

Multimedia Feedback Systems for Engineering

Michael J. McDonald, Eric J. Gottlieb,
Scott Gladwell, and Cara L. Slutter
Sandia National Laboratories
Intelligent Systems and Robotics Center
PO Box 5800-1004
Albuquerque, New Mexico 87185-1004
Email: mack@sandia.gov
Telephone: 1-505-845-9852
Albuquerque, NM 87185-1004

RECEIVED
JAN 21 1999
OSTI

Abstract

The World Wide Web has become a key tool for information sharing. Engineers and scientists are finding that the web is especially suited to publishing the graphical, multi-layered information that is typical of their work. Web pages are easier to distribute than hardcopy. Web movies have become more accessible, in many offices, than videos. Good VRML viewing software, bundled with most new PCs, has sufficient power to support many engineering needs.

In addition to publishing information, science and engineering has an important tradition of peer and customer review. Reports, drawings and graphs are typically printed, distributed, reviewed, marked up, and returned to the author. Adding review comments to paper is easy. When, however, the information is in electronic form, this ease of review goes away. It's hard to write on videos. It's even harder to write comments on animated 3D models. These feedback limitations reduce the value of the information overall.

Fortunately, the web can also be a useful tool for collecting peer and customer review information. When properly formed, web reports, movies, and 3D animations can be readily linked to review notes. This paper describes three multimedia feedback systems that Sandia National Laboratories has developed to tap that potential. Each system allows people to make context-sensitive comments about specific web content and electronically ties the comments back to the web content being referenced. The first system ties comments to specific web pages, the second system ties the comments to specific frames of digital movies, and the third ties the comments to specific times and viewpoints within 3D animations. In addition to the technologies, this paper describes how they are being used to support intelligent machine systems design at Sandia.

1 Introduction

The purpose of Computer Aided Drafting and Engineering (CAD and CAE) is to help engineers develop and refine design concepts and generate information that others can use to produce the designs or further the concepts. CAE tools are typically designed for traditional, sequentially executed design projects. Here, groups of engineers use integrated sets of tools to develop or test ideas, and then pass results on to other engineers with different tools for successive tasks. Compressing these design efforts requires creating highly collaborative multidisciplinary teams. Here CAE tools need to allow people to develop ideas and have their work reviewed while it is still in process. This means that the resultant information must be readily accessible at various midpoints of completion.

Traditionally, engineers who wish to review models must use the software used to create the data. Unfortunately, as the Computer Aided Design Report [CAD-98] points out: "CAD and engineering software is too complex and too costly to put on every worker's desk just to view design data." This is especially true for multidisciplinary teams, where many tools are used to solve a problem. As a result, several commercial software packages have been developed to allow people to view various forms of CAE data. These basically fall under the categories of general viewing software and proprietary or data-specific codes. For example, Adobe Exchange works like a printer driver and viewer that lets people generate and mark-up electronic or virtual paper copies of most program outputs. Proprietary, or data-specific viewers, that work with CAD tools allow users direct access

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, make any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

to CAD files. Here, users can, for example, inspect internal dimensions within models. While less capable, general viewing software is generally easier to use and less expensive to support than the proprietary analogs.

While 2D CAD viewers are well advanced, 3D model viewers are just emerging for CAD systems. As CAD Review notes, however:

After reviewing the 3D viewing software of a number of leading companies, we're forced to conclude that this software is not ready for mass distribution.... We found the user interfaces to be awkward and the on-line help to be baffling. ... The measuring tools leave much to be desired. Most don't handle assemblies well. Most firms will not yet find 3D-model distribution to be a substitute for drawings. [CAD-98]

