DOCUMENT RESUME

ED 418 102 TM 028 196

AUTHOR	Chung, Gregory K. W. K.; Herl, Howard E.; Klein, Davina C. D.; O'Neil, Harold F., Jr.; Schacter, John
TITLE	Estimate of the Potential Costs and Effectiveness of Scaling Up CRESST Assessment Software.
INSTITUTION	National Center for Research on Evaluation, Standards, and Student Testing, Los Angeles, CA.
SPONS AGENCY	Office of Educational Research and Improvement (ED), Washington, DC.
REPORT NO	CSE-TR-462
PUB DATE	1997-12-00
NOTE	33p.
CONTRACT	R305B60002-97
PUB TYPE	Numerical/Quantitative Data (110) -- Reports - Evaluative (142)
EDRS PRICE	MF01/PC02 Plus Postage.
DESCRIPTORS	Computer Assisted Testing; *Computer Software; *Concept Mapping; *Educational Assessment; Integrated Activities; Problem Solving; Scoring; *Simulation; *Technological Advancement; *Test Construction
IDENTIFIERS	Center for Research on Eval Standards Stu Test CA

ABSTRACT

This report examines issues in the scale-up of assessment software from the Center for Research on Evaluation, Standards, and Student Testing (CRESST). "Scale-up" is used in a metaphorical sense, meaning adding new assessment tools to CRESST's assessment software. During the past several years, CRESST has been developing and evaluating a series of software products to measure various types of learning. Collectively, these are called the Integrated Assessment System. The integrated simulation CRESST has developed includes both collaborative and individual concept mapping tasks, a problem-solving search task, an explanation task, and a metacognitive questionnaire. New tools should measure one or more aspects of the CRESST model of learning, be stand-alone or integrate with other tools, have authoring capacity, use automated scoring, and be Internet-deployable. Specifications are listed for eight proposed tools, and each is evaluated for the CRESST requirements. The tools are: (1) Collaborative Concept Mapper, a tool for group use; (2) Flowcharter, a tool to allow students to sequence events; (3) Idea Generator and Evaluator, a tool for group problem solving; (4) Model Simulator, a model building tool; (5) Networked Team Simulator, a team model builder; (6) Multimedia Concept Mapper, a multimedia tool for concept mapping; (7) Outliner, an organizing tool; and (8) Problem Solver, an instrument to measure problem solving processes. (Contains 4 tables and 13 references.) (SLD)

CRESST

National Center for Research on Evaluation, Standards, and Student Testing

Estimate of the Potential Costs and Effectiveness of Scaling Up CRESST Assessment Software

CSE Technical Report 462

Gregory K. W. K. Chung, Howard E. Herl, Davina C. D. Klein
CRESST/University of California, Los Angeles

Harold F. O'Neil, Jr.
University of Southern California/CRESST

John Schacter
CRESST/University of California, Los Angeles

- UCLA Center for the Study of Evaluation

in collaboration with:

- University of Colorado
- NORC, University of Chicago
- LRDC, University of Pittsburgh
- University of California, Santa Barbara
- University of Southern California
- The RAND Corporation

Estimate of the Potential Costs and Effectiveness of Scaling Up CRESST Assessment Software

CSE Technical Report 462

Gregory K. W. K. Chung, Howard E. Herl, Davina C. D. Klein
CRESST/University of California, Los Angeles

Harold F. O'Neil, Jr.
University of Southern California/CRESST

John Schacter
CRESST/University of California, Los Angeles

December 1997

Center for the Study of Evaluation
National Center for Research on Evaluation,
Standards, and Student Testing
Graduate School of Education & Information Studies
University of California, Los Angeles
Los Angeles, CA 90095-1522
(310) 206-1532

The work reported herein was supported under the Educational Research and Development Center Program, PR/Award Number R305B60002, as administered by the Office of Educational Research and Improvement, U.S. Department of Education.

ESTIMATE OF THE POTENTIAL COSTS AND EFFECTIVENESS OF SCALING UP CRESST ASSESSMENT SOFTWARE[1]

Gregory K. W. K. Chung[2], Howard E. Herl[2],
Davina C. D. Klein[2], Harold F. O'Neil, Jr.[3], and John Schacter[2]

In this report, we examine issues in the scale-up of CRESST assessment software. We use the term "scaling up" in a metaphorical sense, meaning what new assessment tools should be added to our assessment software. Criteria for new assessment tools are based upon the need to develop additional evaluation tools that would be useful and relevant to students, teachers and the business community.

