Supporting Manufacturing Process Analysis and Trouble Shooting with ACTS

Michelle Potts Steves, Wo Chang*, and Amy Knutilla
National Institute of Standards and Technology
msteves@nist.gov, wchang@nist.gov, knutilla@nist.gov

*Active Member of the W3C SYMM [1] Working Group

Abstract

This paper presents a National Institute of Standards and Technology (NIST) prototype tool called ACTS (Annotation Collaboration Tool via SMIL – Synchronized Multimedia Integrated Language), which was designed to meet the user requirements of a distributed, collaborative team performing remote analysis of robotic welds. Based on open standards such as W3C (World Wide Web Consortium) SMIL, ACTS provides a framework of temporal and spatial synchronization for multimedia data streams. An annotation facility provides for asynchronous collaboration among colleagues. ACTS uses the NIST-developed S2M2 [2] Java applet-based SMIL player that implements and extends the SMIL 1.0 specification. Future work is also discussed.

Keywords: CSCW, groupware, annotation, SMIL

Introduction

The increasing globalization of manufacturing demands that members of an expanded enterprise (e.g., design, manufacturing, distribution, management) exchange information in a concurrent manner and collaborate throughout the product development life cycle. This creates an increasing dependence on information technology in order to share disparate data among geographically-dispersed staff. Globalization trends and recent advances in information technology provide an opportunity now for Computer Supported Cooperative Work (CSCW) [3] systems, particularly those that are web-enabled. There are many opportunities for CSCW in manufacturing to enable, for example, decision-making, product and process design, and research. Imagine this scenario: an advanced integrated CSCW technology is used to enable the efficient trouble shooting of a manufacturing process problem by one of the few experts available in a highly specialized field:

Jade is a welding engineer that works for Cars-R-Us. While reading through her e-mail, she notices that her icon for the collaborative welding trouble shooting system starts flashing yellow. She opens up the icon, and a popup window informs her that there is a problem in the chassis welding line. Clicking a button opens a window for a collaboration space specifically focussed on the chassis problem area.

One of the main tools at Jade's disposal is a web-based, dynamically updated data table that lists all the welds and associated weld quality (green, yellow, red) that have been performed on that day for each welding workstation Jade oversees. She sees a long list of good welds with two interspersed that have fallen into the yellow range. (Having a second one fall into the yellow range is what generated the alert on her workstation.) Clicking on the data for the first yellow weld, she sees a Virtual Reality Modeling Language (VRML) current-voltage graph plotted over the geometry of the weld. Clicking on a button on that plot causes it to be overlaid with a transparent template for a good weld, with tolerance ranges indicated. She can clearly see that the weld is going bad near the end of the run, and she begins to suspect a heat warping problem.

She goes to the more recent yellow weld and brings up the VRML plot and overlays it with the good template. Surprisingly, this weld goes bad at the beginning, then corrects itself. Jade clicks on the "replay" button, which brings up a synchronized multimedia playback tool for selected data streams for this weld. Jade uses the configuration that allows her to view a sweep moving through a representation of the weld quality graphed over the geometry data, while the audio and video of the weld replay synchronized to the movement of the sweep across the graph. At any time during the replay, Jade can stop or pause the replay to make an annotation about the conditions of the weld or her analysis of the problem. Jade begins to associate some spikes in the graph with some sounds and data she reviewed in the playbacks and begins to form a theory.

She re-examines the weld data table and clicks on a thumbnail image to bring up a snapshot of the finished product. She notices some telltale signs in the weld itself that confirm her theory. She calls up Harry, the job setter for this welding cell, and asks him to use his PC to join her in the collaborative analysis system. She points out the features she sees in the data, and works with Harry on what specifically might be causing the problem and how to fix it. It looks like something might be going wrong with the power supply of the work cell.

They decide to call in an electrician to look at it. Jade suggests that Harry show the electrician the current-voltage graph when he arrives to help explain the problem they suspect with the power supply. As they finish up their call, Jade notices a flashing red signal in her welding collaboration system. She quickly moves on to the problem that has stopped a welding line over in the body assembly area.

The basic technology components exist for enabling this and other types of collaboration that will be common in future global manufacturing environments. One principal challenge is in understanding the collaboration requirements of manufacturers, and identifying and integrating appropriate collaboration technology solutions. In this paper we will present methods for user requirements gathering as well as the collaboration requirements for a specific manufacturing scenario. We will also describe the design of a prototype collaborative annotation tool for synchronized multimedia data streams (ACTS) to support a key collaboration requirement.

