td>
</tr>
<tr>
<td>Susie</td>
<td>Issues of sustainable technology and development</td>
</tr>
</tbody>
</table>

(Data obtained from: Colaric & Jonassen, 2000; Colaric, Turgeon, & Jonassen, 2003; Jonassen, 1996b; Jonassen, Tessmer, & Hannum, 1999; Kolodner & Guzdial, 2000; Kolodner, Owensby, & Guzdial, 2004; Narayanan & Kolodner, 1995; Wang et al., 2003)

The Archie-2 (Zimring et al., 1995) library was created to serve as a design aid for professional architects. It consists of cases relating to the design of public buildings with an emphasis on libraries and courthouses. It helps architects make design decisions. The cases of STABLE – SmallTalk Apprenticeship-Based Learning Environment (Kolodner, Owensby, & Guzdial, 2004) – helps students learn skills relating to object-oriented design and programming. Cases were created from previous students’ work. Case Application Suite (Kolodner, Owensby, & Guzdial, 2004) consists of three parts that respectively support the three stages of applying cases to problem-solving. These parts include: Case Interpretation, Case Application, and Solution Assessment.

Case databases or libraries have been used to present task analyses of the knowledge, skills, and attitudes that need to be taught and learned in domains of interest (Jonassen & Hernandez-Serrano, 2002; Jonassen, Tessmer, & Hannum, 1999). Domains include the effective conduct of parent-teacher conferences and innovative integration of technology in school curricula (Jonassen & Hernandez-Serrano, 2002; Jonassen, Tessmer, & Hannum, 1999). These databases or libraries would be useful not only to instructional designers but also to pre-service and in-service practitioners. The United States Department of Education (USDE) has shown support for the case method by creating a case library of technology integration stories to increase technology use in schools (Wang et al., 2003).
This library was created as part of the Department’s Preparing Tomorrow's Teachers to Use Technology (PT3) program and the Knowledge Innovation for Technology in Education (KITE) project.

Web-based case libraries can invite contributions from the skilled in-service practitioners of a domain. Contributed cases can be used to model best practice and offer vicarious experiences for the benefit of novice practitioners. Novice pre-service practitioners in college courses exploring topics such as turfgrass management can learn domain skills and knowledge from consulting such libraries (Colaric & Jonassen, 2000; Colaric, Turgeon, & Jonassen, 2003; Kolodner & Guzdial, 2000). Multimedia online libraries can be used in college architecture courses to support the problem solving integral to designing buildings (Narayanan & Kolodner, 1995). Novice medical students can theoretically develop expert diagnostic and causal reasoning skills from studying skilled professional experience (including “illness scripts”) represented in cases made accessible in a hypertext-based learning environment (Jonassen, 1996b).

Users of a case library can go beyond simply exploring a library by also contributing to it. For example, graduate students in a CBR course at the Georgia Tech College of Computing both used and contributed to the Susie (SUStainable technology Interactive Education) case library (Narayanan & Kolodner, 1995). The library contained cases about pollution, the natural environment, and industrial accidents as related to sustainable technology and development. These students acquired domain content by using and contributing to the library.

One indicator of libraries’ efficacy is whether or not libraries continue to be used. However, because continued use might reflect inertia more than efficacy, other indicators must be used. Two general findings as to the efficacy of libraries relate to building cases and comparing/contrasting cases (Kolodner & Guzdial, 2000; Kolodner, Owensby, & Guzdial, 2004; Narayanan & Kolodner, 1995). First, students can learn as much or even more from constructing cases as from simply using them, assuming a case structure is specified. Second, comparing and contrasting multiple cases for relevance to problem-solving can be a more effective learning experience than applying a single case. Other findings include improved performance coming from use of STABLE, and persistence of performance in the absence of STABLE (Kolodner, Owensby, & Guzdial, 2004). Compared to non-users of the Case Interpretation Tool in the Case Application Suite, users generally extracted rules of thumb from cases better, interpreted cases well, and wrote better summaries (Kolodner, Owensby, & Guzdial, 2004). In addition, the use of case libraries was found to improve complex/ill-structured problem solving performance by undergraduate novices.
(Hernandez-Serrano & Jonassen, 2003). Finally, writing, revising, discussing, and sharing cases about student teaching dilemmas in the context of domain content was found to advance the professional knowledge and skills (pedagogical theory and practice) of student teachers from naive levels to expert levels (Hammerness, Darling-Hammond, & Shulman, 2001).

Hybrids have been developed combining supports for reflection and case libraries as separately discussed above. SMILE is an example (Guzdial, Hmelo, Hubscher, Nagel, Newstetter, Puntembakar, Shabo, Turns, & Kolodner, 1997; Nagel & Kolodner, 1999). Users of SMILE created presentations, indexed them, and added them to a SMILE-based library for public access and feedback. Presentations were used to document the conducting of investigations, resolution of issues, lessons learned from the experience, and application of the lessons. Although created for science, SMILE was appropriate for general project/inquiry-based inquiry. In addition, a hybrid support for case writing and collecting cases has been developed for use in teacher education (Angeli, Bonk, Supplee, & Malikowski, 1998). A Web-based electronic space was created for undergraduate teacher education students in field placements to share case stories and post peer feedback to stories. Using a reflective approach, placements could thus integrate observations of real events from field experiences with relevant text material. Positive findings included: (a) students were "heavily involved" in electronic writing; (b) students, instructors, and cooperating teachers used the electronic conferencing tool to communicate; and (c) students reflected on field observations in a shared electronic space.

**Characteristics of a Comprehensive CBR Tool**

The use of the CBR process for supporting complex thinking skills, and examples of existing CBR tools have been discussed in this chapter. Seven characteristics of a comprehensive CBR tool have been identified on the basis of this discussion and will be summarized in this last section of the chapter. Each of the existing CBR tools described in this chapter includes one, some, or even many of these characteristics, but not all of the characteristics found in a comprehensive CBR tool. These characteristics reinforce five facilitators of learning effectively from experience (Kolodner, 1997; Kolodner & Guzdial, 2000; Kolodner, Owensby, & Guzdial, 2004) identified, listed, and described in Table 3:
Table 3

Five Facilitators of Learning Effectively from Experience

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Facilitator</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Instructive</td>
<td>Having educationally valuable or <em>instructive</em> experiences</td>
</tr>
<tr>
<td>2. Lesson transfer</td>
<td>Interpreting experiences for identifying <em>lessons</em>, capturing experiences in the format of a <em>case</em>, and <em>transferring</em> cases to new situations</td>
</tr>
<tr>
<td>3. Indexing</td>
<td><em>Indexing</em> a cases to show applicability of the case</td>
</tr>
<tr>
<td>4. Failure</td>
<td>Learning from <em>failure</em> and corrective strategizing</td>
</tr>
<tr>
<td>5. Case-based reasoning</td>
<td>Using cases as a basis for <em>reasoning</em></td>
</tr>
</tbody>
</table>

(Data obtained from: Kolodner & Guzdial, 2000; Kolodner, Owensby, & Guzdial, 2004)

One characteristic of a comprehensive CBR tool is that the tool should support the integration of (a) reasoning, (b) understanding and learning, and (c) dynamic memory. In the CBR process, understanding/learning is explained to be the outcome of two operations: (a) reasoning to achieve problem-solving goals and (b) indexing cases of new problem-solving episodes and saving them to a case library in dynamic memory for advancement of the library and adaptation to future problem-solving needs (Aamodt & Plaza, 1994; Jonassen, Tessmer, & Hannum, 1999; Kolodner, 1993; Kolodner, 1997; Kolodner & Guzdial, 2000; Kolodner & Leake, 1996; Kolodner, Owensby, & Guzdial, 2004; Schank, Berman, & Macpherson, 1999).

A comprehensive CBR tool should also provide the user support for completing the four major steps/tasks that make up the CBR process. These four major steps/tasks are: (a) REtrieve, (b) REuse, (c) REvise, and (d) RETain (Aamodt & Plaza, 1994). Each step/task includes a hierarchy of sub-tasks and methods required for completing the CBR process.

In addition, a comprehensive CBR tool should support application of the CBR process to ill-structured problem solving. Problem solving can be of two basic types: either an algorithmic-type problem-solving, or the more challenging ill-structured-type problem-solving (Gagne, 1985; Jonassen & Tessmer, 1999; Merrill, 2002; Spiro et al., 1988; Spiro et al., 1992a; Spiro et al., 1992b; Spiro et al., 1987). Algorithmic problem-solving skills follow a
set pathway to the problem solution. In contrast, ill-structured problem-solving skills follow a circuitous and unpredictable pathway to solution, and rank at the highest level of intellectual skills.

Moreover, a comprehensive CBR tool should combine a case library with supports for reflection on and interpretation of experience (Kolodner & Guzdial, 2000; Kolodner, Ownesby, & Guzdial, 2004). A case library is simply an indexed database of solved cases, each including the parts of problem, solution, outcome, and lessons learned, and sometimes linked to support materials. Supports for reflection can take the form of prompts that stimulate reflection on and interpretation of experiences and guide the construction of written responses for storage in a database and future application to problem-solving needs.

Additionally, a comprehensive CBR tool should possess a versatility for ready integration of the tool into a range of problem domains such as professional education (Jonassen, Tessmer, & Hannum, 1999), architecture (Zimring et al., 1995), turfgrass management (Colaric, Turgeon, & Jonassen, 2003), or medicine (Jonassen, 1996b). The CBR process is a general problem-solving method that is applicable to the problems encountered in a range of domains (Kolodner, 1993). Representation of the CBR process in the tool should be free of domain-specific content. In addition, the tool should allow customization of the tool to particular domains by allowing addition of domain-specific content within the framework of the CBR process.

Furthermore, a comprehensive CBR tool should include the capability for users of the tool's case library to construct their own cases and contribute them to the library. Cases should be constructed from multiple resources, including past cases, and should meet requirements of the pre-specified CBR process (Carey, 1998; Kolodner & Guzdial, 2000; Kolodner, Owensby, & Guzdial, 2004). The construction process is a learning process for the contributor and it produces a useful product - a case - for use by other problem-solvers (Harel, 1991, 2000; Harel & Papert, 1990; Jonassen, Howland, Moore, & Marra, 2003; Kafai, 1996; Kolodner & Guzdial, 2000; Kolodner, Owensby, & Guzdial, 2004; Papert, 1980, 1993, 1996, 1998, 2000; Resnick, M., 1996; Resnick M., & Ocko, 1990).

Finally, a comprehensive CBR tool should include the capability for collecting cases from domain content experts. A defining tenet of CBR is that the novice practitioners of a domain can look to the skilled performance of experienced practitioners of the domain for guidance in resolving domain issues (Jonassen, Tessmer, & Hannum, 1999; Kolodner, 1993; Kolodner & Guzdial, 2000; Kolodner, Owensby, & Guzdial, 2004). Henceforth, these seven
characteristics of a comprehensive CBR tool will be referred to as the *seven design requirements*. They are listed in Table 4:

Table 4

*Seven Theory-, Research-based Design Requirements*

<table>
<thead>
<tr>
<th>Title</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Integrated model</td>
<td>Reflect a model of cognition that integrates reasoning, understanding and learning, and dynamic memory</td>
</tr>
<tr>
<td>2. Four major steps</td>
<td>Support performing four major steps/tasks of the CBR process</td>
</tr>
<tr>
<td>3. Ill-structured problem solving</td>
<td>Support the ill-structured problem-solving process</td>
</tr>
<tr>
<td>4. Two major parts</td>
<td>Consist of two primary parts: supports for reflection and interpretation of experience and a case library for storing those interpretations</td>
</tr>
<tr>
<td>5. Integration</td>
<td>A versatility for ready integration into a range of domains</td>
</tr>
<tr>
<td>6. Case construction and contribution</td>
<td>Support constructing cases and contributing them to the case library</td>
</tr>
<tr>
<td>7. Expert modeling</td>
<td>Support contribution of cases by domain experts</td>
</tr>
</tbody>
</table>
Chapter 3 – The Design Project

The Research Methodology

This chapter is devoted to answering the second research question by application of developmental research methodology. Research methodology can be generally classified as being basic or applied methodology (Pedhazur & Schmelkin, 1991). Basic methodology is concerned with a quest for knowledge per se, regardless of immediate practical value; applied, with a search for practical solutions. A type of applied methodology, developmental methodology seeks to understand, evaluate, and ultimately improve professional practice (Richey, Klein, & Nelson, 2004; Richey & Nelson, 1996). Application of developmental methodology in instructional technology can involve completing two basic phases of a research process: (a) systematically designing a needs-, theory-based educational product according to validated procedures of instructional design, and (b) abstracting lessons learned from the design experience (Richey, Klein, & Nelson, 2004; Richey & Nelson, 1996). Developmental research studies can validly and productively concentrate on particular aspects of a design project without following through to production (Richey, Klein, & Nelson, 2004; Richey & Nelson, 1996). This study will identify lessons learned from design, development, and evaluation (Richey, Klein, & Nelson, 2004; Richey & Nelson, 1996) of a comprehensive CBR tool for complex thinking. Scope of the design project can vary according to research questions (Richey, Klein, & Nelson, 2004; Richey & Nelson, 1996).

A Generic Design Process and a Design Model

The terms instructional development, instructional design, and instructional systems design have been used synonymously in the literature of instructional technology to denote the standard, systematic process of developing needs-, theory-based instruction and products (Akker, Branch, Gustafson, Nieveen, & Plomp, 1999; Gustafson & Branch, 1997, 2002; Molenda, 2003). In general, the process can include addressing issues related to need, design, development, and utilization and maintenance (Akker et al., 1999; Gustafson & Branch, 1997, 2002; Knirk & Gustafson, 1986; Richey, Klein, & Nelson, 2004; Richey & Nelson, 1996). Numerous instructional system development (ISD) models offer interpretations of and prescriptive procedures for implementing the generic process (Akker et al., 1999; Branch & Gustafson, 1997, 2002). The Dick, Carey, and Carey (2005) model as presented in the authors’ text The Systematic Design of Instruction was selected for this design project. This model consists of the integrated stages listed in Table 5:
Table 5

Stages of the Dick, Carey, and Carey Model

<table>
<thead>
<tr>
<th>Stage #1</th>
<th>Stage</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Analyze Needs</td>
</tr>
<tr>
<td>2</td>
<td>Identify Goals and Conduct Instructional Analysis</td>
</tr>
<tr>
<td>3</td>
<td>Analyze Learners and Contexts</td>
</tr>
<tr>
<td>4</td>
<td>Write Performance Objectives</td>
</tr>
<tr>
<td>5</td>
<td>Develop Assessment Instruments</td>
</tr>
<tr>
<td>6</td>
<td>Develop Instructional Strategy</td>
</tr>
<tr>
<td>7</td>
<td>Develop and Select Instructional Materials</td>
</tr>
<tr>
<td>8</td>
<td>Design and Conduct the First Formative Evaluation</td>
</tr>
</tbody>
</table>

While these stages are consecutive, necessary cyclical revisiting of stages and revision of stage outputs are allowed.

(Data obtained from: Dick, Carey, & Carey, 2005)

The standard systematic design process as presented in this model focuses on design, including a succession of increasingly refined prototypes, and stops short of production (Dick, Carey, & Carey, 2005).

This research study does not concern design of instruction, but, instead, design of a product or tool to support instructional, training, or workplace activities. Therefore, outputs for Stages 4-6 were not created. In lieu of the outputs, descriptions of various outputs supported by the tool are offered as a context for use of the tool. Stage 7 normally concerns creation of instructional materials (Levie & Dickie, 1973). In this design project, materials take the form of a prototype.

Five reasons favored selection of the Dick, Carey, and Carey (2005) model. First, the model has been validated through a record of successful instructional design projects conducted over the years (Dick, Carey, & Carey, 2005). Furthermore, the model is the “most widely cited instructional design model” and is the standard for comparison of instructional design models and alternative design approaches (Gustafson & Branch, 2002, p. 59). Moreover, the generic steps in the model are equally applicable to computer-based and digital multimedia-based systems as well as print-based instructional delivery systems (Dick, Carey, & Carey, 2005; Gustafson & Branch, 1997, 2002). In addition, the model has been viewed to be suitable not only for developing teacher-led instruction, but also educational products (Dick, Carey, & Carey, 2005), particularly in situations where product development requires less analysis of need (Gustafson & Branch, 1997, 2002). Finally, the model maps a potentially effective, valid integration of proven objectivist elements, for example, a given problem-solving method with promising
constructivist elements and application of that method to produce innovative, valid, assessable outcomes from multiple resources (Carey, 1998; Dick, Carey, & Carey, 2005; Ertmer & Newby, 1993).

The Jonassen-Tessmer (1996) taxonomy of advanced, real-world learning outcomes (ill-structured or situated problem-solving skills) and the Tessmer (1993) guide to the expert review type of formative evaluation were used to supplement the model. Stage-by-stage documentation of designing the comprehensive CBR tool follows. Specifications derived from each stage are numbered according to stage and sequence within the stage as illustrated in the following example:

Specification 1.1

Stage 1 – Analyze Needs

Analysis of specific needs is the first step of the standard, systematic instructional design process (Dick, Carey, & Carey, 2005). If an instructional intervention is warranted by the analysis, either a new instructional program or instructional support tool may be required. The Dick, Carey, and Carey (2005) instructional design model recommends a performance technology approach to conducting needs analysis. Needs are defined in terms of targeted learner performances (behaviors) that should be supported by interventions. The need is therefore said to be a learner performance need. Specifically, Dick, Carey, and Carey reduce their procedure to a simple subtractive formula: $\text{Should} - \text{Is} = \text{Need}$. Establishing the $\text{Should}$ factor of the formula involves identifying performances that learners should ideally be able to produce. Establishing the $\text{Is}$ factor involves assessing existing performances that have been achieved without benefit of proposed interventions. Existing performances, if relevant, constitute a subset of ideal performances. The gap separating the ideal performance level ($\text{Should}$) and the existing performance level ($\text{Is}$) represents a learner performance deficiency or $\text{Need}$. Proposed interventions should be designed to collectively target maintenance of relevant existing performance levels while reducing learner performance deficiencies.

The Dick, Carey, and Carey (2005) approach was applied to the findings of the literature review. In general terms, the literature review in Chapter 2 identified a need for (1) complex thinking skills and (2) supporting tools to acquire and enhance such skills. Furthermore, the literature review clearly indicated that the CBR method could be used to engage complex thinking skills and implied that the probability of successful implementation of the CBR method by multiple learners could be increased through the use of a comprehensive CBR tool.
The specific need for, and design of, a comprehensive CBR support tool for curricular integration and engaging complex thinking is the focus of the remainder of this research. The needs analysis determined that this instructional support tool should be integrated in instruction and be capable of promoting and supporting valid and reliable performance of complex thinking skills. The support of the learner's/user’s capability to perform complex thinking is likely to occur first in a curricular context and ultimately in the workplace. In addition, the tool should incorporate seven comprehensive design requirements that were identified in the literature review and summarized at the end of Chapter 2. Finally, the tool design should incorporate five facilitators of learning from concrete experience. The facilitators are reinforced by the design requirements. These design requirements and facilitators are recalled from the literature review, and listed and identified in the next section.

**Specification:**

1.1 The comprehensive CBR tool should support learner performance relative to valid and reliable engagement of complex thinking skills first in curricular and ultimately in workplace contexts.

In Table 6, the comprehensive design requirements are listed and identified first:

Table 6

<table>
<thead>
<tr>
<th>Design Requirements and Specifications</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Comprehensive Design Requirement</strong></td>
</tr>
<tr>
<td>1. Provide support for the integration of: (a) <em>reasoning</em> as a learning process, (b) <em>learning</em> and <em>understanding</em> as outputs of this learning process, and (c) <em>dynamic memory</em> as a structure for holding and integrating evolving content that represents the learning and understanding.</td>
</tr>
<tr>
<td>2. Provide support for validly and reliably performing the four top-level <em>tasks of the CBR method</em> (the four <em>REs</em> or REtrieve, REuse, REvise, and RETain) and their underpinning hierarchy of tasks and methods.</td>
</tr>
<tr>
<td>3. Provide support for application of the CBR method to interpreting and solving <em>ill-structured</em> type problems.</td>
</tr>
</tbody>
</table>
### Comprehensive Design Requirement

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>4. Combine <em>supports</em> for the learner/user reflection on and interpretation of concrete experience with the problem-solving resource of a <em>case library</em>, which results from these reflective and interpretative processes.</td>
<td>1.5 The comprehensive CBR tool should consist of basic two units: (a) supports for reflecting upon and interpreting experience, which can serve to produce new cases for an evolving case library, and (b) a case library.</td>
</tr>
<tr>
<td>5. Equip the tool with a versatility that enables <em>ready integration</em> of the tool into a range of problem/practice domains such as professional education, turfgrass management, and medicine.</td>
<td>1.6 Based on a domain-neutral representation of the CBR process but supporting ready creation and integration of domain-specific content – including instructive cases – the tool allows ready integration into diverse domains thus making the benefits of the CBR method available to a broad range of learners.</td>
</tr>
<tr>
<td>6. Allow and support learner/user construction and contribution of cases.</td>
<td>1.7 The comprehensive CBR tool should support learner/user performance relative to constructing cases and adding those cases to the case library.</td>
</tr>
<tr>
<td>7. Offer the capability for collecting cases that have been constructed and submitted by the content experts who practice in the problem/practice domain in which the comprehensive CBR tool has been integrated.</td>
<td>1.8 The comprehensive CBR tool should support collection of cases constructed and submitted by content experts who practice in the problem/practice domain in which the comprehensive CBR tool has been integrated.</td>
</tr>
</tbody>
</table>

Consistent with the comprehensive design requirements identified and listed in Table 6, the five facilitators (Kolodner & Guzdial, 2000; Kolodner, Owensby, & Guzdial, 2004) are identified and listed in Table 7:

#### Table 7

**Facilitators and Specifications**

<table>
<thead>
<tr>
<th>Facilitator</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Having instructive experiences and thereby learning lessons related to the issues that define a problem/practice domain</td>
<td>See relevant Specification 1.6: Based on a domain-neutral representation of the CBR process but supporting ready creation and integration of domain-specific content – including instructive cases – the tool allows ready integration into diverse domains thus making the benefits of the CBR method available to a broad range of learners.</td>
</tr>
<tr>
<td>2. Interpreting experiences for the following: (a) identification of lessons learned from concrete problem-solving experience: (b) creation of cases that capture experiences, interpretations, and lessons for use as a problem-based learning resource by others; and (c) transference or application of the cases to new situations</td>
<td>1.9 The comprehensive CBR tool should support learner/user performance relative to the task of interpreting and benefiting from experience.</td>
</tr>
<tr>
<td>3. Indexing cases (assigning filing descriptors to cases) to point to the cases’ similarity, consequent relevance, and potential applicability to resolving various newly-arising problem situations</td>
<td>1.10 The comprehensive CBR tool should support learner/user performance relative to the task of indexing cases or resolving the <em>indexing problem</em>.</td>
</tr>
</tbody>
</table>
Operationalization of this needs analysis must now begin with the identification, classification, and analysis of goals in Stage 2.

**Stage 2 – Identify, Analyze, and Classify Goals**

The overall goal of this research study is to design a comprehensive CBR tool for support of complex thinking. The tool should be consistent with the general problem solving process, as well as the seven comprehensive design requirements discussed in Stage 1. This tool should provide a means for current users of the tool to practice the case-based reasoning process by solving problems and learning from experience. To be consistent with the seven design requirements, the tool also should provide a means for current users of the tool to capture their problem-solving experiences in the case-based format. Finally, the tool should provide a means for future users of the tool to access case information provided by previous users of the tool and by outside sources. Defining features of the tool include: (a) opportunities to solve problems using the case-based method and (b) support for effectively transferring the lessons of past experience to new problem-solving situations.

**Specification:**

2.1 The tool supports an instructional goal to promote complex thinking through performance of the case-based reasoning process. *(See also related Specification 1.1: The comprehensive CBR tool should support learner performance relative to valid and reliable engagement of complex thinking skills first in curricular and ultimately in workplace contexts.)*

"The central point of education is to teach people to think, to use their rational powers, to become better problem solvers" (Gagne, 1985, p. 85). General problem solving consists of two phases (Jonassen, 1997). The first phase involves processes of *understanding*, and the latter involves processes of *searching*. In the first phase, an
unsolved problem is analyzed, described, and explained. In the second phase, potential solutions are explored, identified, and ranked. Suggested solutions are developed and implemented. In general, they are iteratively evaluated and repaired until a suggested solution proves successful and becomes a confirmed solution. At times, however, this iterative evaluation and repair cycle never produces a confirmed solution. Such a final outcome is allowed because a well-executed but “unsuccessful” problem-solving attempt can potentially stimulate even greater learning than a successful attempt.

Problems can vary structurally along a continuum ranging from problems that are more well-structured, more clearly defined, to those that are more ill-structured, more poorly defined (Jonassen, 1997). The more poorly defined structure is thought to make ill-structured problems the more intellectually challenging of the two types, and the ill-structured problem-solving process more cognitively rigorous (Ertmer & Newby, 1993; Gagne, 1985; Jonassen & Tessmer, 1996). Although some researchers have posited that different structural types require different problem-solving skills (Jonassen, 1997; Jonassen & Tessmer, 1996; Shin, Jonassen, & McGhee, 2003), little empirical evidence has been collected thus far to support a fine distinction of skill sets. (Shin, Jonassen, & McGee, 2003). Therefore, a general problem-solving process as described in the preceding paragraph – applicable to both well-structured and ill-structured problem types – will be emphasized in this research study.