During the past several years CRESST has been developing and evaluating a series of software products to measure various types of learning. These are collectively called the Integrated Assessment System. This system has been documented in a series of reports to the Department of Defense's Computer Aided Education & Training Initiative (CAETI) (Baker & O'Neil, 1996; Herl & O'Neil, 1996; Herl et al., 1996; Klein, O'Neil, & Baker, 1996) and a recent series of conference papers presented at the 1997 annual meeting of the American Educational Research Association (Chung, O'Neil, Herl, & Dennis, 1997; Herl, O'Neil, Chung, & Dennis, 1997; Klein, O'Neil, Dennis, & Baker, 1997; O'Neil, 1997; Schacter et al., 1997). These products are model-based in the sense of reflecting the five types of learning that are specified in the CRESST model of learning (Baker, 1995; see Figure 1).

CRESST has assembled a suite of performance assessment tasks (our integrated simulation) onto which we have mapped the types of learning expected of students. The design of integrated simulation performance assessment has the following characteristics: (a) relevant, project-based scenarios that include meaningful, real-world tasks; (b) individual and team processes and products; (c) a technology base using Web-based, networked systems; and (d) model-based assessments that integrate types of cognitive learning, grade level, and content area.

---

[1] Authors are listed alphabetically.
[2] CRESST/University of California, Los Angeles
[3] CRESST/University of Southern California

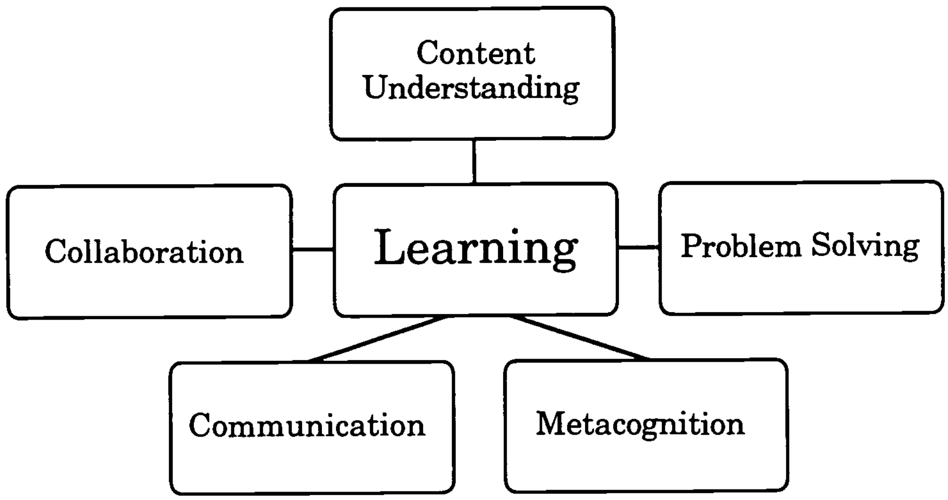

Figure 1. The CRESST model of learning.

The integrated simulation we have developed includes both collaborative and individual concept mapping tasks, a problem-solving search task, an explanation task, and a metacognition questionnaire. It is designed to be useful to students at different ages, given different levels of understanding, and with different content areas. Its administration is computer-based; the software controls and documents progress, while permitting real-time scoring and reporting. For assessing content understanding we use a concept mapping approach that is computer-based. To assess collaborative skills, we use a collaborative mapper (Chung et al., 1997), and to assess problem solving, we use a Web-based approach to support a problem-solving search task (Schacter et al., 1997). To assess metacognition, we use a questionnaire to measure both trait and state metacognition (e.g., see O'Neil & Abedi, 1996, for a description of the state version). In general, the evaluation data for our assessment tools show them to be feasible. However, there is limited reliability and validity information. Such studies are planned. Moreover, there is an issue of scaling up such an integrated simulation: What new assessment tools should be added to the integrated simulation?

New Computer-Based Assessment Tools

The proposed assessment tools will support Project 1.3 goals of the CRESST grant proposal to the Office of Educational Research and Improvement (Baker, Linn, & Herman, 1995), particularly the goals for Advanced Technology and Quality Improvement and Utility Enhancement. We suggest that basic assessment research of such tools through prototype development would be funded by Project 1.3, and product refinement and possible commercialization would be funded through other sources (e.g., NCES or commercial partners).

In general, two classes of software are needed to design and develop computer-based automated assessment tools: (a) individual assessment tool applications software, and (b) application programming interfaces (APIs). Application programming interfaces provide transparent layers for collecting, parsing, managing, analyzing, and outputting data. Application programming interfaces provide assistance for tool evaluation and performance (e.g., data logging and front-end and back-end interface configuration). End users (e.g., students and trainees) interact directly with the application software assessment tools (for example, the Multimedia Concept Mapper) but indirectly with the application programming interfaces.