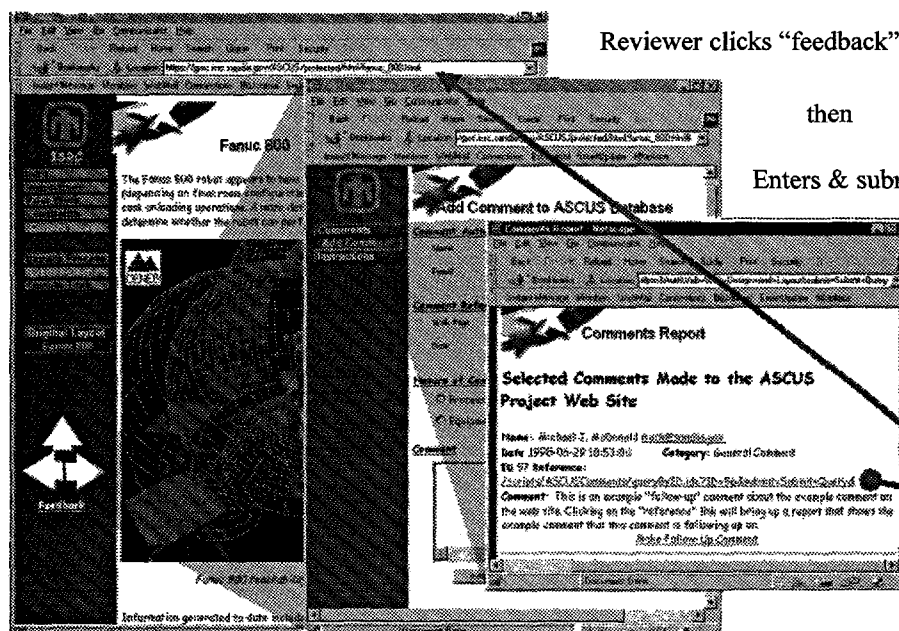
The situation is worse for viewing 3D simulation results. Very specialized software is typically required for viewing 3D simulations developed for 3D-motion analysis. As a result, most 3D-simulation users generate movies (i.e., videos or digital movies) of the simulation results. While valuable, movies lack too much of the original information and are, in addition, difficult to distribute. Beyond the obvious loss of 3D-scene navigation, NTSC video has, for example, the same resolution as the smallest available computer monitors. For practical reasons, digital movies are generated at even lower resolutions than the video versions.

Virtual Reality Markup Language [VRML-97] translators just becoming available for distributing 3D models via the

Internet and viewing them in standard web browsers. VRML (often pronounced 'vermal') is the file format standard for 3D multimedia and shared virtual worlds on the Internet. VRML provides structured graphics and extra dimensions (depth and time) to the online experience. VRML files contain geometric, animation, and program features. VRML files are played in VRML browsers that are typically run within web browsers. Developers have begun using VRML-97 animation features for generating 3D animations of simulation results. Qiming Wang [WANG-98], for example, has developed prototypical software for translating IGRIP [Deneb-98] simulations into VRML animations. These developments are very promising.

Finally, engineered systems are not only about CAD and CAE models. Typically, they include descriptive text, tabular data, pictures, sketches, and process diagrams. This combined information needs to be collected in context and reviewed as collections, and not as individual elements. Currently, the HTML and PDF formats are among the few electronic alternatives for producing this information as collections. Here, HTML has the advantage of providing more web-efficient (smaller files, less data transfer) ways for presenting the collections of movies, images, and other content, while the PDF format has the advantage of presenting the material in easy-to-print representations. Fortunately, the openness of both PDF and HTML supports easy development of hybrid complex documents.

The needs, therefore, are for tools and file formats that allow broad cross sections of engineers to share and review information electronically. The remainder of this document



Reviewer clicks "feedback" button

then

Enters & submits comment via web

Others review comments through custom reports and follow-up via the web

All review comments are linked to source pages

Figure 1: Order of pages user sees when making a comment via the web.

describes three technologies that we developed to satisfy these needs. Section 2 describes an electronic feedback system that interoperates with regular web pages to allow reviewers to comment on web page content. Sections 3 and 4 describe a feedback tool that allows engineers to add annotated bookmarks to normal digital movies and animated 3D VRML models. Sections 5 and 6 discuss applications where these tools have been tested and provide closing remarks about the technologies.

2 Web Page Commenting

Sandia's web page commenting feedback system first appears to reviewers as a "feedback" link on each important web page of a report (see Figure 1). This link has a similar function as typical "mailto" links. Here, however, clicking on this link brings the user to a highly automated "make a comment" web form where new comments can be added or other people's comments can be reviewed. In addition to making comments, users can view other's comments through a linked comment query page. Because the comments are linked to the reference pages, the typically short ambiguous phrases used in hardcopy review (i.e., with a pen) remain understandable in this electronic form.

Figure 2 shows the basic process of submitting and viewing comments. When opened from a Source Page, basic information is automatically filled into the comment form on the Make Comment Page. This information includes the URL of the page the reviewer was viewing when they clicked the feedback button, the date, and the reviewer's name and email address. The reviewer then writes and categorizes the comment, makes any necessary changes (e.g., to add or change their name or email address), and submits the information to the web server.

Later, users submit search terms through a web form on the Report Query Page to generate custom reports of reviewer comments (Report Pages). These reports display the comments and provide hyperlinks to the Source Page to which the comment refers. Additional links let the reviewer respond directly to any prior comment or send email to the reviewer. Comments about comments are entered as above. When reviewed, these comments provide hyperlinks that generate custom reports that show the originating comment (i.e., links to generate pseudo Source Pages), rather than the Source Page being commented on.

Specialized JavaScript code is used on the Source and Make Comment Pages of the commenting system to automatically fill in commenting data. Report source pages access the Make Comment page by posting their internet location within the URL. A JavaScript call is used on the Source Page

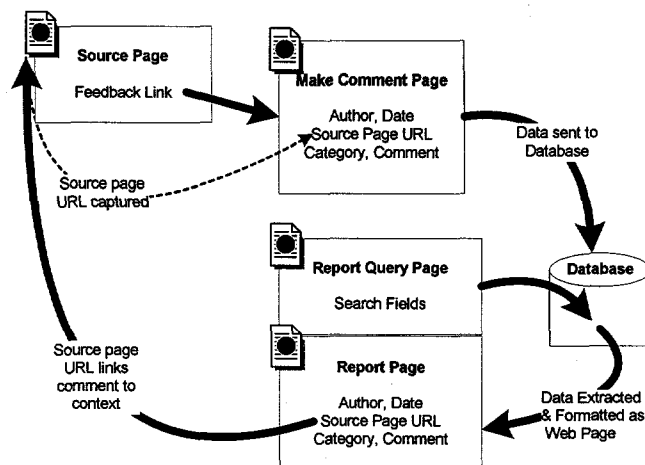


Figure 2: Basic elements of commenting system

construct this link. Related parsing JavaScript routines are used in the Make Comment Page to place this link information into form boxes. Additional cookie-based scripts are used to recall repeated information such as the user's name and email address as well as to fill in helpful date stamp information. Standard web database interactions, including HTML form submission, SQL database access, and special automation functions provided with the web server are used to transfer and store these comments. Similar mechanisms allow users to quickly search for comments made by specific individuals or according to specific categories.

Typical Web database interaction techniques are used to implement the Review Comments page. Reports contain authorship data and hotlinks that allow users to quickly look at the web pages that the comment referred to, email comment authors, or post a follow-up comment. Links for follow-up comments are formed as searches that generate custom reports containing the source comment. Clicking on the reference comment thus generates a custom report showing the originating comment. This linkage, thereby, provides a valuable mechanism for "closing" or documenting responses to open review items.

3 Digital Movie Commenting

Sandia's digital movie commenting system appears as a movie player program with VCR-like movie control buttons, a text input area, a scrolling text display area (see Figure 3) and associated buttons. Clicking on the VCR-like control buttons causes the movie to play, stop, or index as would be expected from any movie player program. Unlike a movie,

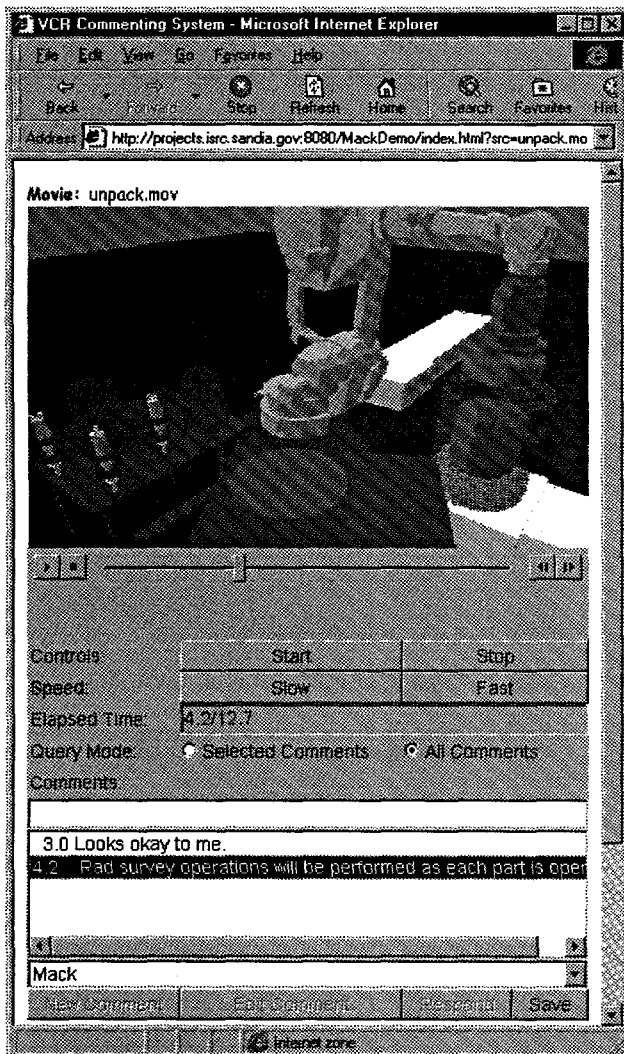


Figure 3: Sandia's Digital Movie Commenting tool.

clicking on text (usually in the form of comments) causes the movie to advance to a special place within the movie and adding text while the movie is paused causes the text to be added (along with identifying information) to the list of other comments. Finally, clicking a "Save" button saves the comments for others to review.

Like the page-by-page commenting system, the people who generally use this tool do so to provide or review detailed information about individual movies (typically made as the result of short simulations). Because the commenting system uses visual real estate, the tool is frequently used in a "stand-alone" manner, apart from broader content pages. That is, the user clicks on a provide/review comments about movie link and is then brought to the movie commenting page.

Within the commenting system application, the commenting area is active whenever the digital movie stops. A name text

box allows users to use simple monikers to identify themselves and a commenting text box allows users to enter comments. When the user presses a return from within the comment box, the name and comment text "move" into a scrollable list of comments. The user can then make additional comments or upload these comments to a central database. Later when any reviewer selects the comment, the movie advances to the place where the comment was made.

The digital movie commenting system is built on several technologies (see Figure 4). Movies are created in any of several formats and stored on a web server. A web page (HTML) defines the page's layout and loads the Java commenting applet. The commenting applet loads and plays the movie, provides user controls, and interacts with the remote database. A Java server program interacts with the commenting applet and retrieves and stores comment data on a network server.

The applet's movie playing functions are implemented Java Media Framework API [JMF-98]. JMF specifies a simple unified architecture, messaging protocol, and programming interface for media players, media capture, and conferencing. The commenting system relies on JMF's ability to play digital movies at the same efficiency as native viewers (i.e., the QuickTime plug-in) while providing hooks that allow the application to seek to a new position in the video file and initiate playback. This allows the applet to capture, store, and reuse the time that the movie is stopped and a comment is made.

The Java applet and server use the Remote Method Invocation [RMI-97] interface to connect the player applet to a remote database. An RMI-based server is used to communicate with the movie commenting system client software and the Java ODBC package is used to let the server interact with an SQL-based database. When the applet is started, comments are loaded across this interface. Clicking a "Save Comments" button transfers new comments to the database.

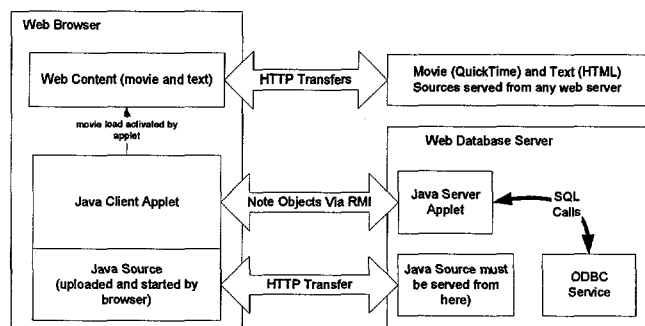


Figure 4: Architecture of Sandia's Digital Movie and VRML Animation Commenting Systems.

4 3D Animation Commenting

Sandia's 3D animation commenting system, shown in Figure 5, appears and works much like the digital movie commenting system. Here, however, the movie is actually 3D graphical animation that can be viewed from any angle or position. Like the digital movie commenting system, users can stop the animation at any point in time, add comments, and have the comments stored in a central repository. Unlike with digital movies, clicking on the comments advances the viewpoint along with the time point in the animation. In addition, the controls and comment editing areas appear in a separate Java applet window from the graphical animation.

The 3D animation commenting system is built on several technologies including:

1. VRML, to define the animation geometry and motions.
2. Cosmo 3D [COSMO-98], a VRML browser with an External Authoring Interface [EIA].
3. A custom Java applet to control and monitor the VRML scene, store comments, and interact with a remote database.
4. The RMI-ODBC-SQL server used in the digital movie commenting system.

Sandia's animations are automatically translated from simulation or other CAE software tools. Sandia has developed translators that produce animations from Deneb Inc.'s Envision [Deneb-98] and from Sandia's Umbra (a simulator developed to support visualization for telerobots and vehicles as well as simulation of robot controllers, industrial machines, and autonomous vehicle swarms). In addition, Sandia has begun development of a translator for its Archimedes assembly analysis code. Sandia's translators work in-core. That is, they compute the translation while the simulation is running rather than translating files that the simulators generate. As a result, the translators are simple and efficient to use.

As with similar simulators, Envision and Umbra use 3D models for both simulation and visualization. The simulators compute the positions and possibly shapes of various geometric elements on a continually increasing time base. (While the simulations only compute the position of elements at specific time steps, the underlying assumption of the simulations is that the real devices operate on a continuous time base.) Due to the geometric nature of the environments being simulated, a significant portion of the simulation results can be conveyed as 3D graphical animations.

Sandia's translators produce VRML files that use general modeling and key-frame animation approaches that are similar to [WANG-98]. Models are constructed with kinematic

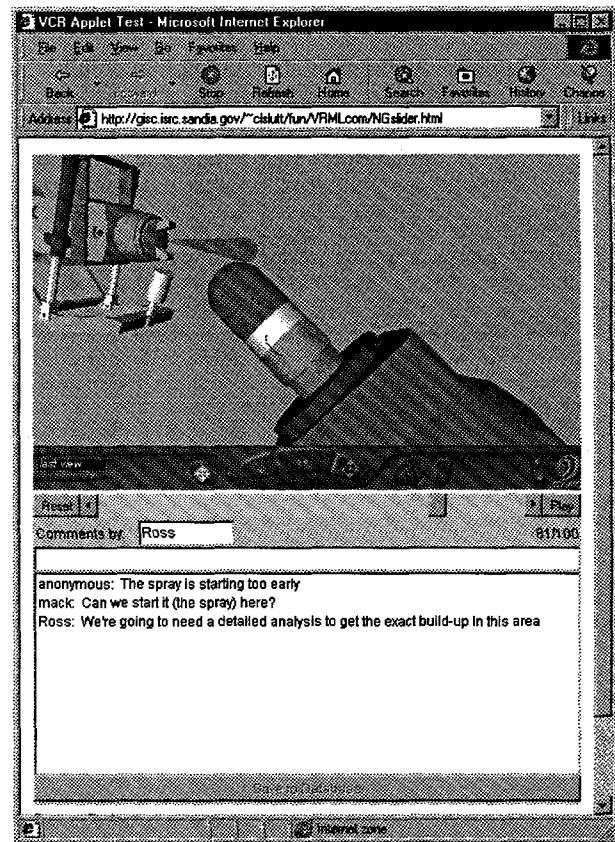


Figure 5: VRML Commenting System User Interface

structures that correspond to the simulated environment. As the simulation progresses, device and joint position and orientation changes are stored as key frame values in position, orientation, and visual scalars. Sandia's VRML files can be played in both stand-alone mode (where the animation plays at a fixed rate whenever the user clicks on any scene element), with VCR-like Java Applets, or with the animation commenting system.

Platinum's Cosmo 3D player, with its EIA interface, is required for Sandia's commenting system. The EIA interface allows browser data to be accessed from within Java applets. In building EIA applications, the browser data structure is accessed and used to retrieve various VRML nodes. From this, the nodes are accessed to retrieve various event interfaces within the nodes. These events are then exercised to store or extract data within the VRML scene. For example, playback speed and time within the animation is controlled or stopped by accessing the animation clock's Start Time and Cycle Interval and adjusting its Current Time and the Fraction of the movie that has played. Other features have been implemented to allow the Java applet to run the animation in reverse.

VCR-like recording functions are implemented by controlling the simulation through the EIA-exposed control nodes.

For example,

- A "Play" button action queries VRML's current time and then sets a time sensor's Start Time to be greater than the current time. If the animation has been partly played, this start time is set so that it will restart the animation with any stored Current Fraction values.
- A "Stop" button queries the movie, saves the fraction of the movie played in a local variable and then stops the animation.
- A slider widget lets users step the movie slowly or drag the time to any point within the movie. Each interaction stops and then adjusts the reference time that the animation uses.

In VRML (as with most 3D animation systems), scenes are rendered with respect to the theoretical position of an eye placed in the world at a specific orientation. These are called viewpoints. In the commenting tool, A ViewPort and a View object are used to record viewpoints and later set the view position to earlier recorded viewpoints. Additional functionality supports the ability to query the current viewpoint's position and orientation and set the viewpoint to past saved viewing positions.

The commenting area is active whenever the animation has stopped. A name text box allows users to use simple monikers to identify themselves and a commenting text box allows users to enter comments. When the user presses a return from within the comment box, the name and comment text "move" into a scrollable list of comments. Within the Java code, animation time and viewpoint data are stored with each comment. When the comment is "selected" by the user, the application uses this data to adjust the animation time and viewpoint to correspond to the stored values.

When the applet is loaded, a database is queried via an RMI server, to populate the scrollable comment box with previously saved comments. Later, reviewers (users) are allowed to upload added comments to the remote database. This uploading function is provided through the same RMI interface and with the same server code described in Section 3.

5 Testing

The technologies described here have both been tested and validated functionally and tested by expert user groups for appropriateness. In addition, Sandia has begun to use the tools to support engineering projects.

The web page feedback system was first tested on a medium-sized engineering project that developed and analyzed

two alternative robotic system designs for a uranium hexa-fluoride shipping container decontamination. Here, web pages were used to present information and review information was collected through the web feedback system (described in Section 2). In addition, VRML models were used to animate various solution concepts. In this deployment, engineers in the field provided review information that was then used by an on-site engineer to correct and update the published information. In addition, the need to have follow-up links connected to review comments (rather than referencing web page) was discovered and implemented.