An integral part of the instructional analysis stage of the Dick, Carey, and Carey (2005) model is classification of goals by learning domain. Classification helps assure alignment of goals, objectives, assessments, and strategy. Taxonomies of learning are used to perform classification. The goal stated at the beginning of Stage 2 (to design a comprehensive CBR tool for complex thinking) can be classified as an intellectual skill. This classification will be refined and extended later in Stage 2 to reflect the specific intellectual skills involved in achieving the goal. A goal analysis has been performed and will serve as basis for that refinement and extension. Because this design project focuses on the development of an instructional support tool, it is necessary to define the relationship between that tool and the learner performance it is intended to support. To do this, it is necessary to identify the steps involved in achieving the overall goal of problem solving as a stimulus of complex thinking. It is also necessary to address the requirements imposed by a review of the literature and by the several analyses that are key features of the Dick, Carey, and Carey (2005) model. These various requirements are addressed first in a
flowchart analysis of the general problem-solving process followed by an analysis of the case-based reasoning process as an extension of that general process.

**Specification:**

2.2 The tool supports performance of highest-order skills (“intellectual skills”) integral to the case-based reasoning process, according to four standard taxonomies of learning.

(See the classification of the goal steps/sub-steps according to these four taxonomies in Table 8.)

Figure 7 presents a flowchart analysis of the general problem-solving process as defined in this paper. Standard flowchart symbols have been used in the figure: ovals to show start and end of the problem-solving process; rectangles, steps of the process; diamonds, decisions that must be made to advance the process; and arrows, direction of the process (Dick, Carey, & Carey, 2005):
Figure 7. A flowchart analysis: The general problem-solving process.
This flowchart has a central core and peripheral elements surrounding that core. The six basic steps that constitute the core of the general problem-solving process also hold the central location and encapsulate the central activity of the chart. These steps are intended to be followed successively through completion of the six-step problem-solving process and are represented by vertically distributed, numbered rectangles. The activity occurring in each step is expressed by an observable behavior in a format similar to a behavioral objective. Steps 1-2 relate to understanding the problem – defining, describing, and explaining it. Steps 3-6 relate to searching for a solution to the problem - designing, developing, implementing, and evaluating a solution.

The ancillary, iterative operations found on the far left and bottom/right sides of the centrally-located steps are represented by unnumbered diamonds and rectangles. The yes/no decision-making questions that drive execution of these operations are strategically presented in the diamonds. The operation found on the far left side of the chart involves making a yes/no decision as to sufficiency of internal memory for performing any one or all of the six problem-solving steps: is internal memory sufficient?. Internal memory is personal memory. If internal memory is not sufficient to perform the step, external memory must be used. External memory represents problem-solving resources and data that lie beyond and supplement internal memory.

The operations found to the bottom/right side of the centrally-located steps are executed according to the outcome of completing the problem-solving steps and process. Three yes/no decision-making questions must be acted on: (a) was problem solved?, (b) make another attempt?, and (c) revise step output?. The question in the first diamond – was problem solved? – is answered on the basis of evaluating the solution in step 6. The question in the second diamond – "make another attempt?" – allows attempting to revise a failed attempt or exiting the problem-solving process without a solution. This exit option is allowed because some ill-structured problems may have no solution at all (Jonassen, 1997). While achieving a solution is certainly preferred and beneficial, it is not essential for productive problem-based learning. In fact, a well-executed but unresolved problem-solving project could theoretically produce more learning than a resolved project. The No option in the second diamond allows the latter benefit to be realized. The third diamond – "revise step output?" – allows the problem solver to holistically and/or incrementally (stepwise) revisit the project and revise the failed solution. In the event of a failed solution, revision can include selecting an alternative feasible solution in Step 3 for implementation and evaluation.
The general problem-solving process has been discussed and graphically analyzed as a context for, and precursor to, discussion and analysis of the CBR process. The flowchart analysis of the CBR process in Figure 8 consists of four integrated levels. Level 1 begins with analyzing and interpreting the new problem to be solved. After reviewing the facts associated with the new problem, the learner uses problem descriptors to begin the search for past cases of similar problem-solving episodes held in internal (personal) and/or external (resource-based) memory for potential application to solving the problem. This iterative search and exploration process ultimately reduces a large, initial set of potentially similar cases to increasingly smaller subsets of more similar and eventually most similar cases. The contents of the most similar cases are retrieved for possible use in the remainder of the CBR problem-solving process.

At the beginning of Level 2 activities, the contents of the retrieved case(s) are closely compared and contrasted with the new, unsolved problem to closely examine relative similarities and/or dissimilarities between retrieved case(s) and the new problem and to determine their potential value in solving the New Case. Three decision-making options for processing this similarity assessment follow at Level 2: (a) reuse cases by copying them to produce a suggested solution as allowed by relevant similarities and no relevant dissimilarities (trivial reuse), (b) adapting them as required by relevant dissimilarities (non-trivial reuse), or, (c) given ultimately non-adaptable case(s), replacing the non-adaptable with adaptable case(s) by revisiting the iterative, reductive search process. Final activities of Level 2 involve creating a suggested, yet unevaluated solution from copied and/or adapted elements, and predicting outcome of that solution.

At the beginning of Level 3 activities, the suggested, yet unevaluated solution of the tentatively solved case from Level 2 is implemented and finally evaluated for one of two possible results: a successful solution or a failed solution. If the solution is successful, the process immediately advances to Level 4. Alternatively, a failed solution is followed by either a re-attempt or an aborted attempt. A re-attempt includes deciding which problem-solving outputs require revisiting and repair. These decision-making options (similar to options in the general problem-solving process) can ultimately allow the case to proceed with resolution and benefits or without resolution but with benefits. At Level 3, the suggested, yet unevaluated solution from Level 2 has become a tested/confirmed solution, or the problem-solving episode is on its way to becoming an unresolved but informative experience.
Level 4 activities begin with reflecting on the problem-solving episode from Levels 1-3 and subsequently extracting potentially useful knowledge for eventual integration in the external, evolving knowledge base memory. The extraction process culminates with creating a representation of the new knowledge. It includes deciding to opt for a new case or for extension or generalization of an existing case as the more appropriate alternative for integration of new knowledge. Indexing the new knowledge for future similarity assessment and retrieval is also addressed here. Indexing involves the use of existing indexes or revising indexes per need. This revision process includes evaluating new indexes and/or deciding to revise ineffective ones.

One final observation regarding the CBR flowchart is useful. As mentioned earlier, creating support for complex thinking is the focus of this research study. Support can include three types of challenging stimuli intended to execute complex thinking: (a) solving a problem, (b) making a decision, or (c) designing an artifact (Iowa Department of Education, 1989; Iowa State University, 1999; Jakovljevic, Ankiewicz, & de Swardt, 2005; Jonassen, 2001; Slangen & Sloep, 2005). The flowchart analysis of CBR in Figure 8 depicts CBR as a method for solving problems and learning from experience (Aamodt & Plaza, 1994; Kolodner, 1993; Kolodner & Leake, 1996; Leake, 1996a, b). As such, CBR includes all three types of recommended stimuli in form of: (a) flowchart rectangles for steps of the case-based reasoning or problem-solving process, (b) diamonds for decision-making to advance that process, and (c) Level 4 steps and decision-making for creation of an artifact and a resource.

**Specification:**

2.3 The tool supports CBR as an extension of the general problem-solving process.
STAGE 1
Interpreting the Problem

START

Analyze the new problem that is to be solved.

Infer problem descriptors (for defining problem attributes) from the analysis

Explore the knowledge base memories of indexed cases and general knowledge for cases and general knowledge similar to the problem descriptors from Step ?, and therefore potentially relevant to solving the problem.

Is internal memory sufficient for performing the step?

Yes

Explore and apply internal memory as appropriate, and proceed.

No

Explore, apply, and integrate external memory as appropriate, and proceed

START
STAGE 1
(Continued)

Are cases found?

No

Revise descriptors?

Yes

May most closely matching cases have been found?

No

Are initial matches found?

Yes

Rate each match between problem descriptors and the retrieved cases and general knowledge from Step ?.

Rank the retrieved cases and general on basis of degree of similarity to problem descriptors.

Select the retrieved cases and general knowledge that are possibly most similar to problem descriptors.

Yes

Initially match the problem descriptors to sufficiently similar cases and general knowledge.

Are cases found?

Yes

Are initial matches found?

No

Revise descriptors?

Yes

Are cases found?

No
STAGE 2
Solving the Problem

Systematically compare the new case problem with the previous case problems to establish exact degree of similarity between new and previous cases.

Systematically contrast the new case problem with the previous case problems to establish exact degree of dissimilarity between new and previous cases.

Yes

Are similarities relevant with no relevant dissimilarities?

Yes

As allowed by relevant similarities and irrelevant dissimilarities between the new case and the previous cases, copy the solution(s) and/or solution method(s) from previous case(s) to the new case, a process which represents trivial reuse of a previous case solution and/or solution method.

Explore potentially feasible solution(s).

Rank solution(s).

Select top-ranking solution for development, implementation, and evaluation.

Develop selected, suggested, yet unevaluated solution.

Predict the outcome of the solution.

No

Were adaptable elements found?

Yes

No

Replace retrieved cases?

No

Yes

As required by relevant dissimilarity(ies) between the new case and the previous cases, attempt to adapt the solution(s) and/or solution method(s) from previous case(s) to the new case, a process which represents nontrivial reuse of a previous case solution and/or solution method.

No

Explore potentially feasible solution(s).

Rank solution(s).

Select top-ranking solution for development, implementation, and evaluation.

Develop selected, suggested, yet unevaluated solution.

Predict the outcome of the solution.
STAGE 3
Evaluating the Solution

Did solution/solution method produce predicted outcome?

Analyze variance between predicted outcome and actual outcome as to nature and explanation of variance.

Make another attempt?

Attempt by reconsidering other potentially feasible solutions or by repairing step output(s)?

Repairing

Reconsidering

Yes

No

Implement the solution.

Yes

No
STAGE 4
Abstracting Lessons Learned

Review the knowledge learned from the new problem-solving episode in Levels 1-3 for potentially useful knowledge that can serve as a problem-solving and learning resource held in external memory.

Analyze the knowledge.

Evaluate the potential usefulness of the knowledge.

Extract potentially useful knowledge.

Assemble the potentially useful knowledge for eventual integration in external memory.

Construct a representation of the potentially useful knowledge.
Integrate the new knowledge in a related, existing case, resulting in an extension of that case.

Which case is the more appropriate means for holding and presenting the new knowledge?

Existing

Copy the representation of new knowledge to a new case.

New
Are current indexes adequate for representing evolution of knowledge base memory resulting from integration of new knowledge?

Revise (tune) current indexes to represent evolution in knowledge base memory resulting from integration of new knowledge.

Assign indexes to new knowledge.

END

Figure 8. A flowchart analysis: The case-based reasoning (CBR) process.
Specification:

2.4 The tool supports performance of steps and sub-steps comprising the case-based reasoning process. (See also related Specification 1.3: The comprehensive CBR tool should support learner performance relative to completing and/or implementing the four theory-based, research-based, top-level tasks of the CBR method, and their underpinning hierarchy of sub-tasks and methods.)

The major steps and sub-steps of the CBR process were analyzed in the flowchart above. In addition to this analysis, instructional analyses normally include an analysis of the skills and knowledge underlying the performance of major steps and sub-steps (Dick, Carey, & Carey, 2005). These skills and knowledge are often referred to as subordinate skills and knowledge. Learners beginning the CBR process may already possess some of these skills and knowledge while others must be acquired through instruction and activities in order to perform the major steps and sub-steps. The subordinate skills and knowledge that must be acquired should be supported by the new instruction or instructional support tool. The remaining skills that should be possessed by all learners before beginning the CBR process are entry behaviors or prerequisite skills and are not supported by the new instruction or instructional support tool.

Two representative analyses separately showing the hierarchy of subordinate skills for one problem-solving step and for one problem-solving decision from the flowchart are presented in Figures 9-10.
Figure 9. A flowchart analysis: Representative subordinate skills for problem-solving steps.

1. Identify the two systematic processes of comparing and contrasting.
2. State the purpose of each process.
3. Describe each process:
   - Comparing: Input = Relevant criteria for comparing two or more objects, and output = identification of similar properties per criteria
   - Contrasting: Input = Relevant criteria for contrasting, and output = identification of dissimilar properties per criteria
4. Demonstrate comparing and contrasting.
5. Identify tools for comparing and contrasting, including the options of: (a) Venn diagram, (b) a T-chart, and (c) a compare-and-contrast matrix.
7. Describe the application of each tool.
8. Demonstrate the application of each tool.

Systematically compare the new case problem with the previous case problems to establish exact degree of similarity between new and previous cases.

Systematically contrast the new case problem with the previous case problems to establish exact degree of dissimilarity between new and previous cases.

Systematically compare the new case problem with the previous case problems to establish exact degree of similarity between new and previous cases.

Systematically contrast the new case problem with the previous case problems to establish exact degree of dissimilarity between new and previous cases.
Which case is the more appropriate means for holding and presenting the new knowledge?

4. Demonstrate the two types of integration as follows:
   
   IF recently created knowledge = existing knowledge, THEN assimilate recently created knowledge within an existing structure
   
   or
   
   IF recently created knowledge ≠ existing knowledge, THEN accommodate recently created knowledge within an extended, existing structure or within a newly created structure

3. Describe each type (how assimilation involves using recently created knowledge to reinforce an existing structure; and how accommodation involves using recently created knowledge to extend an existing structure or to create a new structure).

2. State the purpose of each type.

1. Identify two types of integration as follows: (a) assimilation and (b) accommodation.

Figure 10. A flowchart analysis: Representative subordinate skills for a problem-solving decision.
A fully populated, comprehensive CBR tool should provide comparable information for each step and decision point within the CBR process. Conducting a comprehensive subordinate skill analysis is beyond the scope of this research study. However, this type of information can be generated later and the proposed tool can be designed to enable incremental integration as needed.

**Specification:**

2.5 The tool supports subordinate skills relative to the steps and sub-steps of the CBR process.

In addition to breaking goals down into steps, instructional analysis includes classification of goals by learning domain (Dick, Carey, & Carey, 2005). Classification was introduced earlier in this Stage 2 discussion. Various learning taxonomies can be used to classify goals. Taxonomies by Bloom (Bloom, Englehart, Furst, Hill, & Krathwohl, 1956), Bloom revisionists (Anderson, 1999; Anderson & Krathwohl, 2001), Gagne (1985), and Jonassen and Tessmer (1996) will be used here. Because a comprehensive CBR tool should support complex thinking, it is important to identify the major cognitive skills and knowledge that should be supported by such a tool. As seen in Table 8, the primary activities of the CBR process rank at the higher levels of the cognitive domain in all four taxonomies:
Table 8

Classification of Goal Steps/Sub-steps According to Four Learning Taxonomies

<table>
<thead>
<tr>
<th>Taxonomy</th>
<th>Description</th>
<th>Major cognitive categories within taxonomy (in generally ascending order of difficulty as relevant)</th>
<th>Major cognitive categories relevant to CBR (in generally descending order of difficulty as relevant)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bloom – Original</td>
<td>A standard taxonomy of learning, including the psychomotor, affective, and cognitive domains with a focus on the cognitive domain in this research study</td>
<td>Knowledge, Comprehension, Application, Analysis, Synthesis, Evaluation</td>
<td>6. Evaluation: Judging value 5. Synthesis: Combining existing elements to create an original artifact 4. Analysis: Separating into parts for understanding relationships and organizational pattern</td>
</tr>
<tr>
<td>Bloom – Revised</td>
<td>A standard taxonomy of learning, including the psychomotor, affective, and cognitive domains with a focus on the cognitive domain in this research study</td>
<td>Remembering, Understanding, Applying, Analyzing, Evaluating, Creating</td>
<td>6. Creating: Comparable to Synthesis in original Bloom 5. Evaluating: Comparable to Evaluate in original Bloom 4. Analyzing: Comparable to Analysis in original Bloom</td>
</tr>
<tr>
<td>Gagne</td>
<td>A standard taxonomy of learning, including the psychomotor, affective, and cognitive domains with a focus on the cognitive domain in this research study</td>
<td>Verbal Information, Cognitive Strategy, Intellectual Skills</td>
<td>Intellectual Skills focusing on highest sub-class of this class: Problem solving, including well-structured problem solving and the higher-level ill-structured problem solving</td>
</tr>
</tbody>
</table>
Taxonomy | Description | Major cognitive categories within taxonomy (in generally ascending order of difficulty as relevant) | Major cognitive categories relevant to CBR (in generally descending order of difficulty as relevant)
--- | --- | --- | ---
Jonassen and Tessmer$^4$ | A taxonomy intended to extend standard taxonomies by focusing on and more finely delineating classes of higher-order learning produced by real-world practice | Inclusion of Situated (Ill-Structured) Problem-Solving Skills as the most authentic kind of instruction in the taxonomy, the means for acquiring advanced knowledge, and “one of the most important learning outcomes that can ever be supported by instruction because [it] is what people get paid for in the real world;” other classes of skills include structural knowledge, executive control, ampliative strategies, and self-knowledge | Situated (Ill-Structured) Problem-Solving Skills involving defining and decomposing real-world problem, hypothesizing alternative solutions, and evaluating viability of solutions

1Bloom et al., 1956  
2Anderson, 1999; Anderson & Krathwohl, 2001  
3Gagne, 1985  
4Jonassen & Tessmer, 1996

Specification:

*See relevant Specification 2.2: The tool supports performance of highest-order skills ("intellectual skills") integral to the case-based reasoning process, according to four standard taxonomies of learning.*

Stage 3 – Analyze Learners and Contexts

The comprehensive CBR tool for complex thinking should be integrated in an instructional or training program. A well-designed instructional or training program constitutes a system of components that have been integrated to achieve the purpose of that system (Dick, Carey, & Carey, 2005). Two key components of any instructional or training system are (a) *learners* and (b) *contexts* (Dick, Carey, & Carey, 2005). Contexts include *performance* and *learning contexts*. General aspects of learners and contexts for the CBR tool will be discussed in the following sections. Specifications resulting from the analysis of learners and contexts will also be presented.
Specifications:

3.1 The CBR tool should be integrated in an instructional, training, and workplace management programs across a range of domains. (See also related Specification 1.6: Based on a domain-neutral representation of the CBR process but supporting ready creation and integration of domain-specific content – including instructive cases – the tool allows ready integration into diverse domains thus making the benefits of the CBR method available to a broad range of learners.)

3.2 The tool should be compatible with the learner and context components addressed by the instructional or training system.

Who are the learners using the tool? Theoretically, the CBR tool could be beneficially used by various potential groups of learners. However, for purposes of this research study, it was practically necessary to focus on use of the tool by one particular subset of potential learners. Restriction to this subset can produce informative findings that serve to advance the theory base of instructional technology. Findings can also form the basis for conducting further research involving additional potential groups (as proposed later in this paper). The subset of learners is defined here on the basis of the learner level of educational preparation and experience and consequent level of proficiency. This preparation and experience serve as qualification for applying the tool to cognitively rigorous learning activities.

The subset of potential learners using the tool rank at the advanced level on the continuum of experience and proficiency discussed in Chapter 2. That continuum begins with the pre-novice level and advances to and ends with the expert level (Alexander, 1997; Benner, 1984; Ertmer & Newby, 1993; Jonassen, Mayes, & McAleese, 1993; Ormrod, 1999; Spiro et al., 1987; Spiro et al., 1988; Spiro et al. 1992a; Spiro et al., 1992b). The advanced level is above the novice level on the continuum and the expert level; that is, advanced learners are post-novice/pre-expert learners. Advanced-level learners have been previously engaged in building a foundation of educational preparation and clinical and/or real-world experience in the professional domain of study. As a result, their knowledge has been increasing in quantity and quality. Higher quality knowledge is less fragmented and more integrated. It is more spontaneously recalled and applied to resolving novel situations (Perkins & Salomon, 1992; Salomon & Perkins, 1988). In addition, these learners over time have also been building more effective study strategies. Finally, over time, they have come to demonstrate a professional commitment to the domain. However,
these learners generally fall short of expert-level knowledge, strategies, and commitment. Advanced learners are most consistently found in higher education (Jonassen, Mayes, & McAleese, 1993).

**Specification:**

3.3 The tool should support advanced learners in the application of CBR. *(See also related Specification 2.2: The tool supports performance of highest-order skills [“intellectual skills”] integral to the case-based reasoning process, according to four standard taxonomies of learning.)*

One of the seven comprehensive design requirements was capability for addressing ill-structured issues. The tool will be dedicated to resolving ill-structured, challenging problems that define the issues of a professional domain, and thereby producing advanced understanding of those issues and initiating best practice. Best practice is "an activity or procedure that has produced outstanding results in another situation and could be adapted to improve effectiveness, efficiency, ecology, and/or innovativeness in another situation" (ICH, 2003). Advanced learners are able to productively use the tool because they have the foundation of educational preparation and clinical and/or real-world experience for applying CBR-supported complex thinking to resolving the ill-structured issues (Ertmer & Newby, 1993; Jonassen, Mayes, & McAleese, 1993).

The tool could be broadly applied to solving both well- and ill-structured problems that define the issues of a domain. *(The distinction between these two basic types of problems was discussed earlier in this chapter.)* However, the potential of the tool for engaging complex thinking and for producing advanced understanding is realized through application of the tool to the ill-structured type of problem solving.

**Specification:**

3.4 The tool should support advanced learners in the application of CBR to ill-structured problem solving as a means for engaging complex thinking and producing advanced understanding. *(See also related Specification 2.2: The tool supports performance of highest-order skills [“intellectual skills”] integral to the case-based reasoning process, according to four standard taxonomies of learning.)*

*What contexts are involved in using the tool?* Contexts involved in using the tool are performance and learning contexts (Dick, Carey, & Carey, 2005). The performance context is the setting where newly learned
knowledge and skills will be ultimately used. The workplace generally represents the performance context. The learning context is the formal educational or training setting where knowledge and skills are learned. The classroom generally represents the learning context. The performance context is listed and defined first here because the actual application of newly learned knowledge and skills in the performance context is the fundamental purpose and test of all the teaching and learning activity that occurs in the “classroom” (Ormrod, 1999).

Reliable application of newly learned knowledge and skills in the performance context can be increased by “reconnecting” the performance context with the learning context (Ertmer & Quinn, 2003; Perkins, 1995). Performance contexts and traditional “classroom” contexts are viewed as being disconnected. One means of “reconnecting” performance and learning contexts is to analyze characteristics of the performance context and then using analysis data to making the learning context more realistic (Collins, Brown, & Newman, 1989) or more similar to the performance context. The Dick, Carey, and Carey (2005) instructional system development model speaks to this issue of reconnecting contexts and identifies instructional relevance, learner motivation, and transfer (Perkins & Salomon, 1992; Salomon & Perkins, 1988) as benefits:

Accurate analysis of the performance context should enable the designer to develop a more authentic learning experience, thereby enhancing the learners' motivation, sense of instructional relevance, and transfer of new knowledge and skills to the work setting (Dick, Carey, & Carey, 2005, pp. 103-104).

A practice field (Barab & Duffy, 2000) learning context by definition supports the type of “more authentic learning experience” (Dick, Carey, & Carey, 2005, p. 103) recommended by the Dick, Carey, and Carey model. The relative degrees of realism characterizing the range of standard learning contexts, including the practice field, can be graphically seen by placing types of learning contexts ranging from least realistic/least concrete to most realistic/most concrete contexts as seen in Figure 11:
The practice field learning context is more realistic than the classroom learning context but less realistic than the community of practice context. The community of practice context integrates performance and learning contexts. Despite certain pedagogical benefits offered by the integrated, community of practice context, logistical problems can beleaguer this integration considering present-day organization of the educational system (Ormrod, 1999). The practice field serves as a useful, feasible compromise between the least realistic and most realistic learning contexts and should be used as the context for learning and performance occurring within the CBR tool.

**Specification:**

3.5 The CBR tool should be integrated in a practice field learning context.

*Interaction of learners with performance/learning contexts.* The performance context of any professional domain routinely presents ill-structured, challenging issues to practitioners for resolution. These issues represent opportunities for achieving best practice. Applying CBR-supported complex thinking to resolution of these issues produces advanced understanding of domain issues. Advanced understanding can serve to underpin and initiate recommendations for best practice. The National Council for Accreditation of Teacher Education (2002) requires a reflective type process for achieving best practice in professional education. Advanced learners and soon-to-be practitioners need to acquire knowledge and skills for applying CBR-supported complex thinking in the performance context of their domain. When successfully employed, a CBR approach promotes acquisition of these
skills in a practice field learning context. In such a case, instruction will be relevant, learners will be motivated, and knowledge and skills are likely to be applied in the performance context.

**Specification:**

3.6 The CBR tool should enable advanced learners and soon-to-be practitioners studying issues of professional practice to acquire realistic experience in applying CBR-supported complex thinking to resolving ill-structured issues that define the practice domain. *(See also related Specification 3.1: The CBR tool should be integrated in an instructional, training, and workplace management programs across a range of domains.)*

Advanced learners preparing for practice in a range of professional domains should be able to productively use the comprehensive CBR tool. Use of the tool should revolve around the valid and reliable application of a general, reflective problem-based method of learning called CBR. CBR is different from other methods in looking to a case library of specific, concrete problem-solving episodes as the fundamental problem-solving resource. This general method has been demonstrated to be useful to the problem-based learning needs of a range of professional domains, including professional education, architecture, and medicine (Colaric & Jonassen, 2000; Colaric, Turgeon, & Jonassen, 2003; Jonassen, Tessmer, & Hannum, 1999; Kolodner & Guzdial, 2000; Kolodner, Owensby, & Guzdial, 2004; Wang et al., 2003). CBR tools dedicated to addressing the issues of particular professional domains have been successfully developed and implemented. However, one of the seven comprehensive design requirements calls for creation of a tool that offers ready adaptability to and integration into a range of domains. This adaptability is based on a generalized representation of the CBR process and includes the capability of customizing and extending the tool to better meet domain needs. This ready adaptability offers the professional benefits of CBR to a range of professional domains.