Tool Design Strategy

Our strategy is to design assessment tools that (a) measure one or more components of the CRESST model of learning, (b) are stand-alone or integrate with other tools, (c) have authoring capability, (d) employ automated scoring and reporting, and (e) are Internet-deployable in both Windows and Macintosh environments.

Tool integration. Each tool will be designed as both a stand-alone instrument and part of an integrated assessment suite. To illustrate this point, the software package Microsoft Office is analogous to a tool suite or our integrated simulation, and the applications within Microsoft Office, such as Word, PowerPoint, and Excel, are analogous to individual component tools. To maximize integration, all tools will be designed around a set of application programming interfaces. These application programming interfaces are sets of code and services that are independent of any specific tool. All tools will collect and log users' performance and process data. Instead of creating (and recreating) code to perform data logging on a per-tool basis, the data logging application programming interface will provide one set of standard data logging routines accessible by all tools. Authoring, scoring, reporting, and analysis are other areas that would benefit from the development of application programming interface software.

Authoring. All tools will be authorable. That is, all tools will provide users with the capability to define content, establish scoring criteria, and customize the interface of the tool to adapt to audiences of different ages. For younger users, interfaces would be adjusted to be sparse and iconic, whereas for older, more experienced users, interfaces can be adjusted to show the more complex features of the software. An example of an adjustable interface is the Microsoft Word Tool Bar. Users can customize this tool bar such that either very few or many options appear on it. Our authoring capabilities will take a similar approach.

Automated scoring and reporting. A goal of all CRESST applications is real-time automated scoring and reporting. Automated scoring and reporting processes involve real-time data collection, management, and analysis. Therefore, underlying application programming interfaces will be written to handle these real-time processes for all tools. Although each tool will have specific reporting requirements, many of the underlying functions (e.g., calculating statistics) will be handled by an application programming interface.

Internet-deployable and platform-independent. All tools will be designed to be deployable across the Internet for both the Windows and Macintosh environments. Thus, tools will be developed in an architecture-neutral language such as Java™ or in applications that create downloadable plug-ins that run on both platforms. In addition, all tools will be compatible with Netscape Navigator™ and Microsoft Explorer™ Internet browsers.

Spiral model of software development. Our assumption is that we will be developing software. Given this—and because such software will likely have evolving requirements—we believe a spiral model is the appropriate development model. The spiral model minimizes risks through incremental development of software. In concrete terms, periodic deliveries of prototypes are planned. The goal of each prototype cycle is to design and implement only the highest priority features. Each successive prototype cycle incorporates more features. The benefit of the spiral model is that technical and functional problems are identified early in the development cycle. This early risk identification means that we can make informed decisions about whether to cancel or continue a project. An added benefit of this development model is that there is high product visibility: There is always a working prototype that can be demonstrated.

Proposal assessment tools. Table 1 lists each proposal assessment tool and the component of the CRESST model that serves as the measurement target. Table 2 presents a list of assessment tools and their respective application programming interfaces. Detailed descriptions of each tool are found in the Tool Specifications section.

Tool Specifications

In this section, specifications for a series of tools are proposed. Table 3 provides a first cut of the specifications for each proposed tool, and Table 4 provides a first cut of specifications for each proposed application programming interface.

Next Steps

We need to prioritize the development activities.

Table 1

Tools Proposed and Cognitive Families of Learning

Tools	Content under-standing	Problem solving	Communi-cation	Collabor-ation	Metacog-nition
Collaborative Concept Mapper	✗	✗	✗	✗	✗
The Flowcharter	✗	✗			
Idea Generator and Evaluator		✗	✗	✗	
Model Simulator		✗			✗
Networked Team Simulator		✗	✗	✗	✗
Multimedia Concept Mapper	✗	✗			
The Outliner	✗		✗		
Problem Solver	✗	✗	✗	✗	✗

Table 2

Application Programming Interface (API) Support for Assessment Tools

	Authoring API	Data logging API	Reporting API	Collabor-ative API	Searching API	Simu-lation API	Text classifier API	Scoring API	Process analyzer API
Collaborative Concept Mapper	✗	✗	✗	✗	✗		✗	✗	✗
The Flowcharter	✗	✗	✗				✗	✗	✗
Idea Generator & Evaluator	✗	✗	✗	✗	✗			✗	✗
Model Simulator	✗	✗	✗	✗		✗	✗	✗	✗
Networked Team Simulator	✗	✗	✗			✗		✗	✗
Multimedia Concept Mapper	✗	✗	✗				✗	✗	✗
The Outliner	✗	✗	✗				✗	✗	✗
Problem Solver	✗	✗	✗	✗	✗			✗	✗