Collaboration technologies for manufacturing process analysis and trouble shooting

Researchers at NIST's Manufacturing Engineering Laboratory (MEL) and Information Technology Laboratory (ITL) are assessing collaboration technologies for manufacturing applications. We are particularly interested in the effective use of collaboration tools in manufacturing environments and how manufacturing practices will change as a result of their use. We expect these studies will yield useful insights into future data interchange standards needs, advance the state of the art and practice in CSCW deployment techniques for the manufacturing domain and produce some noteworthy collaboration and data visualization tools.

We are using the user-centered system design [4] concept to develop ACTS. User-centered system design has been shown to increase the likelihood of acceptance, effectiveness and user satisfaction of information technology (IT) systems [5]. Currently, the first prototype of ACTS is nearing completion. A field study methodology is being used to document the work and show where there are changes in manufacturing processes and data exchange requirements as a result of collaboration technology systems' use.

This field study, set in the context of automated gas-metal robotic welding, assesses the deployment and use of collaboration technologies for manufacturing process engineering and trouble shooting scenarios. In industry, there is a relative scarcity of welding engineers. This scarcity is exacerbated by the current requirement that welding engineers be on-site when setting up new welding processes or when trouble arises on a manufacturing welding line. Collaboration technology holds the promise of realizing substantial savings in productivity by allowing geographically dispersed welding teams to trouble shoot bad welds over time and distance, as conceptualized in the "Jade" illustration.

A group of NIST researchers have collaboration requirements similar to those found in the "Jade" operations illustration. As a geographically dispersed team, they are working to define interface standards among welding work cell components, controllers and power supplies. A functioning welding testbed has been implemented for testing new interfaces [6]. The "Analyze Weld" activity (see Figure 1) is performed to verify effective welding cell performance. Effective collaboration over welding data over time and distance is a critical component to this team's success. Just as in the industrial operations scenario, task appropriate collaboration and data visualization technologies hold the promise of effective collaboration.

Our research first targets the research collaboration environment and then the related welding operations environment, in conjunction with our industrial partners. This approach takes advantage of the similarities in the weld quality analysis activity in the research environment and the welding trouble shooting activity in the operations environment. In both cases, user requirements are gathered and analyzed, the work process is documented, and an appropriate set of tools is deployed and evaluated.

User requirements and analysis. To collect users' requirements, we interviewed all of the participants in the research scenario, including one representative industry partner, noting their roles and requirements. Gathering user requirements also entailed understanding the pertinent user processes. This activity had a two-fold benefit. First, we better understood user requirements and occasionally were able to ask questions that brought otherwise hidden requirements to light. Second, by understanding the process, we were able to document it, facilitating later identification of process change. Figure 1 shows the high-level welding process activities for the research scenario.

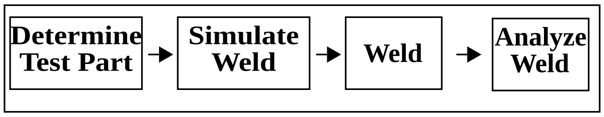

Figure 1: Welding Research Process

The focus of the remainder of this paper is on the requirements for "Analyze Weld" activity and the prototype annotation tool, ACTS, which supports that activity.

Summarized requirements.

- Weld analysis requires collaboration among participants in distant locations and time zones. Asynchronous communications are required.
- The welding process generates data in various formats that multiple people need to access, review, and annotate. Not all formats have been specified to date.
- NIST researchers require a central repository of data, which supports appropriate access permission controls, supports heterogeneous data formats, and is a means for organizing data and interactions around a central principle, e.g., around a particular weld or part.
- Engineers need to divide time among several problems, and therefore do not want the burden of being in lockstep synchrony with each current problem.
- High networking bandwidth requirements can not be imposed because some welding industries and sites do not have high capacity networking infrastructure.
- Multi-platform computer support is needed for all the major computing platforms.
- To analyze welds, a data visualization tool incorporating an overlay of bad welds on a good weld template with delineated tolerance ranges is needed.
- To identify trends and analyze problems, a visualization of a time series of good and bad welds per work cell is needed.
- A synchronized replay of weld audio, video, sensor, and controller data is needed. Further, the capability to make annotations at notable events during the weld data replay is especially important.

ACTS and web technologies

To support the described requirements, particularly the last requirement, researchers at NIST are prototyping a web-based tool to review and annotate synchronized multimedia data streams. A non-web-based approach has been pursued to address these requirements, however, we believe that a web-based approach, using open standards, will ultimately be more universally useful. There are certainly other approaches to this problem, but we are not aware of any existing implementations that are web-based and use open standards. ACTS provides the ability to playback previously captured and synchronized multimedia data streams and allow creation and review of annotations along the replay of such a data stream. The following sections describe advanced web technologies that are used in the initial ACTS prototype: eXtensible Markup Lanugage (XML), SMIL, and Document Object Model (DOM). In addition, Java applets and Java Media Framework (JMF) are also discussed. Figure 2 shows a depiction of the ACTS system components.