**Specification:**

*See relevant Specification 3.1: The CBR tool should be integrated in an instructional, training, and workplace management programs across a range of domains.*

Analyses of needs, goals, learners, and contexts for the comprehensive CBR tool have been presented in Stages 1-3. The needs analysis revealed seven comprehensive design requirements for the tool. Operationalization of
these requirements began in the goal, learner, and context analyses. Performance objectives now need to be written in Stage 4 to further operationalize the seven comprehensive design requirements.

**Stages 4-6 – Write Performance Objectives, Develop Assessment Instruments, and Develop an Instructional Strategy**

Activities carried out in Stages 4, 5, and 6 respectively produce the outputs of performance objectives, assessment instruments, and an instructional strategy (Dick, Carey, & Carey, 2005). Front end analyses from Stages 1-3 of the design process – Stage 1 needs analysis, Stage 2 instructional analysis, and Stage 3 learner/context analysis – serve as the foundation for producing these Stage 4-6 outputs (Dick, Carey, & Carey, 2005), as depicted in Figure 12:

![Diagram](image-url)

**Figure 12.** Relationship between Stage 1-3 outputs (front-end analysis) and Stage 4-6 outputs.

(Data obtained from: Dick, Carey, & Carey, 2005)
Preparing objectives has been traditionally judged to play a critical role in the development of effective instruction (Mager, 1962, 1975, 1997). From this perspective, the core of a typical objective is a specification of the \textit{behavior} to be learned and performed as identified in the Stage 2 goal analysis (Dick, Carey, & Carey, 2005; Mager, 1962, 1975, 1997). The performance situation or \textit{condition} specification, and an acceptable performance \textit{criterion} are added to this core element (Dick, Carey, & Carey, 2005; Mager, 1962, 1975, 1997). The condition specification can include antecedent cue or stimuli, needed resource materials, and scope and complexity of the task (Dick, Carey, & Carey, 2005; Mager, 1962, 1975, 1997). The criterion specification can require a single, correct response, or can allow multiple, feasible responses, depending on intent of the behavior (Dick, Carey, & Carey, 2005; Mager, 1962, 1975, 1997). While agreeing with the need for clearly defined goals, steps, and sub-steps, more recent views hold that efficient, effective project management practices might dictate re-allocating time otherwise spent on creating a comprehensive list of detailed behavioral objectives to more useful pursuits (Cennamo & Kalk, 2005). According to these views, the existing, more general outcomes and subordinate skills from the Stage 2 goal analysis can suffice for meeting the requirement of aligning instructional intent with assessment (Cennamo & Kalk, 2005) without having to perform the intervening step of creating a comprehensive list of behavioral objectives.

Regardless of one’s position with respect to the use of objectives, there is little dispute relative to the need for assessment. If objectives are used, there should be a direct connection between writing objectives in Stage 4 and developing the assessment in Stage 5 (Cennamo & Kalk, 2005; Dick, Carey, & Carey, 2005). Although objectives and assessments have different purposes, they speak to the same behaviors, criteria, and conditions. Objectives can be thought of as specifications for the assessment (Cennamo & Kalk, 2005), or, in some situations, the assessment can be created first and, if desired, used to produce objectives (Cennamo & Kalk, 2005). If objectives are where we are trying to go, then the assessment tells us when we have reached that destination (Cennamo & Kalk, 2005). In other words, the assessment provides feedback on learner achievement and, consequently, feedback on instructional effectiveness (Dick, Carey, & Carey, 2005).

Assessments can address different purposes and assume various forms (Dick, Carey, & Carey, 2005). The purpose can vary from intermediate determination of progress toward achievement of objectives to final determination of achievement. The purpose may relate to measuring knowledge or skills prior to, and following, instruction. Pre- and post- assessments are similar except pre-assessments can include items for assessing entry
behavior skills. As appropriate, alternative types of assessments can supplement or replace traditional, paper and pencil assessments for assessing product and process goals of complex intellectual skills as they are applied in enacting performances and creation of products (Chicago Board of Education, 2004; Dick, Carey, & Carey, 2005; Marzano, Pickering, & McTighe, 1993).

The instructional strategy is developed in Stage 6 (Dick, Carey, & Carey, 2005). While objectives with the accompanying assessments comprise the what, or object of instruction, strategy comprises the how or method (Dick, Carey, & Carey, 2005). If learners are unprepared initially to achieve objectives or perform acceptably on assessments, a plan is required to prepare the learners to achieve the desired performance. That plan, with its various levels, stages, and pieces, constitutes the instructional strategy (Dick, Carey, & Carey, 2005).

Two levels of strategies are macro-strategy or total strategy, and the multiple, interacting micro-strategies that comprise the macro-strategies (Dick, Carey, & Carey, 2005). Generally, the macro-strategy should be developed before micro-strategies can be considered. A typical macro-strategy plan consists of five components: pre-instructional activities, content presentation, participation, assessment, and follow-through activities (Dick, Carey, & Carey, 2005). The organization of this plan is depicted in Table 9:

Table 9

Organization of a Standard Macro Instructional Strategy

<table>
<thead>
<tr>
<th>Major Component</th>
<th>Sub-Components</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Pre-instructional activities</td>
<td>1.1 Motivation</td>
</tr>
<tr>
<td></td>
<td>1.2 Objectives</td>
</tr>
<tr>
<td></td>
<td>1.3 Entry behaviors or recall prerequisites</td>
</tr>
<tr>
<td>2. Content presentation</td>
<td>2.1 Instructional sequence</td>
</tr>
<tr>
<td></td>
<td>2.2 Content</td>
</tr>
<tr>
<td></td>
<td>2.3 Examples</td>
</tr>
<tr>
<td>3. Participation</td>
<td>3.1 Practice</td>
</tr>
<tr>
<td></td>
<td>3.2 Feedback</td>
</tr>
<tr>
<td>4. Assessment</td>
<td>4.1 Pre-assessment</td>
</tr>
<tr>
<td></td>
<td>4.2 Post-assessment</td>
</tr>
<tr>
<td>5. Follow-through activities</td>
<td>5.1 Remediation</td>
</tr>
<tr>
<td></td>
<td>5.2 Enrichment</td>
</tr>
<tr>
<td></td>
<td>5.3 Memorization and transfer</td>
</tr>
</tbody>
</table>

(Data obtained from: Dick, Carey, & Carey, 2005)
Micro-strategies that could be included in these components are activities, group discussions, case studies, simulations, and projects (Dick et al. 2005).

More specifically, a problem-solving macro-strategy is particularly relevant to designing the CBR tool. The CBR method has been presented and analyzed in Stage 2 as an instance and extension of the general problem-solving method. Therefore, a problem-solving macro-strategy dedicated to achieving the goal of solving a problem would be particularly relevant implementing the CBR method in the features of the CBR tool. Jonassen (1999) has proposed a problem-solving “macro-strategy” that includes eight “micro-strategies” as presented in Figure 13:

![Diagram](image)

**Figure 13.** An option for strategy: A macro-strategy and eight supportive micro-strategies.

(Data obtained from: Jonassen, 1999)

Factors determining appropriateness of strategies at either the macro or micro level include characteristics of learning outcomes, learners, and performance and learning contexts (Dick, Carey, & Carey, 2005; Ertmer & Newby, 1993; Jonassen, Mayes, McAleese, 1993; Perkins, 1999).

Although Stage 4-6 outputs are integral to the instructional design process (Dick, Carey, & Carey, 2005), this research study does not include producing them per se. This study instead focuses on designing a
comprehensive CBR tool relevant to the different objectives, assessments, and instructional strategies found in a range of instructional, training, and workplace contexts. In effect, the versatility needed to support these different objectives, assessments, and strategies suggests specifications for design of the tool, akin to the specifications derived from the Stage 1 needs analysis and the Stage 3 learner and context analysis. Therefore, the preceding discussion of Stages 4-6 provides a backdrop for understanding the relevance of the following factors to designing the tool (see Table 10):

Table 10

*Stage 4-6 Design Factors*

<table>
<thead>
<tr>
<th>Factor Identification</th>
<th>Factor Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>General problem-solving behaviors</td>
<td>General problem-solving behaviors remain constant across contexts while specific problem-solving behaviors vary across contexts</td>
</tr>
<tr>
<td>Conditions</td>
<td>Conditions can vary</td>
</tr>
<tr>
<td>Criteria</td>
<td>Criteria may vary</td>
</tr>
<tr>
<td>Assessments</td>
<td>If conditions and criteria vary, assessments must be allowed to vary in order to align with objectives</td>
</tr>
<tr>
<td>Strategies</td>
<td>Strategies also must be allowed to vary in order for strategies to align with context as well as objectives and assessments</td>
</tr>
</tbody>
</table>

Based upon the above considerations, the comprehensive CBR tool should comply with the following Stage 4-6 specifications (see Table 11):
Stage 4-6 Specifications

<table>
<thead>
<tr>
<th>Stage</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stage 4</td>
<td>4.1 The tool will support the general behaviors that comprise the CBR problem-solving approach. (See also related Specifications 1.3: The comprehensive CBR tool should support learner performance relative to completing and/or implementing the four theory-based, research-based, top-level tasks of the CBR method, and their underpinning hierarchy of sub-tasks and methods. See also related specification 2.4: The tool supports performance of steps and sub-steps comprising the case-based reasoning process.)</td>
</tr>
<tr>
<td></td>
<td>4.2 The tool will support specific problem-solving behaviors that are appropriate for particular contexts and learners.</td>
</tr>
<tr>
<td></td>
<td>4.3 The tool will be compatible with a range of specific, relevant behavior performance conditions.</td>
</tr>
<tr>
<td></td>
<td>4.4 The tool will support multiple behavior performance criteria.</td>
</tr>
<tr>
<td>Stage 5</td>
<td>5.1 The tool will support a variety of assessment options and content that is appropriate for specific contexts and specific learners.</td>
</tr>
<tr>
<td>Stage 6</td>
<td>6.1 The tool will function effectively with a range of relevant instructional strategies.</td>
</tr>
</tbody>
</table>

(As depicted in Figure 13, one possible set of integrated strategies consists of a problem-solving macro-strategy and eight micro-strategies.)

Stage 7 – Create a Prototype

Specifications for designing a comprehensive CBR tool were identified and stated but not operationalized in the preceding six stages of the Dick, Carey, and Carey model. The central task in Stage 7 is to operationalize the specifications in a prototype of the tool. Seven factors relevant to designing a tool – characteristics of the tool and, by logical extension, design requirements for it – were identified in the literature review. The subsequent application of Stages 1-6 of the Dick, Carey, and Carey (2005) instructional systems development model to these design requirements generated a total of twenty-nine theory- and/or research-based design specifications, which embodied and operationalized the design requirements for a comprehensive CBR tool.

Operationalization of these specifications in this prototype is documented in the sequence of Table 12, Figures 14-19, and Table 13 on the following pages, and also in Appendix A. The tables, figures, and appendix should be reviewed with the realization that many but not all the features of a fully functioning tool were developed in the prototype. Significant formative evaluation results can be attained using a preliminary version of a tool like this prototype that is not fully developed but representative (Dick, Carey, & Carey, 2005; Virzi, Karis, & Sokolov, 1996; Walker, Takayama, & Landay, 2002).
The status of two particular features requires clarification: resources and a relational database. The prototype did include place holders for task-adjacent links to task-relevant resources but did not include the complete content of these proposed resources. Instead, each of these place-holding links connected the formative evaluator to a brief description of the content that could eventually constitute the proposed resource pending further development beyond this study. Proposed resources for each task included specification of subordinate skills needed for performance of the task behavior, models (so-called “worked examples” presented as exemplary task responses), procedures (cognitive modeling that involves “thinking aloud” about how to construct an exemplary task response), a list of various Web-based resources, and evaluation standards for task performance. These represented, proposed resources would be an integral part of the tool’s facilitated performance environment.

Moreover, the prototype included features that simulated the presence and functionality of a relational database, but did not include the database. Simulation was achieved primarily through ubiquitous data-entry forms that were non-functioning but “database-ready” and through the inclusion of representative domain-specific content. These forms were supplemented by working hyper-links that further simulated database functionality and operations. Creation of a working database for the comprehensive CBR tool would require programming code to make all components fully functional. This process requires a significant investment of time for its design and development, and expenditures for the professional, technical services of a database programmer (Hernandez, 1997). However, knowing that the prototype could inform creation of the needed database and that the prototype would inevitably require revisions following its formative evaluation, the decision was made to use a simulated database until the final programming needs could be fully and accurately specified.

A relational database is a robust information processing system and is thus essential to the concept of the tool that centers on an external mimicking of cognitive activity (Bush, 1945; di Sessa, 2000; Englehardt, 1962; Hull, Bull, Montgomery, May, & Overton, 2000; Nelson, 1978, 1983; Pea, 1985; Salomon, Perkins, & Globerson, 1991). All the content is to be generated from data held in a database, including the CBR-specific content created by the developer and represented in the database during the initial development process, and the domain-specific content created by users (both instructor and students) and represented in the database during routine use of the tool (Antelman, 1999; Element K, 2004c; Hernandez, 1997; McNeil & Robin, 2000a, b; Sullivan, 1999; Tu, 1999, 2000; Westman, 2001).
All the specifications from Stages 1-6 have been collected and re-stated in the second column of Table 12. A title, explanation, and requirements have been added to each specification in this listing. The third column presents a description of the manner in which specifications have been operationalized in the prototype. Examples of the operationalized specifications in the prototype can be seen in Figures 14-19 and also in Appendix A. These figures are component summaries that include name of the component, purpose, and a screen capture of the component’s initial page. The appendix consists of menu links to module contents. These artifacts together provide a comprehensive view of specifications and their operationalization in the prototype. The locations of all operationalized specifications are presented in Table 13, which immediately follows the figures.

Table 12

Operationalization of Specifications in the Prototype

<table>
<thead>
<tr>
<th>Source</th>
<th>Specification Features That Must Be Operationalized</th>
<th>Operationalization Approach Employed in Prototype</th>
</tr>
</thead>
<tbody>
<tr>
<td>STAGE 1: Needs Analysis – Specifications</td>
<td>1.1 Complex Thinking – The Construct: A comprehensive CBR tool should support learner performance relative to valid and reliable engagement of complex thinking skills first in curricular and ultimately in workplace contexts. Complex thinking is a psychological construct defined as becoming operative when three basic types of thinking (content/basic, creative, and critical thinking) are integrated to achieve three types of real-world goals: (a) solve a problem, (b) make a decision, and/or (c) create an artifact. Application of CBR ultimately involves experience achieving all three goal types; a comprehensive CBR tool for complex thinking should therefore support the whole CBR process inclusive of all three goal types.</td>
<td>Performance of CBR involves achieving three goal types. The central part of the prototype supports: (a) integrated procedures and tasks; (b) links to task-relevant resources; and (c) and database functionality, including relevant data-entry forms. Overall, the foregoing procedures, resources, and forms support problem solving. They also support the goal of creating an artifact, specifically, constructing a case. An example of support for decision making is the set of related tasks that involves accepting a failed solution or instead opting to repair the solution. In addition, discussion of subjects and constituent topics/sub-topics provides background information about CBR as a strategy for engagement of complex thinking.</td>
</tr>
<tr>
<td>Source</td>
<td>Specification Features That Must Be Operationalized</td>
<td>Operationalization Approach Employed in Prototype</td>
</tr>
<tr>
<td>--------</td>
<td>---------------------------------------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td><strong>1.2 Integrated Model of Cognition:</strong> A comprehensive CBR tool should support learner performance relative to integration of reasoning, learning/understanding, and dynamic memory.</td>
<td>The prototype mimics and provides support for the integrated model of cognition. Mimicry and support serve as a stimulus for valid, reliable performance of CBR. They further serve as a structure for external memory and thereby a repository for integrated, dynamic learning outputs. The prototype thus augments human reasoning, memory (internal or cognitive memory), and learning. Mimicry and support are operationalized through: (a) integrated procedures and tasks that support performance of CBR behaviors, including links to resources; (b) user’s responses to tasks are held in data entry forms, and eventually saved to a relational database; and (c) the integrated, evolving set of saved responses represents the outcomes of problem-based learning.</td>
<td></td>
</tr>
<tr>
<td>CBR is said to be an integrated model because it interrelates three key factors of cognition: (a) experience-based, resource-supplemented reasoning as a natural learning strategy, (b) learning as newly acquired domain knowledge and skills that result from application of that strategy, and (c) dynamic memory as a container for a useful, retrievable, case-formatted representation of new knowledge and skills, and as a subsequent support for continued experience-based learning. The tool should include a working representation of this model as it can be made to operate within external (resource-based) memory so as to reflect its operation within internal (personal) memory.</td>
<td>Integrated procedures and tasks at the center of the prototype stimulate reliable, valid performance of CBR behaviors relative to the tasks and alternative methods of the process. Furthermore, links to resources are intended to support that performance. In addition, related sets of integrated procedures/tasks support a feedback ↔ response loop for facilitating successful task performance. Finally, a relational database, including data-entry forms, enables saving, manipulating, and retrieving task responses, which collectively and eventually make up the body of a case. The case can be retrieved by search utility form.</td>
<td></td>
</tr>
<tr>
<td><strong>1.3 CBR Task Performance and Method Implementation:</strong> A comprehensive CBR tool should support learner performance relative to completing and/or implementing the four theory-based, research-based, top-level tasks of the CBR method, and their underpinning hierarchy of sub-tasks and methods.</td>
<td>The CBR model employs a hierarchical makeup of tasks and alternative task-performance methods. (For more details, see the discussion and graphic of the CBR hierarchy in Chapter 2 and the Stage 2 goal analysis in this chapter.) The tool should support this hierarchy as it has been reflected and further analyzed in the goal analysis.</td>
<td></td>
</tr>
<tr>
<td>Source</td>
<td>Specification Features That Must Be Operationalized</td>
<td>Operationalization Approach Employed in Prototype</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>1.4 Interpreting and Solving Problems: A comprehensive CBR tool should support learner performance relative to application of the CBR method to interpreting and solving ill-structured type problems.</td>
<td>The prototype includes a procedure and set of tasks dedicated to interpreting problems. It also includes two successive procedures for solving the problem – initially solving it and then either accepting a failed solution and proceeding, or attempting to repair the solution.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>The CBR approach can be applied to solely interpreting a given, undefined, ill-structured problem, or to both interpreting and subsequently solving the problem on the basis of that interpretation, depending on problem context needs.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A comprehensive tool should support both the outcomes of interpreting a problem and solving a problem.</td>
<td></td>
</tr>
<tr>
<td>1.5 Having and Saving Experiences: A comprehensive CBR tool should provide (a) supports for reflecting on and interpreting experience, which can serve to produce new cases for an evolving case library, and (b) a case library.</td>
<td>The prototype includes support for both learning from episodes of concrete problem-solving experience and for saving instructive, effectively represented learning outputs for retrieval and application to future problem-solving needs.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>The standard, general CBR cycle begins with interpreting and solving a given concrete problem through comparison with similar defined, solved, and documented problems, and includes reflecting on and learning from that experience, representing useful new knowledge in a retrievable case, and integrating the new case in a dynamic knowledge base to fuel future problem-solving experiences and further learning.</td>
<td>The procedures and tasks serve as prompts or stimuli for understanding experience and extracting lessons from it. Task performance is supported by links to resources.</td>
</tr>
<tr>
<td></td>
<td>The tool should support both these reflective activities and the saving of cases in order to support the complete CBR cycle.</td>
<td>A relational database, with data-entry forms for task responses and search requests, enables holding, manipulating, saving, and retrieving new cases that result from task responses.</td>
</tr>
<tr>
<td>Source</td>
<td>Specification Features That Must Be Operationalized</td>
<td>Operationalization Approach Employed in Prototype</td>
</tr>
<tr>
<td>--------</td>
<td>------------------------------------------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>1.6 Cross-Domain Versatility: A comprehensive CBR tool should be based on a domain-neutral representation of the CBR process but support ready creation and integration of domain-specific content, including instructive cases. The comprehensive CBR tool should allow ready integration into diverse problem/practice domains thus making the benefits of the CBR method available to a broad range of learners.</td>
<td>The prototype was designed to present a domain/program-neutral representation of CBR completely free of domain-specific references. It thus allows ready cross-domain/program integration, and is normally used within the adoptive domain/program to create domain-specific content (primarily cases). Moreover, the prototype provides the opportunity for inserting relevant domain-specific background information within a hierarchy of subjects/topics/subtopics, and domain-specific problem-solving resources. Database functionality enables the preceding operations through means of a dynamic content base: (1) CBR-specific content contributed and maintained by the developer and (2) domain-specific content contributed and maintained by users, who includes instructor and students.</td>
<td></td>
</tr>
<tr>
<td>A domain-neutral but readily appended representation of CBR as a natural, general approach to learning from domain-specific, concrete problem-solving experience and collecting domain-relevant cases is potentially useful for effectively meeting a particular class of learning needs confronting a range of practice domains, including professional education, law, turfgrass management, medicine, and architecture. The tool should support ready, appended cross-domain integration.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.7 Case Construction: A comprehensive CBR tool should support learner/user performance relative to constructing cases and adding those cases to the case library.</td>
<td>Database functionality enables a process of receiving and holding the user’s task responses in the objects of data-entry forms and subsequently saving them to a relational database, thus constructing a case. This case construction process can occur during and/or after problem-solving activities. Cases thus constructed collectively constitute a case library searchable by means of a data-entry form (a search form) and made accessible as needed for future problem-solving needs.</td>
<td></td>
</tr>
<tr>
<td>CBR includes not only facilitating concrete problem-solving experience and learning from that experience, but, in addition, CBR simultaneously or subsequently supports capturing useful representations of that experience in a retrievable case-based format for application to future problem-solving needs. To support the full CBR process, the tool should support this case construction activity.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Source</td>
<td>Specification Features That Must Be Operationalized</td>
<td>Operationalization Approach Employed in Prototype</td>
</tr>
<tr>
<td>--------</td>
<td>------------------------------------------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>1.8 Expert Practice Modeling: A comprehensive CBR tool should support collection of cases constructed and submitted by content experts who practice in the problem/practice domain where the comprehensive CBR tool has been integrated.</td>
<td>Integrated procedures/tasks with links to resources support constructing useful representations of instructive, concrete problem-solving experience. Expert experience can be particularly instructive. Well-planned, well-executed task prompting can prevent the expert’s perhaps unconscious tendency to over-generalize and require an informative representation of experience identifying all relevant factors. Database functionality enables capturing this prompted, procedural, detailed representation of experience for benefit of the inexperienced.</td>
<td></td>
</tr>
<tr>
<td>1.9 Experience-Centered Analogical Learning: A comprehensive CBR tool should support learner/user performance relative to the task of interpreting and benefiting from experience.</td>
<td>Relational database functionality allows saving useful representations of past problem-solving episodes for retrieval and application to new problem-solving needs. Procedures for defining a new, concrete problem through comparison with a similar, past, concrete problem, support adapting solutions to past problems to solving the new problem.</td>
<td></td>
</tr>
</tbody>
</table>

Seaking to offer vicarious experience for benefit of the novice, the CBR model features capturing experiences of expert practice, carefully avoiding the expert’s likely, automatic, generalized representations of experience, while striving for useful, explicit, coherent presentations of problem-solving experiences.

Faithfulness to the CBR model recommends that the tool should reflect a focus on models of expert practice, including those that can be routinely captured from expert practitioners of a domain.

CBR is the type of analogical reasoning that focuses on systematically deriving problem-solving and consequent educational value from interpreting and appropriately applying informative analogs from collected, retrievable representations of concrete problem-solving experience.