Table 3

Specifications for Assessment Tools

Assessment Tool: Collaborative Concept Mapper

Brief Product Statement: The Java-based Collaborative Concept Mapper will use the Java-based Individual Concept Mapper and build into it both synchronous and asynchronous collaborative support capability. A group of students, business people, parents, and so on, could use this tool collaboratively to develop conceptual understanding of an area. Provisions will be made to negotiate who leads the collaboration and when. A front-end user interface will be developed for entering the concept terms and links of choice. This front-end interface will have synchronous and asynchronous collaborative capabilities such that users can agree on the terms and links to use.

CRESST learning families	Communication, Collaboration, Content Understanding, Problem Solving
Project 1.3 area	Improvement and Utility Enhancement
Rationale	This is a multi-user assessment and learning tool that is domain- and audience-independent. It is adaptable to any subject or content area and can be scored in real time using automated techniques.
Domain	Domain- and audience-independent
Task	Construct a concept map collaboratively
Performance measures	Concept mapping; collaborative and communicative interactions
Process measures	State map construction over time; use of collaborative interactions
Configurability	Adaptable front-end user interface and expert scoring template
Scoring	Automated; expert performance (Herl metric: Herl, Baker, & Niemi, 1996)
Reporting	Automated; final performance, process performance (i.e., use of interaction, map construction over time, productive collaborative behaviors)
Software requirements	Netscape 3.0, Mac OS 7.5.3, Windows 95 or NT
Hardware requirements	Macintosh Power PC or Pentium PC, 16 MB of RAM, 20 MB of disk space
Development cycle	5-10 months production work and testing
Large-scale assessment outlook (scalability)	Completely scalable

Table 3 (continued)

Assessment Tool: Flowcharter

Brief Product Statement: Flowcharting tool that allows students to sequence events. Usable in a variety of content domains and across grade levels (e.g., historical flow, programming, mathematical algorithms, experimental methodology). Student flowcharts can be scored automatically online.

CRESST learning families	Content Understanding, Problem Solving
Project 1.3 area	Advanced Technology
Rationale	Without the complications of natural language scoring, flowcharting allows for inspection of student understanding of basic procedures, principles, sequential ordering, and problem solving.
Domain	Domain- and audience-independent
Task	Students create flowcharts to show the progression of events for a given problem statement.
Performance measures	Student flowchart
Process measures	None
Configurability	Completely adaptable to content and audience needs; adaptable front-end user interface as well as back-end scoring and reporting interface
Scoring	Automated; based on teacher input
Reporting	Automated on-line and off-line reporting available; in addition, student responses may be printed for teacher inspection
Software requirements	Netscape 3.0, Mac OS 7.5.3, Windows 95 or NT
Hardware requirements	Macintosh Power PC or Pentium PC, 16 MB of RAM, 20 MB of disk space
Development cycle	6-10 months; requires working version of essay-scoring API for best results
Large-scale assessment outlook (scalability)	Completely scalable

Table 3 (continued)

Assessment Tool: Idea Generator and Evaluator

Brief Product Statement: The shell for this product is a multi-user Java-based bulletin board. A group problem is posted. Group members are asked to come up with potential solutions or ideas to solve the problem. All ideas are posted to the bulletin board. One to two ideas are then randomly assigned and mailed to an individual group member for evaluation and review. Ideas are evaluated using the CRESST problem-solving rubric. Quality of ideas and quality of evaluations of ideas are assessed.

CRESST learning families	Communication, Collaboration, Problem Solving
Project 1.3 area	Improvement and Utility Enhancement
Rationale	This is a multi-user assessment and learning tool that is domain- and audience-independent. It is adaptable to any subject or content area and can be scored using automated techniques combined with a history database of problems proposed and exemplary solutions and evaluations.
Domain	Domain- and audience-independent
Task	Users propose, review, and evaluate ideas for solving a complex problem.
Performance measures	Written ideas, evaluation of others' ideas
Process measures	Clear communication of ideas; analysis of ideas; development of ideas after analyses
Configurability	Completely adaptable to content and audience needs
Scoring	Automated; expert-based evaluations; compiled history of problem solutions and evaluations library
Reporting	Automated; quality of ideas and quality of evaluations are reported
Software requirements	Netscape 3.0, Mac OS 7.5.3, Windows 95 or NT
Hardware requirements	Macintosh Power PC or Pentium PC, 16 MB of RAM, 20 MB of disk space
Development cycle	4-10 months
Large-scale assessment outlook (scalability)	Completely scalable

