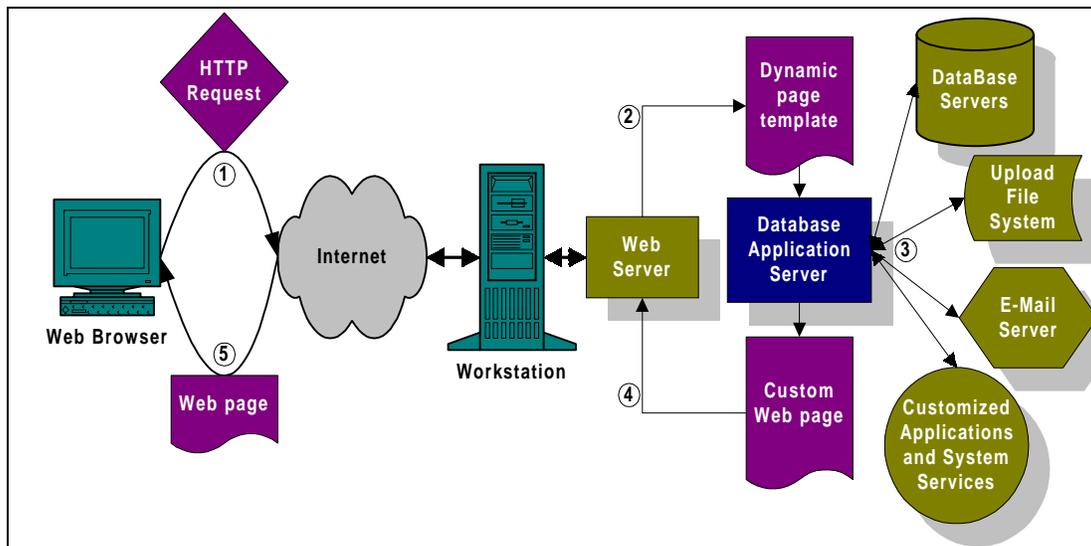




Web-Enabled Design Review and Lessons Learned

by E. William East



The Department of Defense invests significant capital in building new facilities—approximately \$11.2 billion in Fiscal Year 1996. Conventional facility delivery processes and practices that were once satisfactory are increasingly expensive, labor intensive, and not fully automated or integrated. State-of-the-art automation technology is the best hope to keep pace with requirements to reduce design and construction errors, reduce resource requirements, and optimize mission performance. This report discusses one aspect of facility delivery—the design review process.

The Design Review and Checking System (DrChecks) is the next step in the evolution of previous USACERL-developed products to support the technical, design, and Biddability,

Constructibility, Operability (BCO) review process.

This document describes (1) the requirements and constraints considered in this research, (2) requirements for an integrated system of lessons learned, (3) minimum requirements to install and test the distribution version of DrChecks, and (4) the steps required to implement a distributed lessons learned system.

This work was accomplished in accordance with MIL-STD-498. To test the prototype system requires an Intel Pentium processor and a 2 GB hard drive compatible with HTML 2.0. Client systems must be linked to the Internet using TCP/IP protocols with a minimum connection speed of 9600 bps.

Foreword

This study was conducted for Headquarters, U.S. Army Corps of Engineers (HQUSACE) under Project 4A162784AT41, "Military Facilities Engineering Technology"; Work Unit AP7, "Design Reviewer's Support Environment." The technical monitors were Justin Taylor, CEMP-ES, and Stan Green, CEMP-CE.

The work was performed by the Engineering Processes Division (PL-E) of the Planning and Management Laboratory (PL), U.S. Army Construction Engineering Research Laboratories (USACERL). The USACERL principal investigator was E. William East. The USACERL technical editor was Linda L. Wheatley, Technical Information Team. Appreciation is also expressed to the members of the Steering Committee who actively contributed to this project. In alphabetical order, these members are: Drew Anderson, Omaha District; Johnny Baggette, South Atlantic Division; George Brule, Savannah District; Margie Crumley, Omaha District; John Hart, Ohio River Division; Terry Houghton, HQ, Corps; Blaine Kemsley, Albuquerque District; Pat Linsey, Omaha District; Leo Phillips, Mobile District; Joseph Russell, Portland District; Norman Sams, Alaska District; Donald E. Slater, USACE, Transatlantic Programs Center; Stephen Stoner, Sacramento District; Justin Taylor, HQUSACE.

Dr. Michael P. Case is Chief, CECER-PL-E, and L. Michael Golish is Operations Chief, CECER-PL. COL James A. Walter is Commander of USACERL, and Dr. Michael J. O'Connor is Director.

Contents

SF 298	Error! Bookmark not defined.
Foreword	2
List of Figures and Tables	6
1 Introduction	9
Background.....	9
Objectives	10
Approach.....	11
Scope.....	11
Mode of Technology Transfer	11
2 System Overview	13
3 Lessons Learned Perspectives	14
Definition of Lessons Learned	14
Drafting Potential Lessons Learned.....	16
4 Required System Capabilities	17
Creation of Design Review Projects	17
Creation of Design Review Phases	19
Preparation of Design Review Comments	21
Evaluation of Design Review Comments	23
Project Manager Certification	25
Back-Check Review	25
Search for Relevant Past Comments	26
Reference Source Search.....	26
Lessons Learned	28
User Registration	30
CADD Integration.....	31
Electronic Document Transfer	31
Cost-Benefit Study Information	31

5	Internet Technology Platform Selection	33
	Hypertext Markup Language	34
	CGI Scripts and Java	34
	Telephony.....	35
	Internet Relay Chat (IRC)	35
	Virtual Reality Modeling Language	36
	Client-Server Operation	37
	Benefits of an Internet-Based Solution	38
6	Constraints to System Design.....	40
	Support of All Computer Platforms	40
	Cost of Complete System	40
	Cost of System Maintenance	40
	Use of Commercial Software	40
	User Training	41
	User Access to the World Wide Web	41
	System Access Security	41
	Access to Sensitive Project Data	41
	Web-Browser Software	42
	Use of Advanced Design Tools	42
	Integration of DrChecks with ARMS	42
7	Computer Hardware and Software Minimum Requirements	43
	Hardware Requirements	43
	Software Requirements	43
	Communications Requirements.....	43
8	Software Development Methods	45
	Object-Oriented Modeling	45
	Standards for Software Products	45
	Reusable Software Products	45
	ODBC Compliance.....	46
	Security Mechanisms.....	46
	Computer Hardware Resource Utilization	47
	Identification of Project Stakeholders.....	47
	Access to Project, Review, Comment, and Evaluation Data.....	47
	Access to User Registration Information	48
9	Design Review and Checking System (DrChecks)	49
	User Registration	50
	On-Line References.....	51

Manage Projects and Reviews	55
Add Review Comments.....	56
Evaluate Review Comments	58
Evaluate Lessons Learned	58
10 CADD-Integrated Design Review	62
11 The Virtual Design Review	66
Goals for Improving the Design Review Process.....	66
The Virtual Design Review.....	68
Expected Benefits.....	71
12 Conclusion.....	76
References.....	78
Distribution	

List of Figures and Tables

Figures

1	Project based learning	15
2	Corporate learning cycle	15
3	VRML building model	36
4	Client server operations	37
5	DrChecks homepage	49
6	User registration page.....	50
7	On-line reference page.....	52
8	Search CERS lessons learned.....	53
9	Example CERS search results.....	54
10	Knowledge-base search results	55
11	Manage projects.....	56
12	Selecting a review to check.....	57
13	Adding a review comment	58
14	Evaluating a review comment.....	59
15	Adding a proposed lesson.....	60
16	Reviewing lessons learned.....	61
17	Server, project, and review identification.....	63
18	Select CADD object	63
19	CADD-based design review comment	64
20	Confirmation of comment transmission.....	64
21	VRML system model.....	70

Tables

1	Project based learning	15
2	Corporate learning cycle	15
3	VRML building model	36
4	Client server operations	37
5	DrChecks homepage	49

6	User registration page.....	50
7	On-line reference page.....	52
8	Search CERS lessons learned.....	53
9	Example CERS search results.....	54
10	Knowledge-base search results.....	55
11	Manage projects.....	56
12	Selecting a review to check.....	57
13	Adding a review comment.....	58
14	Evaluating a review comment.....	59
15	Adding a proposed lesson.....	60
16	Reviewing lessons learned.....	61
17	Server, project, and review identification.....	63
18	Select CADD object.....	63
19	CADD-based design review comment.....	64
20	Confirmation of comment transmission.....	64
21	VRML system model.....	70

1 Introduction

Background

Army facilities enable 2.5 million active military and civilian personnel to perform their jobs, and the Army and Department of Defense (DOD) continue to invest significant capital in building new facilities—approximately \$11.2 billion in Fiscal Year (FY) 1996. These facilities must support mission performance in a productive and healthy working environment and in an economical and efficient manner. Conventional facility delivery processes and practices that were once satisfactory are increasingly expensive, slow, labor intensive, fragmented and not fully automated or integrated. State-of-the-art automation technology is the best hope to keep pace with requirements to reduce design and construction errors, reduce resource requirements, and optimize mission performance. This report discusses one aspect of facility delivery—the design review process.

A *design review process* is used by the Architect/Engineer/Contractor (A/E/C) community to help ensure the quality of designs. The process is one of iterative redesign. Throughout the building design process, two groups of people work on the building design. The *design team* is responsible for the synthesis of design, from developing functional specifications for the building through interactions with the customers, to finalizing the set of design documents for the construction stage. At certain points in the design process, the design team submits the existing design documents to a *review team*, whose task is to periodically check the documents for inconsistencies, errors, and other suboptimal aspects of the design. Each member of the review team represents a different concern. For example, a building design review team may contain an architect, electrical engineer, structural engineer, mechanical engineer, the builder, the occupant, etc. The design documents pass from member to member of the review team, each critiquing the design according to his/her area of expertise. The critiques are accumulated as a set of textual *review comments* and are returned to the design team along with the design documents. Reviews are usually conducted at least twice during the design process.

The reviewer receives a set of possibly partially-completed drawings and specifications for a building design. The reviewer then produces a set of review comments based on the reviewer's personal experience, the recorded experiences of others, and reference materials. These materials are primarily paper-based, although recent systems like the Reviewer's Assistant (RA) (East et al. 1995) have provided computer support for the storage and retrieval of review comments.

Unfortunately, the design review process is far from perfect. As it is an inherently resource-intensive and time-intensive process, a building design may be reviewed only a few times during the design process, which results in more errors slipping through to the construction phase of the building's life cycle. The design review process is often expensive because of the need for face-to-face meetings between the designers, reviewers, and customers to articulate requirements, problems, and solutions.

Objective

This project was conducted to assist in the understanding of requirements for future design and BCO* review tools. The work builds upon knowledge gained by the U.S. Army Construction Engineering Research Laboratory (USACERL) during the development of the Automated Review Management System (ARMS) and the RA program.

The objective of this research was to develop an integrated support environment for design and BCO reviewers. Specifically, this research was to (1) determine the requirements and constraints of the research, (2) identify requirements for an integrated system of lessons learned, (3) determine minimum requirements to install and test the distribution version of the Design Review and Checking System (DrChecks), and (4) determine the steps required to implement a distributed lessons learned system.

* BCO = Biddability, Constructibility, and Operability.

Approach

The approach used in this research and development effort was the spiral design model (Boehm 1988). There are four steps to each of the cycles in this iterative design approach. The first step of the cycle is to determine the objectives a system is to meet. In addition, alternatives and constraints to system development are considered. In the next step, system developers evaluate the objectives in light of the alternatives and constraints while attempting to maximize the system benefits. The developers provide the results of their work to the system users for evaluation. During the third stage of the cycle, detailed design requirements are developed. The plan to implement these design requirements is developed and carried out in the fourth step of each iteration of the spiral model.

The Design Review Tools Steering Committee—a group of users from across the Army Corps of Engineers—and project technical monitors have provided feedback during the course of this project. System documents detailing requirements were prepared in accordance with the U.S. Military Standard for Software Development and Documentation, MIL-STD-498, and the subsidiary Data Item Definition DI-IPSC-81433, Software Requirements Specification.

Scope

This research considered the review of plans and specifications during the design and BCO reviews conducted for traditional facilities built under the Corps Military Construction Program. The work is also applicable to a wide range of government and private construction programs.

Mode of Technology Transfer

The results of this report apply to any organization, public or private, that produces design drawings. Army Corps of Engineer offices that have fully implemented ARMS may utilize the lessons learned work described in this report.

This work will be transferred through two mechanisms:

1. The complete design review package, including DrChecks and the integrated lessons learned package, is being evaluated for commercial distribution through a Cooperative Research and Development Agreement.

2. An early demonstration version of DrChecks and the lessons learned package may be tested directly via the World Wide Web (WWW) at <http://www.cecer.army.mil/pl/ra/committee>.

2 System Overview

DrChecks extends the expertise developed by USACERL in developing systems to support the design review process and those conducting design reviews. DrChecks takes advantage of the emerging technology of the WWW to create a collaborative environment for the identification and resolution of potential deficiencies in construction plans and specifications. On-line reference materials are also available for the reviewer. Users of the DrChecks system include: private A/E firms, members of local construction offices, project client and project occupant representatives, and design management offices.

Several key elements of DrChecks distinguish the system from previously developed design review tools. These items are described below:

World Wide Access. All members of the design and construction team will conduct design reviews and follow-up evaluations using the WWW. Aside from standard Internet connections and browser software, no special software will be required for this application.

On-Line References. Design reviewers will have access across the Internet to a variety of references, including constructibility issues captured in the Construction Evaluation Reporting System (CERS), Corps guide specifications, and standard design details.

Integrated Lessons Learned. Rather than use separate systems for design review and lessons learned, DrChecks contains an integrated design review lessons learned checking system. Lessons learned are specifically discussed in Chapter 3 of this report.

The design review steering committee's web site (<http://www.cecer.army.mil/pl/ra/committee>) contains a variety of information and links related to DrChecks.

3 Lessons Learned Perspectives

Definition of Lessons Learned

Many review comments occur with such frequency or are of such significant impact that they should be individually documented. These types of comments are typically referred to as “lessons learned.” In considering the development of a design review and lessons learned system, researchers began by exploring what constitutes “lessons learned.” A combination of the two following definitions provides the basic definition of lessons learned for our demonstration system.

A lesson learned is knowledge or understanding gained by experience. The experience may be positive, as in a successful test or mission, or negative, as in a mishap or failure. Successes are also considered sources of lessons learned. A lesson must be significant in that it has a real or assumed impact on operations; valid in that it is factually and technically correct; and applicable in that it identifies a specific design, process, or decision that reduces or eliminates the potential for failures and mishaps, or reinforces a positive result. (NASA 1997)

...good work practice or innovative approach that is captured and shared to promote application. It may also be an adverse work practice or experience that is captured and shared to avoid recurrence. (DOE 1997)

Based on these definitions, this program attempts to demonstrate the capture of successes and failures of experienced design and construction personnel. The items captured must have a real impact on operations, be factually or technically correct, have application to a specific process or component, and have limited management implication. Once captured items become lessons learned, they must be shared with personnel at the time when the lessons can be applied at the least cost (typically during design) to improve success of each new project.

In the traditional project process shown in Figure 1, individual users analyze the results of their project performance. Feedback for the lessons learned is often an informal word-of-mouth or paper process. Turning project-based learning into corporate learning is the basis of the current demonstration system.

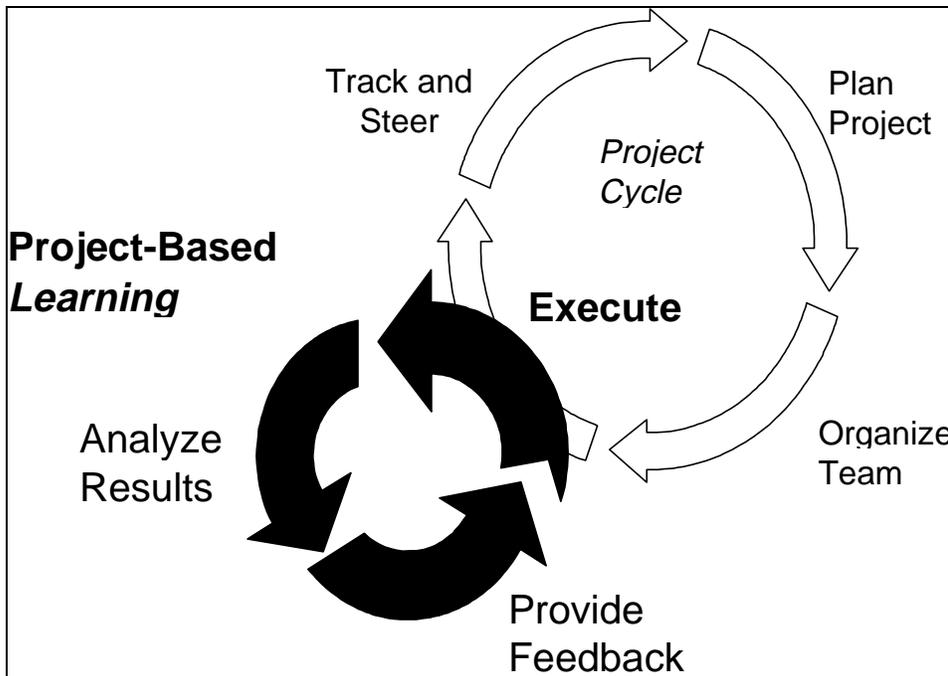


Figure 1. Project based learning.

In the corporate learning process, project-based learning is extracted from the personnel directly involved in the project cycle. The lessons learned by individuals are then shared throughout the organization. Figure 2 illustrates the corporate learning process.

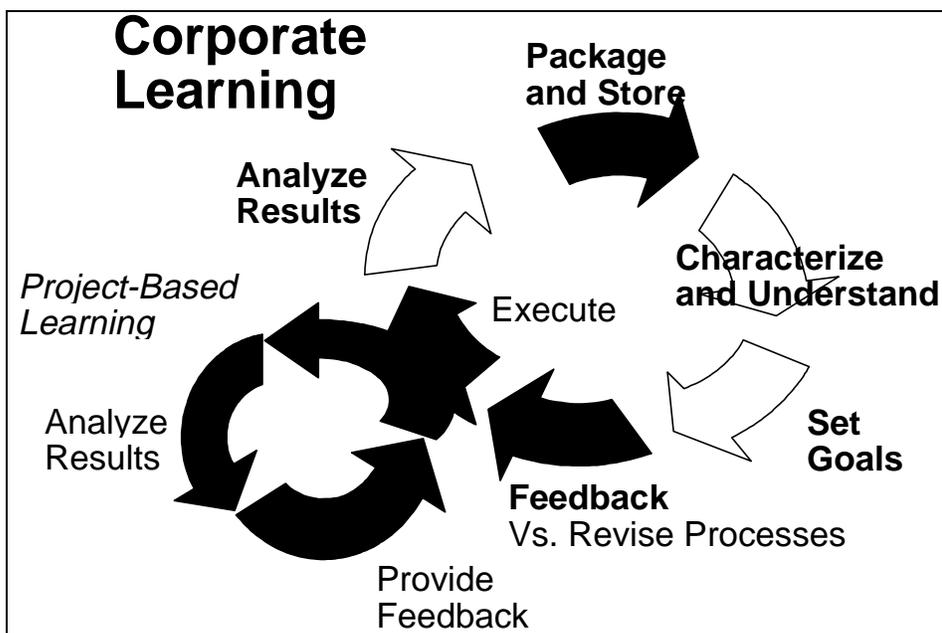


Figure 2. Corporate learning cycle.

The specific elements of the corporate learning process that the demonstration program attempts to model are: (1) capturing raw lessons learned, shown by the “execute” arrow, (2) saving the raw lessons learned and action taken by reviewers, shown by the “package and store” arrow, and (3) *ad hoc* user searches of design review lessons learned, shown by the “feedback” arrow. As the lessons learned demonstration evolves, other aspects of the corporate learning cycle will be investigated.

Drafting Potential Lessons Learned

While lessons learned systems should promote submission of potential items for consideration, many submissions are not stated clearly enough to be interpreted outside the context of the project on which the issues were noticed. Examples of three proposed lessons learned that are poorly formed and should be screened by the reviewer(s) are:

The design team shall consult with the Base Civil Engineering Office to review the installation’s program of architectural compatibility.

Detail C on Sheet A-7 does not clearly show how or if the cab glazing units are anchored at the jambs.

Secure rooms and vaults shall have bars in ducts.

The next paragraph is an example of a well-formed proposed lessons learned. Note that the lesson explains why the item is a problem and also provides a reference.

The specifications indicate cooper roof pan lengths to be approximately 45’ long. The Copper Development Association recommends 30’ max. pan lengths, especially in northern climates. Copper expands 1/8” per 10’ for every 100 °F of temperature change. The 45’ long pans with expansion cleats are theoretically possible but not practical during installation.

DrChecks will allow reviewers to identify a review comment as a potential lessons learned and forward the issue to a project manager for evaluation. Once an issue has been submitted, a project manager will be responsible for the evaluation and approval or disapproval of the potential lessons learned. Once a proposed lesson has been checked, it may be approved for general use, sent back to the author for further clarification, or disapproved. Approved lessons learned will be available to project partners and an authorized set of the general public who may use the lessons learned demonstration during future design reviews.

4 Required System Capabilities

Developing computer support for the design and BCO review process means creating systems which amplify the effectiveness of the individual reviewers and the design review team as a whole. Based on this research, a number of required capabilities should be supported by design review systems. These capabilities provide a procedure for creating, reusing, evaluating, and storing review comments on specific projects and on future similar projects. Issues related to the quality of the design review, as well as the workflow of the review, are addressed by functions identified during the course of this research.

This chapter describes the required components of a design review and integrated lessons learned system that were identified during this research. With each requirement is a description of (1) the general set of data items that need to be maintained to support the capability, (2) who should have access to the data, (3) the way in which data will be entered to meet that capability, and (4) a description and listing of data required to implement the capability. Using this approach and the spiral software development model provides that data item descriptions support a rapid translation of the required capabilities into software.

Discussion of the integration of these functions into ARMS is only briefly highlighted in the discussion below. A detailed integration of these issues within ARMS may be considered at a later time by the ARMS Technical Center of Expertise in conjunction with the Design Review Tools Committee and Headquarters proponents.

Creation of Design Review Projects

The first capability of a design review tool is to create projects that may be reviewed. Design review projects will be created by project managers employed at the office managing the design contract for the project. The data required for this function are: project name and unique identification number. Design start date, design completion, bid opening, and construction contract award dates may also be supported by the tool if suspense tracking is enforced. The use of

such workflow information should be flexible enough to be implemented in a variety of modes, depending on the specific work flows implemented at each design or design management office.

The project manager responsible for a project shall be required to manually enter the limited set of project information into the DrChecks system. The time required to enter this data should be less than 5 minutes per project. Modification of existing project data will be limited to those who have been identified as project managers during the registration process. Projects may only be deleted by those who have been identified as system administrators during the registration process.

The following paragraphs explain in detail several required data elements. Table 1 shows a simplified database table containing the minimum set of recommended data elements. Note that several items such as location, office, designer, and manager may be implemented using a foreign key from a reference table in the database or directly as text fields.

Project Identification Number

A formal definition of the project identification number has not been possible due to the variety of schemes used across Army Corps of Engineers District offices. Two numbers will be used to identify projects. Each project shall have a

Table 1. Minimum set of required data elements for design and review projects.

Attribute	Description	Type	Length (characters)
project_id	Unique project identification key for each project.	Integer	Long
description	A brief project description. The 18 character code used by ARMS may need to be expanded to 30 characters for non-Corps offices.	Text	18
location	Includes Base/State/Country name. The 18 character code used by ARMS may be insufficient for a database containing a large geographical area.	Text	18
office	Name of the office completing the design management. The three character code, called "Design District," used by ARMS may need to be expanded for non-Corps offices.	Text	3
authorization	Number shown on 1391 or equivalent project authorization document. Will be required for future links to other integrated systems and ARMS.	Text	20
funding	Number found on authorizing document or provided in standard list of sources. Standard listing of funding sources is provided in ARMS.	Text	20
manager	Name of person initiating the project. Called "Technical Manager" in ARMS.	Text	20
designer	Name of registered A/E firm who will have access to the system. Called "A/E User Name" in ARMS.	Text	20
start date	The award date for the design contract, defaults to current date. Called "project initiation date" in ARMS. Should be provided by future integrated systems.	Date	
end date	The estimated finish date for the design project. Called "estimated RTA date" in ARMS. Should be provided by future integrated systems.	Date	

unique project key that is created when the project is first added to the database. Users will not be able to modify this project key. An additional project number will be provided for user reference.

Project Description

A number of data items may be required to adequately describe the project and allow potential integration with ARMS. These data items include: project description, location, and funding document.

Project Partners

A number of data items may be required to identify the participants in the project. These data items include, but are not limited to, project management office, project manager, and designer.

Project Schedule

Two data items may be required to identify the overall schedule for the project and all of the activity that is required during the lifetime of the project. These data items are the design contract award date and the construction bid date.

Creation of Design Review Phases

Once a project has been created, a “review phase” allows reviewers to enter individual potential design problems on contractually required design package submissions. Because the number of reviews and scope of each review will vary widely depending on the size and type of project under consideration, project managers must be able to create any number of design reviews and assign or enforce suspense dates as needed. The data required to manage a design review are limited to the name of the design submission that is being reviewed and the start and finish date of the review period.

The project manager responsible for a project shall be required to manually enter into DrChecks the few data items required to initiate a design review. The time required to enter this data should be less than 5 minutes per review. Creation of design reviews will be limited to users who have been identified as project managers during the registration process. Reviews may only be deleted by those who have been identified as system administrators during the registration process.

The following paragraphs explain in detail several required data elements. Table 2 provides a simplified database table containing the minimum set of required data elements.

Review Submission Identification

Each review shall have a unique project key that is created when the review is first added to the database. This key will consist of a combination of the project number and the review number. Users will not be able to modify this project key.

Review Submission Description

A textual description of each review will be provided by the project manager. A brief description will be sufficient for each review, since there are typically between two and four reviews per project. The specific description of the review will be determined by the project manager.

Review Submission

To define the window in which review comments are accepted, the project manager will be able to assign start and end dates for the review. Under the demonstration version of DrChecks, the start and end dates of the review will not be restricted.

Table 2. Minimum set of required data elements for design review phases.

Attribute	Description	Type	Length (characters)
project_id	Foreign key from the project object to allow inheritance	Integer	Long
review_id	Unique review identification key for each review	Integer	Long
name	Name of the review to be conducted. The 38 characters used in ARMS are adequate for describing most reviews	Text	38
submittal_date	Date that the review submittal is expected from the A/E. This date should not be prior to the design award date	Date	
review_date	Date that all comments for the review should be completed. This date should not be later than the construction award date.	Date	

Preparation of Design Review Comments

Once a review has been created, reviewers may begin to document their evaluation of the specific set of plans and specifications distributed for that review. Four types of information are required for design review comments:

1. Project context information defines the location to which the comment applies on the plans and specifications being reviewed.
2. Comment context information provides links to relevant indexes, which will allow others (project managers, other reviewers, and evaluators) to easily find the comment in the future.
3. The comment itself.
4. The identity of the author.

Typically the text of a comment will be entered manually by a reviewer. For other data elements required to completely describe a comment, Graphical User Interface (GUI) tools such as drop down list boxes, check boxes, and radio buttons will be provided. In addition to manual data entry of comment text, users may paste information copied from other data sources. The ability to cut and paste comments from references or past review comments saves significant time for the reviewer.

Creation of design review comments will be limited to users who have been identified as reviewers during the registration process. All project managers will also be identified as reviewers. Once submitted, comments may not be individually modified or deleted. Deleting projects and reviews, which may only be done by those who have been identified as system administrators during the registration process, will also delete all associated comments.

The following paragraphs explain in detail several required data elements. Table 3 shows a simplified database table containing the minimum set of required data elements. Note that several items such as discipline, customer, location, and specification section may be implemented using a foreign key from a reference table in the database or directly as text fields.

Table 3. Minimum set of required data elements for design review comments.

Attribute	Description	Type	Length (characters)
comment_id	Unique comment identification key for each comment.	Integer	Long
project_id	Foreign key from the project object to which this comment belongs to allow inheritance.	Integer	Long
review_id	Foreign key from the review object to which this comment belongs to allow inheritance.	Integer	Long
author_id	Foreign key from the person object who created the comment to allow inheritance.	Integer	Long
created	The date that the comment was created.	Date	
discipline	Discipline of the consultant who should review the comment. The three character standard codes provided by ARMS may not be sufficient for non-Corps offices. A reasonable user interface requires that an English list of disciplines be provided. A foreign key would be more efficient.	Text	3
sheet	Location in drawing on document to which the comment applies.	Text	5
detail	Provides exact location where comment is applicable.	Text	5
spec	Specification to which the comment applies.	Text	5
customer	If customer specific, enter the customer's name. This should be a foreign key from the list of offices.	Text	50
location	If location specific, enter the location name. This should be a foreign key from the list of locations.	Text	50
reference	If there is a citation of code or standards, the data may be placed in this field. This should be a foreign key from the list of available reference materials.	Text	50
lesson	Is the item to be submitted as a potential lessons learned? If so, additional processing will occur if the user selects Y.	Y/N	
text	Statement of what is to be fixed to correct the potential problem.	Memo	

Project Context

Each comment shall have a number that is created when the comment is first added to the database. This key will consist of a combination of the project number, the review number, and the comment number. Users will not be able to modify this project key.

Comment Context

Each comment shall be identified by a set of relevant indexes, selected by the comment author, that will allow the user and others to retrieve the comment in the future. This information will include specification number, drawing sheet number, room number, and design discipline. References, when appropriate, should also be cited.

Comment Information

The text of the comment should be provided in two parts. In the first section, the reviewer should identify the potential problem that has been addressed. The second part of the comment should indicate the recommended change. While text-based evaluation of contract documents has been used successfully, the exchange of graphics should also be possible through DrChecks.

Author Identification

Every comment generated on a project must have an author that is a registered user of DrChecks. The author's name, telephone number, and e-mail address will be provided to comment evaluators if clarifications are needed.

Evaluation of Design Review Comments

Once comments are provided for a given design review, A/E firms and engineering or other consulting firms will evaluate those comments to determine if the issues addressed are actually problems with the current design and to explain what action, if any, is to be taken to resolve the problem.

The designer or consultant will respond to each comment with the following required information: (1) concur/nonconcur, (2) explanation for nonconcurring responses, (3) general category of reasoning for nonconcurrency. While the date of creation and author of each response will be tracked, suspense tracking for designer responses will not be explicitly included in this system. The designer and system administrator may generate reports of comments that were generated and resolved at a given design review phase.

Three types of information are to be maintained related to review comment evaluations. The first item is a link between the evaluation and the original comment. In the demonstration version of DrChecks, one evaluation field will be provided for a given comment. The next set of information documents the designer's evaluation of the comment. Finally, the name, phone number, and e-mail address of the A/E firm or consulting company representative who completed the evaluation will be appended to the evaluation.

The A/E or consulting firm identified as the responsible party by the comment author shall identify if the issue is to be resolved ("concur") or if the issue is

irrelevant to the current design (“nonconcur”). Links between the evaluation and the author of the evaluation will be created automatically.

Evaluation of design review comments will be limited to users who have been identified as designers for specific projects during the registration process. Once an action has been taken to evaluate a comment that comments may not be individually modified or deleted. Deleting projects and reviews, which may only be done by those who have been identified as system administrators during the registration process, will also delete all associated comments and evaluations.

The following paragraphs explain in detail several required data elements. Note that the items provided in Table 4 are items that should be included as part of the review comment table.

Comment Evaluation Context

Blank evaluation fields will be created when a comment is created. Each comment shall have a number that is created when the comment is first added to the database. This key will consist of a combination of the project number, the review number, and the comment number. Users will not be able to modify this project key.

Evaluation Specifics

Radio buttons will be used to identify if the evaluator agrees with or does not agree with the comment in question. A text field will be required for the evaluator to identify the action to be taken.

Table 4. Minimum set of required data elements for evaluation of design review comments.

Attribute	Description	Type	Length (characters)
comment_id	Unique comment identification key for each comment.	Integer	Long
project_id	Foreign key from the project object to which this comment belongs to allow inheritance.	Integer	Long
review_id	Foreign key from the review object to which this comment belongs to allow inheritance.	Integer	Long
reviewer	Name of designer who has checked this item.	Text	50
reviewed	Date of the last update to the review portion of the comment record.	Date	
concur	Identification of concur or nonconcur with recommendation.	Y/N	
review	Text of comments provided by the evaluator.	Memo	

Author Identification

The registration information from the evaluator will be automatically appended to the evaluation when the evaluation is completed. There is a single evaluation field for each comment, so the most recent evaluator's identification will be that which is saved during an update of the database. The test of the demonstration version of DrChecks will identify if a more sophisticated approach, to allow multiple evaluators, is needed or practical.

Project Manager Certification

The project manager or designated representative may log into the system and print a list of all outstanding review comments. The project manager certification indicates that the review is closed, and all comments have been adequately addressed. Many offices require such a certification before awarding a construction contract. The data required to allow electronic certification of design reviews is simply the name of the review and the date of certification.

The project manager responsible for a project will be the only one authorized to enter certification data. The time required to enter this data should be less than 5 minutes per review. Table 5 provides a simplified database table containing the minimum set of required data elements.

Back-Check Review

Reviewers may check the designer's responses to their comments and, if appropriate, request that the issue be raised to management for consideration. If the reviewer does not concur with the designer's comments, the reviewer must begin official correspondence using electronic mail to redress the issue.

Table 5. Minimum set of required data elements for project manager certification.

Attribute	Description	Type	Length (characters)
project_id	Foreign key from the project object to allow inheritance	Integer	Long
review_id	Unique review identification key for each review.	Integer	Long
certifier	Name of manager certifying the review has been completed.	Text	50
certified	Identification of certify or noncertify that review has been completed.	Y/N	
certify_date	Date that all comments for the review should be completed. This date should not be later than the construction award date.	Date	

Reviewer-generated reports will be provided to reviewers to allow them to review A/E evaluations of their comments. No additional data elements within the tool are proposed for A/E evaluations that are in dispute. While several additional data elements could be included within the proposed design review tool, dispute resolution issues are beyond the scope of the current study.

Search for Relevant Past Comments

During the execution of a design review, comment authors frequently find issues similar to those that have appeared on previous projects. Reviewers should be able to search past review comments, created by other authors, on the current or any other project contained in the local database.

The reviewer may, while creating a comment, search for related past comments by means of selecting one or more of the following indexes: specification number, plan sheet number, detail number, room number, or keyword. The reviewer will be required to manually type the information of interest into the search screen. No additional data items need to be maintained for this component.

Reference Source Search

During the execution of a design review, comment authors frequently need to access reference materials to confirm pending questions. Reviewers should be able to search available references and apply the result to the current project. The reviewer may, while creating a comment, search for related material contained in on-line references by means of one or more of the following indexes: Specification section, building component, building materials, and key words. The reviewer may select the valid indexes from drop down list boxes. Keywords will be entered manually by the user.

The search form used to prompt users for their needed search information will be created by searching the appropriate reference database for all valid index values. These values will be automatically placed on the search screen.

The reference source object model provided in Table 6 is based on a knowledge base of low slope roofing construction developed under the RA project (East et al. 1995, Appendix B). This information is provided as a description of a reference source that may be used by reviewers as they conduct reviews. This data model is not, explicitly, part of the proposed prototype DrChecks system, since any

number of references could be provided. Tables 7 through 10 are the lookup tables that show indexes used to describe individual items in the knowledge base and links to photographic files that relate to specific information.

Table 6. Reference source table.

Attribute	Description	Type	Length
kbase_id	Unique identification key for each item in the reference source.	Integer	Long
component	The component to which the reference item refers - foreign key.	Text	
material	The material to which the reference item refers - foreign key.	Text	
function	The function about which the reference item occurs - foreign key.	Text	
query	A question posed to identify if the specific reference item is applicable for a specific project.	Memo	
comment	If the question posed for this specific item is relevant for a given project, the comment explains the steps needed to consider the issue being discussed.	Memo	

Table 7. Component lookup table.

Attribute	Description	Type	Length
component_id	Unique identification key for the component item.	Integer	Long
component	A component to be addressed in the knowledge-base.	Text	

Table 8. Material lookup table.

Attribute	Description	Type	Length
material_id	Unique identification key for the material item.	Integer	Long
material	A material to be addressed in the knowledge-base.	Text	

Table 9. Function lookup table.

Attribute	Description	Type	Length
function_id	Unique identification key for the function item.	Integer	Long
function	A function to be addressed in the knowledge-base.	Text	

Table 10. Photo lookup table.

Attribute	Description	Type	Length
photo_id	Unique identification key for the photo item.	Integer	Long
photo	The name of a photo file that is related to the item.	Text	

Lessons Learned

The majority of the information required to fully define a potential lesson learned will have been provided by the reviewer when drafting the design review comment. Additional information required will be solicited by manual input and selection of options provided using GUI tools.

Several additional items must be maintained to support lessons learned. These items describe the context of the design review comment in greater detail. Also, information regarding the evaluation of the potential lessons learned must be captured.

The following paragraphs explain in detail several required data elements. Table 11 provides a simplified database table containing the minimum set of required data elements. Note that several items such as discipline, customer, location, and specification section may be implemented using a foreign key from a reference table in the database or directly as text fields.

Lessons Context

Blank evaluation fields will be created when a lesson is submitted. Each comment will have a number that is created when the comment is first added to the database. This project key will consist of a combination of the project number, the review number, and the comment number. Users will not be able to modify this key.

Lesson Evaluation

Radio buttons will be used to identify the evaluator's agreement or disagreement with the comment in question. A text field will be required for the evaluator to identify the action that is to be taken.

Evaluation Author Identification

The registration information from the evaluator will automatically append to the evaluation when the evaluation is completed. Since each comment has a single evaluation field, the most recent evaluator's identification will be that which is saved during an update of the database. In the demonstration version of DrChecks, the lessons learned object model is based upon data obtained from the Headquarters, U.S. Army Corps of Engineers' (HQUSACE's) CERS.

Table 11. Minimum required data elements for lessons learned.

Attribute	Description	Type	Length (characters)
lesson_id	A unique key for each lesson item.	Integer	Long
project_id	Foreign key from the project object to which this lesson belongs.	Integer	Long
comment_id	Foreign key from the comment object from which this lesson was submitted.		
author_id	Foreign key from the person object who created the comment.	Integer	Long
description	A brief name of the item being addressed.	Text	50
catcode	Department of Defense standard category code.	Text	30
location	If the item is location specific, then this data field should have data.	Text	25
client	Name of the office for which the project is being completed.	Text	15
office	Name of the office conducting the project.	Text	15
spec	The specification number for the item.	Text	5
discipline	The discipline that should be responsible for correcting the issue.	Text	5
code	A code contained in CERS, where the sample data base originated. The code identifies if the issue is related to design, construction, or operations.	Text	3
url	Allows the author of the lesson to add a relevant URL to the lesson.	Text	50
problem	A complete description of the problem that has been encountered.	Memo	
solution	A complete description of the solution to the problem identified in the record.	Memo	
error	Identifies if the issue being submitted is a potential design error.	Y/N	
omission	Identifies if the issue being submitted is a potential design omission.	Y/N	
coordination	Identifies if the issue being submitted is a potential design coordination problem.	Y/N	
cost	Identifies if the issue being submitted may result in construction cost growth.	Y/N	
time	Identifies if the issue being submitted could result in construction time growth.	Y/N	
quality	Identifies if the issue being submitted could result in construction quality problems.	Y/N	
design	Identifies if the issue being submitted results in problems that occur during the design phase.	Y/N	
construction	Identifies if the issue being submitted results in problems that occur during the construction phase.	Y/N	
operation	Identifies if the issue being submitted results in problems that occur during the operations phase.	Y/N	
regulation	Identifies if the issue being submitted may be resolved by a change to the applicable regulations.	Y/N	
guidespec	Identifies if the issue being submitted may be resolved by a change to applicable guide specifications.	Y/N	
created_on	The date that the item was inserted into the database.	Text	6
act_date	The date that action was first taken on the issue.	Date	
act_author	The author of the action that was taken.	Text	15
act_code	The status of the item as it moves from a "P" pending to an "E" evaluation.	Text	1
act_org	The organization with responsibility to resolve the issue.	Text	25
act_office	The specific office with responsibility to resolve the issue.	Text	19
act_officer	The action officer to whom the issue has been assigned.	Text	21
act_taken	The action taken to resolve the issue.	Text	50
act_descr	Additional description of any items needed to resolve the issue.	Memo	

User Registration

Users of the demonstration version of DrChecks must self-register before using any project manager, reviewer, evaluator, or lessons learned function. Users will manually provide registration information through an input form. Following self-registration, the system administrator will manually assign one or more access rights for each user. Assignment of the access rights will be password protected. The Internet Protocol (IP) address of the user will be obtained automatically through the use of Common Gateway Interface (CGI) variable calls and used to validate the user identification once.

Basic address information will be captured in the registration form. The registrant's system CGI variables will also be captured through submission of the form. The system must be able to limit users to those who have first registered with the designated system administrator. The user may provide the necessary registration information on-line, and the administrator will check the potential user's information and assign a password that will be e-mailed back to the user. The administrator may also identify when various users have been logged into the system.

The following paragraphs explain in detail several required data elements. Table 12 provides a simplified database table containing the minimum set of required data elements. Note that several items such as office may be implemented using a foreign key from a reference table in the database or directly as text fields.

Table 12. Minimum required data elements for user registration.

Attribute	Description	Type	Length (characters)
person_id	The unique key to identify each person.	Integer	Long
Title	The title of the person.	Text	50
First	The first name of the person.	Text	50
Last	The last name of the person.	Text	50
Office	The name of the office to which the person belongs.	Text	50
address1	The first line of address for the person.	Text	50
address2	The second line of address for the person.	Text	50
City	The city in which the person's office is located.	Text	50
State	The state in which the person's office is located.	Text	50
Phone	Telephone number of the person.	Text	50
Email	Internet e-mail address of the person	Text	50
Ipaddress	The IP address of the user's computer identified automatically during the registration process.	Text	15
reviewer	Is the individual authorized as a reviewer?	Y/N	
manager	Is the individual authorized as a manager?	Y/N	
client	Is the individual authorized as a client?	Y/N	
designer	Is the individual authorized as a designer?	Y/N	
password	User provided password. This item may be used at a later time to assist in maintenance of user accounts.	Text	10

CADD Integration

Since some users of the DrChecks tool may be reviewing design documents in electronic formats from within Computer Aided Design and Drafting (CADD) systems, an interface between the CADD system and the tool should be developed. Native CADD system forms and input widgets will be used to develop the CADD interface. These forms will capture user-provided and system-provided data to support review comment generation. To the extent possible, based on the underlying CADD model, the CADD Interface will capture relevant design review information and provide this information automatically to DrChecks. The user should have the opportunity to evaluate the computer selected criteria and change those criteria, if needed. No additional data requirements are expected to support this capability.

Electronic Document Transfer

Some users will be accessing plans and specifications electronically, so the ability to transmit marked-up drawings should be available using DrChecks. In addition, field conditions identified during BCO reviews and captured using digital cameras should be included as part of a design review.

At the time of this report, requirements describing electronic document transfer have not been fully developed; however, additional fields could be provided on the tools review comment input screen to provide file names and remote URLs for a given comment. A description of the file may also be required as a data field separate from the comment description. The technology exists within the current implementation of the prototype to transfer files from remote users to the server.

Cost-Benefit Study Information

To the extent reasonable, the costs and benefits of using the tool should be tracked. Costs include: user connection problems, user training at their local offices, time required to register as a user, time required to access DrChecks, time required to enter comments, and other related costs. Benefits associated with the use of DrChecks include improved design quality and decreased construction and operations cost and time.

Costs of using DrChecks cannot be effectively captured through data from within the DrChecks system. Interview information, provided by users testing DrChecks should supply this data at the conclusion of the DrChecks test. Benefits data may be captured during the use of DrChecks by the addition of several data fields which can identify the benefit of including the indicated item in the finished design.

Additional fields will be provided on the DrChecks comment input screen. These fields will allow the user to add necessary information describing the qualitative and quantitative benefits of including the comment in the design. Additional information provided by A/Es may also corroborate the data included by the reviewer.

5 Internet Technology Platform Selection

The WWW serves as the foundation of the prototype DrChecks system, and is an extension of the fundamental capabilities of the Internet. It may be described as a hypermedia-based distributed information system. It provides easy access to many existing network resource tools, such as multimedia players, file transfer protocols, newsgroups, and electronic mail, and is rapidly becoming the major means of access to Internet resources (Boutell 1996). Using the WWW for developing a design review system has several notable benefits. The two most important benefits are: (1) useful tools for information storage and retrieval, communications, and modeling already exist for the WWW, avoiding the need to develop applications which perform similar functions “from the ground up” and (2) a standardized interface for basic interactions between the human and computer already exists. The WWW operates under the client-server paradigm, which means information providers place hypertext documents and other media on servers that accept connections from clients using software browsers. The most common form of hypermedia found on the web is the hypertext document. Hypertext is text with pointers to other text, and hypertext documents for the WWW are written in the Hypertext Markup Language (HTML) (NCSA 1996a). Besides HTML, another markup language, Virtual Reality Modeling Language (VRML), is being developed for virtual reality applications (SDSC 1996). VRML can depict realistic worlds as well as otherworldly places.

Web-enabled design review paradigms use the hypermedia information storage, retrieval, and transmission technologies developed for the WWW. Use of these technologies has at least three advantages over developing systems “from scratch”: (1) since these technologies are available for a wide range of platforms, the system will be robust for users with different computational resources, (2) existing web technologies constitute powerful tools in the design of multi-reviewer, interactive design review paradigms and should reduce the cost of investigating and creating new technologies, and (3) with increased use of the WWW, more people will be familiar with the method of interaction of systems based on the same interaction paradigms, reducing the training costs of familiarizing users on individual user interfaces for each new software system.

The variety of web-based technologies available to developers has expanded rapidly over the past 5 years. This section attempts to capture the rapidly evolving state-of-the-art technologies relevant to web-enabled design review paradigms and to point out needs in the existing tools. While it is not a comprehensive nor an exhaustive survey of web technologies, it highlights some of the relevant areas emerging on the WWW for discussion of the capabilities of web-enabled design review systems.

Hypertext Markup Language

As stated earlier in this chapter, the WWW uses the client-server paradigm for information transfer, and a server contains information that can be accessed by clients with the appropriate software browser. Hypertext documents defined by the HTML are the most common form of information found on the web. HTML contains commands to make ordinary text behave in certain ways (i.e., to make text appear in boldface, in italics, or to draw lines on the page, to include graphics, and to act as a reference to another HTML document). HTML alone is best suited for creating static displays of information.

Limited forms of whiteboarding, or the sharing of documents, are also available over the web. Although the current forms of whiteboarding primarily involve the transmission of graphical images (e.g., annotated screen or window captures), the concurrent use of application documents like spreadsheets or word-processing documents will become readily available soon.

Retrieval of information on the WWW may be handled by using a Wide Area Information Server (WAIS) (WAIS 1994). The server has databases containing hypermedia (although most existing databases contain primarily text-based documents). The databases may be organized in different ways using various database systems, but the client is not required to learn the query languages of the different databases. Instead, a client may search for relevant information using natural language queries.

CGI Scripts and Java

Because HTML alone does not have the ability to perform actions dynamically (e.g., to perform calculations, process user input, display data about the system, etc.), a number of extensions have been developed to provide these capabilities.

Two types of extensions that can run from HTML documents are CGI scripts (NCSA 1996b) and Java™ (Sun 1996).

A CGI script is designed to be run by a server's web page. The CGI itself is a set of rules about how the web server and the program exchange information across the web. Examples of CGI scripts include counters for web page access, forms for inputting information, and even animations.

Java is an object-oriented programming language developed by Sun Microsystems, Inc. It shares many superficial similarities with the C++ object-oriented programming language but, in fact, was developed from the ground up incorporating many ideas from other programming languages as well. Java was designed to allow secure execution of code across networks. Code intended for this manner of use is called an applet, and is run on browsers with the capability of executing Java code.

Telephony

Fully duplex audio connections are available between pairs of networked users. These programs allow real-time voice transmission between two parties, much like a person-to-person call. Multiparty voice conferencing over the Internet is still being developed, although several vendors offer specialty hardware for audio conferencing over local area networks (e.g., Ethernet or FDDI networks). The Multicast Backbone, or MBONE (1994), is used by researchers for developing protocols and applications for group communication. Multicast technology provides one-to-many and many-to-many network delivery services for both multi-user video and audio conferencing, but various technological issues still need to be overcome, including limited bandwidth for information transfer, and the design of efficient and robust routing protocols.

Internet Relay Chat (IRC)

IRC (Ohio State 1996) is a multi-user text-based chat system for the Internet. It is based on a client-server architecture where clients with an Internet connection and the IRC client software running on their machine connect to a server that contains one or more "channels" for conversations with other clients. Each channel is a virtual place, usually with a topic of conversation.

Virtual Reality Modeling Language

The VRML is a markup language to provide interactive, three-dimensional environments. The first version of VRML (SDSC 1996) defined the representation of three-dimensional worlds and movement within them. Figure 3 shows an example of a VRML building model. Packages are currently available to translate CADD drawings into VRML files. Directions for later VRML standards have included better interaction between users and between users and the environment (SGI 1996).

The development of virtual worlds and interactive environments are useful for the virtual design review. Together these environments have established several common modalities for interaction. One mode is the visual representation or avatar concept. In virtual reality multi-user environments, interaction between users is enhanced through the use of avatars of the users. The avatars may “talk” with each other, exchange information, and even bump into each other. Currently the standard form of inter-avatar communication is textual in nature. For example, anything you type may appear in a speech balloon above your avatar’s “head.”



Figure 3. VRML building model.

VRML-based virtual worlds are relevant to this discussion. Existing virtual worlds generally feature a subset of the capabilities envisioned for the virtual design review. An example of a virtual world is Alphaworld by Worlds Chat.* Alphaworld is a multi-user community where users that “immigrate” to Alphaworld select a representational avatar and are able to acquire property, construct their own buildings, and interact with other users’ avatars. Communication between users is textual in nature only and appears in speech balloons over the avatars’ heads as well as in an IRC-like dialog box.

Client-Server Operation

One of the most important aspects of the Internet, from a business perspective, is that data may be served to users from corporate databases. This service expands the ability of offices to share data and communicate with each other. A variety of commercial database server systems exist. The operation of a client-server database system operating across the Internet is illustrated in Figure 4 and the steps following.

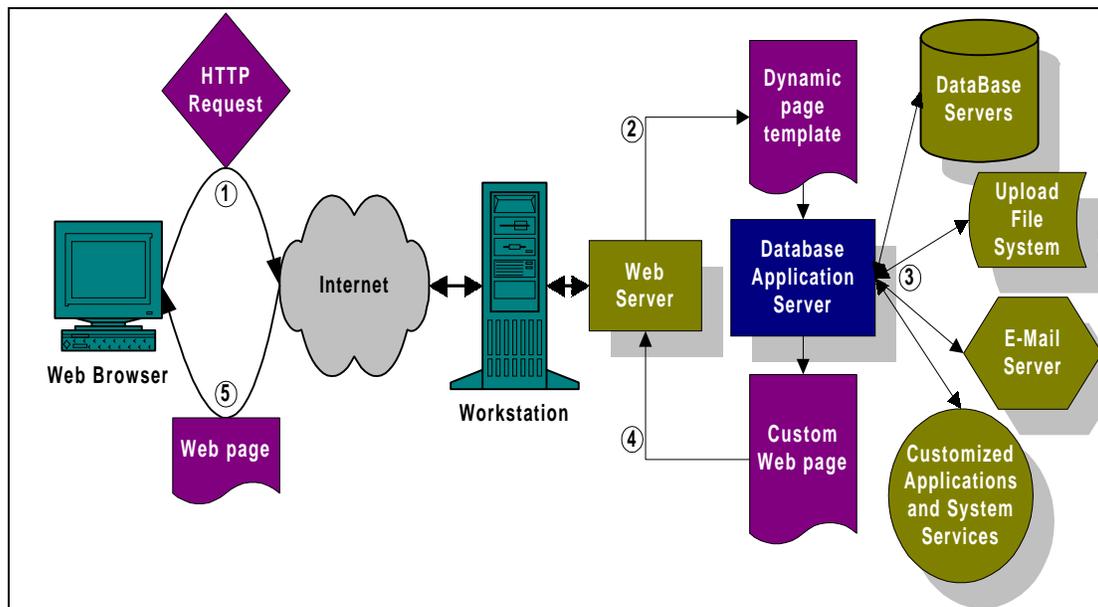


Figure 4. Client server operations.

* Worlds Chat website: <http://www.worlds.net/wc/>

Step 1. When a user clicks a “Submit” button on a form or a hypertext link on a web page, the user’s web browser sends an HTTP request to the web server via the Internet.

Step 2. The web server passes the data submitted by the client and the appropriate template file to the database server through a server API. CGI programming may also be used; however, the application of CGI programming is more cumbersome than that required when using the API.

One important consideration in this step is that the database server may reside on the same physical machine as the web server, or it may reside on a completely different web server.

Step 3. The database server reads the data from the client and processes specialized HTML tags used in the template. Based on the data command tags, the application server interacts with the database server, the file system, mail servers, and potentially with other applications and extensions.

Step 4. The database server dynamically generates an HTML web page which is returned to the web server. These web pages do not physically exist, but are recreated each time a user request is received.

Step 5. The web server returns the HTML page to the user’s browser.

Benefits of an Internet-Based Solution

The use of client/server technology through the web has been widely supported through the development of web services within an organization (Intranets) as well as web services that support the organization and its customers (Extranets). The benefits of these systems include:

- Systems based on web technology use a commercially supported software system as the basis for all transactions and data transmission.
- No licensing fees are assessed for using the application-specific portions of web technology.
- User’s software does not need to be constantly upgraded. As Internet technology improves, the technology may be delivered to users desktops as

part of upgrades to basic operating systems in the context of upgrades to e-mail or other communication systems.

- Users do not have to learn “yet another piece of software” to perform design review. Once they know how to use a web browser, they can use systems based on the WWW.
- Most offices already have a personal computer that can be used as a design review/lessons learned server.
- Widely available commercial software is used for the basis of the proposed tools. The cost to most offices will be the purchase of the database server, costing about \$500.
- Maintenance costs for the server will be under \$1,000 per year. Typically, these costs will be for the purchase of commercial software upgrades, not for proprietary software. Local personnel may be used to backup and check the web server as part of their current tasks.
- The simplicity of developing web-based solutions also means that offices may customize basic applications to suit specific work practices at that office, while complying with a national database standard.

6 Constraints to System Design

Support of All Computer Platforms

One of the greatest impediments to developing software systems before the advent of the WWW was that of multiple computer platforms. Users of DrChecks may use any of the currently available hardware and software platforms, including IBM-compatible personal computers (PCs), MacIntoshes, and Unix systems. This platform independence is vital because it enables all Corps project partners to contribute to DrChecks.

Cost of Complete System

The cost of DrChecks must be such that a Corps of Engineer District is able to implement the system at a very small initial cost. If typical existing equipment is used, the first cost of DrChecks drops to below \$1,000. The total first cost of DrChecks, for new hardware and software, is under \$5,000. The minimum DrChecks server is an Intel 486 running Windows NT 3.51 and the shareware EMWAC web server.

Cost of System Maintenance

Maintenance of the product should be such that the duties can be incorporated into the office without requiring additional staff. Offices interested in DrChecks will be able to use in-house talent to operate and maintain DrChecks.

Use of Commercial Software

The adaptation of commercial-grade software for use in DrChecks is the underlying foundation of the system. DrChecks has been developed using the most

current, commercially available software. As vendors upgrade their software, additional functionality may be incorporated into DrChecks and tested.

User Training

All users of DrChecks should have sufficient general computing skills to allow them to start their own web-browser software, use typical types of web-forms, and use electronic mail.

User Access to the World Wide Web

To support application development on the WWW, all users of DrChecks must have access to the Internet. Corps of Engineers and other government offices will need to work with their information management departments to obtain the necessary network connections.

Private companies needing to access DrChecks will need to obtain Internet access through a local Internet service provider (ISP). It is suggested that the provider that is selected assign the user a permanent IP address.

System Access Security

Access to data in the DrChecks system is restricted to only those who are working on the project under review. System access is conditional upon the user's initial registration and subsequent assignment of access. The system administrator may grant access to users at one or more of the following levels: design project manager, design reviewer, client, designer, or consultant.

Access to Sensitive Project Data

Access to project data by U.S. Army Corps of Engineers' clients and private design firms should be limited to only those projects on which the client or designer has specific contractual requirements.

Web-Browser Software

To access the WWW, users must have one of a number of commercially developed browsers. The browser interprets the information provided in Hypertext Transfer Protocol (HTTP) from web servers to the user's computer when the user visits a given web site. These programs are widely available and free of charge from a variety of software vendors.

Army Corps of Engineers and other government offices will need to work with their information management departments to obtain the necessary web-browser software. Private companies needing to access DrChecks will obtain web-browser software through their local ISP.

In the course of other business with DrCheck's users, it is assumed that local information management offices and ISPs will provide support and maintenance for the web-browser software. No additional user maintenance cost will be required to support DrChecks.

Use of Advanced Design Tools

The pages displayed on the WWW are developed in the HTML, which allows any of the large number of commercial browsers to view a given page. As with the rest of the evolving WWW, the HTML standard changes over time. To support the widest variety of users, pages comprising DrChecks will contain only the commonly used HTML elements.

The practical effect of this constraint means, at this time, that DrChecks will not contain any of the elements called "tables" and "frames." In addition, browser-specific or plug-in features will not be supported in the prototype DrChecks.

Integration of DrChecks with ARMS

Lessons learned from the development of DrChecks may be adopted by the Automated Review Management System's Technical Center of Expertise (ARMS TCX). The use of the DrChecks prototype by ARMS TCX has not been determined at this time.

7 Computer Hardware and Software Minimum Requirements

This chapter provides the standards for all hardware and software to be used in the testing of the prototype system. In general, the required hardware and software is already in place at most offices. The stumbling block appears to be connection and management acceptance of the Internet and WWW as business-related tools.

Hardware Requirements

The server computer operating the demonstration version of DrChecks evaluation will be a personal computer with an Intel™ Pentium® Processor and a 2 GB hard drive. During the evaluation of the demonstration version of DrChecks, alternative platforms will be discussed. The only restriction on the computer hardware of clients is that the hardware selected be compatible with HTML version 2.0.

Software Requirements

The server will be a PC operating with the Microsoft Windows™ NT version 3.51 operating system.

Client systems may use any operating system, provided that the clients have access to an HTML-2.0-compliant WWW browser. Browser programs supporting this industry standard include Netscape™, version 3.0 or better, or Internet Explorer™, version 3.0 or better.

Communications Requirements

The server system must be linked to the Internet using Transmission Control Protocol/Internet Protocol (TCP/IP). The recommended connection speed for the

server system will be developed through the test of demonstration version of DrChecks.

Client systems must be lined to the Internet using TCP/IP protocols. Connection to the server will be accomplished through the WWW. Based on the anecdotal experience of the authors, the minimum connection speed for clients testing DrChecks should be 9600 bits per second (bps). A recommended speed of client connections to the web will be identified through the test of the demonstration version of DrChecks.

8 Software Development Methods

This chapter provides references for the software development methods and techniques used during the execution of the development effort. These references are provided for those who, after obtaining copies of the prototype system, wish to make modifications.

Object-Oriented Modeling

Software requirements identified in this project were modeled using the object-oriented modeling technique as described in Rumbaugh et al. (DATE). These models were translated into the relational database-style tables shown in Chapter 3 (Tables 1–12). A draft version of the object-oriented design may be found at <http://www.cecer.army.mil/pl/ra/drchecks/model.htm>.

Standards for Software Products

The basic language of the prototype system is HTML, an application of ISO Standard 8879:1986, Information Processing Text and Office Systems; Standard Generalized Markup Language (SGML). Future developers should have no problem finding a wealth of information on HTML on the Internet.

Communication standards and more technical information may be found through the official organizing body of the WWW. Information on this body may be found at <http://www.w3.org/pub/WWW/Consortium>.

Reusable Software Products

One of the major focuses in software design is that of reusable software components. Care has been taken in prototype development to ensure that pages may be used on a variety of servers and viewed with browsers of version 3.0 or later. With the exception of the database engine that provides the

capability of querying from HTML forms and returning information in HTTP format, the entire prototype may be delivered on a variety of commercial Windows-NT-based platforms.

ODBC Compliance

The commercial product which drives the HTTP database queries of the prototype's pages uses the software-industry standard "ODBC" database protocols. All code developed in the prototype may be easily used for any ODBC-compliant database provided that field and table names are the same as those found in the prototype database.

Security Mechanisms

To be accessible by our customers and clients, the prototype systems must be available to the entire Internet community. Installing the prototype behind a firewall of network security will significantly limit the usefulness of the tool to other project stakeholders.

One of the issues to be resolved during the testing of the prototype is the use of IP addresses for security purposes. Many business persons with e-mail addresses have a unique IP number assigned to their PC. Portions of this IP address can be used to identify each user, their organization, and their Internet domain. For example, the "army.mil" section of the e-mail address "b-east@cecer.army.mil" corresponds to an IP address that ends in "129.229."

The proposed registration scheme is to automatically capture users' IP addresses as they register. The IP address will be stored to identify the user for subsequent uses of the prototype. The real benefit of this system is that, after registration, the user will never have to log into the system again. Each page that contains sensitive data will check to ensure that the user has registered with the system.

While the above scheme should be useful for those with permanent IP addresses, some users of the Internet, particularly small A/E firms, may have IP addresses that are dynamically assigned by their ISP. An alternative method for logging into the system will have to be included in the current prototype to allow users without permanent IP addresses to use the system.

Another approach that is also being tested is for the server system to assign identification numbers to individual browser software. These server-assigned and client-stored variables are referred to as “cookie” variables. To accept such variables from servers, users must set permission to assign these variables in their browser software. Additional information regarding cookie variables may be found through any WWW search engine.

Computer Hardware Resource Utilization

The extent to which a typical Windows NT 3.51 or 4.0 web server is able to scale-up to a full test with hundreds of users and thousands of items has not been specifically tested in this project due to time constraints. Based on comments of other users of the same database and web server platform used for the prototype, the selected platform is well within performance levels expected by users.

In terms of the resources needed by the client, no additional software will be required. Browser software should be installed as part of future operating system upgrades or as part of an office’s concurrent implementation of e-mail and other Internet services.

Identification of Project Stakeholders

Project participants self-register with the system at the beginning of a project. A system administrator or manager will then identify each participant as a project manager, reviewer, customer, or consultant. It is also possible that all registered users can be given a guest login that allows users to search restricted resources.

Access to Project, Review, Comment, and Evaluation Data

Write access for all design review data contained within the prototype will be restricted those people responsible for authoring that data. Project managers will have access only to project and review creation. Reviewers may only author new comments. Specific individuals may have both project manager and reviewer access. Customers will have author-level access. Consultants will be able to evaluate comments.

Access to these pages will be limited by checking the incoming user's IP address through browser-provided CGI programming variables against the set of registered users. To use this system efficiently under this scheme, users must access the prototype from the same equipment with fixed IP addresses.

In the prototype version, only the system administrator will have the ability to delete projects or reviews within a project. Once a project is deleted, all comments associated with that project will also be removed from the database.

Access to User Registration Information

The public will be able to obtain a list of persons registered to use the DrChecks system. This list will include the name, company affiliation, telephone number, and e-mail address. The name, telephone number, and e-mail addresses of each comment author or evaluator will be included with each review comment or review comment evaluation. IP address information will only be provided to individual users to allow them, or system administrators, to validate registration information.

9 Design Review and Checking System

This chapter provides a brief description of DrChecks, which was developed to test the usefulness of the Internet in creating simple and effective tools for design review and lessons learned. To explain the operation of the system, several screen captures have been included here.

To find the DrChecks demonstration system, users must begin at the page for the steering committee who assisted in this research. The URL for this page is <http://www.cecet.army.mil/pl/ra/committee>. Once at the committee homepage, the link to DrChecks can be found. Clicking that link opens the homepage, shown in Figure 5. All of the tasks that may be accomplished by DrChecks can be found through this single homepage. Access to these functions is based on user registration and subsequent administration approval by the system.



Figure 5. DrChecks homepage.

One of the issues discussed during demonstrations of DrChecks, and other web-based client-server applications, is the need for homepages for persons with different levels of access. Not all functions of each client-server system are available to all users, so serving only those portions of the program which are meaningful to a given user may be a very useful feature of commercial quality web-based systems.

User Registration

When accessing DrChecks for the first time, users are required to register if they wish to access any of the design review or lessons learned submission functions. Unregistered users may review on-line references.

Figure 6 shows the user registration page, which includes postal and electronic address information as well as the “name” of the computer on which the user has accessed the system. The IP address information is one of a number of pieces of data that is transmitted from the client’s browser to the server.

The screenshot shows a web browser window titled "Design Review and Checking System - Microsoft Internet Explorer". The address bar contains the URL "http://www.2.cecei.army.mil/cgi-bin/da/lex?Template=search/searchLesson.dbr". The main content area has the heading "Check Lessons Learned" and a sub-heading "Use the form below to review DrChecks' on-line lessons learned sources. You may also return to DrChecks' homepage or get help." Below this is a paragraph of text: "Over 4,000 items gathered from projects across the Corps of Engineers are contained in this resource. The first fifty matching items will be returned to you. Note that all text fields will match on uncompleted data. For example the location 'CA' for California may also return lessons from 'CAiro' Egypt!" The form is divided into four sections: "Part 1. Identify Selection Criteria:" with fields for "Keywords:", "Location:", "Spec Number:", and "Related To:" (with radio buttons for "Design", "Construction", and "Operations"); "Part 2. Sort By:" with radio buttons for "Specification" and "Location"; "Part 3. Identify Type of Search:" with radio buttons for "Matching All" and "Matching Any"; and "Part 4. Submit Your Query:" with a "Search Now" button.

Figure 6. User registration page.

Corporate users have a unique IP address assigned to their desktop computer, so the IP address is the basis of security within DrChecks. A database lookup is conducted on each DrChecks page to determine if the user's IP address is known. If the IP address was not previously recorded in the DrChecks user table, then user access to the pages is denied and an error message is generated.

Commercial or future implementations of DrChecks should revise this security arrangement in lieu of an arrangement that assigns cookie variables to a user's browser. This revision will be required since many commercial services (such as those used by consulting engineer or architectural firms) dynamically assign IP addresses to users as they connect by modem.

On-Line References

During the development of past design and BCO review tools, a number of users indicated the need to have electronic access to reference materials. As an example of the types of references currently available through the Internet, a reference page was included within DrChecks.

All users, those who have registered and those who have not, are able to access a variety of on-line references through DrChecks. Figure 7 shows the page that serves these references to users. The two references developed in conjunction with DrChecks, the "CERS Lessons Learned" and "Roofing Knowledge-Base," are described in the following paragraphs.

The first three references shown in Figure 7 are those developed in conjunction with DrChecks. "Check Previous Review Comments" searches those comments that have been entered directly into DrChecks by users testing the system. The second button "Check Approved Lessons Learned" provides access to over 4,000 design-related items extracted from CERS maintained at HQUSACE, Military Programs Directorate, Construction Evaluation Branch (CEMP-CE). The final reference is a knowledge base of low-slope roofing information developed under an eight-man-month effort. This knowledge base also includes photographs of actual roofing construction appropriate to various knowledge-base items.

Other references on this page include search routines that run against the standard library of CADD details and Corps of Engineers Guide Specifications. Local references such as the Mobile District's lessons learned system are also linked through this page.

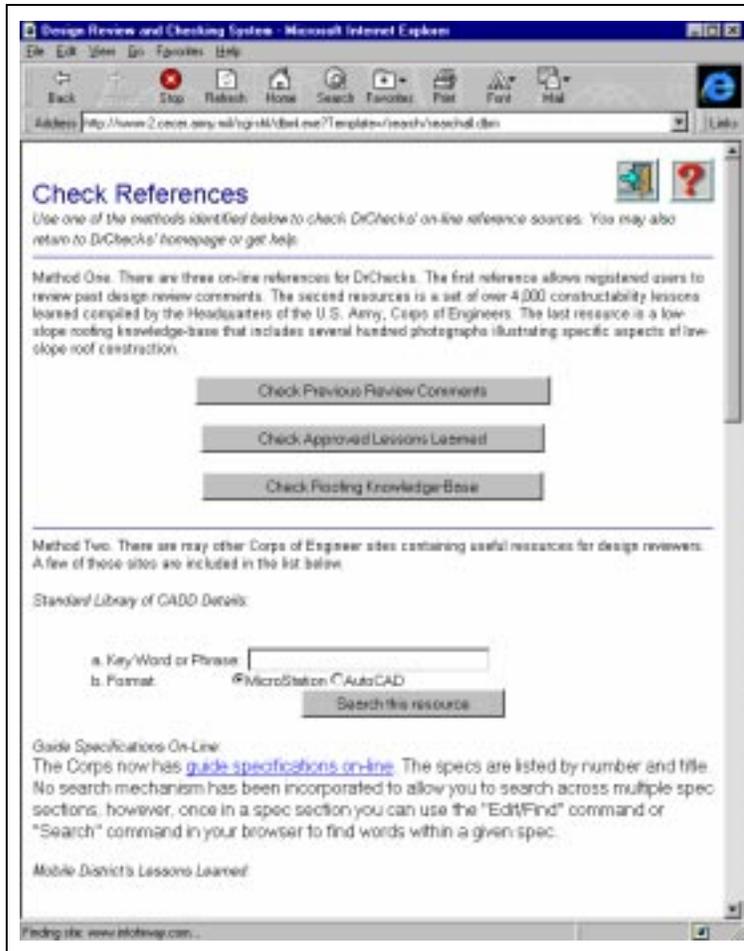


Figure 7. On-line reference page.

CERS Lessons Learned

CEMP-CE conducts periodic visits to construction projects to assess the quality and effectiveness of the Corps Quality Assurance Program. Items that are in violation of guide specifications or stated Corps policy are documented by teams of inspectors and provided to the local office for action. The items that have been documented are actually captured in CERS. While CERS contains many items related to repetitive construction deficiencies, a large number of items also relate to design and design criteria. Those CERS items that were noted as design related have been provided as a set of sample “lessons learned” within the DrChecks demonstration.

To find items of interest within the CERS data, users enter data in appropriate fields in the search screen shown in Figure 8. The DrChecks database server builds a database query from all the information provided by the user and returns that information for the user to cut and paste into a current design review comment.

Check Lessons Learned

Use the form below to review DrChecks' on-line lessons learned sources. You may also return to DrChecks' homepage or get help.

Over 4,500 items gathered from projects across the Corps of Engineers are contained in this resource. The first fifty matching items will be returned to you. Note that all text fields will match on uncompleted data. For example the location "CA" for California may also return lessons from "CAko" Egypt!

Part 1. Identify Selection Criteria:

Keywords:

Location:

Spec Number:

Related To: Design Construction Operations

Part 2. Sort By:

Specification Location

Part 3. Identify Type of Search:

Matching All Matching Any

Part 4. Submit Your Query:

Figure 8. Search CERS lessons learned.

Figure 9 shows the results of a search conducted on the CERS database using the two keywords “concrete” and “placement.” Details of relevant returned items may be explored if the user clicks on the item number in the top left corner of each comment.

While the use of *ad hoc* queries is very useful within such a large database, improved indexing mechanisms may also be useful to provide a measure of the quality of the match achieved by each returned database record. Most readers will be familiar with this type of search, which is often found on generalized web search engines. These engines return a percent match for all items found. Such a ranked searching mechanism should be included in future or commercial versions of DrChecks.

Roofing Knowledge Base

The other on-line reference source developed during the course of the work on DrChecks was the low-slope roofing knowledge base. Since problems with flat roofs are second only to problems with mechanical systems, and since the number of components of flat roofs is relatively small, it was felt that the development of a tool to assist in the design review of this building system would be very useful.



Figure 9. Example CERS search results.

The low-slope roofing knowledge base requires that the user indicate three pieces of information before any data is returned: (1) the roofing component of interest, (2) the material of which the component is made, and (3) which specific function of that component is being investigated. The material type and function questions may be answered by selecting the desired values from the list of relevant items, or by selecting the default value of any material type and all functions.

The results provided from this knowledge base are illustrated by the following example: a user wishes to check design of a steel roof deck, of the type common found on many light industrial facilities, and wants to make sure that the deck is able to support all design and superimposed loads. Based on these search criteria, a list of items is returned that the user should check. One such item is provided in Figure 10.



Figure 10. Knowledge-base search results.

The photographs associated with many of the items in the knowledge base provide a valuable reference source for the reviewer, particularly for the junior engineer who may not be familiar with details of specific roofing components.

Manage Projects and Reviews

Figure 11 shows an example of the project listing page. From this page, new projects may be added, using the form at the bottom of the page. Reviews for each project may be added by “drilling down” on the project identification number associated with a project.

* Drilling down is a reporting style whereby one line of a record is displayed; when the user clicks on that line, the entire record is displayed.

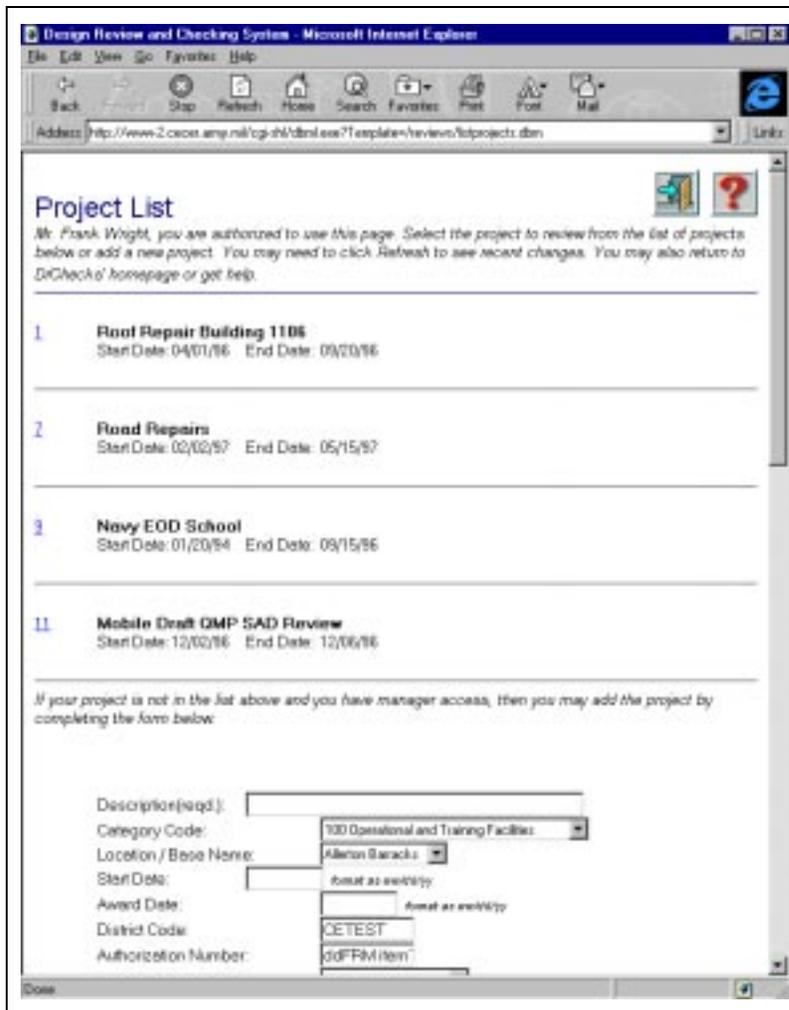


Figure 11. Manage projects.

Add Review Comments

Before adding review comments, the reviewer must select the project and review to which the comments should be added. Figure 12 shows an example of a page used to select from one of the reviews currently set up for a selected project. The reviewer may either add new comments to a given review or search for comments that may have been acted upon in previous reviews.

In the prototype, no restrictions are placed on suspense dates for reviews. Options could be set to restrict the addition of new comments to reviews for which end dates have been passed. These options have, however, proven to be unpopular with many reviewers because drawings are often not received by the reviewer until after the stated completion date of the review period.

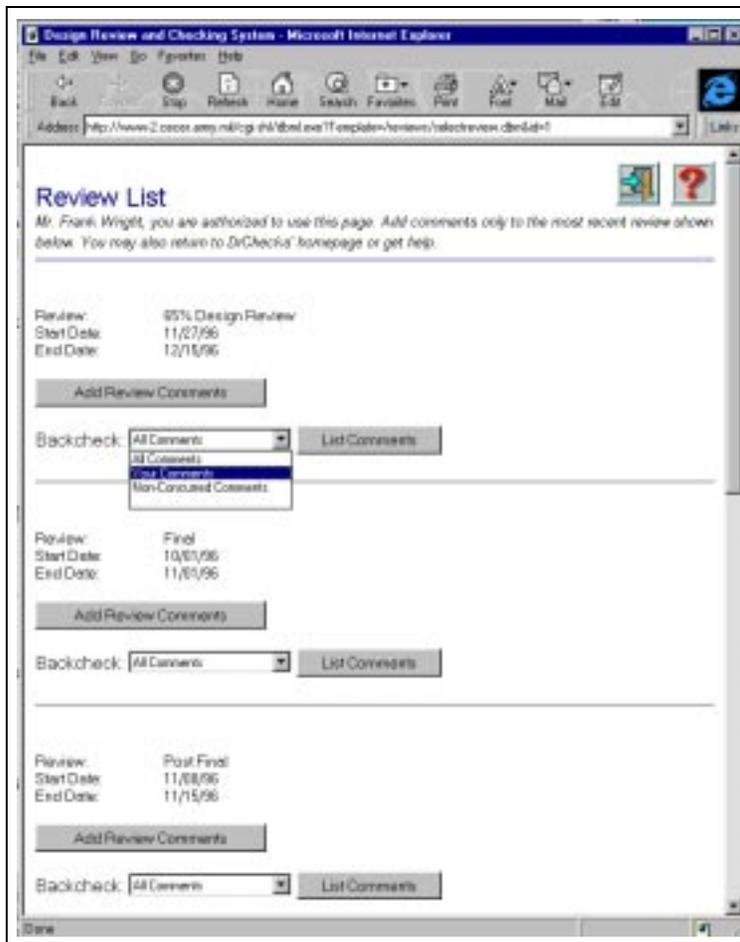


Figure 12. Selecting a review to check.

Once the project and review phase are selected, then the reviewer may add new review comments for the designers to evaluate. Figure 13 gives an example of the page used by reviewers to add new comments. While the set of indexing information above the comment text (discipline, specification section, sheet, etc.) were developed to be consistent with ARMS, this set of indexes has been shown, by previous attempts at automated evaluation of ARMS comments, to be inadequate. Given the large number of comments currently in the ARMS database (approximately 7 million) having indexes needed to identify and extract specific sets of comments is essential and should be considered in future system developments.

Another aspect of the review comment form is that a comment may be identified as a potential lessons learned as well as a review comment on the specific job. If the comment is considered by the reviewer to be a repetitive deficiency, success story, or other important information, then the reviewer may flag the comment using the radio button provided.

Figure 13. Adding a review comment.

Once an item is indicated to be a lessons learned, then the item is inserted as a pending lessons learned into a designated lessons learned database. The screen used to process this lessons learned item is described later.

Evaluate Review Comments

The designer or consultant for a project may select some subset of comments to review and will arrive at the page shown in Figure 14. On the comment evaluation page, the designer or consultant is able to view each comment, in detail, and respond to that comment.

Evaluate Lessons Learned

Comments that have also been identified as lessons learned are transmitted to the lessons learned page. Figure 15 shows a blank form that contains, in the case of design review comments, the comment text and associated indexing information. This information is required to more fully identify the context, potential impacts, and steps to mitigate the proposed item.

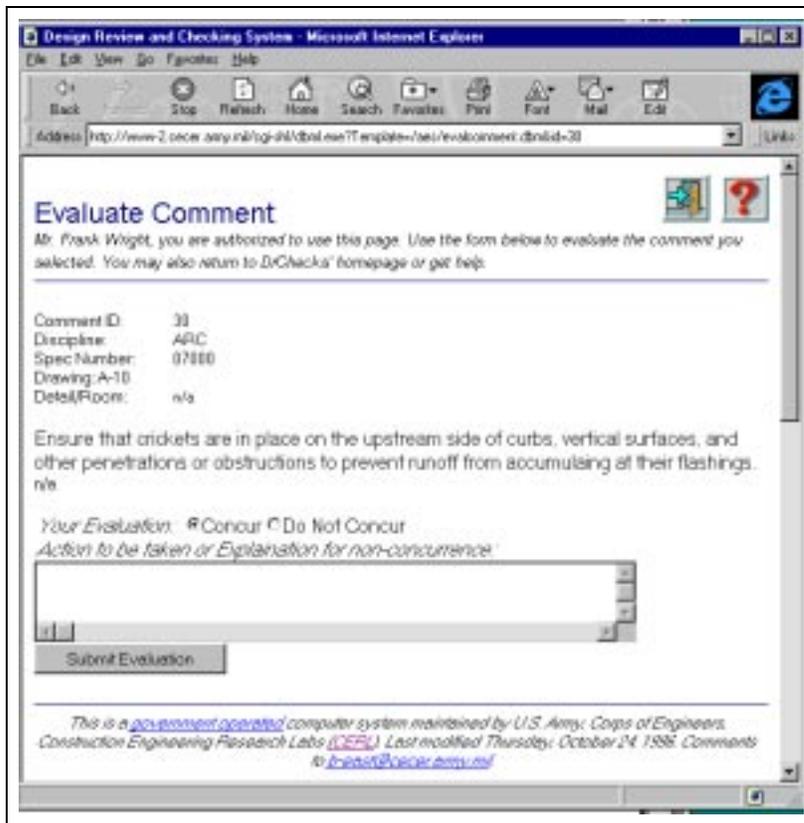


Figure 14. Evaluating a review comment.

A complete work flow of lessons learned processing was not included in the prototype since the purpose of the prototype was simply to demonstrate the effectiveness of integrating design review and lessons learned generation. Future versions of the prototype may have a variety of optional methods to process lessons learned. These methods may include an automated submission of request to change guide specifications through organizational levels from a district office through a division to headquarters.

The flow of lessons learned processing that has been incorporated into the program allows users and reviewers to check on the current status of submissions (see Figure 16). Manager screens contain approval fields not found on the pages displayed to the authors.

The screenshot shows a Microsoft Internet Explorer browser window with the title 'Design Review and Checking System - Microsoft Internet Explorer'. The address bar contains the URL: <http://www.2.usace.army.mil/cgi-bin/ldm.exe?Template=lessons/new.htm>. The main content area is titled 'Propose a Lessons Learned Item' and includes a sub-header: 'Use the form below to enter your proposed lessons learned. You may also return to DChecks' homepage or get help.' The form is organized into five parts:

- Part 1. Identify the Project:** Includes text input fields for 'Project Name', 'Location', 'Client', and 'Office'. A dropdown menu for 'Cat. Code' is set to '100 Operational and Training Facilities'.
- Part 2. Describe the Problem:** Includes a 'Problem Title' text field, a 'Problem Type' section with checkboxes for 'Error', 'Omission', and 'Coordination', a 'Results In' section with checkboxes for 'Cost Growth', 'Time Growth', and 'Quality Problems', an 'Occurs During' section with checkboxes for 'Design', 'Construction', and 'Operations', a 'Spec Number' text field, and a 'Discipline' dropdown menu set to 'Architect'.
- Part 3. Describe the Problem:** A large text area for describing the problem.
- Part 4. Recommended A Solution:** Includes a 'Change' section with checkboxes for 'Regulation' and 'Guide Specs', and a 'Solution' text area.
- Part 5. Submit the Lesson:** A 'Submit Lessons' button.

The browser's status bar at the bottom shows the word 'Done'.

Figure 15. Adding a proposed lesson.



Figure 16. Reviewing lessons learned.

10 CADD-Integrated Design Review

One of the requirements identified in Chapter 4 of this report is the ability of the design review tool to generate comments within the CADD environment. To demonstrate the capability of the prototype system to accept information across the Internet in a variety of sources, a program module was written in the Modular Design Language (MDL) to be used within the MicroStation™ CADD software system. This chapter provides several screen captures illustrating the operation of this program module.

The first phase of the CADD-based design review tool is shown in Figure 17. The user begins each CADD-based design review session by providing the name of the server to which comments will be sent and the number of the design and review. Of course, the server information could be stored in a configuration file, and the design and reviews could have been selected from pull down lists; however, the goal of the demonstration was to test Internet transmission of comments through customized programs developed for MicroStation.

Once the server, project, and review have been identified for the design review session, the reviewer can create new design review comments based on the objects on the drawing. For drawings in native MicroStation, the tag number for the CADD item is automatically identified (see Figure 18). The sheet number of the item identified is also automatically detected and entered into the comment form.

Drawings that are developed through the Modular Design System (MDS) provide additional information such as the standard module number and specification sections. This information is also added automatically to the design review comment, freeing the reviewer from selecting many of the needed comment indexes. Figure 19 shows an example of the comment generation screen developed for the CADD-based design review prototype.

Once a comment is created, that comment may be submitted to the DrChecks server. Figure 20 shows the screen that confirms the comment was properly submitted and imported as a new design review comment.

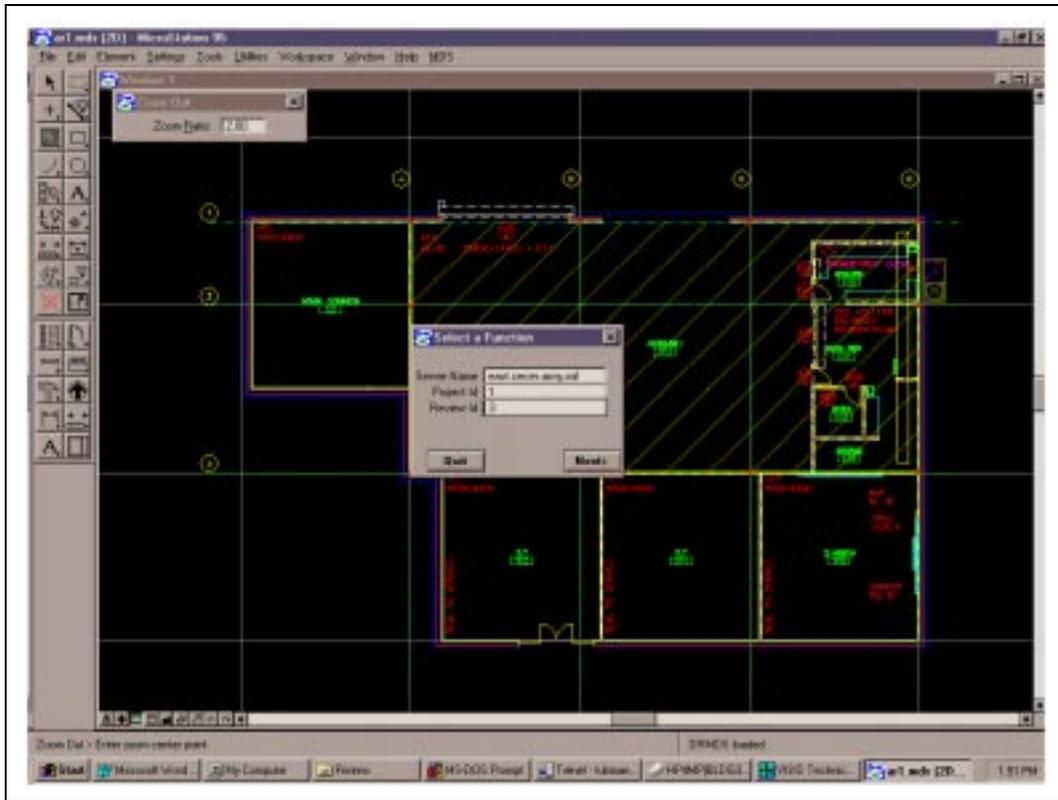


Figure 17. Server, project, and review identification.

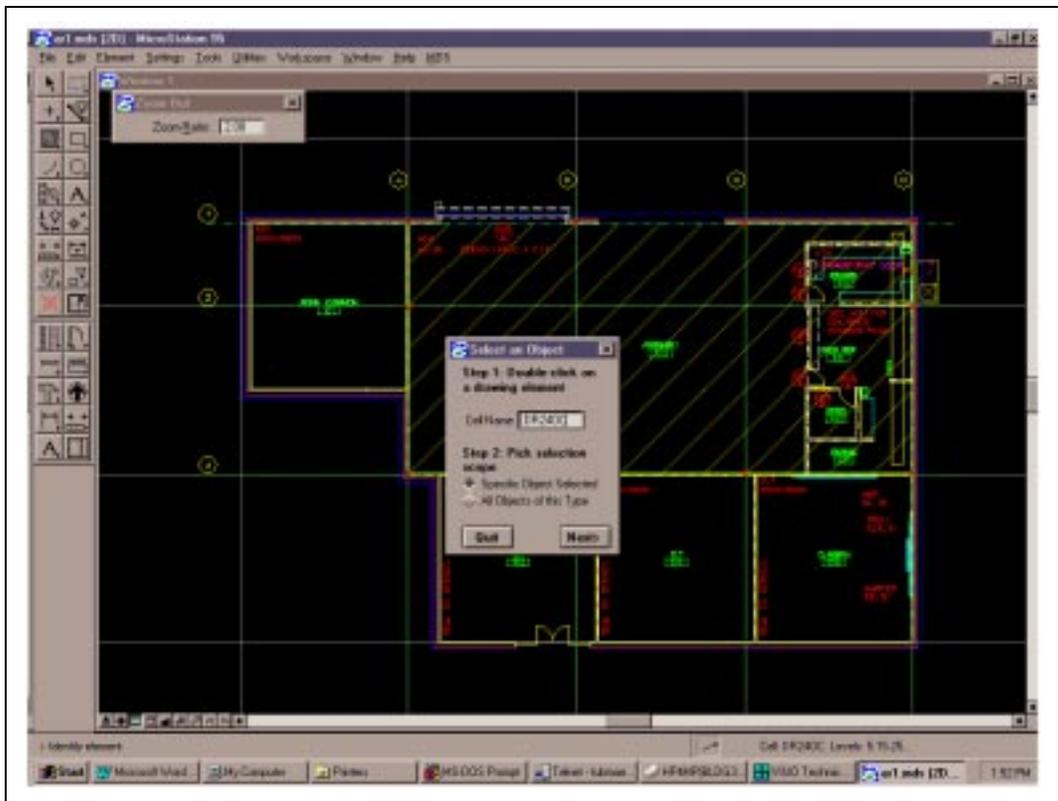


Figure 18. Select CADD object.

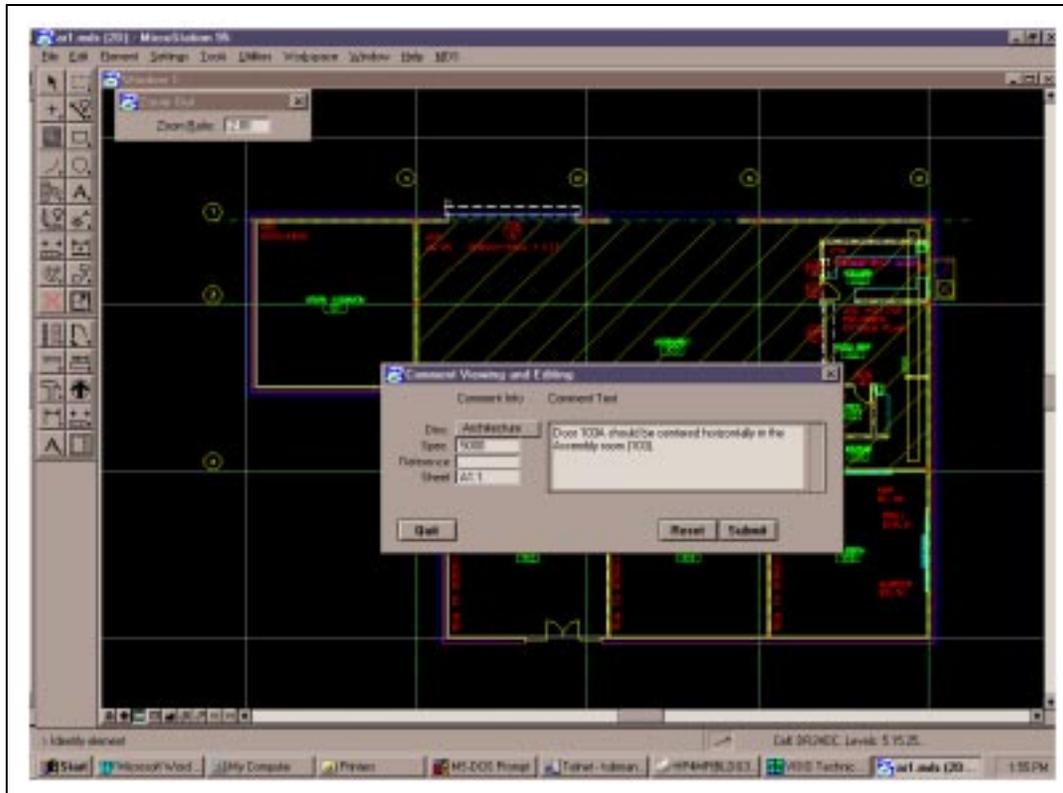


Figure 19. CADD-based design review comment.



Figure 20. Confirmation of comment transmission.

Prototype code for the CADD-based design review is available free of charge through the WWW. Select the CADD-based design review tool from the free software link from the Computing in Construction Committee Homepage, <http://www.cecer.army.mil/asce>.

During the development of the CADD-based design review tool, several additional features were tested but not included with the prototype code. These features included searching the server for relevant past comments and including “red-lined” portions of the drawings with the review comment text.

11 The Virtual Design Review

This chapter presents a new approach to collaborative design review called *virtual design review*, which allows a group of reviewers to evaluate a building design during a real-time interaction set within a three-dimensional representation of the building. The chapter begins with a discussion of several reasons why text-based design reviews are insufficient for real improvement in the design process.

Goals for Improving the Design Review Process

The requirements described in this report represent minimum criteria for systems that support the design review process. Additional criteria for improving the design review process include: (1) handling (e.g., viewing, retrieving, and storing) information, (2) facilitating interaction between the members of the review team and possibly the design and review teams, and (3) changing the role of the reviewer in the team.

Handling Information

The first issue, handling information (typically paper documents in the current design review process), involves improving how information is viewed, retrieved, and stored. This issue is closely related to the concept of helping manage a common information space for the team members (Schmidt and Bannon 1992).

The design plans and specifications given to the reviewer in the current design review process are paper-based, two-dimensional, and textual. Often the reviewer needs to cross-reference drawing sheets and the specification documents to visualize necessary information. In the virtual design review, reviewers will be able to index and view both three-dimensional and traditional plans of the building.

Retrieval of previous review knowledge to apply to the current project is often done by having paper deficiency or “lessons-learned” lists. Editing, storage, and retrieval of these paper deficiency lists is problematic. In the virtual design

review paradigm, reviewers may use A/E-created indexes based on standardized object definitions. This capability allows for more uniform, efficient, and comprehensive indexing. Using concepts developed in the RA and Lessons-Learned Generator (East et al. 1995), USACERL researchers plan to facilitate the generation, storage, and retrieval of review comments in this object-oriented environment.

Facilitating Interaction

Interactions between members of the review team should enhance the coherency of reviews, allowing for the coordination of comments and suggestions from the review team. Current methods require most communications to be in writing. This process severely limits the bandwidth of information transfer between the participants in the design review. Because of the time pressures involved and the sometimes limited writing skill of the authors, written commentaries are usually poorly formed. Interactions within the review team and between the design and review teams are severely hampered by this asynchronous process. Interaction between the design and review team would help the reviewers understand the rationale for design decisions, thus improving the relevance of review comments. Conversely, increased interaction would improve designer understanding of user requirements and operability issues. As interaction during a face to face meeting of the design and review teams is a rich and complex process, care must be taken to provide a flexible and open forum for these behaviors. Experiments with the research prototype of GROVE, a concurrent multi-user text editor, showed that an open, robust environment allowed its users to coordinate naturally using human social protocols left to the control of the participants rather than technological protocols enforced by the environment (e.g., turn-taking protocols and other floor control mechanisms). In the virtual design review, participants may confer verbally using the audio conferencing facility, visually through the three-dimensional representation of the building, and by sharing documents using the environment's whiteboarding facility.

Changing the Role of Reviewers

Reviewers are subject to a number of pressures in the design review process. First, design review is inherently a time-consuming process. Second, review is a resource-constrained process; the assignment of experts to "more critical" design and construction tasks create resource bottlenecks. Third, time constraints due to backlogs of unreviewed drawings and specifications may force reviewers to sacrifice the thoroughness of their reviews. Also, time constraints on project

funding cycles tend to constrain the ability of the A/E to change or fix design review items found. As a result, jobs may be bid with known mistakes. Experienced reviewers might look at a few critical design issues to assess the overall quality of the plan; if the design meets the reviewer's standards on these critical aspects, the reviewer checks the remainder of the project less thoroughly. This practice has been the subject of recent work by Fu (1995), which automates review assistance to the design reviewer by checking and commenting using encoded spatial and constructibility knowledge. The virtual design review system allows the addition of software agents that may act as associates, assistants, or tools to the reviewers.

The Virtual Design Review

The virtual design review is based on the client-server architecture of the WWW. Members of the design review team use software browsers that connect with a central information server containing the VRML building model to be reviewed. Within the virtual design review each client is represented by a character or avatar, which can move about the VRML building model and interact with other clients' avatars. The virtual design review also supports real-time multiparty voice conferencing between clients. The interactions with other clients and with the building model are presented through the client's software browser.

Besides the VRML building model, two other building models are contained in the server: a traditional two-dimensional plan view and the specification documents for the building. The VRML building model is generated from the plan and specification documents. Each of these models may be only partially complete. Additional databases of review comments are either available on the server itself or can be acquired on other information servers.

Using client software browsers, the reviewers connect to the server to perform their reviews. The browser allows the user to view the VRML building representation and the original plans and specification documents. Review comments made by individual reviewers are stored as HTML documents accessible directly from the VRML building model. By providing these visualization capabilities, the virtual design review improves how information is viewed, stored, and handled. The virtual design review supports interactions among the review team members and between the design and review teams by providing facilities for visual and audio interaction. Thus the virtual design review supports what Schmidt and Bannon (1992) call the *management of workflows* and the *management of a common information space*.

Management of workflows requires a balance between supporting formalized procedures required by the process and providing an effective, robust channel for communications between team members. The success of previous work on systems that provide an open environment for rich social interactions, like GROVE (Ellis 1991), motivates the multiparty audio conferencing facility in the virtual design review. The common information space in the virtual design review consists of the three building models and the databases of review comments accessible to the team members. This information space provides a three-dimensional model that helps designers and reviewers visualize and correct problems in the building design, and provides a formal procedure for adding new lessons learned to the review comment database.

Finally, the virtual design review provides an environment for software agents to critique the design according to design codes or automate the design process. These agents may reduce the amount of repetitive work that human reviewers now perform and change their role in the design review process.

A web-enabled design review paradigm, the virtual design review provides an environment that addresses the requirements and goals discussed earlier in this chapter. It supports the five requirements of web-enabled design reviews: restricted access, review comment generation, designer's response generation, back-check review generation, and project manager certification. Besides meeting these requirements, the virtual design review represents the attempt to improve the design review process along the three dimensions of handling information, facilitating interaction, and changing the role of reviewers. This paradigm should improve the efficiency, accuracy, and completeness of design reviews, resulting in an improvement in the quality of the constructed facility.

The virtual design review follows the client-server paradigm. The server (Figure 4) contains a number of databases, including a building database which contains the building drawings and specification documents. A three-dimensional VRML model is maintained of the building drawings and specifications. The server also contains a database of review comments in the form of HTML documents, semi-autonomous or autonomous agents, and audio communication facilities for the system users. Each system user accesses the server using a software browser (Figure 21), which includes a list of reviewers currently accessing the server, an HTML editor for review comment generation, a dialog window for control of audio and textual conferencing, the VRML model display window, and a two-dimensional building map that shows the locations of the reviewers within the building. Each client is equipped with three input devices (keyboard, mouse, microphone) and two output devices (monitor and speakers).



Figure 21. VRML system model.

Example of the Virtual Design Review System in Use

Suppose that five reviewers (Reviewers A through E) and three designers (Designers A through C) are using the virtual design review environment to perform a design review of a building project. Initially, Reviewers A and B and Designers A and B are connected to the server. Reviewer A and Designer A are architects, Reviewer B is a mechanical engineer, and Designer B is a structural engineer.

Upon entering the project being reviewed, Reviewer A notices that the bulk of the electrical wiring for the computers runs through a hole cut in the floor of the laboratory. However, the pipe chase is shown outside the wall and may result in operational problems. Reviewer A asks Designer A to look at the situation. Designer A, who is in another part of the building, uses the two-dimensional map to see that Reviewer A is in the computer laboratory and moves there using a “jump to location” button. Reviewer A points out the problem to Designer A,

who confirms that a problem exists. Reviewer A then uses the HTML editor to type a review comment, and links the comment to the pipe chase in the VRML model. Designer A receives a notification of the comment and is asked to correct the potential hazard.

Reviewer B uses the comment search facility to look up past lessons learned pertaining to fire sprinkler systems. After looking at the previous lessons, Reviewer B views the sprinkler layout of the building. A potential problem in covering all portions of an irregularly shaped room is identified. The mechanical engineer of the design team is not logged on, so Reviewer B uses the HTML editor to write the review comment related to an item, such as a pipe, in the drawing. A notification is sent to the appropriate design team member.

Reviewer C, who is an electrical engineer, logs on to the system. While viewing the wiring layout of the basement, she notices that an electrical distribution panel has insufficient clearance to allow access by building tenants. Reviewer C asks Reviewer B (mechanical engineer) and Designer B (structural engineer) to move to the basement for a quick conference. When the two arrive, Reviewer C uses her avatar's pointing tools to show the wall and ductwork that could be slightly moved to avoid the problem. Both Reviewer B and Designer B concur with Reviewer C, who then writes the appropriate review comment and links it to the building model. To make the situation clearer, Reviewer C also uses the graphics tool to draw a rough sketch of the desired situation, and includes it as a part of the review comment.