The tool should support an analogy-based reasoning and learning process that compares a given undefined, unsolved concrete problem with the analogs offered by defined and solved problems from the past.
<table>
<thead>
<tr>
<th>Source</th>
<th>Specification Features That Must Be Operationalized</th>
<th>Operationalization Approach Employed in Prototype</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1.10 Indexing Experience:</strong> A comprehensive CBR tool should support learner/user performance relative to the task of indexing cases or resolving the indexing problem.</td>
<td>Procedures/tasks with relevant resources and database functionality support the indexing experience: analyzing the case elements and accordingly assigning tentative indexing descriptors (by student) or final indexing descriptors (by instructor) to cases. Data-entry forms (search forms) incorporate the descriptors as means for identifying and retrieving cases from the database.</td>
<td></td>
</tr>
<tr>
<td>Resolving the indexing problem is a major contributor to achieving a successful CBR outcome: (a) the indexing of cases and the consequent indexing descriptors form the basis for performing similarity assessment at the heart of CBR, and (b) user participation in an indexing activity constitutes an abstracting process that facilitates case-based problem interpretation, problem solving, and learning. The tool should support two aspects of indexing: (a) instructor-performed indexing to ensure optimal operation of the tool and (b) user-performed indexing to facilitate the user’s case-based problem interpretation, problem-solving, and learning experience.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>1.11 Failure-Based Learning:</strong> A comprehensive CBR tool should support learner/user performance relative to learning from failed and corrective problem-solving strategies.</td>
<td>The prototype includes a sub-set of tasks and data-entry forms for evaluating the outcome of a solution and subsequently accepting a failed outcome or attempting to repair the solution, before proceeding to identification, analysis, and description of lessons learned from the problem-solving experience.</td>
<td></td>
</tr>
<tr>
<td>CBR views a well-planned, well-executed, but failed solution strategy to be a significant learning opportunity depending on iterative analysis and interpretation of the failure, repair of the strategy, execution of the repaired strategy, and evaluation. The tool should support an iterative, documented process of understanding, resolving, and learning from well-planned, well-executed, but failed efforts to solve a problem.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>1.12 Using Cases for Interpreting and Solving Problems:</strong> A comprehensive CBR tool should support learner/user performance relative to using cases as the basis for reasoning.</td>
<td>The prototype’s collection of cases in the relational database and case library serves as a problem-solving resource. Also, database functionality supports the capability to construct, save, index, retrieve, and apply usefully-formatted cases of instructive, past problem-solving episodes to defining and solving similar, new problems.</td>
<td></td>
</tr>
<tr>
<td>Useful episodes of concrete problem-solving experience authentically represented in a retrievable case-based format forms a basis for reasoning and solving problems. The tool should provide a means for collecting cases and probing them for relevant problem-solving strategies.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Source</td>
<td>Specification Features That Must Be Operationalized</td>
<td>Operationalization Approach Employed in Prototype</td>
</tr>
<tr>
<td>--------</td>
<td>--------------------------------------------------</td>
<td>------------------------------------------------</td>
</tr>
<tr>
<td>STAGE 2: Instructional Analysis – 5 specifications</td>
<td><strong>2.1 Complex Thinking – The Goal:</strong> A comprehensive CBR tool should support an instructional goal to promote complex thinking through performance of the case-based reasoning process. A tool should be designed to support achievement of an instructional goal for performance of complex thinking skills with specific regard to three types of sub-goals: (a) solving a problem, (b) making a decision, and (c) designing an artifact. The tool should support an instructional goal for performance of complex thinking skills with specific regard to three areas.</td>
<td>An effective strategy for teaching complex thinking skills ultimately extends to authentic practice of these skills, specifically, skills applied to achieving the three specified types of sub-goals. Sub-sets of tasks prompt performance of all three types of complex thinking skills.</td>
</tr>
<tr>
<td></td>
<td><strong>2.2 Highest-Order Skills:</strong> A comprehensive CBR tool should support performance of highest order skills integral to the case-based reasoning process, according to four standard taxonomies of learning. A tool should be designed to support acquisition of highest-order skills (“intellectual” skills) as classified by four standard taxonomies of learning, including well-structured problem-solving skills but emphasizing the more cognitively rigorous ill-structured problem-solving skills. The tool should support acquisition of highest-order skills.</td>
<td>The prototype’s skill performance tasks involve resolving ill-structured issues, such as predicting the outcome of a solution, analyzing why the solution failed, and repairing the solution.</td>
</tr>
<tr>
<td></td>
<td><strong>2.3 General Problem-Solving Context:</strong> A comprehensive CBR tool should support performing steps/sub-steps of the case-based reasoning process as an extension of its precursor the general problem-solving process. A tool should be designed to support acquisition of CBR-specific problem-interpretation and problem-solving skills within the context of the general problem-solving process. The tool should support acquisition of CBR-specific problem-interpretation and problem-solving skills.</td>
<td>The Stage 2 instructional analysis, which was based on a standard task/method analysis of CBR, identified CBR-engaged complex thinking as a goal and situated that goal within the context of the general problem-solving process. Development of integrated sets of procedures/tasks within the prototype reflects the instructional analysis and ultimately the task/method analysis. Therefore, these procedures/tasks of the prototype likewise conform to the basic requirements of the general problem-solving process.</td>
</tr>
<tr>
<td>Source</td>
<td>Specification Features That Must Be Operationalized</td>
<td>Operationalization Approach Employed in Prototype</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------</td>
<td>-----------------------------------------------</td>
</tr>
<tr>
<td>2.4 CBR Steps and Sub-Steps: A comprehensive CBR tool should support performance of steps and sub-steps comprising the case-based reasoning process.</td>
<td>The prototype’s integrated procedures/tasks reflect the Stage 2 goal step/sub-step analysis. Procedures generally coincide with goal steps, and tasks with sub-steps.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A tool should be designed to support performance of goal step and sub-step behaviors making up the CBR process.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>The tool should support performance of goal step and sub-step behaviors</td>
<td></td>
</tr>
<tr>
<td>2.5 Subordinate Skills: A comprehensive CBR tool should support subordinate skills relative to performing the steps/sub-steps of the case-based reasoning process.</td>
<td>The design of the procedure/task environment dominating the prototype appropriately reflects the standard CBR task/method analysis and the subsequent Stage 2 goal step/sub-step analysis. The Stage 2 goal analysis includes two representative sets of subordinate skills, and indicates that a comprehensive analysis of subordinate skills could be conducted later and incorporated in a fully-functioning tool. The prototype features a task-adjacent bar of links to task-relevant resources. One of these links connects the user to task-relevant subordinate skills.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A tool should be designed to include specification of subordinate skills for performance of goal step and sub-step behaviors represented in the tool.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>The tool should specify relevant subordinate skills.</td>
<td></td>
</tr>
</tbody>
</table>
### STAGE 3: Learner and Context Analysis – 6 specifications

<table>
<thead>
<tr>
<th>Source</th>
<th>Specification Features That Must Be Operationalized</th>
<th>Operationalization Approach Employed in Prototype</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1 Program Relevance, Adoption, and Integration: A comprehensive CBR tool should be integrated in an instructional, training, and workplace management programs across a range of domains.</td>
<td>Tools can be designed to allow narrow integration only in a certain domain, or broad integration in a range of domains. A comprehensive tool allows broad integration. This tool is designed to effectively, readily meet both cross-domain, general needs and domain-specific, particular needs. A flexible structure is essential to creating this capability.</td>
<td></td>
</tr>
</tbody>
</table>

A tool should be designed for usefulness in instructional, training, and/or workplace management programs across a range of domains such as professional education and architecture, thereby making it an alternative for adoption and integration.

The tool should offer a capability for ready integration in various types of programs across a range of domains.

3.2 Learner and Context Compatibility: A comprehensive CBR tool should be compatible with the integrated learner and context components of the instructional or training system.

To ensure the integrity of an instructional system, a tool should be designed to be compatible with learner characteristics, such as proficiency level, and performance and learning context characteristics, such as degree of realism.

Multiple cases are required to address differing learner characteristics and to reflect accurately a variety of contexts. Although the prototype contains limited examples of cases, its design allows the addition of an unlimited numbers of cases dealing with a wide range of topics. Also, because the tool is designed around the use of a relational database, cases can be retrieved based upon their relevance to both learners and contexts.

Design of the tool should be compatible with learner and context characteristics.
3.3 Advanced Learning: A comprehensive CBR tool should support advanced learners in the application of CBR.

A tool should be designed to support advanced learners in achieving advanced learning outcomes according to explicit definitions of these variables in the literature.

The tool should support advanced learners in achieving advanced outcomes.

<table>
<thead>
<tr>
<th>Source</th>
<th>Specification Features That Must Be Operationalized</th>
<th>Operationalization Approach Employed in Prototype</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.3 Advanced Learning: A comprehensive CBR tool should support advanced learners in the application of CBR.</td>
<td>The advanced learners of a domain of study have progressed to a level of experience and education that enable them to engage the ill-structured, advanced issues of their domain. The prototype can serve as a tool for effectively engaging those issues.</td>
<td>The advanced concepts of a domain tend to be ill-structured per se or in their application. Over time, the evolving collection of different cases accessible in the prototype’s relational database and case library is likely to offer the multiple perspectives essential to fully comprehending any ill-structured concepts of a domain.</td>
</tr>
<tr>
<td></td>
<td>The prototype’s indexing and formatting of a case-documented perspective is intended to clarify its applicability to engaging particular ill-structured issues of a domain.</td>
<td>In addition, the prototype’s procedures and tasks, supported by relevant resources, guide the analysis and probing of the case-documented perspectives to determine their applicability to understanding and resolving particular issues of a domain. Thus comparing and contrasting the various perspectives with each other and with the problem situation, and concurrently evaluating their applicability, produce interconnected, flexible, advanced knowledge that is likely to transfer from the learning context to performance contexts.</td>
</tr>
</tbody>
</table>
3.4 Real-World or Ill-Structured Problem Solving: A comprehensive CBR tool should support advanced learners in the application of CBR to ill-structured problem solving as a means for engaging complex thinking and producing advanced understanding.

A tool should be designed for use by advanced learners in applying CBR to solving real-world or ill-structured problems and thereby both engaging complex thinking and achieving advanced learning outcomes.

The tool should support advanced application of CBR.

3.5 Authentic Learning Context: A comprehensive CBR tool should be integrated in a practice field learning context.

Consistent with CBR’s interest in the problem-solving and educational value of well-executed, authentic problem-solving experience, a tool should be designed for productive integration in an authentic learning context, with options generally ranging from more authentic contexts (a practice field or simulated context) to most authentic context (the actual workplace).

The tool should be adequate to achieving the particular educational potential of an authentic learning context.

In general, the prototype’s database and case library of problem-based learning experiences center on situated or ill-structured problems, introducing a scenario of multiple, divergent solution options with best solution depending on situational factors.

The prototype’s case library, probing procedures and tasks, task-relevant resources, and data-entry forms support higher-order analysis, interpretation, and application of past ill-structured problem-solving experiences to solving a new problem: identifying solution options, evaluating their feasibility, tentatively selecting the “best” feasible option, justifying the selection, predicting the outcome of the selected option, testing the selection, comparing prediction to outcome and explaining any failure, revising or replacing the option as appropriate, and identifying lessons learned.

This facilitated task performance involves all three of the basic complex thinking behaviors (solving problems, making decisions, and creating an artifact). In addition, such a process of identifying relevant factors and analyzing relationships among them naturally produces interconnected, advanced-level knowledge that transfers from the learning context to performance contexts.

A real-world, authentic learning context naturally presents actual, challenging problems as the primary stimuli for an advanced learning process. The prototype, with its CBR-supportive background information, procedures and tasks, task-relevant resources, and database of cases demonstrates a means of responding to these learning stimuli (solving the problem) and saving each instructive experience to spur a continuing problem-driven learning process (capturing the problem solving in a case). The prototype could thus support the user in realizing the potential of an authentic learning environment.
<table>
<thead>
<tr>
<th>Source</th>
<th>Specification Features That Must Be Operationalized</th>
<th>Operationalization Approach Employed in Prototype</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>3.6 Practitioner Candidate Training Relevance:</strong> A comprehensive CBR tool should enable advanced learners and soon-to-be practitioners studying issues of professional practice to acquire realistic experience in applying CBR-supported complex thinking to resolving ill-structured issues that define the domain.</td>
<td>The learning process supported by the prototype is consistent with a standards-, research-recommended approach for achieving best practice in a domain of interest. Practitioners-in-training are expected to master this approach as a requirement for becoming an effective practitioner who contributes to advancement of the domain. Specifically, the prototype’s facilitated, problem-driven procedures and tasks prompt the learner to reflect on case-based, issue-centered experience – which could be instructive episodes of practice occurring within the domain – and identify, describe, and explain lessons learned from that experience and practice. Lessons could be applied to improving practice and ultimately achieving best practice. Advanced, practitioners-in-training could thus construct cases from useful, issue-centered experiences occurring within the context of clinical experiences, and save them to the database for benefit of peers as well as in-service practitioners. In addition, a directed teaching strategy could call on case-captured practice to illustrate knowledge and skills of the domain and to stimulate emulation of effective practice.</td>
<td></td>
</tr>
<tr>
<td><strong>STAGE 4: Objectives – 4 specifications</strong></td>
<td><strong>4.1 General CBR Performance:</strong> A comprehensive CBR tool should support the general behaviors that comprise the CBR problem-solving approach. A tool should be designed to support the general step and sub-step behaviors of the CBR process as they have been incorporated in program objectives from across a range of domains and program contexts, including instructional, training, and/or workplace management contexts. The tool should support general goal step and sub-step behaviors of the CBR process.</td>
<td>General CBR performance as initially identified in the standard CBR task/method analysis and subsequently in the Stage 2 goal step/sub-step analysis is reflected in the design of the procedure/task environment that dominates the prototype. In addition, the database functionally, including data-entry forms, provides the dynamic memory required by CBR. Finally, background information for general CBR performance is provided in a hierarchical presentation of subjects/topics/sub-topics.</td>
</tr>
<tr>
<td>Source</td>
<td>Specification Features That Must Be Operationalized</td>
<td>Operationalization Approach Employed in Prototype</td>
</tr>
<tr>
<td>--------</td>
<td>----------------------------------------------------</td>
<td>-------------------------------------------------</td>
</tr>
<tr>
<td><strong>4.2 Specific Performance Compatibility:</strong> A comprehensive CBR tool should support specific problem-solving behaviors that are appropriate for particular contexts and learners.</td>
<td>Support for general CBR performance in 4.1 provides a framework for specific performance supported in 4.2.</td>
<td></td>
</tr>
</tbody>
</table>

A tool should be designed for compatibility with a range of specific performance behaviors that occur within the framework of general CBR performance in 4.1 and vary according to program needs.

The tool should offer compatibility with a range of specific performance behaviors.

| **4.3 Performance Condition Compatibility:** A comprehensive CBR tool should be compatible with a range of specific, relevant behavior performance conditions. | Condition compatibility can be clarified through an example. The procedure/task environment dominating the prototype includes links to resources. Both the content of these resources and access to them could be varied as a means for achieving condition compatibility. Content would likely be more comprehensive and accessible in an instructional or training situation but less so in a workplace situation. This content/access variability could be readily accomplished through capabilities of the database technology. |

A tool should be designed for compatibility with a range of performance conditions that vary according to program needs.

The tool should offer compatibility with a range of performance criteria.

| **4.4 Performance Criteria Compatibility:** A comprehensive CBR tool should support multiple behavior performance criteria. | The sets of integrated procedures/tasks with integrated resources and data-entry forms allow a range of performance criteria depending on domain/program needs, similar to performance condition variability. In general, high-stakes workplace integrations of the prototype would require more stringent criteria than instructional or training integrations. |

A tool should be designed for compatibility with a range of performance criteria that vary according to program needs.

The tool should offer compatibility with a range of performance criteria.
<table>
<thead>
<tr>
<th>Source</th>
<th>Specification Features That Must Be Operationalized</th>
<th>Operationalization Approach Employed in Prototype</th>
</tr>
</thead>
<tbody>
<tr>
<td>STAGE 5: Assessments – 1 specification</td>
<td>5.1 <em>Assessment Compatibility:</em> A comprehensive CBR tool should support a variety of assessment options and appropriate for specific contexts and specific learners.</td>
<td>Database functionality, including data-entry forms, supports creation of a dynamic assessment instrument applicable as appropriate to meeting both formative and summative needs. Specifically, developer-created and-entered, unalterable, CBR-relevant evaluation standards could be extended by user-created, -edited, and -inserted domain-specific standards.</td>
</tr>
<tr>
<td></td>
<td>A tool should be designed for compatibility with a range of assessment options that vary according to domain/program needs.</td>
<td>On the other hand, the instructor’s choice of an alternative evaluation format could be substituted for this prototype-based format, and the space of the prototype dedicated to an instrument could be left empty and disregarded.</td>
</tr>
<tr>
<td></td>
<td>The tool should offer compatibility with a range of assessment options.</td>
<td>Regardless of instrument format, assessment level must be matched to outcome level in order to produce valid evaluation data. For example, while the prototype’s integrated sets of procedures/tasks emphasize advanced learners and outcomes, instructor-created and -formatted lower-order assessments could be used to evaluate lower-level outcomes related to basic knowledge of the steps and sub-steps that make up the CBR process, or to particular case-based domain knowledge of interest.</td>
</tr>
<tr>
<td>Source</td>
<td>Specification Features That Must Be Operationalized</td>
<td>Operationalization Approach Employed in Prototype</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------</td>
<td>-------------------------------------------------</td>
</tr>
<tr>
<td>STAGE 6: Strategy – 1 specification</td>
<td>6.1 <em>Strategy Compatibility</em>: A comprehensive CBR tool should function effectively with relevant instructional strategies, including but not restricted to a problem solving macro-strategy supported by multiple micro-strategies. A tool should be designed for compatibility with a range of strategy options that vary according to program needs. The tool should offer compatibility with a range of strategy options.</td>
<td>The prototype employs a problem solving macro-strategy supported by eight micro-strategies: one micro-strategy inspired by the CBR method and seven more inspired by the general problem-solving method. Three different examples of prototype-strategy compatibility can clarify this specification and approach. One strategy could consist of directly teaching the content of cases for demonstration of domain principles. Cases are held in a relational database, and access to them is achieved through a search form. Cases are structured according to a specified, recommended format for highlighting their lessons and usefulness. Another strategy could be directly teaching and demonstrating a recommended format for creating useful representations of concrete problem-solving experience. This could be an important strategy: usefulness of vicarious experience depends on effective representation of that experience. Primarily procedures and tasks/subtasks, and links to problem-solving resources would be relevant to this strategy, but relevant subjects and topics/sub-topics could also play a role. A third strategy could be a problem-based approach primarily relying on procedures and tasks/subtasks and links problem-solving resources. While all these strategies would productively utilize the prototype, implementation of this third strategy would more fully tap the educational potential of the prototype.</td>
</tr>
</tbody>
</table>
approach to thinking, problem solving, and learning. The following overview consists of a listing of the components with the name of each component, purpose, and screen capture of the component’s initial page:
Figure 14. Background component: Screen capture and purpose. The first of the six components is Background. The purpose of Background is to present background information for understanding the CBR process and professional applications of the process to achieving best practice.
Figure 15. Operations component: Screen capture and purpose. The second of the six components is Operations. The purpose of Operations is to specify, guide, and support active, realistic practice of the CBR process by means of procedures/tasks, links to task-relevant resources, and data-entry forms.
Figure 16. Help component: Screen capture and purpose. The third of the six components is Help. The purpose of Help is to provide direct search access to Models, Procedures, and Web Resources (which can also be accessed through linking in the procedure/task environment of the Operations component), and to Glossary Definitions (which can also be accessed through linking within case content).
Figure 17. Evaluation component: Screen capture and purpose. The fourth of the six components is Evaluation. The purpose of Evaluation is to provide direct access to evaluation instruments (which can also be accessed through linking in the procedure/task environment of the Operations component).
**Figure 18.** Exploration component: Screen capture and purpose. The fifth of the six components is Exploration. The purpose of Exploration is to provide search access to the cases that make up this case library component of the tool, which consists both of cases created during and subsequent to the problem-solving process.
Figure 19. Administration component: Screen capture and purpose. The sixth of the six components is Administration. The purpose of Administration is to facilitate practice of the CBR process in the Operations component through a feedback ↔ response loop, through editing new cases, and adding customized task-relevant resources.
Finally, with the overview of the prototype’s components as a backdrop, locations of prototype structures dedicated to enabling operationalization of specifications are presented in Table 13. The references to structure locations found in column two include component name and module number only. Detailed module menus are presented in Appendix A:

<table>
<thead>
<tr>
<th>Specification</th>
<th>Location of Operationalized Specifications in Prototype</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1 Complex Thinking – The Construct</td>
<td>Background 1-2; Operations 2-5, 7; Help 1-3; Evaluation 1; Exploration, 1-2; Administration 1.</td>
</tr>
<tr>
<td>1.2 Integrated Model of Cognition</td>
<td>Background 2-3; Operations 2-5; Exploration 1-2; Administration 1-2.</td>
</tr>
<tr>
<td>1.3 CBR Task Performance and Method Implementation</td>
<td>Background 2-3; Operations 2-5, 7; Help 1-3; Exploration 1-3; Administration 1-2.</td>
</tr>
<tr>
<td>1.4 Interpreting and Solving Problems</td>
<td>Interpreting a problem: Operations 2; Help 1-3; Exploration 1-2; Administration 1-2.</td>
</tr>
<tr>
<td></td>
<td>Solving a problem: Operations 3-4; Help 1-3; Exploration 1-2; Administration 1-2.</td>
</tr>
<tr>
<td>1.5 Having and Saving Experiences</td>
<td>Background 2; Operations 2-5; Help 1-3; Exploration 1-2; Administration 1-3.</td>
</tr>
<tr>
<td>1.6 Cross-Domain Versatility</td>
<td>Background 1, 4; Operations 1-7; Help 1-3; Exploration 1-2; Administration 1-7.</td>
</tr>
<tr>
<td>1.7 Case Construction</td>
<td>Background 2; Operations 1-7; Help 1-3; Exploration 1-2; Administration 1-3.</td>
</tr>
<tr>
<td>1.8 Expert Practice Modeling</td>
<td>Background 2; Operations 1-7; Help 1-3; Administration 1-3.</td>
</tr>
<tr>
<td>1.9 Experience-Centered Analogical Learning</td>
<td>Operations 2; Help 1-3; Exploration 1-2.</td>
</tr>
<tr>
<td>1.10 Indexing Experience</td>
<td>Operations 2-5, 7; Administration 1-2.</td>
</tr>
<tr>
<td>1.11 Failure-Based Learning</td>
<td>Background 2; Operations, 4, 7; Help 1-3; Administration 1.</td>
</tr>
<tr>
<td>Specification</td>
<td>Location of Operationalized Specifications in Prototype</td>
</tr>
<tr>
<td>-------------------------------------------------</td>
<td>------------------------------------------------------</td>
</tr>
<tr>
<td>1.12 Using Cases for Interpreting and Solving Problems</td>
<td>Background 1-2; Operations 1-7; Help 1-3; Exploration, 1-2; Administration 1; Operations 1-7.</td>
</tr>
<tr>
<td>2.1 Complex Thinking – The Goal</td>
<td>Background, 1-5; Operations, 1-7; Help 1-3; Evaluation 1; Exploration, 1-2; Administration 1-3.</td>
</tr>
<tr>
<td>2.2 Highest-Order Skills</td>
<td>Operations 1-7.</td>
</tr>
<tr>
<td>2.3 General Problem-Solving Context</td>
<td>Operations 1-7; Administration 1.</td>
</tr>
<tr>
<td>2.4 CBR Steps and Sub-Steps</td>
<td>Operations 2-5.</td>
</tr>
<tr>
<td>2.5 Subordinate Skills</td>
<td>Operations 2-5.</td>
</tr>
<tr>
<td>3.1 Program Relevance, Adoption, and Integration</td>
<td>Background 1-5; Operations 1-7; Help 1-3; Evaluation 1; Exploration 1-2; Administration 1-7.</td>
</tr>
<tr>
<td>3.2 Learner and Context Compatibility</td>
<td>Background 1-5; Operations 1-7; Help 1-3; Evaluation 1; Exploration 1-2; Administration 1-7.</td>
</tr>
<tr>
<td>3.3 Advanced Learning</td>
<td>Background 1-5; Operations 1-7; Help 1-3; Evaluation 1; Exploration 1-2; Administration 1-7.</td>
</tr>
<tr>
<td>3.4 Real-World or Ill-Structured Problem Solving</td>
<td>Background 1-5; Operations 1-7; Help 1-3; Evaluation 1; Exploration 1-2; Administration 1-7.</td>
</tr>
<tr>
<td>3.5 Authentic Learning Context</td>
<td>Background 1-5; Operations 1-7; Help 1-3; Evaluation 1; Exploration 1-2; Administration 1-7.</td>
</tr>
<tr>
<td>3.6 Practitioner Candidate Training Relevance</td>
<td>Background 1-5; Operations 1-7; Help 1-3; Evaluation 1; Exploration 1-2; Administration 1-7.</td>
</tr>
<tr>
<td>4.1 General CBR Performance</td>
<td>Background 2; Operations 2-5; Help 1-3; Evaluation 1; Exploration 1-2; Administration 1.</td>
</tr>
<tr>
<td>4.2 Specific Performance Compatibility</td>
<td>Background 2; Operations 2-5; Help 1-3; Evaluation 1; Exploration 1-2; Administration 1.</td>
</tr>
<tr>
<td>4.3 Performance Condition Compatibility</td>
<td>Operations 1-7; Administration 4-7.</td>
</tr>
<tr>
<td>4.4 Performance Criteria Compatibility</td>
<td>Operations 1-7; Help 1-3; Evaluation 1; Exploration 1-2; Administration 1.</td>
</tr>
</tbody>
</table>
### Specification | Location of Operationalized Specifications in Prototype

<table>
<thead>
<tr>
<th>5.1 Assessment Compatibility</th>
<th>Evaluation 1; Administration 4.</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1 Strategy Compatibility</td>
<td>Background 1-5; Operations 1-7; Help 1-2; Evaluation 1; Exploration 1-2; Administration 1-7.</td>
</tr>
</tbody>
</table>

The foregoing documentation of specifications in the prototype answered the second research question and thus provided one view of the prototype. Acquiring expert feedback relative to the prototype in the upcoming formative evaluation provides another view.

*Stage 8 – Design and Conduct a Formative Evaluation*

Tools such as comprehensive CBR tool should eventually undergo both formative and summative types of evaluation (Dick, Carey, & Carey, 2005). Robert Stake (1991, p. 169) has been credited with using a soup metaphor for distinguishing between these two types of evaluation:

> When the cook tastes the soup, that is formative evaluation; when the guest tastes it, that is summative evaluation.