Table 3 (continued)

Assessment Tool: Model Simulator

Brief Product Statement: This tool will let students build a conceptual model and include a simulation option. Students will build some system (a system is a set of components that relate to each other causally). With the simulation option they will be able to run the system and get feedback on their model. The simulation (or "run") option can also be used to score the model. The students will build their model visually using a toolkit of widgets and objects.

CRESST learning families	Problem Solving, Metacognition
PROJECT 1.3 area	Advanced Technology
Rationale	Most assessment measures do not require students to construct something that they have to get working. Usually, the assessment measure requires them to represent their understanding in static terms as a set of interrelated concepts. Rarely are students required to *apply* their understanding.
Domain	Any area that can be represented as a cause-effect system of interacting components
Task	Students are given widgets representing major components of system. They are required to demonstrate their understanding of the system by building a working model.
Performance measures	(1) Runnability of system; (2) number of components used; (3) quality and significance of relationships between components; (4) parsimony of model; (5) company to expand model
Process measures	(1) Trials to success; (2) path to solution (trial-error, systematic); (3) use of feedback; (4) problem approach (bottom-up, top-down)
Configurability	Able to set parameters of simulation model; able to configure components of model; able to assign icons to model
Scoring	(1) Runnability of model; (2) comparison with expert
Reporting	Automated; quality of model
Software requirements	Netscape 3.0, Mac OS 7.5.3, Windows 95 or NT
Hardware requirements	Macintosh Power PC or Pentium PC, 16 MB of RAM, 20 MB of disk space
Development cycle	3 months for build-only; 12 months for simulation component
Large-scale assessment outlook (scalability)	Scalable but use is limited by simulation content

Table 3 (continued)

Assessment Tool: Networked Team Simulator

Brief Product Statement: This tool will be a networked simulation within a teamwork framework that will require group members to build a runnable model of a system. The target system is comprised of a set of interrelated "black boxes." Each member gets a black box to build and has to integrate with other team members. The system works only if each black box works.

CRESST learning families	Collaboration, Problem Solving, Metacognition
Project 1.3 area	Advanced Technology
Rationale	The current teamwork measures of processes and skills are based on tasks that weakly couple group members—the success of the team is not dependent on all members. A black box/system simulation means that the success of the team depends on the success of each member (high face validity for measuring workforce readiness).
Domain	Anything that can be represented as a network of causally related components. Probably science/engineering-type problems. May also be applicable to classic survival problems.
Task	Group must build a working system. Each member is assigned one component of the system. Members work together to integrate components into system. System runs when all components are working.
Performance measures	(1) Whether system works; (2) quality of solution (e.g., tolerance of system—does system run under all conditions or does system work only under certain conditions); (3) if system doesn't work, score by quality of components.
Process measures	Teamwork processes; trials to solution; type of hypothesis/testing strategy
Configurability	Able to set parameters of simulation model; able to configure components of model; able to assign icons to model
Scoring	Automated; high- versus low-comparison
Reporting	Teamwork processes, simulation results
Software requirements	Netscape 3.0, Mac OS 7.5.3, Windows 95 or NT
Hardware requirements	Macintosh Power PC or Pentium PC, 16 MB of RAM, 20 MB of disk space
Development cycle	9-12 months
Large-scale assessment outlook (scalability)	Scalable but use is limited by simulation content

Table 3 (continued)

Brief Product Statement: Multimedia Concept Mapper

Brief Product Statement: This concept mapping tool is a variation of the Individual Concept Mapper. Students construct concept maps using the individual concept mapper's construction system. Multimedia images (graphics, sound, and video) can be imported by students to serve as instances of concepts and links. Multimedia images (i.e., physical space representations, geographical maps, and timelines) can also be used to add dimensionality to the mapping area.

CRESST learning families	Content Understanding
Project 1.3 area	Advanced Technology
Rationale	Concept maps' construction systems are currently text-based. This tool will allow students to construct and enhance their concept maps through the use of graphics and sound. Students possessing more limited vocabularies may benefit from visual and/or aural stimuli.
Domain	Domain-independent
Task	Students use the individual concept mapping construct system. They can access and import images using the multimedia image database builder to elaborate their concept maps. Students constructing history concept maps may be prompted to position concepts within timeline and geographical dimensions. Students constructing science maps may be prompted to map concepts onto physical space dimensions.
Performance measures	Semantic content and organizational structure scores can be obtained using expert-based scoring criteria.
Process measures	None
Configurability	Adaptable to content domain
Scoring	Expert-constructed concept maps can be used to provide automated scoring of students' maps.
Reporting	Automated reporting of individual map scores
Software requirements	Netscape 3.0, Mac OS 7.5.3, Windows 95 or NT
Hardware requirements	Macintosh Power PC or Pentium PC, 16 MB of RAM, 100 MB of disk space
Development cycle	6-9 months to develop software
Large-scale assessment outlook (scalability)	Dependent upon user hardware

Table 3 (continued)

Assessment Tool: Outliner

Brief Product Statement: Outlining tool to allow students to organize important information within the framework of a prewriting activity. Usable in a variety of content domains and across grade levels. Student outlines can be scored automatically online.

CRESST learning families	Content Understanding, Communication
Project 1.3 area	Advanced Technology
Rationale	Similar to essay writing tasks, but scoring issues are simplified by the outline structure. Key terms can be highlighted by instructor, as can relevant ordering, links, and so on.
Domain	Domain- and audience-independent
Task	Given a topic (teacher-inputted), students create outlines (which could subsequently be used to write essays, lab reports, problem solutions, etc.).
Performance measures	Outcome outline
Process measures	Possibly steps towards creation of outline
Configurability	Completely adaptable to content and audience needs; outline topic, amount of information available to the student, student level of expertise, and range of appropriate responses all adaptable; adaptable front-end user interface as well as back-end scoring and reporting interface
Scoring	Automated; based on teacher input (see automated essay scoring)
Reporting	Automated on-line and off-line reporting available; in addition, student responses may be printed for teacher inspection
Software requirements	Netscape 3.0, Mac OS 7.5.3, Windows 95 or NT
Hardware requirements	Macintosh Power PC or Pentium PC, 16 MB of RAM, 20 MB of disk space
Development cycle	6-8 months; needs simple working version of essay-scoring API for outline scoring
Large-scale assessment outlook (scalability)	Completely scalable

Table 3 (continued)

Assessment Tool: Problem Solver

Brief Product Statement: Problem solving involves extracting relevant information, simplifying and organizing it, and being able to then present and explain the solution clearly to others. Measurement of these processes independent of the content or domain is the goal of this highly adaptable tool. Models and information can be placed into this tool (e.g., diagrams, text, images, video), or individuals can use this tool to access existing informational resources such as the Internet. This tool will measure how well individuals extract relevant information, simplify and organize it, and then present it to others. Determinants for each of these categories are based either on expert performance or on automated scoring techniques. These processes are joined with performance in the form of an individual concept map, collaborative concept map, essay task, presentation, short-answer task, multiple-choice task, diagram construction, or oral presentation. Integration of any of these performance outcome measures will fit within this tool.

CRESST learning families	Problem Solving
Project 1.3 area	Advanced Technology
Rationale	This generic problem-solving tool measures the processes of information seeking (e.g., determining relevant from spurious information; organization and simplification of information). It can fit with other assessment tools to measure these processes against outcome performance.
Domain	Domain-independent
Task	Users search, analyze, extract, simplify, and reorganize information.
Performance measures	Information seeking; relevance determinants; information simplification and organization
Process measures	Information seeking; information simplification and organization; attention
Configurability	Adaptable to content and audience needs
Scoring	Automated; expert performance; lexical libraries integrated with latent semantic analyses
Reporting	Automated; problem-solving processes reported along with outcome measure performance
Software requirements	Netscape 3.0, Mac OS 7.5.3, Windows 95 or NT
Hardware requirements	Macintosh Power PC or Pentium PC, 16 MB of RAM, 20 MB of disk space
Development cycle	9-12 months; this tool needs to be developed concurrently with other tools to ensure that other performance tools work within it. 14-24 months before this tool is fully operable with other tools across assessment suite
Large-scale assessment outlook (scalability)	Completely scalable

Table 4

Specifications for Assessment Application Programming Interfaces (APIs)