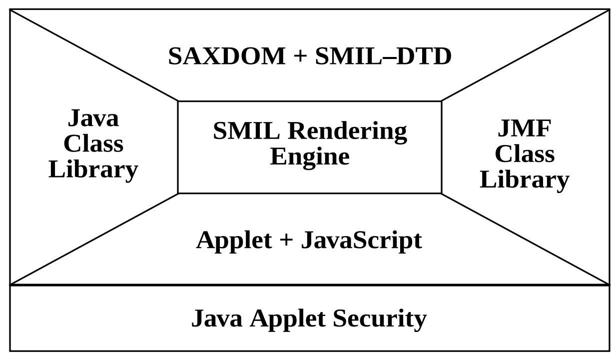

Figure 2: ACTS System

The eXtensible Markup Language [7]. XML is a simplified version of the Standard Generalized Markup Language (SGML) [8] that allows document structures to be defined, levels of subdivisions to be created, and content data to be stored and retrieved hierarchically. XML is centered around how data is structured rather than on how data is displayed. XML provides the mechanism to define new markup tags so that new attributes can be created for specific needs and data to be searched more effectively and efficiently.

Synchronized Multimedia Integrated Language [9]. SMIL is a text-based, user-readable and editable markup format. Like other XML-based markup languages, SMIL has its own grammar and rules that are defined in its Document Type Definition (DTD). This DTD provides a set of rules that specify which tags and values are allowed within a SMIL document. The documents can be served through any Web server by its Multipurpose Internet Mail Extensions (MIME) type (application/smil). Since SMIL is based on the XML model, it conforms to the basic structures and rules of XML, such as a DTD structure and the Well-Formed rule.

An important feature of SMIL is its ability to enable Web pages to position visual media objects (e.g., location, size, z-index ordering) and assign temporal attributes (e.g., begin time, duration or end time) between one another.

Additionally, it provides capabilities to create basic interactive multimedia presentations (via hyperlinks) similar to those on computer-based interactive compact discs (CDs), except the content data can either reside locally or be distributed over the Web. However, unlike components of interactive CDs, SMIL media objects can be updated independently and remotely. ACTS uses SMIL as the underpinnings for its content infrastructure so that multi-source multimedia objects can be played synchronously.

Document Object Model [10]. DOM provides the middle layer interface between domain applications and the document content objects, so that document content objects can be organized dynamically. ACTS uses SAXDOM [11] (DOM on top of Simple Application Protocol Interface (API) for XML) to modularize the DTD parsing for the SMIL player. SAXDOM separates the underlying parsing functions into a set of callback routines from the SAXDOM driver. When a given SMIL document is ready for rendering, the SAXDOM driver uses the SMIL DTD to parse the document content to get the node names and attribute values, and pass them back to the application callback routines. Then the application decides what actions to execute.

Java applets. Java applet technology was chosen for the client implementation due to its tight integration with web browsers. A Java-based collaborative annotation tool can use the full capabilities offered by current popular web browsers without the need to download any additional plug-ins.

Java applets come with strong security restrictions that disallow access to local resources as well as other network hosts. Since ACTS needs to access multimedia clips from other hosts other than the host where the applet was downloaded, it is necessary to relax the applet security. ACTS employs the signed certificate method that uses a trusted applet of all the ACTS classes, bitmaps, and other supporting files.

Java Media Framework [12]. Due to the strong requirements for a high performance multimedia (audio and video) engine in Java, a group of vendors created a framework called JMF. JMF provides the basic binding between a set of cross-platform Java application protocol interfaces (APIs) to the underlying operating system (OS) for handling audio and video functions. Instead of improving the Java Virtual Machine or providing a set of new Java routines to handle audio and video streams, JMF utilizes the underlying OS functions to handle specific audio and video streams via Java Native Interface (JNI). With this architecture, all the common audio/video rendering routines will not need to be downloaded each time a Java applet runs (or have the above classes locally available for Java applications). In addition, locally optimized OS-level audio/video routines can be used for optimal system performance.