As Stake’s (2007) metaphor and the name *summative* (Merriam-Webster, 2006b) suggest, this type of evaluation, when applied to tools, is a validation process that comes after production and deployment of a tool (Dick, Carey, & Carey, 2005). Summative evaluation seeks to determine how well a deployed tool plays a purported role in supporting achievement of an adopting organization’s program goals (Dick, Carey, & Carey, 2005). It consequently determines the longevity of the tool in that organizational context (Dick, Carey, & Carey, 2005). Oftentimes then, the organization or its paid, outside agents will conduct a summative evaluation (Dick, Carey, & Carey, 2005). Although important, a summative evaluation was beyond the scope of the systematic design process defined and carried out in this research study (Dick, Carey, & Carey, 2005). Therefore, such an evaluation was not designed and/or conducted here.
Formative evaluation, unlike summative evaluation, plays an integral role in the process of forming or designing a tool (Dick, Carey, & Carey, 2005). A systematic, iterative process of more prototype revisions followed by more evaluations constitutes a primary mechanism for improving and shaping an evolving, increasingly refined design (Dick, Carey, & Carey, 2005). The designer as chief stakeholder in guiding the design process appropriately conducts this type of evaluation (Dick, Carey, & Carey, 2005).

The formative evaluation. In this design project three specific areas required formative evaluation. The first area involved the instantiation of theory or research-based specifications within the prototype. The second area dealt with the visibility of the support strategies (Jonassen, 1999) incorporated into the prototype and the third area related to the overall usability (Head, 1999; Nielsen, 1994; Nielsen & Molich, 1990; Norman, 1988; Pierotti, 2005) of the user interface.

Activities carried out in Stage 7 were designed to achieve the goal of operationalizing specifications in the prototype. Operationalization is documented in Table 12, Figures 14-19, and Table 13. This operationalization was performed by the researcher and provided one "test" of the concept’s validity (Rieber, 1994; Tripp & Bichelmeyer, 1990). It also served a formative evaluation function by confirming that all specifications were addressed in the prototype.

The remaining formative evaluation areas dealt with the visibility of the support strategies incorporated into the prototype and the overall usability of the user interface (Head, 1999). Both of these evaluations employed an expert review type of evaluation (Tessmer, 1993). Available experts were asked to contribute evaluation data relating to two dimensions of the prototype: (a) the instructional strategies incorporated into the design, and (b) the usability of the user interface. The experts are generally described in Table 14:
Table 14

Experts Serving as Formative Evaluators

<table>
<thead>
<tr>
<th>Professional Responsibilities</th>
<th>Relevant Expertise</th>
<th>Focus</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assistant Professor of</td>
<td>Multicultural Education, Science Education</td>
<td>Strategy visibility</td>
</tr>
<tr>
<td>Education</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Associate Professor of</td>
<td>Instructional Technology, Problem- and Case-Based Learning</td>
<td>Strategy visibility</td>
</tr>
<tr>
<td>Education</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Associate Professor of</td>
<td>Educational Psychology, Problem-Based Learning, Multimedia Technology Instructional Supports for Higher-Order Cognition</td>
<td>Usability</td>
</tr>
<tr>
<td>Education</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Freelance Editor of</td>
<td>Problem-Based Methods in PK-12 Education, Educational Research</td>
<td>Strategy visibility</td>
</tr>
<tr>
<td>Educational Publications</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Associate Professor of</td>
<td>Literacy Education, Problem-Based Learning, Multicultural Education</td>
<td>Strategy visibility</td>
</tr>
<tr>
<td>Education</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Information Technology</td>
<td>Design of Dynamic (Database-Enabled) Web Applications</td>
<td>Usability</td>
</tr>
<tr>
<td>Support Manager</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Data were collected over a four-week period. The researcher managed the evaluation process primarily by email communication with evaluators. First, an invitation to participate in the evaluation was emailed to the experts and prospective evaluators. It consisted of the invitation, a brief description of the tool concept, the evaluation procedure, and identification of the prospective evaluator’s relevant expertise. A second email announced the start of the evaluation period. That email expressed appreciation for serving, and included information relating to purpose of the evaluation, the foci of evaluation, extent of the evaluation period, and evaluation artifacts (the online prototype created in Stage 7 and two online instruments described later and included in Appendices B and C). This email also included relevant links and directions for logging into the prototype and instruments, along with the researcher’s contact information for questions and/or reporting problems. Evaluators could complete their evaluations anytime during the scheduled evaluation period. Two follow-up emails including basically the same information as the latter email were sent during the evaluation period to encourage completion of the evaluation.

Separate instruments were designed and created for evaluating the visibility and appropriate application of strategies in the prototype and usability of the prototype interface (see copies of the instruments in Appendices B and C). A Web-based survey tool was used to produce and deploy the instruments, and to collect evaluator responses. The instrument for strategy visibility (see Appendix B) was based on the researcher’s analysis of Jonassen’s (1999) model of problem-based instructional strategies. This instrument contained questions dealing with one macro-strategy and eight supportive micro-strategies. As explained in Stage 6, these strategies are directly
relevant to realizing the full potential of the tool (Kolodner & Guzdial, 2000), and thus constitute relevant evaluation criteria. Furthermore, as depicted in Figure 20, an overall congruence of strategy criteria with specifications was established by entering each in a relational database and associating the two:

![Table 15](image)

**Figure 20.** Associating evaluation elements and design specifications using a relational database.

The usability instrument (see Appendix C) was based on the researcher’s adaptation of a Xerox instrument (Pierotti, 2005). The adaptation process involved selecting relevant items from the Xerox instrument. The Xerox items were generally derived from Nielsen’s ten usability heuristics (rules of thumb) or principles (Nielsen, 1994; Nielsen & Molich, 1990), and therefore, constituted relevant evaluation criteria.

Both instruments had the same format as can be seen by reviewing the content of the instruments in Appendices B and C and the depiction of formatting presented in Table 15:
Table 15

*Format for the Two Formative Evaluation Instruments*

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction:</td>
<td>1. Purpose of the research study</td>
</tr>
<tr>
<td></td>
<td>2. Objectives of the research study</td>
</tr>
<tr>
<td></td>
<td>3. Intended users of the tool</td>
</tr>
<tr>
<td></td>
<td>4. Representative scenarios for use of the tool</td>
</tr>
<tr>
<td></td>
<td>5. Focus of the evaluation</td>
</tr>
<tr>
<td></td>
<td>6. Evaluation procedure</td>
</tr>
<tr>
<td></td>
<td>7. Contact information for the researcher</td>
</tr>
<tr>
<td>Directions:</td>
<td>1. Directions for responding to the items of the instrument</td>
</tr>
<tr>
<td></td>
<td>2. Theoretical basis for designing the instrument</td>
</tr>
<tr>
<td>Evaluation Items:</td>
<td>Each of several evaluation items consisted of the following elements:</td>
</tr>
<tr>
<td></td>
<td>● Item: A design principle related to strategy visibility or usability</td>
</tr>
<tr>
<td></td>
<td>● Set of questions relevant to the principle:</td>
</tr>
<tr>
<td></td>
<td>Responding to a question involves making a Likert rating of the tool’s</td>
</tr>
<tr>
<td></td>
<td>compliance with the principle on the basis of the criterion specified in</td>
</tr>
<tr>
<td></td>
<td>the question</td>
</tr>
<tr>
<td></td>
<td>● An opportunity to make discretionary comments that elaborate on</td>
</tr>
<tr>
<td></td>
<td>ratings for this set of questions, including:</td>
</tr>
<tr>
<td></td>
<td>(a) Perceived positive and negative aspects of the design</td>
</tr>
<tr>
<td></td>
<td>(b) Recommended improvements</td>
</tr>
</tbody>
</table>

Evaluators could respond to either web-based or word-processed versions of the instruments. Three of the five evaluators responded directly to the individual items in the Web-based versions of the instruments. The remaining two evaluators opted to vary somewhat from the instruments provided to them. Rather than responding to all individual items, these evaluators grouped their reactions in a limited number of clusters. Although the latter approach prevented direct comparison of all responses, it did provide meaningful information that was included in the formative evaluations.

*Formative evaluation data.* Each of the ten two-table sets in this section (Tables 16-35) generally include: (a) a table stating a strategy (Jonassen, 1999), listing elements of the strategy, and presenting Likert ratings for each element, and (b) a table summarizing comments relative to visibility of specified elements in the prototype.
Table 16

Visibility of Macro Strategy: Problem-Solving – Likert Ratings

<table>
<thead>
<tr>
<th>Strategy</th>
<th>Element</th>
<th>None</th>
<th>Low</th>
<th>Somewhat Low</th>
<th>Somewhat High</th>
<th>High</th>
</tr>
</thead>
<tbody>
<tr>
<td>The environment should be centered on a “problem”-driven process, where “problem” is a generic term that includes a question, case, issue, problem, or project (Jonassen, 1999).</td>
<td>1. Problem solving drives the learning of domain content</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2. Ill-structured problem solving produces ownership</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3. The problem situation is analyzed</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>4. A product is created or a decision is made</td>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

Table 17

Visibility of Macro Strategy: Problem Solving – Comments

<table>
<thead>
<tr>
<th>Element</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A. Support for case-based problem solving was less clear than support for some other features, such as editing and adding new cases to the tool. Even though the tool does ask the user to identify adaptable cases, adaptable solution elements, models, feasible and infeasible solutions, with regard to problem solving support, one evaluator said, “I did not see how it did this.”</td>
</tr>
<tr>
<td>2</td>
<td>A. The capability of the tool to support solving ill-structured problems was not clear to one evaluator. According to this evaluator, if adding a new case is to be a “problem,” it would be a well-structured rather than an ill-structured “problem”. Solving this well-structured problem may produce ownership of learning, but meaningfulness of the problem solving would depend on whether it was elective or imposed. Meaningfulness would be assumed with elective well-structured problem solving. Meaningfulness would depend on learner characteristics with imposed problem solving.</td>
</tr>
<tr>
<td>3</td>
<td>A. According to one evaluator, the tool does not provide support for problem situation analysis unless links to models, procedures, Web resources, and assessment standards present in the tool might provide such support. While the prototype did include links to these resources and general descriptions of their content, it does not include that content. Seeing that content was needed to make a definitive assessment of support for problem situation analysis. B. The tool probably supports situation analysis for adding a new case by means of prompts and expert feedback.</td>
</tr>
<tr>
<td>4</td>
<td>A. Evidence for creating a product (related to adding a new case) and making a decision (related to problem solving) was seen in the prototype.</td>
</tr>
</tbody>
</table>
Table 18

Visibility of Micro-Strategy: Cases – Likert Ratings

<table>
<thead>
<tr>
<th>Strategy</th>
<th>Element</th>
<th>None</th>
<th>Low</th>
<th>Somewhat Low</th>
<th>Somewhat High</th>
<th>High</th>
</tr>
</thead>
<tbody>
<tr>
<td>The problem-driven environment provides cases for support of problem-solving performance (Jonassen, 1999).</td>
<td>1. Problem-related, indexed cases are provided as a problem-solving resource</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2. Cases present a range of perspectives to produce cognitive flexibility and transfer of learning</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 19

Visibility of Micro-Strategy: Cases – Comments

<table>
<thead>
<tr>
<th>Element</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A. Through prompting and a search utility for retrieving cases, the tool does support use of cases for scaffolding or supplanting learner memory for both adding a case and solving a problem, according to one evaluator.</td>
</tr>
<tr>
<td></td>
<td>B. An evaluator notes that the tool does support indexing of cases.</td>
</tr>
<tr>
<td>2</td>
<td>A. One evaluator noted that the tool supports transferring lessons from past cases to solving new problems.</td>
</tr>
<tr>
<td></td>
<td>B. Transferring lessons from past cases to the “problem” of adding a new case holds questionable value, concluded an evaluator.</td>
</tr>
</tbody>
</table>
Table 20

**Visibility of Micro-Strategy: Information Resources – Likert Ratings**

<table>
<thead>
<tr>
<th>Strategy</th>
<th>Element</th>
<th>None</th>
<th>Low</th>
<th>Somewhat Low</th>
<th>Somewhat High</th>
<th>High</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The problem-driven environment provides <em>information resources</em> such as Web repositories of text documents and graphics for support of problem-solving performance (Jonassen, 1999).</td>
<td>1. Rich, directly-related, well-organized information resources for just-in-time use</td>
<td>2</td>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 21

**Visibility of Micro-Strategy: Information Resources – Comments**

<table>
<thead>
<tr>
<th>Element</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A. An evaluator noted that the tool supports uploading and “attaching” exhibits (case relevant documents) to newly added cases, which the evaluator viewed as information resources.</td>
</tr>
</tbody>
</table>
Table 22

**Visibility of Micro-Strategy: Cognitive Tools – Likert Ratings**

<table>
<thead>
<tr>
<th>Strategy</th>
<th>Element</th>
<th>None</th>
<th>Low</th>
<th>Somewhat Low</th>
<th>Somewhat High</th>
<th>High</th>
</tr>
</thead>
<tbody>
<tr>
<td>The environment provides <em>cognitive tools</em> such software templates and database shells for support of problem-solving performance (Jonassen, 1999).</td>
<td>1. Problem representation tools: For creating models, such as a visualization tool</td>
<td></td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2. Knowledge modeling tools: For articulating what has been learned, such as a hypermedia or database authoring tool</td>
<td></td>
<td>2</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3. Performance-support tools: For freeing the learner from repetitive, lower-order tasks, such as a database shell</td>
<td></td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>4. Information searching tools: For gathering information, such as a search engine</td>
<td></td>
<td>2</td>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 23

**Visibility of Micro-Strategy: Cognitive Tools – Comments**

<table>
<thead>
<tr>
<th>Element</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A. One evaluator did not see tools for creating models.</td>
</tr>
</tbody>
</table>
| 3 | A. An evaluator did see template support for guiding a particular way of thinking about problems, and observed: “it’s important to think carefully about what you ask them to do when they fill out the form.”

B. This same evaluator thought templates might in some cases require lower-order thinking rather than free the learner from that level of thinking, for example, where templates involve entry of demographic information. The evaluator felt this use of templates would be more appropriate in case construction than in case-based problem solving. |
| 4 | A. “A nice search tool” provides definite support for information gathering, said one evaluator. |
Table 24

Visibility of Micro-Strategy: Conversation and Collaborative Tools – Likert Ratings

<table>
<thead>
<tr>
<th>Strategy Element Comment</th>
<th>None</th>
<th>Low</th>
<th>Somewhat Low</th>
<th>Somewhat High</th>
<th>High</th>
</tr>
</thead>
<tbody>
<tr>
<td>The problem-driven environment provides conversation and collaborative tools for support of problem-solving performance (Jonassen, 1999).</td>
<td>1. Dialog and collaboration for solving problems and building a computerized knowledge base</td>
<td>2</td>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 25

Visibility of Micro-Strategy: Conversation and Collaborative Tools – Comments

<table>
<thead>
<tr>
<th>Element</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A. &quot;I was very unclear about this being a computer-mediated conversation.....,” said an evaluator.</td>
</tr>
<tr>
<td></td>
<td>B. Dialogic/collaborative capabilities &quot;could be a useful feature,&quot; speculated one evaluator.</td>
</tr>
<tr>
<td></td>
<td>C. Strictly speaking, a learning community is more egalitarian than the hierarchy of expertise operative in the tool, one evaluator said. (Researcher’s note on relevance of this comment: learning communities emphasize dialogic and collaborative learning strategies.)</td>
</tr>
<tr>
<td></td>
<td>D. The preceding evaluator gave the tool high marks for support of a computerized knowledge base where contributions are shared, evaluated, revised, extended, synthesized, and reformulated, saying, &quot;this seems to be what it [the tool] does best.&quot;</td>
</tr>
</tbody>
</table>
Table 26

Visibility of Micro-Strategy: Social and Contextual Support – Likert Ratings

<table>
<thead>
<tr>
<th>Strategy</th>
<th>Element</th>
<th>None</th>
<th>Low</th>
<th>Somewhat Low</th>
<th>Somewhat High</th>
<th>High</th>
</tr>
</thead>
<tbody>
<tr>
<td>The problem-driven environment provides social and contextual support such as purpose statement and scenarios to facilitate effective functioning of the environment (Jonassen, 1999).</td>
<td>1. Provide information about purpose of the tool</td>
<td>2</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2. Provide information about intended users</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3. Provide information about intended usage scenarios</td>
<td>2</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>4. Provide information about prerequisite knowledge and skills</td>
<td>2</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>5. Provide information about the tool’s hardware/software requirements</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 27

Visibility of Micro-Strategy: Social and Contextual Support – Comments

<table>
<thead>
<tr>
<th>Element</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-5</td>
<td>A. One evaluator generally rated the tool high in this category, especially with regard to provision of information about intended users and scenarios, and allocation of space to listing hardware/software requirements in a FAQ section,</td>
</tr>
<tr>
<td>5</td>
<td>B. This same evaluator did not see where pre-requisite skills and knowledge are identified in the tool.</td>
</tr>
</tbody>
</table>
Table 28

Visibility of Micro-Strategy: Modeling – Likert Ratings

<table>
<thead>
<tr>
<th>Strategy</th>
<th>Element</th>
<th>None</th>
<th>Low</th>
<th>Somewhat Low</th>
<th>Somewhat High</th>
<th>High</th>
</tr>
</thead>
<tbody>
<tr>
<td>The problem-driven environment provides <em>modeling</em> for support of problem-solving performance (Jonassen, 1999).</td>
<td>1. Behavioral (overt) modeling to demonstrate experienced, skilled, problem-solving performance, such as worked example</td>
<td></td>
<td></td>
<td>1</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2. Cognitive (covert) modeling to demonstrate experienced, skilled problem-solving performance, such as “thinking aloud”</td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

Table 29

Visibility of Micro-Strategy: Modeling – Comments

<table>
<thead>
<tr>
<th>Element</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-2</td>
<td>A. The tool keeps modeling at a set level regardless of increasing learner expertise, noted one evaluator.</td>
</tr>
<tr>
<td>1</td>
<td>A. One evaluator sees the tool’s potential for support of the worked example type of modeling, but does not see this support in the first iteration of the tool.</td>
</tr>
<tr>
<td>2</td>
<td>A. An evaluator acknowledges that users are, in fact, prompted to articulate their problem-solving processes, but prefers to reserve judgment on the tool’s support for cognitive modeling pending future iterations of the prototype.</td>
</tr>
</tbody>
</table>
Table 30

Visibility of Micro-Strategy: Coaching – Likert Ratings

<table>
<thead>
<tr>
<th>Strategy</th>
<th>Element</th>
<th>None</th>
<th>Low</th>
<th>Somewhat Low</th>
<th>Somewhat High</th>
<th>High</th>
</tr>
</thead>
<tbody>
<tr>
<td>The problem-driven environment should offer coaching support for problem-solving performance (Jonassen, 1999).</td>
<td>1. Shows relevance, stimulating confidence, motivating</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2. Supports self-monitoring and self-regulating performance</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3. Supports reflection on performance</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>4. Corrects flawed understanding by questioning</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>5. Corrects flawed understanding by prompting reflection</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>6. Corrects flawed understanding by clarifying an outcome</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>7. Corrects flawed understanding by probing confidence</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>8. Corrects flawed understanding by suggesting dissonant views</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
</tr>
</tbody>
</table>
Table 31

Visibility of Micro-Strategy: Coaching – Comments

<table>
<thead>
<tr>
<th>Element</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-8</td>
<td>A. An evaluator observed that instructor and/or system feedback needs to occur throughout case-based learning activities, such as by “interjecting some questions as my students thought through cases.”</td>
</tr>
<tr>
<td>1</td>
<td>A. One evaluator judged the tool to do an &quot;excellent&quot; job on positively influencing perception of relevance, confidence, and motivation, saying &quot;I especially like your discussion of the use of historical cases being similar to what you do in the grocery store.”</td>
</tr>
<tr>
<td>2</td>
<td>A. An evaluator did not see support for self-monitoring and self-regulation of problem solving performance, and wishes to await the next iteration for evaluating support for self-monitoring and self-regulation of case construction performance.</td>
</tr>
</tbody>
</table>
| 3       | A. The tool prompts reflecting on and monitoring of experience in problem solving: “it asks them to judge the adequacy of their solution, reflect on aspects of new problem that are similar to old problem, etc.,” according to an evaluator.  
B. An evaluator did not see support for reflection on and monitoring of performance, although the evaluator thought the tool provided performance feedback by means of rubric and allowed performance modification. |
| 6       | A. One evaluator did not see support for clarification of causes. |
| 7       | A. An evaluator did not see support for probing degree of confidence. |
| 8       | A. An evaluator did not see support for dissonant views, although the evaluator observed that the tool asked the user to consider alternative solutions in problem solving. |

Table 32

Visibility of Micro-Strategy: Scaffolding – Likert Ratings

<table>
<thead>
<tr>
<th>Strategy</th>
<th>Element</th>
<th>None</th>
<th>Low</th>
<th>Somewhat Low</th>
<th>Somewhat High</th>
<th>High</th>
</tr>
</thead>
<tbody>
<tr>
<td>The problem-driven environment should provide scaffolding (general or systemic) support for learner problem-solving performance (Jonassen, 1999).</td>
<td>1. Adjust task difficulty</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2. Supplant learner performance</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3. Offer alternative assessments</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 33

Visibility of Micro-Strategy: Scaffolding – Comments

<table>
<thead>
<tr>
<th>Element</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-3</td>
<td>A. The evaluator asked, “did I miss [the elements of adjusting task difficulty, supplanting learner performance, and offering alternative assessments]?”</td>
</tr>
</tbody>
</table>
Table 34

Overall Visibility of Strategies – Likert Ratings

<table>
<thead>
<tr>
<th>Strategy Element</th>
<th>None</th>
<th>Low</th>
<th>Somewhat Low</th>
<th>Somewhat High</th>
<th>High</th>
</tr>
</thead>
<tbody>
<tr>
<td>An evaluation of the overall visibility of the one macro-strategy and eight micro strategies.</td>
<td>Overall visibility</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 35

Overall Visibility of Strategies – Comments

<table>
<thead>
<tr>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>A. One evaluator rated the tool’s resource and learning potential to be “high” with “great potential” for appropriate learner populations.</td>
</tr>
<tr>
<td>B. “I was very impressed by the operational capabilities of the tool,” said one evaluator.</td>
</tr>
<tr>
<td>C. The tool can be productively used with diverse learner populations, according to an evaluator.</td>
</tr>
<tr>
<td>D. Elective, as-needed use of the tool will favorably affect intrinsic motivation, observed an evaluator.</td>
</tr>
<tr>
<td>E. One evaluator wanted to see a second iteration with more sample user-entered content before drawing a final conclusion as to whether the tool promotes a teacher-centered or learner-centered approach.</td>
</tr>
<tr>
<td>F. One evaluator was &quot;impressed with the evaluation section&quot;: (a) &quot;I believe you provide users with a lot of feedback on how to write a case for the [prototype], and (b) “there is the potential for strong modeling and coaching in the development of that [case construction] skill set.”</td>
</tr>
<tr>
<td>G. An evaluator saw merit in abandoning the tool’s problem-solving component while continuing the design and development of its case construction component to produce a tool that might fill a niche among similar Web-based tools: “it might be best to explore the option of simplifying the tool.”</td>
</tr>
</tbody>
</table>
Each of the eleven two-table sets in this section (Tables 36-57) generally includes: (a) a table stating a usability principle (Nielsen, 1994; Nielsen & Molich, 1990) and listing representative elements of that principle (Pierotti, 2005), and (b) a table summarizing comments relative to visibility of the specified elements in the prototype.