API	Description
Authoring API	**Purpose**: This API will provide authoring services to tools. Authoring services include create/edit/delete capability for specifying assessment criteria, content, and tool parameters. **Development Cycle:** 6-12 months
Data logging API	**Purpose**: This API will provide data logging services for tools. The data will be logged to a database in a common log format. Services will be provided to log, extract, parse, sort, and filter data. **Development Cycle:** 2-6 months
Reporting API	**Purpose**: This API will provide reporting services to tools. Reporting services include retrieving performance data and process data, reporting statistics on the retrieved data, and reporting scoring results. **Development Cycle:** 6-12 months
Collaborative networked API	**Purpose**: This API will provide collaborative services to tools. Network services are required for tools that require online collaboration, teamwork, or other network-based functions. **Development Cycle:** 6-8 months
Search API	**Purpose**: This API will provide search services to tools. Search services include a search engine with keyword searching capability, Boolean operators, and results returned as keyword hits or similarity rankings. **Development Cycle:** 4-8 months
Simulation API	**Purpose**: This API will provide simulation services to tools. Simulation services include setting and retrieving model parameters, running the model, querying components for state information, and providing access to widget objects and libraries. **Development Cycle:** 8-18 months

28

Table 4 (continued)

Text classifier API	**Purpose:** This API will provide text classification services to tools. Given a set of text, this API will provide an estimate of similarity between the given text and other text in the database. This API will most likely be used for matching student written texts to prescored texts and categorizing short answers or user-typed messages. **Development Cycle:** 18-24 months. This API will need a program of research. **Comments**: Any automated, online system must have text processing capability. Otherwise, the system will lack the capability to handle a very common and important form of assessment. May not be scalable if the number of requested comparisons is large, or if the textbase used for comparison is large.
Scoring API	**Purpose**: This API will provide scoring services to tools. Scoring services include computing performance data and process statistics, carrying out scoring functions (e.g., comparing student and expert concept maps). **Development Cycle:** 6-12 months
Process analyzer API	**Purpose**: This API will provide the capability to characterize the kinds of cognitive processing students are using as they interact with computer-based tools. This API will provide estimates of students' processing. If even moderately successful, this effort will provide substantial utility for real-time feedback, diagnosis, and prescription. Will advance state of the art in computer-based cognitive process research. **Development Cycle:** 18-36 months. Need to gather think-aloud protocol data and link it to computer-based data. Need to develop model of processes first at the tactical level and then at the strategic level. This API will need a program of research. **Comments**: We know very little about what students actually do while they take tests in computerized environments. The inferences we draw about cognitive processing are rarely based on strong process data. Yet conclusions about cognitive processing and its relationship to performance are typically based on self-reported data or data aggregated over the task. Estimating learners' processes will be tractable on tasks that require complex reasoning, decision making, evaluation, and self-regulation. The task must provide opportunities for these kinds of processes to occur. A strong coupling is required between overt action and these processes.

References

Baker, E. L. (1995, September). *Finding our way*. Presentation at the annual conference of the Center for Research on Evaluation, Standards, and Student Testing, University of California, Los Angeles.

Baker, E. L., Linn, R. L., & Herman, J. L. (1995). *Institutional grant proposal for OERI Center on Improving Student Assessment and Educational Accountability: Integrated assessment systems for policy and practice: Validity, fairness, credibility, and utility*. Los Angeles: University of California, Center for Research on Evaluation, Standards, and Student Testing.

Baker, E. L., & O'Neil, H. F., Jr. (1996). *Design report for CAETI assessment plans* (Year 1) (Report to ISX/DODEA). Los Angeles: University of California, Center for Research on Evaluation, Standards, and Student Testing.

Chung, G. K. W. K., O'Neil, H. F., Jr., Herl, H. E., & Dennis, R. A. (1997, March). Use of networked collaborative concept mapping to measure team processes and team outcomes. In H. F. O'Neil, Jr. (Chair), *An integrated simulation approach to assessment*. Symposium presented at the annual meeting of the American Educational Research Association, Chicago.

Herl, H. E., Baker, E. L., & Niemi, D. (1996). Construct validation of an approach to modeling cognitive structure of U.S. history knowledge. *Journal of Educational Research, 89*, 206-219.

Herl, H. E., & O'Neil, H. F., Jr. (1996). *Final report. CAETI STS1 technologies baseline data collection* (Year 1) (Report to ISX/DODEA). Los Angeles: University of California, Center for Research on Evaluation, Standards, and Student Testing.

Herl, H. E., O'Neil, H. F., Jr., Chung, G. K. W. K., & Dennis, R. A. (1997, March). Feasibility of an on-line concept mapping construction and scoring system. In H. F. O'Neil, Jr. (Chair), *An integrated simulation approach to assessment*. Symposium presented at the annual meeting of the American Educational Research Association, Chicago.