Expected Benefits

The method of addressing the five functional requirements (i.e., restricted access, reviewer comment generation, designer's response generation, back-check review generation, and project manager certification) is similar to the solutions developed for the web-enabled design review system discussed in the previous section. Access is restricted to known designers and reviewers for the project through the use of a log-in screen. To generate review comments, the reviewer uses the HTML editor to type in the comment text and to include illustrative graphics or other forms of multimedia. Designers logging into the system are presented with the set of review comments and are required to generate a response to the review comments. After the designer's response, each comment is back-checked by the reviewer, and, if the comment is approved, it may be added to the lessons learned review comment database. Project managers may also access the server to print lists of outstanding or protested

design review comments, and to make the appropriate certifications for disputed comments.

In addition to the five functional requirements, the three original goals in improving web-enabled design review systems are discussed in the context of the virtual design review.

Handling Information

The first goal for improving the design review process was to enhance the reviewer's ability to handle information: visualization, retrieval, and storage. In the virtual design review paradigm, these three issues are intertwined. While systems have addressed the latter two issues, the greatest contribution of web technologies is to provide better visualization resources for the reviewer. Enhanced information handling capabilities include: (1) using VRML to provide a three-dimensional walk-through of the building, while still providing representations of the traditional building plans and specifications, (2) supporting multiple perspective views of the building design, (3) allowing annotations to be made on both the two-dimensional and three-dimensional building representations, and (4) providing the reviewer with a suite of standard tools (e.g., a tape measure/ruler, a protractor/angle measure, etc.). Visualization of building information is enhanced through the three-dimensional building view linked to the two-dimensional plans and textual specifications. Retrieval of information is enhanced by: (1) providing visual HTML review comment links from within the three-dimensional model and (2) providing searchable and linkable indices to previous review comments from reviewers, reference materials, and the building plans and specifications. Storage of information is maintained by the review comment database.

Interactions Among Stakeholders

In the virtual design review paradigm, interaction between reviewers and between the design and review teams is supported through a combination of visual and verbal collaborative capabilities. To support visual interaction, all clients viewing the building model are represented as avatars. The three-dimensional view of the building is from a first-person perspective, as shown in Figure 10. The avatars can interact visually as well as verbally. Visual interaction should include avatar gesticulations and actions beyond simply updating the positions

of other avatars in the first person perspective. A richer vocabulary of avatar gesticulations and actions should include:

- The ability to “point” at features. If the reviewer wishes to draw the attention of other personnel to a particular feature of the building, then the ability to visually designate the object would be useful. For example, an avatar may be associated with a three-dimensional arrow that acts as a pointer, highlighting a feature or area of the building. This capability can be extended to include other gesticulation abilities if necessary.
- The ability to leave viewable tools or visual artifacts. Visual communication may also include the display of tools (e.g., measuring tape, protractor) and other visual artifacts to draw the attention of other personnel. The visual artifacts may also be linked to HTML documents for additional descriptions (as in the case of linking a visual cue to a textual review comment).
- The ability to share application documents between participants in conference (whiteboarding) allows for greater collaboration than that realized from verbal communication alone. Document sharing may also reduce the administrative burden of collecting and distributing the review comments. For example, reviewers that enter the virtual design review at different times will be able to find all of the distributed documents.

Verbal interaction in the virtual design review paradigm is handled through the multi-user voice conferencing capability. As was discovered in GROVE, a flexible and open verbal conversation architecture that does not constrain the rich social interactions among system users is more beneficial than a system which limits the types of interactions that can take place.

Changing the Role of Reviewers

Reviews are sometimes conducted under time and resource constraints, so the virtual design review architecture supports the addition of software agents that support reviewers for certain tasks. Agents may either exist on the server or may be linked to the server as clients. An example of an agent is the SEDAR* system, which helps reviewers perform constructibility reviews of flat and low-

* SEDAR = Support Environment for Design and Review.

slope roof systems. The system checks the existing roof design exhaustively for violations of published constructibility codes (Fu 1995). The reviewer selects a roof subsystem from a list and directs SEDAR to perform an exhaustive check of existing constructibility codes related to that subsystem on the existing roof design. SEDAR returns a sequence of textual/graphical critiques that describe violations according to the codes.

The number and type of agents using this building model is left open. Besides semi-autonomous agents like SEDAR, agents may range from quantitative evaluation tools (i.e., cost analysis or construction scheduling agents) to more autonomous agents (i.e., an agent that generates specification documents).

Technological Issues

Despite the variety of tools available for the WWW, several technological issues must be addressed for an implementation of the virtual design review. First and foremost is the issue of bandwidth on the Internet. Currently many of the components of the virtual design review are single applications. For example, multi-user virtual reality worlds are now commonplace, and certain types of interactions are supported (e.g., interactions with other avatars, IRC-style chat). Advances in data compression have improved the voice transmission quality of Internet telephony radically over the past few years.

A few critical technologies are not yet viable, including multiway audio conferencing and application document whiteboarding. Of the two, audio conferencing is the more difficult because of the demanding data flow requirements of conversation-quality voice transmission. The advent of intranets (Internet-like networks within companies) with greater data transmission bandwidth and shorter, deterministic routes or "hops" between the sender and recipient of data packets in companies, may help to alleviate this problem. Second, the division of computation between server and client requires that the client have access to a medium- to high-end PC. Existing recommendations for the IBM-PC-compatible family of computers include a medium to high-end Pentium® processor, at least 16 MB of memory, and a high-resolution color display. Additional issues include: (1) fidelity of the VRML building model, (2) responsiveness of navigation, and (3) security. Since the usage of the virtual design review is for reviewers to methodically check the dimensions, spatial arrangements, and equipment in the building, the three-dimensional model of the building must accurately reflect the building drawings (fidelity of the VRML model). Furthermore, the display of the building on each client's screen should accurately reflect the VRML building model (fidelity of the browser). Without

sufficient fidelity for the model representation and display, users would not be able to conduct thorough and accurate reviews of the building.

The responsiveness of navigation within the building model is also important; unresponsive navigation will quickly lead to disuse of the system. Navigation is a difficult issue to address because it depends on three factors: (1) the ability of the server to handle updates of the various clients, (2) the bandwidth of the connection between the server and the client, and (3) the processing power of the client's computer. The cost of building and maintaining these computer models (e.g., VRML) of the building should also be considered. Several commercial products exist which translate a two-dimensional CAD drawing into a full-fledged VRML representation. Finally, secure communication of information between the server and clients is necessary to ensure that the integrity of reviews is maintained. Recent advances in WWW transaction capabilities, including authentication certificates, have improved this aspect greatly.

A number of human-to-computer and cooperative work issues need additional clarification. One of these issues is that of overwhelming new users with audio and visual information. As the experience of the system user grows, so does his or her capacity for information processing within the system environment. Thus, web-enabled design review systems like the virtual design review environment should provide a hierarchical menu system that allows users to increase the amount of information presented by the browser as the user gains experience with the environment. Providing too much information initially to new users may cause confusion and slow the learning process.

12 Conclusion

The best environment for a design or BCO reviewer is one in which a variety of reference materials are readily available when needed by the reviewer. If reviewers are required to open file cabinets or refer to indexed material libraries, the quality of the reviews that can be conducted is decreased because the reviewers do not have time to fully utilize these important resources.

Virtual libraries, made possible through the WWW, have been shown by the DrChecks demonstration to be a viable alternative to paper reference libraries. The references developed for or linked with DrChecks include: a multi-media knowledge base for building systems, standard sets of CADD details, Corps of Engineers guide specifications, lessons learned, and past design review comments.

The reference felt to be the most useful by those who have tested DrChecks is the lessons learned. This catch-all type of reference source captures into one source important information about a variety of building systems, components, customer requirements, and local constraints. The ways in which these lessons should be indexed for retrieval is a significant contribution of this research.

A major constraint on a fielded system is that it have a very nominal first cost. The DrChecks system may be implemented using current personnel on existing hardware and usually about \$500 of additional software. Maintenance costs for the DrChecks system depend on the amount of customization needed to map the system into existing business processes. Because the DrChecks system development tools are easily used, these changes take very little time compared with tradition client/server programming projects.

While the future technologies used to support design and BCO reviews, such as the VRML system described in this report, appear promising, any such future development should aim to reduce or eliminate text-only review comments. The best approaches should facilitate some type of virtual conferencing that will support the full bandwidth of information needed between reviewers and designers.

DrChecks is envisioned as a new type of nonmandatory, but potentially Corps-wide system. It will be a locally customized, fully distributed system for which the offices responsible for generating and keeping the information also have the capability to operate and maintain the system. Since the communication tools provided through the Internet and WWW have alleviated the most difficult problems associated with distributed computing, it is only a matter of time before these simple and inexpensive tools are used by local offices to modify prototypes such as DrChecks for their own office use.

References

Cited

- Boehm, Barry W., "A Spiral Model of Software Development and Enhancement," *Computer*, Vol. 21, No. 5 (1988), pp 61-72.
- Boutell, T., Boutell.Com, Inc., World Wide Web Frequently Asked Questions (FAQs) (1996), <http://www.boutell.com/faq>.
- East, E. William, Timothy L. Roessler, Mark D. Lustig, and Michael Chin-Ming Fu, "The Reviewer's Assistant System: System Design Analysis and Description," Technical Report FF-95/09/ADA294605 (U.S. Army Construction Engineering Research Laboratories [USACERL], 1995).
- East, E. William, Timothy L. Roessler, and Mark D. Lustig, "Improving the Design Review Process: The Reviewer's Assistant System," American Society of Civil Engineers, *Journal of Computing in Civil Engineering*, Vol 9, No. 4 (1994).
- Ellis, C., "Groupware: Some Issues and Experiences," *Communications of the ACM*, Vol. 34, No. 1 (1991), pp 38-58.
- Fu, Michael, "Using a task-based model of design in an expert critiquing system," Master's thesis, University of Illinois at Urbana-Champaign, 1995.
- Lutz, James D., Donn E. Hancher, E. William East, "Framework for Design-Quality Review Database System," American Society of Civil Engineers, *Journal of Management in Engineering*, Vol 6, No. 3 (1990).
- Military Standard (MIL-STD) 498, *Software Development and Documentation* (Department of Defense, 5 December 1994).
- Multicast Backbone, Frequently Asked Questions (FAQs) on the Multicast Backbone (MBONE), <http://www.mediadesign.co.at/newmedia/more/mbone-faq.html>, 1994.
- NASA (1997) <http://envnet.gsfc.nasa.gov/ll/definition.html>.
- National Center for Supercomputing (NCSA) Applications, A Beginner's Guide to HTML (1996a) <http://www.ncsa.uiuc.edu/General/Internet/WWW/HTMLPrimer.html>.
- NCSA, The Common Gateway Interface, (1996b), <http://hoohoo.ncsa.uiuc.edu/cgi>
- Ohio State University, IRC Frequently Asked Questions (FAQ), 1996, <http://www.cis.ohio-state.edu/hypertext/faq/usenet/irc-faq/faq.html>.

- Rumbaugh, James, Michael Blaha, William Premerlaini, Fredrick Eddy, and William Lorenson (1991) *Object-Oriented Modeling and Design*, Prentice Hall, New Jersey.
- Schmidt, K., and L. Bannon, "Taking CSCW Seriously," *Computer Supported Cooperative Work*, Vol. 1 (1992), pp 7-40.
- San Diego Supercomputer Center Inc. (SDSC), The VRML Repository, 1996, <http://www.sdsc.edu/vrml>.
- SGI, The Virtual Reality Modeling Language Specification Version 2.0 (1996), Draft #3, ISO/IEC 14772, <http://vrml.sgi.com/moving-worlds>.
- Sun, *The Java Platform*, Sun Microsystems Inc. (1996), <http://www.java.sun.com/java.sun.com/aboutJava/index.html>.
- Wide Area Information Server (WAIS) FAQ, 1994, <http://sunwww.informatik.uni-tuebingen.de:8080/Archive/WAISFAQ.html>.

Uncited

- Brauer, Roger L., and Martin Koch, *A Method for Users to Review Facility Concept Designs*, USACERL Technical Report, P-117/ADA112870 (1981).
- Department of Energy (DOE) SAD-TMP-23-94, *Lessons Learned Technical Standard* (DOE, 1997).
- DrCheck's Homepage, USACERL, 1996, <http://www.cecer.army.mil/pl/ra/drchecks>.
- East, E. William, Michael Chin-Ming Fu, "Abstracting Lessons-Learned from the Design Review Process," *Journal of Computing in Civil Engineering* (American Society of Civil Engineers, 1996).
- East, E. William, "A Design Review Lessons Learned Demonstration," U.S. Army Corps of Engineers, Proceedings of the 1996 Computer Aided Design and Drafting (CADD) Conference (1996) (<http://www.cecer.army.mil/pl/ra/committee/papers/p-lessons.htm>).
- Kirby, Jeffrey G., Robert P. Cannalate, Donald K. Hicks, and Edward J. Japel, *Constructibility and Design Reviews: Analysis and Recommendations for Improvement*, Technical Report P-89-15/ADA212602 (USACERL, 1989).
- Kirby, Jeffrey G., Donald K. Hicks, Douglas A. Furry, and Jeffrey A. Koenke, *Automated Review Management System*, ADP Report P-87/08 (USACERL, 1988).
- Phillips, J. Leo, "Enhancing Engineering Project Designs by Linking Lessons Learned," Masters Thesis (University of Florida, 1996) (<http://www.sam.usace.army.mil/sam/cd/masrep.html>).
- Sears, The Internet Telephony Page (1996), <http://rpcp.mit.edu/~asears/main.html>.
- Suchman, L., *Plans and Situated Actions: The Problem of Human-Machine Communication* (Cambridge University Press, New York, NY, 1987).

U.S. Army Corps of Engineers, *Engineering and Design: Quality Management*, Engineering Regulation 1110-1-12 (Headquarters, U.S. Army Corps of Engineers [HQUSACE], 1993).

U.S. Army Corps of Engineers, *Architect/Engineer User Manual*, Sacramento District (CESPK)-PAM-1-2 (CESPK, 1992).

U.S. Army Corps of Engineers, *Review Manager User Manual*, CESPK-PAM-1-4 (CESPK, 1992).

U.S. Army Corps of Engineers, *Reviewer User Manual*, CESPK-PAM-1-1 (CESPK, 1992).

U.S. Army Corps of Engineers, *Technical Manager User Manual*, CESPK-PAM-1-3 (CESPK, 1992).

Distribution

Chief of Engineers

ATTN: CEHEC-IM-LH (2)
ATTN: CEHEC-IM-LP (2)
ATTN: CECC-R
ATTN: CEMP-C
ATTN: CEMP-CE (2)
ATTN: CEMP-E
ATTN: CEMP-ES (2)
ATTN: CERD-L

US Army Engr District

ATTN: Library (42)
ATTN: Civil Engineer (42)

US Army Engr Division

ATTN: Library (8)
ATTN: Civil Engineer (8)
ATTN: Civil Construction/Civil Con-Ops (8)

US Army Transatlantic Program Center

ATTN: TAC 22604
ATTN: TAE 09096

US Army Engineering and Support Center

ATTN: CEHND 35807-4301

Defense Tech Info Center 22060-6218

ATTN: DTIC-O (2)

125
+12
01/98