Table 36

**Usability Principle 1: Visibility of System Status**

<table>
<thead>
<tr>
<th>Usability Principle</th>
<th>Representative Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>The system should continuously keep the user informed as to what is going on (system status) through timely feedback (Nielsen, 1994; Nielsen &amp; Molich, 1990; Pierotti, 2005).</td>
<td>1. Titling or labeling</td>
</tr>
<tr>
<td></td>
<td>2. Appropriate location of menu items</td>
</tr>
<tr>
<td></td>
<td>3. Domain-consistent menu terminology</td>
</tr>
<tr>
<td></td>
<td>4. Feedback: Item selection status</td>
</tr>
<tr>
<td></td>
<td>5. Feedback: System status</td>
</tr>
<tr>
<td></td>
<td>6. Feedback: Operation status</td>
</tr>
<tr>
<td></td>
<td>7. Multi-screen navigational aids</td>
</tr>
</tbody>
</table>

Table 37

**Visibility of Usability Principle 1 – Comments**

<table>
<thead>
<tr>
<th>Element</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A. One evaluator suggested adding just a bit of introductory text (a sentence or two) to pages and page elements such as menus. In the latter case, the text would state that menu options link to specified learning modules (sets of tasks). This text would be helpful on first visits while less needed on repeat visits.</td>
</tr>
<tr>
<td>7</td>
<td>A. A different evaluator commented that labels are needed for guiding the user from task to task on pages of the learning modules, although the layout of these pages was judged to be “good.”</td>
</tr>
</tbody>
</table>
Table 38

Usability Principle 2: Match between System and the Real World

<table>
<thead>
<tr>
<th>Usability Principle</th>
<th>Representative Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>The system should speak the user’s language, using logical, natural world words, phrases, concepts, conventions, and order (Nielsen, 1994; Nielsen &amp; Molich, 1990; Pierotti, 2005).</td>
<td>1. Icons</td>
</tr>
<tr>
<td></td>
<td>2. Menus: Parallel titles; logical organization</td>
</tr>
<tr>
<td></td>
<td>3. Related parts connected</td>
</tr>
<tr>
<td></td>
<td>4. Colors</td>
</tr>
<tr>
<td></td>
<td>5. Prompts</td>
</tr>
</tbody>
</table>

Table 39

Visibility of Usability Principle 2 – Comments

<table>
<thead>
<tr>
<th>Element</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>A. An evaluator suggested making clear to users that internationalized terms are being used on data entry forms, for example, terms like “province” referring to the region of a country.</td>
</tr>
</tbody>
</table>

Table 40

Usability Principle 3: User Control and Freedom

<table>
<thead>
<tr>
<th>Usability Principle</th>
<th>Representative Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>The user should be able to exercise user control and appropriately select and sequence tasks (Nielsen, 1994; Nielsen &amp; Molich, 1990; Pierotti, 2005).</td>
<td>1. Forward/backward navigation among parts</td>
</tr>
<tr>
<td></td>
<td>2. Multiple undo and cancel functions</td>
</tr>
<tr>
<td></td>
<td>3. Arranging multiple windows</td>
</tr>
<tr>
<td></td>
<td>4. Confirmation of high-stakes commands</td>
</tr>
<tr>
<td></td>
<td>5. User commands control operations</td>
</tr>
<tr>
<td></td>
<td>6. Broad rather than deep menus</td>
</tr>
<tr>
<td></td>
<td>7. Field copying and editing</td>
</tr>
</tbody>
</table>

Table 41

Visibility of Usability Principle 3 – Comments

<table>
<thead>
<tr>
<th>Element</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-7</td>
<td>A. An evaluator judged that the user, in general, had control over navigation.</td>
</tr>
<tr>
<td>3</td>
<td>A. Another evaluator suggested that multiple windows open at once might adversely affect user control and suggested that all content be kept in a single window.</td>
</tr>
</tbody>
</table>
Table 42

Usability Principle 4: Consistency and Standards

<table>
<thead>
<tr>
<th>Usability Principle</th>
<th>Representative Elements</th>
</tr>
</thead>
</table>
| Basic computer platform standards should be followed in the design of the system to be consistent with user experience and avoid confusion (Nielsen, 1994; Nielsen & Molich, 1990; Pierotti, 2005). | 1. Formatting  
  2. Titling  
  3. Windows  
  4. Menus  
  5. Fields  
  6. Attention-getting techniques  
  7. Colors  
  8. Prompts  
  9. Navigation |

Table 43

Visibility of Usability Principle 4 – Comments

<table>
<thead>
<tr>
<th>Element</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>A. An evaluator noted some perceived inconsistencies in form button labeling, specifically, use of both <em>Save</em> and <em>OK/Done</em> as labels.</td>
</tr>
</tbody>
</table>
| 9       | B. One evaluator “liked” the site map serving as a conceptual rather than the typical access map, noting that it was helpful in showing orientation of each component within the tool: “when I saw this [conceptual site map], it helped me put some pieces together ... so I like this.” The evaluator thought that [the prototype’s] dynamically-generated pages may pose a challenge for creating the typical Map access page.  
  C. According to one evaluator, the [tool’s] logo should link to the home page rather than the Map page.  
  D. *Help, Map,* and *Log Out* links need to be brought together at the top of each or most page(s) in a new menu and standardized location, thus removing *Help* from the main menu, according to another evaluator.  
  E. One evaluator recommended using the same rendition of the main menu on both a component’s introductory and its menu-based navigational pages in order to reinforce a theme throughout the pages of the tool.  
  F. One evaluator felt the Web-standard horizontal rather than vertical configuration of bread crumb navigational elements at the top of module pages would more quickly communicate the “you are here” location. |
Table 44

Usability Principle 5: Error Recognition, Diagnosis, and Recovery

<table>
<thead>
<tr>
<th>Usability Principle</th>
<th>Representative Elements</th>
</tr>
</thead>
</table>
| The system should help the user recognize, diagnose, and recover from errors with error messages written in plain language with no codes (Nielsen, 1994; Nielsen & Molich, 1990; Pierotti, 2005). | 1. Aural error signal  
2. Prompts: Tone, user control, etc.  
3. Error messages: Severity, remedy, etc. |

Table 45

Visibility of Usability Principle 5 – Comments

<table>
<thead>
<tr>
<th>Element</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-3</td>
<td>A. One evaluator did not know how or where to activate error messages in the prototype in order to evaluate them.</td>
</tr>
</tbody>
</table>

Table 46

Usability Principle 6: Error Prevention

<table>
<thead>
<tr>
<th>Usability Principle</th>
<th>Representative Elements</th>
</tr>
</thead>
</table>
| The system is designed to prevent occurrence of errors in the first place with good, plainly written error messages, which is a better strategy than helping the user recover from them (Nielsen, 1994; Nielsen & Molich, 1990; Pierotti, 2005). | 1. Multi-window navigation  
2. Higher/lower menu coordination  
3. Fewer-screen data entry  
4. Case blindness  
5. Least qualifier keys  
6. Field default values |

Table 47

Visibility of Usability Principle 6 – Comments

<table>
<thead>
<tr>
<th>Element</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>No comments.</td>
<td></td>
</tr>
</tbody>
</table>
Table 48

**Usability Principle 7: Recognition Rather than Recall**

<table>
<thead>
<tr>
<th>Usability Principle</th>
<th>Representative Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Make objects, actions, and options visible. The user should not have to remember</td>
<td>1. Window activity status</td>
</tr>
<tr>
<td>information from one part of the dialogue to another. Instructions for use of the</td>
<td>2. Layout</td>
</tr>
<tr>
<td>system should be visible or easily retrievable whenever appropriate (Nielsen, 1994;</td>
<td>3. Distinguishing elements</td>
</tr>
<tr>
<td></td>
<td>5. Mapping controls to actions</td>
</tr>
<tr>
<td></td>
<td>6. Attention getting</td>
</tr>
<tr>
<td></td>
<td>7. Type</td>
</tr>
<tr>
<td></td>
<td>8. Color</td>
</tr>
</tbody>
</table>

Table 49 - Visibility of Usability Principle 7 – Comments

**Visibility of Usability Principle 7 – Comments**

<table>
<thead>
<tr>
<th>Element</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>No comments.</td>
</tr>
</tbody>
</table>

Table 50

**Usability Principle 8: Flexibility and Efficiency of Use**

<table>
<thead>
<tr>
<th>Usability Principle</th>
<th>Representative Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>The system should be designed to meet the needs of various users differing on such</td>
<td>1. Flexibility: Mouse/keyboard alternatives</td>
</tr>
<tr>
<td>criteria as experience, cognitive ability, culture, and language (Nielsen, 1994;</td>
<td>2. Flexibility: User level error messages</td>
</tr>
<tr>
<td></td>
<td>4. Efficiency: Skip unneeded detail</td>
</tr>
</tbody>
</table>

Table 51 - Visibility of Usability Principle 8 – Comments

**Visibility of Usability Principle 8 – Comments**

<table>
<thead>
<tr>
<th>Element</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>One evaluator noted that the Web platform does not support the creation of shortcut- or macro-accelerators allowing experienced users to bypass unnecessary detail, but went on to say that the Web platform does support the presenting a range of numerous alternatives to the user.</td>
</tr>
</tbody>
</table>
### Usability Principle 9: Aesthetic and Minimalist Design

<table>
<thead>
<tr>
<th>Usability Principle</th>
<th>Representative Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dialogs should contain only minimal (all/only) information and uncluttered by non-essential information, and should be aesthetically presented (Nielsen, 1994; Nielsen &amp; Molich, 1990; Pierotti, 2005).</td>
<td>1. Aesthetics including use of white space</td>
</tr>
<tr>
<td></td>
<td>2. Minimalism including brief titles, labels</td>
</tr>
</tbody>
</table>

#### Table 53

**Visibility of Usability Principle 9 – Comments**

<table>
<thead>
<tr>
<th>Element</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-2</td>
<td>A. One evaluator thought that the interface design was “aesthetically pleasing,” with “fields [that] are well defined and easy to use.”</td>
</tr>
<tr>
<td></td>
<td>B. An evaluator gave high marks to the design of the Background component’s introductory page (or Home page), and presumably, all component introductory pages since they share the same design. The evaluator said: “you've done a very nice job on the first page ... the look and feel is really nice ... aesthetically it is great ... professional orientation ... laid out well ... it is clean ... it is clear ... good palette ... good discrimination next to your background ... font is good ... buttons all work ... nice button orientation ... nice margins ... nice graphic design.&quot; (However, the evaluator suggested that the page be re-sized from 1024x768 to 800x600 screen resolution.) This evaluator recommended carrying the design of this page forward to all pages of the prototype.</td>
</tr>
<tr>
<td></td>
<td>C. This same evaluator drew a contrast between design of the Background component’s introductory page discussed above and that component’s menu-based navigational page. Specifically, the evaluator thought that the latter page’s menu for learning modules contained too many attention-getters, such as <em>both</em> checkmark-type icons for marking menu items <em>and</em> boldfaced font for titling and describing menu items. This menu also contained nonessential information, and was thereby rendered less visually comprehensible: “there is lots of stuff [on this menu].” The evaluator concluded.</td>
</tr>
<tr>
<td></td>
<td>D. An evaluator suggested that all pages should open in a single window rather than multiple windows, which might confuse the user’s orientation within the tool.</td>
</tr>
<tr>
<td></td>
<td>E. One evaluator felt purpose, usage, and architecture of the tool needed clarification in the Background component’s introduction to the tool, including the distinction between instructor and student sides of the tool and the two levels of the tool – each component’s introduction followed by that component’s modules – and not three levels as implied in the discussion of three “page types.”</td>
</tr>
<tr>
<td></td>
<td>F. The number of <em>Save</em> buttons on forms needs to be reduced, according to one evaluator.</td>
</tr>
</tbody>
</table>
Usability Principle 10: Help and Resources

<table>
<thead>
<tr>
<th>Usability Principle</th>
<th>Representative Elements</th>
</tr>
</thead>
</table>
| Although a system that can be used without help is preferable, the system should offer help and documentation that is task- and procedure-oriented, searchable, and not very large (Nielsen, 1994; Nielsen & Molich, 1990; Pierotti, 2005). | 1. Visually distinct  
2. Layout  
3. Relevant, goal-oriented  
4. Accurate, complete, clear  
5. Descriptive, procedural, interpretive  
6. Navigational  
7. Context-sensitive  
8. Help/work coordination |

Visibility of Usability Principle 10 – Comments

<table>
<thead>
<tr>
<th>Element</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A. One evaluator noted that the Help link conceptually belongs in a new menu made up of Help, Map, and Log Out options at a standard, top-of-the-page location more than in the main menu, which has links to major components (Background, Operations, Help, Evaluation, Exploration, and Administration), and therefore suggested thus relocating the Help link.</td>
</tr>
</tbody>
</table>

Overall Visibility of Usability Principles

An evaluation of the overall visibility of usability principles 1-10.

Overall Visibility of Usability Principles – Comments

<table>
<thead>
<tr>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>A. One evaluator noted both positive aspects of the design and recommended improvements. Positive aspects included: (a) an overall “very pleasing and well thought out” interface, (b) well-labeled parts, (c) pleasing and compatible colors, and (d) “very easy to read” text. Suggested improvements included: (a) reducing the number of windows simultaneously open and (b) making the tool’s logo to link back to the Background component’s introductory page or Home page rather than the Map page.</td>
</tr>
</tbody>
</table>

Lessons can be learned from the experience of designing and validating the prototype in Stage 7, and the formative evaluation data in Stage 8. These lessons are the topic of the next chapter, which ends with a suggestion of implications for future directions.
Chapter 4 – Lessons Learned and Future Directions

In the previous chapters, two research questions motivating this developmental research study have been asked and answered. Factors relevant to designing a comprehensive CBR tool for complex thinking were identified (Question 1). These factors formed the foundation for conducting a design project, including identification of specifications, creation and validation of a prototype, and conduct of a formative evaluation (Question 2). The current chapter is devoted to summarizing lessons learned from this research study. These lessons are based on the process of creating a product (the prototype), the product itself, and interaction between the two. Twenty-four lessons are classified according to the three major steps of the development research methodology that has guided the conduct of this study: lessons relating to design, to development, and to evaluation. The chapter concludes with implications for future directions, with specific regard to production of the comprehensive CBR tool, additional research, and practice.

Lessons Learned

Lessons Relating to Design

1. An effort can be highly-motivated, theory-/research-based, and systematically and productively executed, but over-attention to immediately producing the ultimate outcome can prolong development and finally produce an intricate outcome requiring simplicity. A realistic, recommended approach for eventually achieving the ultimate outcome shares these positive characteristics of motivation, information, and execution, but is also fundamentally incremental. This incremental strategy is recommended within the procedures of formative evaluation and rapid prototyping (Dick, Carey, & Carey, 2005; Tessmer, 1993). Time spent in this research study on at once producing the ultimate outcome could have alternatively been spent on reducing the scale of the design project, and fitting the manageable reduction into a long range research agenda that included all the many variables of interest to the researcher.

2. Divergent and convergent processes should be alternated at a rate to prevent an initial, extended divergent phase from prolonging development. Divergent thinking is essential to identifying a range of innovative options. However, there is an intellectual enticement to the divergent process that can, if yielded to, postpone an equally necessary convergent review of those options – making feasible/infeasible determinations, selection, trial implementation, and evaluation –followed in turn by more needed divergent thinking, and so on. Less extended
divergent processes and more frequently occurring convergent processes would have achieved a more timely completion of the design project.

3. Although theoretical speculation can be intellectually productive, theory’s value is realized in its timely application. A significant amount of time and effort was spent on performing the literature review. Information from hundreds of books, refereed journals, and Web-based publications was identified, digitized, collected, organized, and integrated. This extensive background will undoubtedly contribute to a long-term research agenda. However, while this research study presented significant needs for information, the breadth and depth of retrieved information exceeded those needs and even distracted from most efficiently meeting those needs. The time would have been better spent on expeditiously applying the theory, completing the design project, and leaving some of the theoretical speculations for future studies.

4. Conducting a developmental research study provided extensive, in-depth, and reflective experience in systems analysis (Akker et al, 1999; Dick, Carey, & Carey, 2005; Gustafson & Branch, 1997, 2002). Systems analysis is a basic skill for instructional designers. The developmental research study can be thought of as a system, and the design project within it a sub-system, each with a specified purpose and integrated components for efficiently and effectively achieving that purpose. The overarching purpose of the study was twofold: (a) stimulate the doctoral student’s learning process and (b) articulate and report the learning outcomes as they directly benefitted the researcher’s professional development and as they can potentially benefit practice in the field. Within this framework, two research questions were used to drive and guide the research process. The parts of the study were two research questions driving the research process, a literature review for answering the first question, a design project for answering the second question, and the current chapter to culminate the process by identification and description of lessons learned. The immediate purpose of the design project was to create a comprehensive CBR tool for complex thinking. The secondary purpose was to serve as the learning stimulus for the research study. Parts of that tool consisted of specifications for each of the needs, goals, objectives, assessments, and strategies; a specification-based prototype; and a formative evaluation examining visibility of specification-enabling structures in the prototype.

5. Summarizing specifications at the beginning of Stage 7 provided useful experience in a reductionist, analytical process applicable to clarifying any set of variables. This process involved analyzing relationships among
thirty-two original specifications and subsequently combining equivalent specifications within a particular stage, thereby reducing the number of specifications. Closely related but discrete specifications from different stages were cross-referenced. Combinations and cross references served to clarify the design and evaluation of the prototype.

**Lessons Relating to Development**

Different types of lessons are presented in this section. The first lesson relates to gaining professional experience in a basic process of instructional design. Lessons 2-6 note proposed prototype improvements informed by formative evaluation findings. Lessons 7-15 constitute a component-by-component analysis of perceived prototype viability reevaluated in light of formative evaluation findings and the general design experience.

1. *Answering the research questions provided comprehensive, in-depth, and reflective experience in the instantiation process (Reigeluth, 1999).* A fundamental principle of instructional design is to ground design in educational theory (Dick, Carey, & Carey, 2005; Ertmer & Newby, 1993). Instantiation (Reigeluth, 1999) is therefore a basic process of instructional design. Accordingly, instantiation occurred along the route mapped in Figure 21:

- **Question 1** (answered in Chapter 2)
  - Seven theory- and/or researched-based factors or characteristics and, by extension, design requirements, from the literature review

- **Question 2** (answered in Chapter 3)
  - 29 specifications generated by applying the Dick, Carey, and Carey (2005) model to the requirements (Stages 1-6 of the model)
  - Prototype structures operationalizing the specifications (Stage 7)
  - Formative evaluation examining visibility of strategies in the structures (Stage 8)

*Figure 21.* The instantiation process implemented in this research study.

2. *Despite the documented validity of specifications and demonstrated operationalization of them in a prototype, unforeseen prototype-related factors can hinder an expected evaluation outcome.* In general,
incorporation of specifications in structures of the prototype was demonstrated in Stage 7. However, Stage 8 evaluation data suggested that evaluators did not always see these structures. A combination of factors contributed to this situation. The prototype and all of its elements consisted of approximately one hundred fifty files and one hundred twenty pages. Visibility of structures was at times obscured by a technique of rapidly presenting large amounts of information. In addition, while the broad outlines of the prototype were drawn and many of its structures were fully developed, the development of some structures was beyond the scope of this research study. Content in these latter structures would have perhaps helped evaluators better understand the functioning of the existing prototype. Finally, use of the prototype involves regularly performing task behaviors by entering responses to task prompts in the textboxes, radio buttons, and other objects of ubiquitous data-entry forms. Therefore, hypothetical, representative user responses included in the form objects of a preliminary iteration of the prototype but not included in the final prototype could have also helped evaluators better understand the operation of prototype structures.

3. A simple, elegant design is more usable than a complex, intricate design, although the latter may be theory-/research-based, valid, and systematically executed. Akin to the proverbial phenomenon of trees obscuring the forest, a complex design can overwhelm the user with information and thereby obscure major points and frustrate the user. A simple, elegant design may still present a large amount of information, but it gradually leads into the large amount of information using a general-to-specific or top-down technique rather than creating a sudden overload of information. The users should have been more gradually led into the large amounts of information presented in the prototype. Furthermore, the prototype could have been simplified by reducing the number of levels, number of windows, amount of information in menus, and number of buttons on some forms.

4. Any departure from established Web formatting standards may increase cognitive load. Simple, easily achieved changes such as making the tool logo link to the home page rather than the map page; gathering and placing log out, map, and help links at the tops of pages; and changing bread crumb navigational elements from vertical to horizontal format would improve this aspect of the design.

5. Never assume an understanding of the “obvious.” A designer immersed in the design and sometimes forgetting the user’s inexperience with the design is naturally prone to over-assuming. The tops of menus were initially labeled using only the component name, such as Background Navigator. The word Modules should be added to this labeling – Background Modules Navigator – to more exactly specify the domain of the menu. This
type of revision is needed by first-time visitors, while less needed by repeat visitors, according to one evaluator. Since a production version of the tool would be Web-based and thus available worldwide, data-entry forms in the prototype include international terminology such as province for a political sub-division of some countries. One evaluator suggested pointing out this internationalized terminology of the form to avoid confusing unsuspecting, uninformed users.

6. Users appreciate a design that is aesthetic, theory-/research-based, systematically executed, and professional. The home page of the Background component received high marks in this area from one evaluator. Presumably this appreciation extended to the home pages of the other five components since they all shared the same design.

7. The original concept can, in fact, lead to development of a production tool. However database/Web platform functionality that was simulated by the prototype is indispensable to this actualization of the concept. A major contribution of this study was the matching of learning needs with ideally supportive technological characteristics. In general, if cognitive (internal) dynamic memory capabilities are to be mimicked by the tool (Bush, 1945; di Sessa, 2000; Englebardt, 1962; Nelson, 1978, 1983; Pea, 1985; Salomon, Perkins, & Globerson, 1991), robust technology-based (external) dynamic memory capabilities offered by the database/Web platform are needed. Specifically, realization of the original concept requires the usability, interactivity, flexibility, capacity, and data processing capabilities offered by this platform for both developer-entered and user-entered content.

8. In general, users’ comprehension of the original concept could be increased by consolidating the numerous elements of the prototype. Representative examples of beneficial changes will serve to clarify this need for consolidation. A component’s introduction and its module menus located on separate pages need to be consolidated on the same page. Component introductions can be seen on the component initial pages depicted in Figures 14-19, and module menus are outlined in Appendix A. Moreover, according to formative evaluation data, reducing the number of windows could minimize chances of navigational confusion. In addition, procedures and accompanying tasks of case editing and content customization modules in the Administration component should be appropriately combined to occupy fewer pages. Finally, the number of components needs to be reduced from six to four by eliminating one (Evaluation) and demoting another (Help) from the level of component to the subordinate level of component support.
9. Although the original concept was comprehensively and systematically researched, planned and mapped, its effectiveness and visibility was clouded by content bombardment. With the close proximity and immersion that naturally come in a design project, the designer can sometimes over-assume and unconsciously forget users’ novice status and consequently besiege or bombard users with content – perhaps very unfamiliar content. The content for the prototype was systematically mapped from the books, book chapters, refereed journal articles, and Web-based resources that define the CBR field. Therefore, a major improvement that could be made in the prototype would be to replace any instances of content bombardment with a gradual, simple-to-complex or broad-to-specific presentation of content. A simplified representation of content would quickly and clearly show the “forest” – the major points – while providing ready, contextualized access to the “trees” as needed – the more complex, analytical representations. In addition to this holistic-to-analytical strategy, multimedia elements could be implemented to clarify key concepts of CBR.

10. The Operations component was originally conceived to be the heart of the proposed tool. Evaluator feedback indicated the component achieved its purpose, but it still needs to be further emphasized, developed, and refined. The Operations component was clearly distinguished in the Map page’s configuration graphic, which was found to be useful in finding orientation and understanding the workings of the tool. In all other representations of the tool’s configuration, the Operations component should be visually and/or spatially separated and emphasized to distinguish its central role in the tool.

In view of the tool’s purpose to support performance of CBR, the general, facilitated learning format of the Operations component should be retained, further developed, and refined. Further development of the template-enabled format should include extending task prompts in accordance with the detailed Stage 2 goal step and sub-step analysis. For example, the prototype includes a two-part task that prompts the user to state a lesson learned from the problem-solving experience, and to trace the process of abstracting the lesson (general/abstract knowledge) from concrete events of the case (specific/concrete knowledge). Further development will include breaking this two-part task down into six steps or behaviors outlined in the Stage 2 goal analysis: review, analyze, evaluate, extract, assemble, and construct. In addition, major time must be spent on developing context-sensitive (task relevant) content for Models (behavioral modeling) and Procedures (cognitive modeling), refining it, and inserting it. Finally, context-sensitive specifications of task-relevant subordinate skills must be developed and inserted.
Other context sensitive additions may be theoretically desirable and interesting, and probably practically useful. However, feasibility of many of these features is questionable. For example, development of context-sensitive Web resource lists is probably desirable but, in view of the Web’s dynamic content base, an intranet would offer a more stable environment for these resources. (An intranet is an organization’s, usually a corporation’s, private network that operates like the Internet [Webopedia, 2007]). Also, although perhaps beneficial, cost would probably outweigh benefit. Similarly, context sensitive assessment standards is a theoretically desirable and interesting feature, and perhaps a practically useful feature but, in view of the tool’s compatibility with a range of domain and program assessment alternatives, this feature is not critical to the immediate utilization of the tool and could be held as an option for long-range exploration and development.

11. Search utilities are needed for the stated purposes, but must be reconfigured, further developed, and refined. The prototype contains two separate search utilities for cases and help topics. Both utilities are needed for performing necessary, basic data retrieval operations according to specified criteria. However, the separate sets of search criteria need further development and refinement. The status of the help utility should be changed from the level of a component to the subordinate level of component-support. This could be achieved through insertion of integrated “help” links. The case utility deserves both component and component-support status. Component status would incorporate guided exploration of content; component-support status, a search capability only.

12. The basic, originally-conceived notion of a Background component is still seen to address an integral, performance-support role by providing subject/topic-based information prior to and during performance of procedures and tasks. However, this component – retaining its current organization – requires extension, expansion, enhancement, and integration. Topic-based information in a Background component is still needed to support the Operations component which is at the heart of the tool. Offering modular, database-friendly flexibility, the current subject/topic hierarchical organization should be retained but the five existing subjects should be extended to increase depth of coverage in their respective areas. (See five existing subjects in Appendix A, item 1, Background component.) Additional topics need to be added primarily for providing a more in-depth, up-front treatment of CBR as a psychological construct, a cognitive model, and a learning strategy. Both basic topics and just-in-time reference topics should be included, reflecting the existing topic typology. More multimedia enhancements should be added
for quickly clarifying technical, unfamiliar concepts such as similarity assessment. Background content should be integrated in the all-important Operations component by inserting context-sensitive links in the latter.

13. In general, the Administration component in its present form could serve as the instructor “side” of a production tool with some further development. For this to occur, its two existing, basic functions need to be reconfigured, extended, expanded, and refined. The now-dispersed elements of each of the two basic functions – case editing and content editing – should be consolidated on one page. The general outlines of the feedback mechanism in the case editing function require more detailed delineation. An element should be added to the case editing function for entering domain and/or program-specific content in the topic-based modules of the Background component.

14. The Map page beneficially served an orientation function in the prototype and could do the same in a production tool. If the reconfigurations described in the preceding sections are made, the spatial and visual orientation function of the Map page also must be revised to reflect these changes. Also, the function might be enhanced by adding brief descriptive information to each component representation on the page and color for showing separation and emphasis.