Herl, H. E., O'Neil, H. F., Jr., Chung, G. K. W. K., Dennis, R. A., Klein, D. C. D., Schacter, J., & Baker, E. L. (1996). *Measurement of learning across five areas of cognitive competency: Design of an integrated simulation approach to measurement. Year 1 report* (Report to ISX/DODEA). Los Angeles: University of California, Center for Research on Evaluation, Standards, and Student Testing.

Klein, D. C. D., O'Neil, H. F., Jr., & Baker, E. L. (1996). *A cognitive demands analysis of innovative technologies* (Report to ISX/DODEA). Los Angeles: University of California, Center for Research on Evaluation, Standards, and Student Testing.

Klein, D. C. D., O'Neil, H. F., Jr., Dennis, R. A., & Baker, E. L. (1997, March). The five families of cognitive learning; A context in which to conduct cognitive demands analyses of innovative technologies. In H. F. O'Neil, Jr. (Chair), *An integrated simulation approach to assessment*. Symposium presented at the annual meeting of the American Educational Research Association, Chicago.

O'Neil, H. F., Jr. (Chair). (1997, March). *An integrated simulation approach to assessment*. Symposium presented at the annual meeting of the American Educational Research Association, Chicago.

O'Neil, H. F., Jr., & Abedi, J. (1996). Reliability and validity of a state metacognitive inventory: Potential for alternative assessment. *Journal of Educational Research, 89*, 234-245.

Schacter, J., Herl, H. E., Chung, G. K. W. K., O'Neil, H. F., Jr., Dennis, R. A., & Lee, J. J. (1997, March). Feasibility of a Web-based assessment of problem solving. In H. F. O'Neil, Jr. (Chair), *An integrated simulation approach to assessment*. Symposium presented at the annual meeting of the American Educational Research Association, Chicago.

U.S. Department of Education
Office of Educational Research and Improvement (OERI)
Educational Resources Information Center (ERIC)

ERIC®

REPRODUCTION RELEASE

(Specific Document)

I. DOCUMENT IDENTIFICATION:

Title: Estimate of the Potential Costs and Effectiveness of Scaling Up CRESST Assessment Software

Author(s): Gregory K.W.K. Chung, Howard E. Herl, Davina C.D. Klein, Harold F. O'Neil, Jr., John Schacter

Corporate Source: UCLA Center for the Study of Evaluation

Publication Date: December 1997

II. REPRODUCTION RELEASE:

In order to disseminate as widely as possible timely and significant materials of interest to the educational community, documents announced in the monthly abstract journal of the ERIC system, *Resources in Education* (RIE), are usually made available to users in microfiche, reproduced paper copy, and electronic/optical media, and sold through the ERIC Document Reproduction Service (EDRS) or other ERIC vendors. Credit is given to the source of each document, and, if reproduction release is granted, one of the following notices is affixed to the document.

If permission is granted to reproduce and disseminate the identified document, please CHECK ONE of the following two options and sign at the bottom of the page.

☑ Check here **For Level 1 Release:** Permitting reproduction in microfiche (4" x 6" film) or other ERIC archival media (e.g., electronic or optical) *and* paper copy.

The sample sticker shown below will be affixed to all Level 1 documents

PERMISSION TO REPRODUCE AND DISSEMINATE THIS MATERIAL HAS BEEN GRANTED BY

Sample

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

Level 1

The sample sticker shown below will be affixed to all Level 2 documents

PERMISSION TO REPRODUCE AND DISSEMINATE THIS MATERIAL IN OTHER THAN PAPER COPY HAS BEEN GRANTED BY

Sample

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

Level 2

☐ Check here **For Level 2 Release:** Permitting reproduction in microfiche (4" x 6" film) or other ERIC archival media (e.g., electronic or optical), but *not* in paper copy.

Documents will be processed as indicated provided reproduction quality permits. If permission to reproduce is granted, but neither box is checked, documents will be processed at Level 1.

"I hereby grant to the Educational Resources Information Center (ERIC) nonexclusive permission to reproduce and disseminate this document as indicated above. Reproduction from the ERIC microfiche or electronic/optical media by persons other than ERIC employees and its system contractors requires permission from the copyright holder. Exception is made for non-profit reproduction by libraries and other service agencies to satisfy information needs of educators in response to discrete inquiries."

Sign here→ please

Signature: Kim Hurst	Printed Name/Position/Title: Kim Hurst/Admin. Assistant	
Organization/Address:	Telephone: 310-206-1532	FAX: 310-825-3883
	E-Mail Address: Kim@cse.ucla.edu	Date: 2/25/98