Additional testing of the revised web page feedback system was performed on a variety of projects. Currently unresolved issues include better management and easier separation of comments between projects that utilize the same database. Here, it has been found that it is beneficial to use one database to support several projects. Additional coding needs to be implemented to separate comments between projects.

The digital movie and VRML animation commenting tools have been operationally tested and tested in controlled environments by a small set of potential users. Deployment has been slowed as these two systems rely on new software tools that have not yet been widely deployed. Additional problems include

- Firewall blocking of Java: Many companies, including Sandia, use software to block some of their employee's access to Java applets.
- Java applet incompatibility between browsers: A complete set of support Java applet tools that would execute within both Netscape and Internet Explorer could not be found. A single browser was selected.
- Uncertainty of Cosmo VRML browser: In spring 1998, SGI (who developed Cosmo) formed a company, Cosmo Software, to develop and sell the Cosmo browser. By summer 1998, SGI dissolved the company.

It is hoped that recent trends and events will alleviate these problems. Blocking policies appear to be relaxing. Corporations are finding that blocked users are turning to less secure alternatives like Active X and downloading executable files. Compatibility issues and Microsoft's Java-related development practices are being investigated by the US Department of Justice. In the fall of 1998, Platinum Technology purchased the Cosmo libraries and announced plans to merge the Cosmo product with its existing VRML browser. Thus, market maturity appears to be the current limiting factor. It is expected that usage will increase as these deployment issues are resolved.

6 Conclusions

Engineering projects generate heterogeneous collections of data that must be reviewed by a diverse set of customers and peers. Due to the complexity of the information generated and shared, it is becoming necessary to electronically present this data, rather than rely on traditional (i.e., paper) forms. Sufficient electronic presentation technologies are only now becoming sufficiently advanced to support individual presentation needs. In addition, web-based technologies have potential to provide the glue to bind a large diversity of engineering information into unified presentations.

The need that electronic presentation addresses begs solving the new problem of electronic review. To-wit, data that is too complex to present in paper form is likely too complex to review in paper form. This paper presented three prototype technologies for enabling the needed review. The first system ties comments to specific web pages, the second system ties the comments to specific frames of digital movies, and the third ties the comments to specific times and view-points within 3D animations. Each allows people to make context-sensitive comments about specific web content and electronically ties the comments back to the web content being referenced. All strongly rely on standard web software tools and development approaches. Initial tests indicate that using these technologies will improve quality and reduce costs in robotic system design and development efforts.

Acknowledgements

Sandia's Laboratory-Directed Research and Development Program funded this work. Sandia National Laboratories is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000.

References

[CAD-98] Computer Aided Design Report, Vol. 18, No. 4, April 1998,

[COSMO-98] Cosmo 3D was developed by Silicon Graphics, licensed to Cosmo Software Inc., and, sold to Platinum Technology. Cosmo 3D is available at .

[Deneb-98] IGRIP, published by Deneb, is a simulation tool for design, evaluation, and off-line programming of robotic workcells. Envision is an Open architecture version of the IGRIP product. Deneb Robotics Inc. (<http://www.deneb.com>), is a subsidiary of Dassault Systèmes.

[JMF-98] The JMF 1.0 API was developed by Sun Microsystems, Inc., Silicon Graphics Inc., and Intel Corporation. Sandia's commenting system uses Intel's version of the API. More information is available at <http://java.sun.com/products/java-media/jmf>.

[RMI-97] Java Remote Method Invocation Specification, Copyright 1997, Sun Microsystems (available at <ftp://ftp.javasoft.com/docs/jdk1.2/rmi-spec-JDK1.2.pdf>)

[VRML-97] VRML Consortium (Web 3D Consortium) sources. Available at <http://www.vrml.org>.

[WANG-98] Qiming Wang, Translation of Deneb Robotics formats to VRML
<http://www.nist.gov/itl/div894/ovrt/projects/mfg/SIMA/deneb2vrml/deneb2vrml.html>