15. Useful experience was gained in balancing the design potential and limitations of a particular technology. Basic knowledge of relational database design and development was a prerequisite to creating the prototype in Stage 7 (Element K a-f; Ensor, 1997; Forta, 1998; Getz, Litwin, & Gilbert, 1999; Gruber, M., 2000; Hernandez, 1997; Macromedia, 2003; Microsoft Corporation, 2003; Pitt, Updike, & Guthrie, 2002). This basic knowledge included identifying and describing fields (object attributes in fixed-sequence columns), including field data types; identifying and creating tables, primary keys, and table relationships; constructing queries; data entry; and defining user classes, permissions, and rights. Designing a prototype simulation of a database-enabled application does not require database programming expertise, but does require understanding database programming issues that both enable and constrain design of the simulation.

Lessons Relating to Evaluation

1. Selection of appropriate data collection and processing methods should be based on study needs, and not on generally innovative characteristics of a particular method. Overall, the Web-based survey tool used in this design project offers an innovative method for creating online instruments and collecting and processing data.
However, use of the tool required a certain commitment of time and effort while the project did not require the advanced capabilities of the tool. There were only six evaluators and they were accessible in the local area. Use of the survey tool thus distracted from most efficiently meeting study objectives. Certainly, in general, experience using the tool will potentially contribute to a long-term research agenda. However, in retrospect, simpler, email- or paper-based instrumentation and data collection and spreadsheet-based data processing could have better served needs of this particular project.

2. **Effective instrumentation offers practical implementation.** While an instrument can be generally valid (Nielsen, 1994; Nielsen & Molich, 1990; Pierotti, 2005) and systematically executed, greater length, detail, and technical language can render the instrument impractical and thus distract from best meeting study objectives. This situation was reflected in the usability instrument, but not the strategy instrument. The usability instrument had 176, sometimes technical, Likert-based questions plus eleven prompts for discretionary comments. The shorter, more general, and less technical alternative usability instrument (Brinck, Gergle, & Wood, 2002) substituted by one evaluator similarly addressed usability issues, but did so in a more manageable format and thus effectively met study objectives. That instrument consisted of only nine sets of criteria, each including five to 13 straightforward, yes/no questions that, as implemented, included useful written and oral comments elaborating on the yes/no responses.

3. **An in-depth explanation of evaluation criteria and items, including graphical enhancement, may be required for helping achieve validity of an instrument.** Providing a simple overview of criteria/items may be insufficient. Introducing the strategy instrument with the in-depth type of explanation, making clear the relationships among the macro-strategy and the eight supportive micro strategies, may have increased the validity of this instrument.

4. **Thoughtful written and/or oral comments (qualitative data) can prove particularly useful for clarifying and extending quantitative data, and thereby serve to significantly improve the design of a tool.** Written comments from the two prescribed instruments as well as the alternative usability instrument proved especially useful to improving the prototype. The interview held in conjunction with use of the alternative usability instrument produced notably useful written and oral comments. Therefore, somehow enabling and influencing the evaluator to provide more than discretionary comments in both prescribed instruments may have added to the decidedly useful qualitative
data already collected. In addition, more planned face-to-face evaluations would have likely produced especially useful data.

**Future Directions**

**Implications for Production**

Production and deployment of the tool ultimately resulting from a continued development of the specification-based prototype would involve further elaboration and refinement of content. It would also involve creating a working relational database and associated Web-page templates to support the functionality simulated in the prototype, making a needed connection between database and templates, and installing them on Web-accessible computing hardware. In general, all these activities would require the professional services of instructional designers, subject matter experts (SMEs), professional database programmers, and Web developers. They would involve contributions by more formative evaluators, potentially including both experts (Tessmer, 1993) and users, the latter in one-on-one, small-group, and field testing formats (Dick, Carey, & Carey, 2005). With a large but scalable structure, all, or selected elements of the tool could be developed according to program needs, resource allocation, and time availability. The estimated timeline for full production is approximately fifteen weeks, while that for a smaller-scale option could be significantly less and would vary according to scale. Rates for services of SME and database programmers can vary considerably by geographic region.

**Implications for Additional Research**

Research to validate the various components of a completed version of the tool could immediately, logically follow the production described in the preceding section. This research could include exploring the option of “seeding” the tool’s database and case library as appropriate with cases from existing libraries, such as those identified and listed in Table 2 (Kolodner, Owensby, & Guzdial, 2004). Longer-range research could include analyzing the completed tool for identification of particular qualities, characteristics, or attributes (Levie & Dickie, 1973; Lockee, Moore, & Burton, 2001), and examining their instructional impact. A consideration of attributes could include type of information representation, sensory modalities, level of realism, and feedback capability (Levie & Dickie, 1973). For example, this research could examine an attribute’s impact on improving a particular problem-solving skill (Hernandez-Serrano & Jonassen, 2003).
Implications for Practice

The tool could be integrated in various educational settings. For example, it could incorporate multicultural education elements of a professional education program delivered in the context of content courses, multicultural courses, and/or clinical components as appropriate. Potential users would rank at the advanced undergraduate, graduate, and post-graduate levels.

The selection of topics used in the tool should be based on needs established through vigorous research. In the above case, research results indicate that professional education programs have insufficiently and/or ineffectively addressed pedagogical issues arising from the increasingly multicultural PK12 classroom (Barksdale, Richards, Fisher, Wuthrick, Hammons, Grisham, & Richmond, 2002; Grant, cited in Neuharth-Pritchett, Reiff, & Pearson, 2001; National Center for Education Statistics cited in Neuharth-Pritchett, Reiff, & Pearson, 2000; National Council for Accreditation of Teacher Education, 2002; Neuharth-Pritchett, Reiff, and Pearson, 2000; Neuharth-Pritchett, Reiff, and Pearson, 2001; Pai and Adler, 2001; Tozer, Ukpokodu Violas, & Senese, 2002; Ukpokodu, 2002; United States Census Bureau cited in Potthoff, Dinsmore, & Moore, 2001). Lack of access to diverse PK-12 populations (Virginia Department of Education, 2006; Wood, Schroeder, & Guterbock, 2002) could constrain multicultural education initiatives.

Consistent with research-based recommendations (Barksdale et al., 2002) and professional education standards (National Council for Accreditation of Teacher Education, 2002), the tool could be applied to supporting a case-based, advanced, and useful type of learning ultimately leading to best practice of multicultural education. Cases of vicarious multicultural education experiences provided by the tool would likely offer compensatory benefit to culturally homogenous areas.

Conclusion

This research study produced positive responses to both research questions posed in Chapter 1. The first question was:

(1) Is it possible to identify a relevant set of theory- and/or research-based, operationalizable factors that should be incorporated into a comprehensive CBR tool for engaging complex thinking?
In response to this question, seven factors or characteristics – and by extension, design requirements – were identified and summarized in the literature review. These collectively served as the foundation for designing the comprehensive CBR tool.

The second research question was:

(2) Given these factors, is it possible to create a comprehensive CBR tool that is both consistent with the factors and has the potential to engage complex thinking?

In response to this second question, Stages 1-6 of the Dick, Carey, and Carey (2005) model were subsequently applied to these requirements and consequently generated twenty-nine stage-by-stage specifications for defining the design of the tool. Identifying and describing specifications included combining duplicate specifications from the same stage and cross-referencing closely related, but non-duplicating specifications from different stages. All the specifications from Stages 1-6 were summarized at the beginning of Stage 7 and then applied to creating a prototype of the tool. The researcher’s subsequent internal analysis of the prototype found structures that operationalized the specifications, and thereby answered the second research question. However, formative evaluators did not always see these structures, due mainly to their visibility and to amount of hypothetical, representative task responses.

Despite this unexpected outcome, the collection of specifications represents a major contribution of the study. Unlike the existing, more narrowly defined CBR tools, the comprehensive tool was intentionally created to fill a need for a cross-domain tool readily offering capabilities for generation of both CBR-specific and domain-specific content. In general, experience and data show this to have been a comprehensively, systematically researched and executed but overly informative and analytical study that succeeded with a simplification of aims and a focus on major points at its conclusion. In the end, prominence of these major points established the contribution of the specifications clarified through operationalization and formative evaluation.
References


Iowa Department of Education. (1989, April). *A guide to developing higher order thinking across the curriculum.* (ERIC Document Reproduction Service No. ED 306550)


Lockee, B., Moore, D., & Burton, J. (2001). Old concerns with new distance education research; media comparison studies have fatal flaws that taint distance learning research relying on this strategy. *Educause Quarterly, Number 2*, 60-62.


Appendix A

Module Menus

1. Background

- Module 1 Subject: Professional Issues [issues to be resolved using CBR]
  - Topic 1.1 Your Problem Domain and Domain Issues
- Module 2 Subject: Application
  - Topic 2.1 Achievement of Best Practice
- Module 3 Subject: Sample Output [a sample case]
  - Topic 3.1 How Do You Grow Rice?
- Module 4 Subject: Program Integration [integration of the tool in instructional, training, and/or workplace contexts]
  - Topic 4.1 Pre-service Educators Use *caseThinker at Loyola College for Teachers
- Module 5 Subject: Basic Facts
  - Topic 5.1 FAQ

2. Operations

- Module 1 Procedure: Providing Case-Relevant Personal Data
  - Task 1.1 Personal ID
- Module 2 Procedure: Interpreting the Problem
  - Task 2.1 Problem Analysis (Problem Interpretation)
  - Task 2.2 Problem-Solving Goals
  - Task 2.3 Goal Constraints
  - Task 2.4 Summary
  - Task 2.5 Recommended Problem Indexing Descriptors
  - Task 2.6 Applicable Historical Cases
- Module 3 Procedure: Solving the Problem
  - Task 3.1 Adaptable Elements and Methods
  - Task 3.2 The Solution Concept and Rationale
  - Task 3.3 Non-selected Feasible Solutions and Rationales
  - Task 3.4 Non-selected Infeasible Solutions and Rationales
  - Task 3.5 Projected Results
  - Task 3.6 Summary
  - Task 3.7 Recommended Indexing Descriptors
- Module 4 Procedure: Evaluating the Solution
  - Task 4.1 Solution Rating
  - Task 4.2 Outcome Analysis (including expected and unexpected results)
  - Task 4.3 Solution Repair (including avoidance strategy)
  - Task 4.4 Repaired Solution Rating
  - Task 4.5 Repaired Solution Outcome Analysis (including expected and unexpected results)
  - Task 4.6 Summary
  - Task 4.7 Recommended Indexing Descriptors
- Module 5 Procedure: Abstracting Lessons Learned
  - Task 5.1 Lessons Learned
  - Task 5.2 Summary
  - Task 5.3 Recommended Indexing Descriptors
- Module 6 Procedure: Adding Case-Related Materials
  - Task 6.1 Exhibits
- Module 7 Procedure: Responding to Feedback and Revising the Case
  - Task 7.1 Feedback Response (Revisions)
3. Help

- Module 1 Search Utility: Finding Models and Procedures
- Module 2 Search Utility: Finding Web Resources [links within the tool to Web resources outside the tool]
- Module 3 Search Utility: Finding Glossary Definitions

4. Evaluation

- Module 1 Evaluation Instrument: Evaluating Operations Component Performance

5. Exploration

- Module 1 Search Utility: Finding a Set of related Cases
- Module 2 Search Utility: Finding a Particular Case

6. Administration

- Module 1 Procedure: Evaluating the Case and Providing Feedback [evaluating data entered by user in the Operations component and consequently providing feedback to that user]
  - Task 1.1 Evaluation and Feedback
- Module 2 Procedure: Assigning the Case Indexing Descriptors
  - Task 2.1 Assigning Descriptors for the Problem
  - Task 2.2 Assigning Descriptors for the Solution
  - Task 2.3 Assigning Descriptors for the Outcome
  - Task 2.4 Assigning Descriptors for Lessons Learned
- Module 3 Procedure: Enhancing Case Content [by linking the case to relevant Web resources and glossary definitions]
  - Task 3.1 Adding or Deleting Enhancements
- Module 4 Procedure: Editing Evaluation Standards
  - Task 4.1 Creating, Revising, or Deleting Standards
- Module 5 Procedure: Editing Web Resources [lists of links to Web resources external to the tool]
  - Task 5.1 Creating, Revising, or Deleting Web Resources
- Module 6 Procedure: Editing Model/Procedure Sets
  - Tasks 6.1 Creating, Revising, or Deleting Sets
- Module 7 Procedure: Editing Glossary Definitions
  - Task 7.1 Creating, Revising, or Deleting Entries
Appendix B

Formative Evaluation Instrument: Visibility of Strategies

Thank you for agreeing to participate in the formative evaluation of *caseThinker. This is the very first evaluation. Evaluation results will be used to create an action plan for improving the next iteration.

Overview of the Design Project

► Purpose and Objectives

The *caseThinker concept and design resulted from a developmental research study conducted to meet the dissertation requirement for the PhD degree in Curriculum and Instruction (Instructional Technology).

*caseThinker is a dynamic Web application designed to support achieving best practice in a range of problem domains through the case-based approach to interpreting and resolving problems. The case-based approach centers on applying stories or cases of problem interpretation and problem solving to resolving similar issues arising in the professional setting. In a case-based reasoning cycle, each subsequent episode of addressing a new problem is fully documented, stored, and indexed for application to future, similar problems.

Potential *caseThinker implementation domains include professional education, instructional design, medicine, nursing, law, and architecture.

*caseThinker features a Web-based user interface. The content accessed through this interface is stored in and retrieved from a relational database. The database foundation significantly increases *caseThinker's functionality. (A relational database is a powerful computer application for organizing, storing, revising, and retrieving data.)

The relational database is not yet active in *caseThinker. For the formative evaluation, you have been asked to review a prototype of *caseThinker. The relational database would be added at a later time.

► Targeted Users

The expected users of *caseThinker are pre-novice, novice, and experienced practitioners (both pre- and in-service practitioners) in any of the preceding problem domains who seek to achieve best practice through the case-based approach.

► Scenarios

Users could utilize *caseThinker in a variety of scenarios including the following:

• Getting an overview of applying the case-based method to achieving best practice, and how the *caseThinker concept and design support that application.

• Adding a new case to the *caseThinker case library, which includes reviewing relevant evaluation standards, integrated help topics, cases stored and indexed in the *caseThinker case library, and feedback from the instructor.

• Adapting solutions from the past to resolving a new issue of the present, which includes reviewing relevant evaluation standards, help topics, cases, and feedback from the instructor.

• Providing feedback (coaching) to facilitate case-based practice.
• As allowed by the *caseThinker design, customizing evaluation standards and help resources, which include Web
resources, model/procedure sets, and glossary entries.

The Evaluation

 ► Two Evaluations

Two dimensions of *caseThinker will be evaluated:

• Visibility of strategies
• Usability principles

You will be participating in either or both of these evaluations according to evaluation instructions you received by
email. This evaluation instrument includes an Evaluation Checklist for Visibility of Strategies (see below).

 ► The Evaluation Procedure

Important note: As explained below, this formative evaluation is conducted online and must be finished in a single
online session.

Completion of this formative evaluation involves using two online artifacts: (a) this formative evaluation instrument
and (b) the *caseThinker prototype.

To get started, leave this instrument window open and go to the prototype now by clicking the link above. Log into
the prototype using your existing username and password. (If you do not have these, you have been provided
alternative login information.)

With both the instrument and prototype windows now open, look over all the items of the instrument and quickly
click through all the prototype features to see the scope of the instrument and prototype. Each instrument item
consists of a strategy followed by questions for evaluating visibility of the specified strategy in the prototype. The
prototype consists of a Map page and six components.

Iteratively revisit instrument items and *caseThinker features for an in-depth analysis and evaluation of
*caseThinker compliance with design principles. Finalize and record your evaluation on the instrument form. Click
the “Submit” button at the end of the instrument when finished. You need to complete your evaluation in a single
online session. Please submit your evaluation by October 14 if at all possible.

 ► Your Questions

Please direct any questions about your participation in this formative evaluation to [the designer’s/researcher’s
name], the designer and researcher, as follows:

• Email: [the designer's/researcher’s email address]
• Voice Telephone: [the designer's/researcher’s voice telephone number]

The Evaluation Checklist for strategies is presented in the next section.
Evaluation Checklist

Directions

Nine strategies are identified, listed, and briefly stated below. Evaluate visibility of the strategies in the prototype by answering the questions that follow the strategies. For each of the nine strategies:

(1) After each question, rate the visibility of the strategy on a scale of 0-5

(2) After the set of questions under each principle, add any discretionary comments to clarify your ratings; for example, these comments might include justification for your ratings, noting both negative and positive aspects of the design, and recommendations for improvement

At the end of the nine strategies and in Item 10, rate the overall visibility and add discretionary comments to the overall rating.

Click the “Submit” button at the end of the Checklist when finished.

Note: Creation of this checklist was based on the researcher’s analysis of Jonassen’s model for designing online problem-based learning environments:


Questions for Items 1-10
(a total of 31 questions with discretionary comments)

1. The Question, Case, Issue, Problem, or Project (The “Problem”)

● The environment should be centered on a challenging question, case, issue, problem, or project (collectively and generically termed the “problem”) as the goal that drives the learning process.

1.1 Is a problem at the center of the environment, driving the activity, and requiring the exploration, assemblage, and application of domain content (that is, problem-solving is used to drive the learning of content and not just to practice previously learned content)?
0-NA 1-None 2-Low 3-Somewhat low 4-Somewhat high 5-High

1.2 Does the ill-structured or ill-defined problem (not a "textbook" problem) that drives the learning process produce learner ownership of the problem and meaningful learning?
0-NA 1-None 2-Low 3-Somewhat low 4-Somewhat high 5-High

1.3 Does the tool support the analysis of the problem situation (the problem context), including the community of participants and the physical, socio-cultural, and organizational climate of the problem?
0-NA 1-None 2-Low 3-Somewhat low 4-Somewhat high 5-High

1.4 Does the tool support activities of creating a product or making a decision?
0-NA 1-None 2-Low 3-Somewhat low 4-Somewhat high 5-High

Discretionary comments for questions 1.1 to 1.4:

2. Related Cases

● The problem-driven environment should provide related cases for support of problem-solving performance.
2.1 Will the fully functioning tool (the production version) provide related, indexed cases or stories that recount concrete, skilled, problem-solving experience (vicarious experience and advice) for supporting or supplanting the novice problem-solver’s memory of useful problem-solving experience?  
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

2.2 Will the fully functioning tool (the production version) provide related, indexed cases or stories that present multiple perspectives on problems (problem complexity) for achieving cognitive flexibility and transfer of learning from the learning context (the "classroom") to real-world settings?  
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

Discretionary comments for questions 2.1 to 2.2:

3. Information Resources

● The problem-driven environment should provide information resources for support of problem-solving performance.

3.1 Will the fully functioning tool (the production version) provide rich, relevant, well-organized information resources for just-in-time selection and application to problem-solving activities, including the information resources of Web-based information banks or repositories of text documents, graphics, sound resources, video, and animations?  
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

Discretionary comments for question 3.1:

4. Cognitive Tools

● The problem-driven environment should provide cognitive tools for support of problem-solving performance.

4.1 Does the tool provide computer-based cognitive tools (problem/task representation tools) for creating models (visualizations) helpful in understanding concepts?  
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

4.2 Does the tool provide computer-based cognitive tools such as a database and hypermedia authoring or simulation modeling applications for representing or articulating what has been learned and what it means?  
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

4.3 Does the tool provide computer-based cognitive tools (performance support tools) such as software templates, note-taking space, calculator utilities, or database shells that automate repetitive, algorithmic tasks and thereby free learners from these lower-order tasks for spending more time on higher-order, problem-solving activities?  
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

4.4 Does the tool provide information-seeking, gathering, or searching tools such as a search engine for finding information applicable to successfully completing problem-solving activities?  
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

Discretionary comments for questions 4.1 to 4.4:

5. Conversation and Collaborative Tools

● The problem-driven environment should provide computer-mediated conversation and collaboration tools for support of problem-solving performance.
5.1 Does the tool support dialog and collaboration among members of a learning community for solving problems and building a computerized knowledge base where member contributions are shared, evaluated, revised, extended, synthesized, and reformulated?

0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

Discretionary comments for question 5.1:

6. Social and Contextual Support

- The problem-driven environment should provide social/contextual support for effective functioning of the environment.

6.1 Does implementation of the tool include providing information relating to purpose of the tool, intended users, intended scenarios, prerequisite knowledge and skills (including required computer skills), and hardware/software requirements?

6.1.1 Providing information relating to purpose of the learning tool?

0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

6.1.2 Providing information relating to intended users?

0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

6.1.3 Providing information relating to intended scenarios?

0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

6.1.4 Providing information relating to prerequisite knowledge and skills?

0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

6.1.5 Providing information relating to hardware/software requirements?

0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

Discretionary comments for questions 6.1.1 to 6.1.5:

Preface to Items 7-9:

Learning strategies (student behaviors) are to be distinguished from instructional strategies (teacher behaviors and/or tool “behaviors”).

Accordingly, learning strategies are listed:

(a) Exploration
(b) Articulation
(c) Reflection

Instructional strategies are listed:

(a) Modeling
(b) Coaching
(c) Scaffolding

Items 7-9 address the latter.
7. Modeling

● The problem-driven environment should provide modeling support for problem-solving performance.

7.1 Does the tool demonstrate experienced, skilled problem-solving performance or “how do I do this?/show me” through an example of overt activity (behavior), such as the step-by-step, discrete actions/decisions represented in a worked example?
0-NA 1-None 2-Low 3-Somewhat low 4-Somewhat high 5-High

7.2 Does the tool demonstrate skilled problem-solving performance through articulation of covert, cognitive activity, such as thinking aloud or articulating reasoning/decision-making during performance?
0-NA 1-None 2-Low 3-Somewhat low 4-Somewhat high 5-High

Discretionary comments for questions 7.1 to 7.2:

8. Coaching

● The problem-driven environment should provide coaching support for problem-solving performance.

8.1 Does the tool communicate the relevance of learning activities and boost learner confidence to promote motivation?
0-NA 1-None 2-Low 3-Somewhat low 4-Somewhat high 5-High

8.2 Does the tool monitor and regulate the learner performance through such means as: hints/helps; prompting for reasoning steps; and prompting for application of helpful cognitive tools?
0-NA 1-None 2-Low 3-Somewhat low 4-Somewhat high 5-High

8.3 Does the tool provoke and question learners to reflect on or monitor/analyze personal performance, and engage appropriate problem-solving strategies?
0-NA 1-None 2-Low 3-Somewhat low 4-Somewhat high 5-High

8.4 Does the tool challenge flawed understanding and models through the following means: (a) questioning (“does your model explain ...?”), (b) prompting for reflection on actions (“why did you ...?”), (c) clarifying an outcome (“why did you get this result?”), (d) probing for degree of confidence in response (“how sure are you?”), and (e) suggesting dissonant views or interpretations (“what about this other view on the issue?”)?

8.4.1 Through questioning?
0-NA 1-None 2-Low 3-Somewhat low 4-Somewhat high 5-High

8.4.2 Through prompting for reflection on actions?
0-NA 1-None 2-Low 3-Somewhat low 4-Somewhat high 5-High

8.4.3 Through clarifying an outcome?
0-NA 1-None 2-Low 3-Somewhat low 4-Somewhat high 5-High

8.4.4 Through probing for degree of confidence?
0-NA 1-None 2-Low 3-Somewhat low 4-Somewhat high 5-High

8.4.5 Through suggesting dissonant views or interpretations?
0-NA 1-None 2-Low 3-Somewhat low 4-Somewhat high 5-High

Discretionary comments for questions 8.1 to 8.4.5:
9. Scaffolding

- The problem-driven environment should provide scaffolding support for problem-solving performance.

9.1 Does the tool provide as-needed systemic support for learner problem-solving performance through the following means: (a) adjusting task difficulty level (provide an easier task); (b) supplanting learner performance (redesigning a task or allowing short-term use of a cognitive tool); or (c) offering alternative assessments appropriate for problem-driven learning (providing a worked problem to depict the assessment target)?

9.1.1 Through the means of adjusting task difficulty level?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

9.1.2 Through the means of supplanting learner performance?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

9.1.3 Through the means of offering alternative assessments?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

Discretionary comments for questions 9.1.1 to 9.1.3:

10. Overall Visibility of Strategies

10.1 Are specified strategies visible in the prototype?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

Discretionary comments for question 10.1:
Appendix C

Formative Evaluation Instrument: Usability Principles

A Note on the Organization of the Instrument

This instrument is divided into five parts as follows:

Part I: Items 1-3 of 11 items (a total of 39 questions with discretionary comments)
Part II: Item 4 of 11 items (a total of 37 questions with discretionary comments)
Part III: Items 5-6 of 11 items (a total of 30 questions with discretionary comments)
Part IV: Items 7-8 of 11 items (a total of 41 questions with discretionary comments)
Part V: Items 9-11 of 11 items (a total of 29 questions with discretionary comments)

Thank you for agreeing to participate in the formative evaluation of *caseThinker. This is the very first evaluation. Evaluation results will be used to create an action plan for improving the next iteration.

Overview of the Design Project

► Purpose and Objectives

The *caseThinker concept and design resulted from a developmental research study conducted to meet the dissertation requirement for the PhD degree in Curriculum and Instruction (Instructional Technology).

*caseThinker is a dynamic Web application designed to support achieving best practice in a range of problem domains through the case-based approach to interpreting and resolving problems. The case-based approach centers on applying stories or cases of problem interpretation and problem solving to resolving similar issues arising in the professional setting. In a case-based reasoning cycle, each subsequent episode of addressing a new problem is fully documented, stored, and indexed for application to future, similar problems.

Potential *caseThinker implementation domains include professional education, instructional design, medicine, nursing, law, and architecture.