ACTS system architecture

The ACTS architecture was established on the client-server model. It is based on an HyperText Transfer Protocol (HTTP) protocol to get the SMIL document with its associated multimedia clips and data plots from the HTTP server to the client browser application. ACTS uses

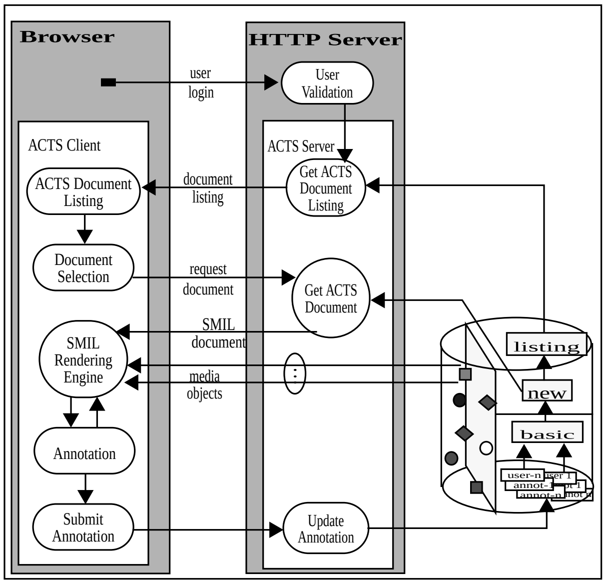

Figure 3: ACTS Architecture

the concept of check-in and check-out to view the materials and the annotations. Each annotator will download the pre-generated cache directory listing and the newly generated master SMIL annotation document for viewing. Once all new annotations are created at the local machine, the annotator can submit them back to the ACTS server (each user will have a dedicated directory to host his/her annotations). See Figure 3 for a depiction of the architecture.The following subsections discuss some of the key components.

The ACTS client, a Java-enabled browser. ACTS allows annotators to create and access annotation materials through the browser interface. Since annotation materials are associated with audio and video streams and still images with synchronized sweeps moving across them, it is important to have a system that provides a cohesive presentation. Additionally, since all required media types have not been specified to-date, a solution that readily accommodates new media types is desirable.

ACTS server authentication. Any network-based information editing system must have strong authentication to ensure the modified information is returned to the server for access by other users. Instead of creating our own authentication routines within ACTS, ACTS fully utilizes the HTTP server authentication. HTTP servers provide different levels of authentication.

ACTS annotation subsystem. Since ACTS is envisioned to be a general purpose collaborative annotation tool, it is anticipated that various users could reasonably expect to view and annotate several different jobs simultaneously. The annotation storage facility must account for this possibility. The ACTS server uses a directory hierarchy to store annotations. Each user will have his own dedicated directory, and each annotation will be stored in a separate file. This way, adding, deleting, and modifying annotations will be handled at the file level, which facilitates record locking, etc., on some platforms, e.g., Unix systems. Additionally, the ACTS server keeps a “master” annotation file, which is updated whenever an annotation is created. A sample excerpt of an annotation using SMIL mark-up follows, illustrating that annotations can begin at any point in time for a specified duration:

```
<textstream id=”User1” title=”annotator 1” region=”t1”
begin=”17s” dur=”10s” fill=”freeze” src=”http://
smil.nist.gov/text/ant1.txt” />
```

For any collaboration tool, it is important to avoid race conditions among multi-users when dealing with synchronous updates. It is also critical to provide fast access by using pre-generated, cached information whenever it is possible. The ACTS annotation subsystem design uses a dedicated directory/file structure with each annotation stored in a separate file and “master” annotation files are updated by the ACTS server itself. This architecture prevents file-locking and race conditions in a multi-user environment.

ACTS prototype features. The following highlight some of the ACTS key features:

- Automatic text scrolling when viewing annotations,
- Ability to associate annotations with still images or video/audio streams,
- Ability to hyperlink to other Universal Resource Locator (URL) resources,
- Multiple anchors within images,
- Graphical timeline annotation view to scan where the annotations are stored,
- Various datatype charts (timeline, box, linear, waveform) to represent user data,
- Automatic pause when editing an annotation,
- Allows use of a time slider to locate and play a particular segment of the viewing material (thus avoiding the need to play the entire stream to locate a particular segment),
- Total time frame and elapsed time of all viewing materials, and
- Multimedia objects can be included in annotations, which can also be annotated.

Anticipated proposed SMIL extensions

For ACTS to provide the required functionality for the case study, a few modified and new SMIL tags are needed. Generally, these extensions pertain to presenting data plots and associated objects within a presentation, e.g., a plot of an acoustic waveform with its actual audio sound. The following describes the modification of some existing SMIL tags and defines a few new SMIL tags for our ACTS annotation collaboration tool.

New attribute for the visual object tag. To provide the required functionality, it is necessary to have extra visual object control. For this work, it is desirable to control the width specification of a visual object’s border and the border color. For example, when presenting a picture image, a border frame may be desirable, whereas when presenting a button image, it is not.

New attributes for the <textstream> tag. To normalize the behavior between SMIL players with respect to the presentation of text streams, additional specifications should be allowed with the SMIL document. These behaviors are: (a) the delay time for how fast (in seconds) the text is supposed to scroll, (b) the direction (horizontal or vertical), and (c) the orientation (left-to-right/right-to-left or top-to-bottom/bottom-to-top) the text is supposed to scroll. To do this, we created delay, direction, and orientation attributes for the <textstream> tag.

New <dataplot> and <datatype> Elements. The graphical data plot presentation form provides a visualization of the data instead of a long listing of numerical values. Although there are many different types of data plots (line chart, XY, scatter chart, bar chart, column chart, etc.), for the ACTS implementation several data plot types were identified that were needed to make the visualization aspects of the tool be more complete. An element called dataplot was created that contains various chart types that are called datatype elements. Multiple datatypes can use one dataplot timeline for synchronization. This way, multiple datatypes can be synchronously presented during the playback.

Conclusions and future work

In this paper we presented methods for user requirements gathering as well as the collaboration requirements for a specific manufacturing scenario. We also presented the design of a prototype collaborative annotation tool for synchronized multimedia data streams (ACTS) to support a key collaboration requirement. Our goal was to build a web-based solution built on open standards that supports the stated user requirements. We believe this has been accomplished.

ACTS's major concepts.

- It is possible to use SMIL content technology as the infrastructure for a collaborative annotation tool. SMIL provides core functionality for temporal and spatial synchronization.
- By adding and modifying SMIL 1.0 tags, ACTS can include other data graphs along with multimedia clips for a fully functional collaborative analytical tool.
- In addition to annotating texts, ACTS can annotate and playback synchronized hypermedia annotations.

Future work includes:

- Deploy ACTS in the described case study and assess ACTS's effectiveness in addressing the requirements.
- Test scalability with respect to the number of users simultaneously using the system.
- Expand the current ACTS prototype to use remotely controllable agents, so that each user can exchange annotations among colleagues in real time. This would avoid the potential scalability problem wherein clients are repeatedly downloading the same media object(s) potentially causing annotation server and network bandwidth bottlenecks.
- Enable one agent to take control of how the playback should behave (play, pause, stop) between other agents when viewing a specific segment of the presentation among multiple users.

Acknowledgments

ACTS is part of the Streaming Synchronized Multimedia project at the National Institute of Standards and Technology. We particularly thank Dr. John Tang of Sun Microsystems for his contributions to this project during his tenure as a guest researcher at NIST. This work has been funded in part by the Systems Integration for Manufacturing Applications (SIMA) program, the Advanced Technology Program (ATP) and the National Advanced Manufacturing Testbed (NAMT) program at NIST, with additional support by the Defense Advanced Research and Projects Agency (DARPA).

References

[1] SYMM: SYnchronized MultiMedia Working Group. http://www.w3.org/AudioVideo.

[2] W. Chang and J. Yu. S2M2 - A Java Applet-Based SMIL Player, International Conference on Multimedia & Telecommunication Management, December, 1998.

[3] Menon, J., "Collaborative Visualization and Modeling", Proceedings of the International Conference on Shape Modeling and Applications, Aizu Wakamatsu, Japan, March 1997, pp.178-187.

[4] D. Norman and S. Draper (editors), User Centered System Design, Lawrence Erlbaum Associates, Inc. 1986

[5] Marcus, A., van Dam, A., "User Interface Developments for the Nineties", Computer, (September 1991), Vol. 24 9, pp.49-57.

[6] Rippey, W., Falco, J., "The NIST Automated Arc Welding Testbed", Proceedings of the Seventh International Conference on Computer Technology in Welding, July 1997.

[7] T. Bray, J. Paoli, C. Sperbrg-McQueen (editors). Extensible Markup Language. http://www.w3.org/TR/REC-xml.

[8] ISO 8879: 1986(E), SGML: Standard Generalized Markup Language, Information processing - Text and Office Systems.

[9] P. Hoschka (editor). Synchronized Multimedia Integrated Language (SMIL). http://www.w3.org/TR/REC-smil.

[10] L. Wood (chair). Document Object Model. http://www.w3.org/TR/WD-DOM.

[11] SAXDOM: The Simple API for XML for DOM. http://www.megginson.com/SAX/

[12] JMF: Java Media Framework. http://java.sun.com/products/java-media/jmf/.