*caseThinker features a Web-based user interface. The content accessed through this interface is stored in and retrieved from a relational database. The database foundation significantly increases *caseThinker’s functionality. (A relational database is a powerful computer application for organizing, storing, revising, and retrieving data.)

The relational database is not yet active in *caseThinker. For the formative evaluation, you have been asked to review a prototype of *caseThinker. The relational database would be added at a later time.

► Targeted Users

The expected users of *caseThinker are pre-novice, novice, and experienced practitioners (both pre- and in-service practitioners) in any of the preceding problem domains who seek to achieve best practice through the case-based approach.

► Scenarios

Users could utilize *caseThinker in a variety of scenarios including the following:
• Getting an overview of applying the case-based method to achieving best practice, and how the *caseThinker concept and design support that application.

• Adding a new case to the *caseThinker case library, which includes reviewing relevant evaluation standards, integrated help topics, cases stored and indexed in the *caseThinker case library, and feedback from the instructor.

• Adapting solutions from the past to resolving a new issue of the present, which includes reviewing relevant evaluation standards, help topics, cases, and feedback from the instructor.

• Providing feedback (coaching) to facilitate case-based practice.

• As allowed by the *caseThinker design, customizing evaluation standards and help resources, which include Web resources, model/procedure sets, and glossary entries.

The Evaluation

► Two Evaluations

Two dimensions of *caseThinker will be evaluated:

• Visibility of strategies
• Usability principles

You will be participating in either or both of these evaluations according to evaluation instructions you received by email. This evaluation instrument includes an Evaluation Checklist for Usability Principles (see below).

► The Evaluation Procedure

Important note: As explained below, this formative evaluation is conducted online and must be finished in a single online session.

Completion of this formative evaluation involves using two online artifacts: (a) this formative evaluation instrument and (b) the *caseThinker prototype.

To get started, leave this instrument window open and go to the prototype now by clicking the link above. Log into the prototype using your existing username and password. (If you do not have these, you have been provided alternative login information.)

With both the instrument and prototype windows now open, look over all the items of the instrument and quickly click through all the prototype features to see the scope of the instrument and prototype. Each instrument item consists of a usability principle followed by questions for evaluating *caseThinker’s compliance with the specified principle. The prototype consists of a Map page and six components.

Iteratively revisit instrument items and *caseThinker features for an in-depth analysis and evaluation of *caseThinker compliance with principles. Finalize and record your evaluation on the instrument form. Click the “Submit” button at the end of the instrument when finished. You need to complete your evaluation in a single online session. Please submit your evaluation by October 14 if at all possible.
Your Questions

Please direct any questions about your participation in this formative evaluation to [the designer’s/researcher’s name], the designer and researcher, as follows:

- Email: [the designer’s/researcher’s email address]
- Voice Telephone: [the designer’s/researcher’s voice telephone number]

The Evaluation Checklist for usability is presented in the next section.

Evaluation Checklist

Directions

Ten usability heuristics are identified, listed, and briefly stated below. Evaluate how well the *caseThinker user interface design complies with the heuristics by answering the questions that follow the heuristics. For each of the ten heuristics:

1. After each question, rate how well the *caseThinker user interface design complies with the heuristic on a scale of 0 to 5
2. After the set of questions under each heuristic, add any discretionary comments to clarify your ratings; for example, these comments might include justification for your ratings, noting both negative and positive aspects of the design, and recommendations for improvement

At the end of the ten heuristics and in Item 11, rate the overall compliance and add discretionary comments to the overall rating.

Click the “Submit” button at the end of the Checklist when finished.

Creation of this checklist was based on J. Nielsen's user interface usability heuristics and D. Pierotti's operationalization of the heuristics:


Questions for Items 1-11
(a total of 176 questions with discretionary comments)

1. Visibility of System Status

- The system should always keep user informed about what is going on, through appropriate feedback within reasonable time.

1.1 Does every display begin with a title or header that describes screen contents?

0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

1.2 Do menu instructions, prompts, and error messages appear in the same place(s) on each menu?

0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High
1.3 In multi-page data entry screens, is each page labeled to show its relation to others?  
0-NA  1-None   2-Low   3-Somewhat low   4-Somewhat high   5-High  

1.4 Is there some form of system feedback for every operator action?  
0-NA  1-None   2-Low   3-Somewhat low   4-Somewhat high   5-High  

1.5 After the user completes an action (or group of actions), does the feedback indicate that the next group of actions can be started?  
0-NA  1-None   2-Low   3-Somewhat low   4-Somewhat high   5-High  

1.6 Is there visual feedback in menus or dialog boxes about which choices are selectable?  
0-NA  1-None   2-Low   3-Somewhat low   4-Somewhat high   5-High  

1.7 Is there visual feedback in menus or dialog boxes about which choice the cursor is on now?  
0-NA  1-None   2-Low   3-Somewhat low   4-Somewhat high   5-High  

1.8 If multiple options can be selected in a menu or dialog box, is there visual feedback about which options are already selected?  
0-NA  1-None   2-Low   3-Somewhat low   4-Somewhat high   5-High  

1.9 Is there visual feedback when objects are selected or moved?  
0-NA  1-None   2-Low   3-Somewhat low   4-Somewhat high   5-High  

1.10 Is the menu-naming terminology consistent with the user's task domain?  
0-NA  1-None   2-Low   3-Somewhat low   4-Somewhat high   5-High  

1.11 Does the system provide visibility: that is, by looking, can the user tell the state of the system and the alternatives for action?  
0-NA  1-None   2-Low   3-Somewhat low   4-Somewhat high   5-High  

1.12 Do GUI menus make obvious which item has been selected?  
0-NA  1-None   2-Low   3-Somewhat low   4-Somewhat high   5-High  

1.13 Do GUI menus make obvious whether deselection is possible?  
0-NA  1-None   2-Low   3-Somewhat low   4-Somewhat high   5-High  

1.14 If users must navigate between multiple screens, does the system use context labels, menu maps, and place markers as navigational aids?  
0-NA  1-None   2-Low   3-Somewhat low   4-Somewhat high   5-High  

Discretionary comments for questions 1.1 to 1.14:  

2. Match between System and the Real World  
● The system should speak the user's language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.  

2.1 Are icons concrete and familiar?  
0-NA  1-None   2-Low   3-Somewhat low   4-Somewhat high   5-High  

2.2 Are menu choices ordered in the most logical way, given the user, the item names, and the task variables?  
0-NA  1-None   2-Low   3-Somewhat low   4-Somewhat high   5-High
2.3 If there is a natural sequence to menu choices, has it been used?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

2.4 Do related and interdependent fields appear on the same screen?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

2.5 Do the selected colors correspond to common expectations about color codes?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

2.6 When prompts imply a necessary action, are the words in the message consistent with that action?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

2.7 For question and answer interfaces, are questions stated in clear, simple language?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

2.8 Do menu choices fit logically into categories that have readily understood meanings?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

2.9 Are menu titles parallel grammatically?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

Discretionary comments for questions 2.1 to 2.9:

3. User Control and Freedom

● Users should be free to select and sequence tasks (when appropriate), rather than having the system do this for them. Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Users should make their own decisions (with clear information) regarding the costs of exiting current work. The system should support undo and redo.

3.1 In systems that use overlapping windows, is it easy for users to rearrange windows on the screen?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

3.2 In systems that use overlapping windows, is it easy for users to switch between windows?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

3.3 When a user's task is complete, does the system wait for a signal from the user before processing?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

3.4 Are users prompted to confirm commands that have drastic, destructive consequences?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

3.5 Is there an "undo" function at the level of a single action, a data entry, and a complete group of actions?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

3.6 Can users cancel out of operations in progress?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

3.7 Can users reduce data entry time by copying and modifying existing data?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

3.8 Are character edits allowed in data entry fields?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High
3.9 Are menus broad (many items on a menu) rather than deep (many menu levels)?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

3.10 If the system has multiple menu levels, is there a mechanism that allows users to go back to previous menus?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

3.11 If users can go back to a previous menu, can they change their earlier menu choice?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

3.12 Can users move forward and backward between fields or dialog box options?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

3.13 If the system has multipage data entry screens, can users move backward and forward among all the pages in the set?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

3.14 If the system uses a question and answer interface, can users go back to previous questions or skip forward to later questions?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

3.15 Can users easily reverse their actions?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

3.16 If the system allows users to reverse their actions, is there a retracing mechanism to allow for multiple undos?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

Discretionary comments for questions 3.1 to 3.16:

4. Consistency and Standards

● Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.

4.1 Have industry or company formatting standards been followed consistently in all screens within a system?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

4.2 Has heavy use of all uppercase letters on a screen been avoided?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

4.3 Do abbreviations not include punctuation?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

4.4 Are there salient visual cues to identify the active window?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

4.5 Does each window have a title?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

4.6 Are vertical and horizontal scrolling possible in each window?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

4.7 Does the menu structure match the task structure?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High
4.8 Have industry or company standards been established for menu design, and are they applied consistently on all menu screens in the system?
0-NA 1-None 2-Low 3-Somewhat low 4-Somewhat high 5-High

4.9 Are menu choice lists presented vertically?
0-NA 1-None 2-Low 3-Somewhat low 4-Somewhat high 5-High

4.10 Are menu titles either centered or left-justified?
0-NA 1-None 2-Low 3-Somewhat low 4-Somewhat high 5-High

4.11 Are menu items left-justified, with the item number or mnemonic preceding the name?
0-NA 1-None 2-Low 3-Somewhat low 4-Somewhat high 5-High

4.12 Do embedded field-level prompts appear to the right of the field label?
0-NA 1-None 2-Low 3-Somewhat low 4-Somewhat high 5-High

4.13 Do on-line instructions appear in a consistent location across screens?
0-NA 1-None 2-Low 3-Somewhat low 4-Somewhat high 5-High

4.14 Are field labels and fields distinguished typographically?
0-NA 1-None 2-Low 3-Somewhat low 4-Somewhat high 5-High

4.15 Are field labels consistent from one data entry screen to another?
0-NA 1-None 2-Low 3-Somewhat low 4-Somewhat high 5-High

4.16 Are fields and labels left-justified for alpha lists and right-justified for numeric lists?
0-NA 1-None 2-Low 3-Somewhat low 4-Somewhat high 5-High

4.17 Do field labels appear to the left of single fields and above list fields?
0-NA 1-None 2-Low 3-Somewhat low 4-Somewhat high 5-High

4.18 Are attention-getting techniques used with care, such as intensity, size, font, blink, and color?
0-NA 1-None 2-Low 3-Somewhat low 4-Somewhat high 5-High

4.19 Are attention-getting techniques used only for exceptional conditions or for time-dependent information?
0-NA 1-None 2-Low 3-Somewhat low 4-Somewhat high 5-High

4.20 Are there no more than four to seven colors, and are they far apart along the visible spectrum?
0-NA 1-None 2-Low 3-Somewhat low 4-Somewhat high 5-High

4.21 Have pairings of high-chroma, spectrally extreme colors been avoided?
0-NA 1-None 2-Low 3-Somewhat low 4-Somewhat high 5-High

4.22 Are saturated blues avoided for text or other small, thin line symbols?
0-NA 1-None 2-Low 3-Somewhat low 4-Somewhat high 5-High

4.23 Is the most important information placed at the beginning of the prompt?
0-NA 1-None 2-Low 3-Somewhat low 4-Somewhat high 5-High

4.24 Are user actions named consistently across all prompts in the system?
0-NA 1-None 2-Low 3-Somewhat low 4-Somewhat high 5-High

4.25 Are system objects named consistently across all prompts in the system?
0-NA 1-None 2-Low 3-Somewhat low 4-Somewhat high 5-High
4.26 Do field-level prompts provide more information than a restatement of the field name?
0-NA 1-None 2-Low 3-Somewhat low 4-Somewhat high 5-High

4.27 For question and answer interfaces, are the valid inputs for a question listed?
0-NA 1-None 2-Low 3-Somewhat low 4-Somewhat high 5-High

4.28 Are menu choice names consistent, both within each menu and across the system, in grammatical style and terminology?
0-NA 1-None 2-Low 3-Somewhat low 4-Somewhat high 5-High

4.29 Does the structure of menu choice names match their corresponding menu titles?
0-NA 1-None 2-Low 3-Somewhat low 4-Somewhat high 5-High

4.30 Do abbreviations follow a simple primary rule and, if necessary, a simple secondary rule for abbreviations that otherwise would be duplicates?
0-NA 1-None 2-Low 3-Somewhat low 4-Somewhat high 5-High

4.31 Is the secondary rule used only when necessary?
0-NA 1-None 2-Low 3-Somewhat low 4-Somewhat high 5-High

4.32 Are abbreviated words all the same length?
0-NA 1-None 2-Low 3-Somewhat low 4-Somewhat high 5-High

4.33 Is the structure of a data entry value consistent from screen to screen?
0-NA 1-None 2-Low 3-Somewhat low 4-Somewhat high 5-High

4.34 Is the method for moving the cursor to the next or previous field consistent throughout the system?
0-NA 1-None 2-Low 3-Somewhat low 4-Somewhat high 5-High

4.35 If the system has multipage data entry screens, do all pages have the same title?
0-NA 1-None 2-Low 3-Somewhat low 4-Somewhat high 5-High

4.36 If the system has multipage data entry screens, does each page have a sequential page number?
0-NA 1-None 2-Low 3-Somewhat low 4-Somewhat high 5-High

4.37 Are high-value, high-chroma colors used to attract attention?
0-NA 1-None 2-Low 3-Somewhat low 4-Somewhat high 5-High

Discretionary comments for questions 4.1 to 4.37:

5. Help Users Recognize, Diagnose, and Recover From Errors

- Error messages should be expressed in plain language (no codes).

5.1 Is sound used to signal an error?
0-NA 1-None 2-Low 3-Somewhat low 4-Somewhat high 5-High

5.2 Are prompts stated constructively, without overt or implied criticism of the user?
0-NA 1-None 2-Low 3-Somewhat low 4-Somewhat high 5-High

5.3 Do prompts imply that the user is in control?
0-NA 1-None 2-Low 3-Somewhat low 4-Somewhat high 5-High
5.4 Are prompts brief and unambiguous?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

5.5 Are error messages worded so that the system, not the user, takes the blame?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

5.6 Are error messages grammatically correct?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

5.7 Do error messages avoid the use of exclamation points?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

5.8 Do error messages avoid the use of violent or hostile words?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

5.9 Do error messages avoid an anthropomorphic tone?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

5.10 Do all error messages in the system use consistent grammatical style, form, terminology, and abbreviations?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

5.11 Do messages place users in control of the system?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

5.12 If an error is detected in a data entry field, does the system place the cursor in that field or highlight the error?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

5.13 Do error messages inform the user of the error's severity?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

5.14 Do error messages suggest the cause of the problem?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

5.15 Do error messages provide appropriate semantic information?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

5.16 Do error messages provide appropriate syntactic information?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

5.17 Do error messages indicate what action the user needs to take to correct the error?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

5.18 If the system supports both novice and expert users, are multiple levels of error-message detail available?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

Discretionary comments for questions 5.1 to 5.18:

6. Error Prevention

- Even better than good error messages is a careful design which prevents a problem from occurring in the first place.

6.1 If the database includes groups of data, can users enter more than one group on a single screen?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High
6.2 Have dots or underscores been used to indicate field length?
0-NA 1-None 2-Low 3-Somewhat low 4-Somewhat high 5-High

6.3 Is the menu choice name on a higher-level menu used as the menu title of the lower-level menu?
0-NA 1-None 2-Low 3-Somewhat low 4-Somewhat high 5-High

6.4 Are menu choices logical, distinctive, and mutually exclusive?
0-NA 1-None 2-Low 3-Somewhat low 4-Somewhat high 5-High

6.5 Are data inputs case-blind whenever possible?
0-NA 1-None 2-Low 3-Somewhat low 4-Somewhat high 5-High

6.6 If the system displays multiple windows, is navigation between windows simple and visible?
0-NA 1-None 2-Low 3-Somewhat low 4-Somewhat high 5-High

6.7 Has the use of qualifier keys been minimized?
0-NA 1-None 2-Low 3-Somewhat low 4-Somewhat high 5-High

6.8 Does the system prevent users from making errors whenever possible?
0-NA 1-None 2-Low 3-Somewhat low 4-Somewhat high 5-High

6.9 Does the system warn users if they are about to make a potentially serious error?
0-NA 1-None 2-Low 3-Somewhat low 4-Somewhat high 5-High

6.10 Does the system intelligently interpret variations in user commands?
0-NA 1-None 2-Low 3-Somewhat low 4-Somewhat high 5-High

6.11 Do data entry screens and dialog boxes indicate the number of character spaces available in a field?
0-NA 1-None 2-Low 3-Somewhat low 4-Somewhat high 5-High

6.12 Do fields in data entry screens and dialog boxes contain default values when appropriate?
0-NA 1-None 2-Low 3-Somewhat low 4-Somewhat high 5-High

Discretionary comments for questions 6.1 to 6.12:

7. Recognition Rather Than Recall

- Make objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.

7.1 For question and answer interfaces, are visual cues and white space used to distinguish questions, prompts, instructions, and user input?
0-NA 1-None 2-Low 3-Somewhat low 4-Somewhat high 5-High

7.2 Does the data display start in the upper-left corner of the screen?
0-NA 1-None 2-Low 3-Somewhat low 4-Somewhat high 5-High

7.3 Are multiword field labels placed horizontally (not stacked vertically)?
0-NA 1-None 2-Low 3-Somewhat low 4-Somewhat high 5-High

7.4 Are all data a user needs on display at each step in a transaction sequence?
0-NA 1-None 2-Low 3-Somewhat low 4-Somewhat high 5-High
7.5 Are prompts, cues, and messages placed where the eye is likely to be looking on the screen?
0-NA   1-None   2-Low   3-Somewhat low   4-Somewhat high   5-High

7.6 Have prompts been formatted using white space, justification, and visual cues for easy scanning?
0-NA   1-None   2-Low   3-Somewhat low   4-Somewhat high   5-High

7.7 Do text areas have "breathing space" around them?
0-NA   1-None   2-Low   3-Somewhat low   4-Somewhat high   5-High

7.8 Is there an obvious visual distinction made between "choose one" menu and "choose many" menus?
0-NA   1-None   2-Low   3-Somewhat low   4-Somewhat high   5-High

7.9 Does the system gray out or delete labels of currently inactive soft function keys?
0-NA   1-None   2-Low   3-Somewhat low   4-Somewhat high   5-High

7.10 Is white space used to create symmetry and lead the eye in the appropriate direction?
0-NA   1-None   2-Low   3-Somewhat low   4-Somewhat high   5-High

7.11 Have items been grouped into logical zones, and have headings been used to distinguish between zones?
0-NA   1-None   2-Low   3-Somewhat low   4-Somewhat high   5-High

7.12 Are zones no more than twelve to fourteen characters wide and six to seven lines high?
0-NA   1-None   2-Low   3-Somewhat low   4-Somewhat high   5-High

7.13 Have zones been separated by spaces, lines, color, letters, bold titles, rules lines, or shaded areas?
0-NA   1-None   2-Low   3-Somewhat low   4-Somewhat high   5-High

7.14 Are field labels close to fields, but separated by at least one space?
0-NA   1-None   2-Low   3-Somewhat low   4-Somewhat high   5-High

7.15 Are long columnar fields broken up into groups of five, separated by a blank line?
0-NA   1-None   2-Low   3-Somewhat low   4-Somewhat high   5-High

7.16 Are symbols used to break long input strings into "chunks"?
0-NA   1-None   2-Low   3-Somewhat low   4-Somewhat high   5-High

7.17 Is reverse video or color highlighting used to get the user's attention?
0-NA   1-None   2-Low   3-Somewhat low   4-Somewhat high   5-High

7.18 Is reverse video used to indicate that an item has been selected?
0-NA   1-None   2-Low   3-Somewhat low   4-Somewhat high   5-High

7.19 Are size, boldface, underlining, color, shading, or typography used to show relative quantity or importance of different screen items?
0-NA   1-None   2-Low   3-Somewhat low   4-Somewhat high   5-High

7.20 Are borders used to identify meaningful groups?
0-NA   1-None   2-Low   3-Somewhat low   4-Somewhat high   5-High

7.21 Has the same color been used to group related elements?
0-NA   1-None   2-Low   3-Somewhat low   4-Somewhat high   5-High

7.22 Is color coding consistent throughout the system?
0-NA   1-None   2-Low   3-Somewhat low   4-Somewhat high   5-High
7.23 Is color used in conjunction with some other redundant cue?
0-NA 1-None 2-Low 3-Somewhat low 4-Somewhat high 5-High

7.24 Is there good color and brightness contrast between image and background colors?
0-NA 1-None 2-Low 3-Somewhat low 4-Somewhat high 5-High

7.25 Have light, bright, saturated colors been used to emphasize data and have darker, duller, and desaturated colors been used to de-emphasize data?
0-NA 1-None 2-Low 3-Somewhat low 4-Somewhat high 5-High

7.26 Is the first word of each menu choice the most important?
0-NA 1-None 2-Low 3-Somewhat low 4-Somewhat high 5-High

7.27 Does the system provide mapping: that is, are the relationships between controls and actions apparent to the user?
0-NA 1-None 2-Low 3-Somewhat low 4-Somewhat high 5-High

7.28 Are inactive menu items grayed out or omitted?
0-NA 1-None 2-Low 3-Somewhat low 4-Somewhat high 5-High

7.29 Are there menu selection defaults?
0-NA 1-None 2-Low 3-Somewhat low 4-Somewhat high 5-High

7.30 Do GUI menus offer affordance: that is, make obvious where selection is possible?
0-NA 1-None 2-Low 3-Somewhat low 4-Somewhat high 5-High

7.31 Are there salient visual cues to identify the active window?
0-NA 1-None 2-Low 3-Somewhat low 4-Somewhat high 5-High

7.32 Do data entry screens and dialog boxes indicate when fields are optional?
0-NA 1-None 2-Low 3-Somewhat low 4-Somewhat high 5-High

Discretionary comments for questions 7.1 to 7.32:

8. Flexibility and Minimalist Design

- Accelerators-unseen by the novice user-may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions. Provide alternative means of access and operation for users who differ from the “average” user (e.g., physical or cognitive ability, culture, language, etc.)

8.1 If the system supports both novice and expert users, are multiple levels of error message detail available?
0-NA 1-None 2-Low 3-Somewhat low 4-Somewhat high 5-High

8.2 Does the system provide function keys for high-frequency commands?
0-NA 1-None 2-Low 3-Somewhat low 4-Somewhat high 5-High

8.3 For data entry screens with many fields or in which source documents may be incomplete, can users save a partially filled screen?
0-NA 1-None 2-Low 3-Somewhat low 4-Somewhat high 5-High

8.4 If menu lists are short (seven items or fewer), can users select an item by moving the cursor?
0-NA 1-None 2-Low 3-Somewhat low 4-Somewhat high 5-High
8.5 If the system uses a pointing device, do users have the option of either clicking on fields or using a keyboard shortcut?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

8.6 Does the system offer "find next" and "find previous" shortcuts for database searches?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

8.7 On data entry screens, do users have the option of either clicking directly on a field or using a keyboard shortcut?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

8.8 In dialog boxes, do users have the option of either clicking directly on a dialog box option or using a keyboard shortcut?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

8.9 Can expert users bypass nested dialog boxes with either type-ahead, user-defined macros, or keyboard shortcuts?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

Discretionary comments for questions 8.1 to 8.9:

9. Aesthetic and Minimalist Design

● Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.

9.1 Is only (and all) information essential to decision-making displayed on the screen?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

9.2 If the system uses a standard GUI interface where menu sequence has already been specified, do menus adhere to the specification whenever possible?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

9.3 Are meaningful groups of items separated by white space?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

9.4 Does each data entry screen have a short, simple, clear, distinctive title?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

9.5 Are field labels brief, familiar, and descriptive?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

9.6 Are prompts expressed in the affirmative, and do they use the active voice?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

9.7 Is each lower-level menu choice associated with only one higher level menu?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

9.8 Are menu titles brief, yet long enough to communicate?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

9.9 Are there pop-up or pull-down menus within data entry fields that have many, but well-defined, entry options?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High
Elective comments for questions 9.1 to 9.9:

10. Help and Documentation

- Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user’s task, list concrete steps to be carried out, and not be too large.

10.1 Are online instructions visually distinct?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

10.2 Do the instructions follow the sequence of user actions?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

10.3 Are data entry screens and dialog boxes supported by navigation and completion instructions?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

10.4 Is the help function visible; for example, a key labeled HELP or a special menu?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

10.5 Is the help system interface (navigation, presentation, and conversation) consistent with the navigation, presentation, and conversation interfaces of the application it supports?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

10.6 Navigation: Is information easy to find?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

10.7 Presentation: Is the visual layout well designed?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

10.8 Conversation: Is the information accurate, complete, and understandable?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

10.9 Is the information relevant?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

10.10 Is the information goal-oriented – what can I do with this program?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

10.11 Is the information descriptive – what is this thing for?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

10.12 Is the information procedural – how do I do this task?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

10.13 Is the information interpretive – why did that happen?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

10.14 Is the information navigational – where am I?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

10.15 Is there context-sensitive help?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High
10.16 Can the user change the level of detail available?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

10.17 Can users easily switch between help and their work?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

10.18 Is it easy to access and return from the help system?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

10.19 Can users resume work where they left off after accessing help?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

Discretionary comments for questions 10.1 to 10.19:

11. Overall Compliance

11.1 Does the *caseThinker design comply overall with recommended interface usability heuristics?
0-NA  1-None  2-Low  3-Somewhat low  4-Somewhat high  5-High

Discretionary comments for question 11.1: