Real Time Streaming Data Grid Applications

Geoffrey C. Fox, Mehmet S. Aktas, Galip Aydin, Hasan Bulut, Shrideep Pallickara, Marlon Pierce, Ahmet Sayar, Wenjun Wu, and Gang Zhai

Community Grids Lab, Indiana University
501 North Morton Street, Suite 224
Bloomington, IN 47404
{gcf, maktas, gaydin, hbulut, spallick, mpierce, asayar, wewu, gzhai}@cs.indiana.edu

Abstract. We review several aspects of building real-time streaming data Grid applications. Building on general purpose messaging system software (NaradaBrokering) and generalized collaboration services (GlobalMMCS), we are developing a diverse set of interoperable capabilities. These include dynamic information systems for managing short-lived collaborative service collections ("gaggles"), stream filters to support the integration of Geographical Information Systems services with data analysis applications, streaming video to support collaborative geospatial maps with time-dependent data, and video stream playback and annotation services to enable scientific collaboration.

1 Introduction

This paper describes research work of the Community Grids Laboratory on Grids built around streaming data sources. This work builds upon general purpose messaging middleware (NaradaBrokering [1, 2, 3]) and incorporates a diverse set of services that include audio/video conferencing (GlobalMMCS [4]) and Geographical Information System services [5]. The architecture and core services are summarized in a recent companion publication [6]. Here, we examine more closely applications and additional functionality that are being integrated into the overall system.

A critical idea in our approach is to view both services and messages (and streams as ordered set of messages) as "first class" entities. The law of the millisecond [7] suggests one should use this type of message oriented middleware (software overlay network) when one can afford latencies of a millisecond or more. This is characteristic of all systems with significant geographic distribution and non-specialized interconnect. As discussed in [6-8], NaradaBrokering is capable of supporting millisecond messaging in a diverse range of applications, ranging from binary data streams to XML-based Web Service messages.

For scientific applications, the data deluge [8] suggests the growing importance of real time data assimilation with the integration of sensors, databases and simulation codes. Similarly, the geographically distributed nature of much current research requires collaboration tools. Our research thus focuses on the reuse of concepts and software as well as integration of data-driven and collaboration-driven real-time problems. As discussed in [6], systematic use of Grid and web services gives us interoperability and access to commodity (industry) capabilities. In our system, we generalize the well-known "system of systems" concept to a "Grid of Grids" [9] and show how one can build Grid applications by using appropriate services drawn from Grid service families.

In this paper, we concentrate on Grid service families that are applicable to real-time data Grid applications. These include general purpose metadata and discovery services, Geographical Information System services suitable for streaming data and information, and services to support playback and shared annotation of video streams. These diverse services may be integrated into Grid of Grids applications through the use of management, orchestration, and workflow services, such as described in [10].

We have identified the importance of supporting both the worldwide Grid and smaller sessions or "gaggles of grid services" that support local dynamic action. We discuss in Sec. 2 the meta-data services optimized for these different requirements. This work also shows the need for three distinct types of XML data/metadata services. UDDI exemplifies a scalable repository; WS-Context a general dynamic store and Web Feature Service, a domain specific repository. Our implementations of these do not use XML databases but for efficiency convert the XML to SQL and store in a conventional MySQL database. This illustrates the important difference between semantics and representation. We preserve the XML Infoset (semantic meaning) but for efficiency do not use a conventional XML representation.

This paper surveys several aspects of real time streaming data grids. We first discuss the metadata management requirements of these systems (Sec. 2). Collaborative streaming systems involve both large, mostly static information systems as well as much smaller, highly dynamic information systems. We refer to these latter collections as "gaggles." We next review the integration of streaming data and Geographical Information systems. This may involve both streaming data (suitable for data mining and other applications), described in Sec3, as well as streaming map imagery (suitable for user interfaces), described in Sec 4. Map servers with streaming video capabilities can also be integrated with GlobalMMCS's general purpose collaborative infrastructure. We may thus inherit many additional features such as replay and collaborative annotation and whiteboard systems, as described in Sec. 5. We summarize this paper and future research activities in Sec. 6.

2 Gaggle-Like Metadata Support in GIS and Sensor Grid

Geographical Information Systems and Sensor Grids present an environment where many geo-resources and geo-processing applications are packaged as services and put together for a particular functionality such as forecasting earthquakes [10]. Here, Grid Information Services [11] maintains metadata about these geo-services and provides standardized methods for publishing and discovery. An Information Service can be thought of as a solution to general problem of managing information about Grid/Web Services, yet it should also support domain-specific information requirements such as geospatial domain information requirements.

GIS/Sensor Grid information may be classified as either a) session metadata or b) static, interaction-independent metadata [12]. Session metadata is the dynamically generated information as result of interactions of Grid/Web Services. Static and interaction-independent metadata is the information describing Grid/Web Service characteristics. Information Services should support handling and discovery of both session-based and static metadata associated to services.

The GIS/Sensor Grids may be thought of as an actively interacting (collaborating) set of managed services where services are put together for particular functionality [12]. Here each collection of services maintains most dynamic information which is the session related metadata. Handling and discovery of such dynamic information requires high performance, fault tolerant and distributed systems.

We use and extend UDDI [13] and WS-Context [14] specifications to build Information Services supporting extensive metadata requirements of GIS Grids [15], [16]. We also design distributed metadata management architecture to support dynamically assembled GIS/Sensor Grid applications where metadata is widely-scattered and dynamically generated [17]. Our Information Services implementation has been focused in two main branches. First, we designed and implemented an advanced UDDI Information Model to support metadata-oriented service registries. Here, we base this on an implementation of Web Service interfaces for publishing and discovering metadata associated to Grid/Web services. We extend basic UDDI functionality by implementing advanced functionalities in order to process service metadata and keep the Registry entries up-to-date. Some examples of these functionalities are a) monitoring capabilities such as leasing, b) Geographical Information System-specific taxonomies to support geo-spatial services, and c) XPATH query capabilities to support domain-specific Information Services.

For dynamic session data, we have designed and implemented a Context Service that deals with handling and discovery of dynamic, session-related metadata. Here, session related metadata is short-lived and dependent on the client. We base this on the WS-Context specification from the standards organization OASIS. We extended WS-Context Specifications to provide advanced capabilities to manage session metadata between multiple participants in Web Service interactions. To decentralize the Context Service, we currently implement a distributed and dynamic metadata management architecture which is described in [16] in greater detail.

3 High Performance Web Service Grid Architecture to Support Real-Time Sensor Streams

Recent technological developments have allowed sensors to be deployed in a variety of application domains. Environmental monitoring, air pollution and water quality measurements, detection of the seismic events and understanding the motions of the Earth’s crust are only a few areas where extent of the deployment of sensor networks can easily be seen. Extensive use of sensing devices and deployment networks of sensors that can communicate with each other to achieve a larger sensing task will fundamentally change information gathering and processing [17]. However, the rapid proliferation of sensors presents unique challenges different than the traditional computer network problems.

Several studies have discussed the technological aspects of the challenges with the sensor devices, such as power consumption, wireless communication problems, autonomous operation, adaptability to the environmental conditions, etc [18] [19]. Here we describe architecture to support real-time information gathering and processing from Global Positioning System (GPS) sensors by leveraging principles of service oriented architectures and open GIS standards.

3.1 GPS Networks

The Global Positioning System has been used in geodesy to identify long-term tectonic deformation and static displacements while Continuous GPS has proven very effective for measurement of the interseismic, coseismic and postseismic deformation [20]. Today networks of individual GPS Stations (monuments) are deployed along the active fault lines, and data from these are continuously being collected by several organizations. One of the first organizations to use GPS in detection of the seismic events and for scientific simulations is Southern California Integrated GPS Network (SCIGN) [21]. One of the collaborators in SCIGN is Scripps Orbit and Permanent Array Center (SOPAC) [22] which maintains several GPS networks and archives high-precision GPS data, particularly for the study of earthquake hazards, tectonic plate motion, crustal deformation, and meteorology. Real time sub-networks maintained by SOPAC include Orange County, Riverside County (Metropolitan Water District), San Diego County, and Parkfield. These networks provide real-time position data (less than 1 sec latency) and operate at high rate (1 – 2 Hz).

3.2 Real-Time Streaming Access to GPS Position Messages

As shown in Fig. 1, raw data from the SOPAC GPS stations are continuously collected by a Common Link proxy (RTD server) and archived in RINEX files. The RTD server outputs the position of the stations in real-time in a binary format called RYO. To receive the position messages, clients are expected to open a socket connection to the RTD server. An obvious downside of this approach is the extensive load this might introduce to the server when multiple clients are connected. These data streams are collections from entire networks rather than individual stations, and for many applications will require per station separation and reformatting. As described below, we are developing general purpose solution to these filtering problems.

To make the position information available to the clients in a real-time streaming fashion we are using the NaradaBrokering messaging system. Additionally we developed filters to serve position messages in ASCII and Geography Markup Language [23] formats.

3.3 Chains of Filters

Since the data provided by RTD server is in a binary format we developed several filters to decode and present it in different formats. Once we receive the original binary data we immediately publish this to a NaradaBrokering topic (null filter), another filter that converts the binary message to ASCII subscribes to this topic and publishes the output message to another topic. We have developed a GML schema to describe the GPS position messages. Another filter application subscribes to ASCII message topic and publishes GML representation of the position messages to a different topic. This approach allows us to keep the original data intact and different formats of the messages accessible by multiple clients in a streaming fashion.

Our GML Schema is based on RichObservation type which is an extended version of GML 3 Observation model [24]. This model supports Observation Array and Observation Collection types which are useful in describing SOPAC Position messages since they are collections of multiple individual station positions. We follow strong naming conventions for naming the elements to make the Schema more understandable to the clients.

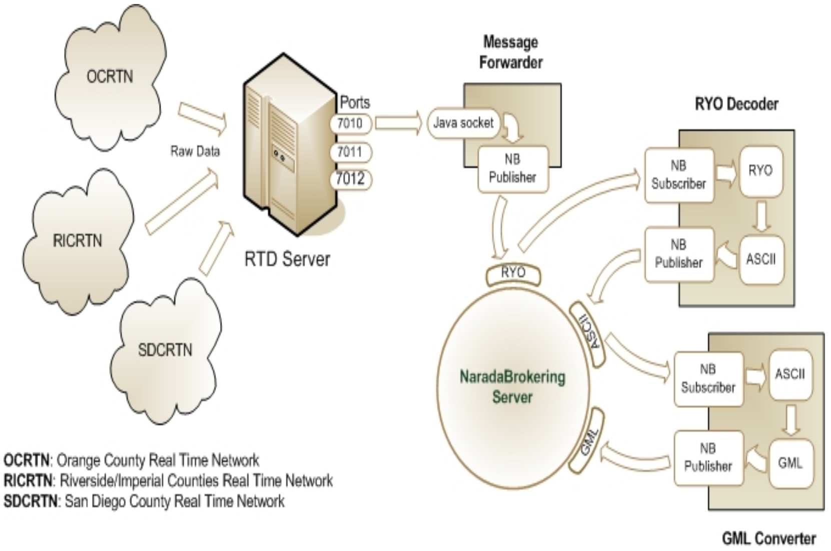

Fig. 1 – SOPAC GPS Services

Currently the system is being tested for the following networks: San Diego Counties, Riverside/Imperial Counties and Orange County GPS networks. For more information, see [5]

3.4 SensorGrid Research Work

We are developing a Web Service-based Grid called SensorGrid for supporting archived and real-time access to sensor data. GPS services we mentioned above will be part of this system. SensorGrid will help us integrate sensor data with scientific applications such as simulation, visualization or data mining software by leveraging GIS standards and Web Services methodologies. The system will use NaradaBrokering as the messaging substrate and this will allow high performance data transfer between data sources and the client applications. The Standard GIS interfaces and encodings like GML will allow data products to be available to the larger GIS community. We will extend OGC's [25] Sensor Collection Service to support streaming data. The Sensor Collection Service provides sensor metadata encoded in SensorML and the actual data products. We use a Web Service version of OGC WFS to distribute archived geospatial data.

3.5 Negotiation Protocol for High Performance Data Transport

Several studies [26, 27, 28, 29] have shown that transport of XML and SOAP messages encoded in conventional "angle-bracket" representation is too slow for applications that demand high performance. At the same time several groups are developing ways of representing XML in binary formats for fast message exchange [28, 29].

We are developing a Web Service negotiation language for higher performance Web Services, a protocol for Web Services to negotiate several aspects of the data transportation such as representation scheme (Fast XML, Binary XML) and transport protocol (TCP, UDP). Initial negotiation will be done using standard angle-bracketed messages to determine the supported representation and transport capabilities. We will employ handlers to take care of the conversion and transport issues, which will make the negotiation and transport process transparent to the services. Once the services agree on the conditions of the data exchange, handlers will convert XML data into an appropriate binary format and stream it over a high performance transport protocol (such as UDP) using NaradaBrokering.

4. Collaborative Web Map Services

The Open Geospatial Consortium defines several related standards for the representation, storage, and retrieval of geographic data and information. The Web Map Service (WMS) [30] produce maps from the geographic data. Geographic data are kept in Web Coverage Server (WCS) or Web Feature Service (WFS). WCS stores raster data in image tiles and WFS stores feature vector data in GML formats. WMS produces maps from these raw geographic data upon requests from the WMS clients. These maps are the static representations of geospatial data. Representations are in pictorial formats such as PNG, SVG, JPEG, GIF, etc. [31]

4.1 Streaming Video and Map Servers

Standard map servers produce static images, but many types of geographic data are time dependent. In order to understand geographic phenomena and characteristics of temporal data it is necessary to examine how these patterns change over time for these types of data. We are therefore investigating the problems of creating streaming video map servers based upon appropriate standard collaboration technologies [4].

In our approach, visualizing changes over time is achieved by integrating temporal information on a map. Usually the result is a series of static maps showing certain themes at different moments. In addition to creating static maps, WMS also has the ability to combine the static maps correspond to a specific time interval data and combine them in an animated movie. Movies created by WMS are composed of a certain number of frames. Each frame represents a static map that corresponds to a time frame defined in request.

WMS is not able to create movie for all of its supported layers listed in its capabilities file. If WMS supports movie functionality for a layer it adds some attributes under this layer. Before making the request, the client first makes a "get capabilities" request and after getting information about the WMS, it makes requests according to capabilities of the WMS. If a client makes a request to get a movie for a specific layer, to succeed, this layer should have a time dimension defined under this layer element in the capabilities file. Clients should make the "get map" standard request to WMS to get the movie for a specific layer. WMS does not provide any other request types for the movie creation functionalities. Clients set the "format" variable to string "movie/<movietype>" and "time" variable to a value in an appropriate format to make a request to get movie from the WMS.

We initially examined two approaches to creating movies: one is client oriented and the other is server oriented. For the sake of performance issues, we have chosen the second one for the collaborative map movies. Since our aim is to create collaborative map movies, we do not need to get the movie back to the client. As soon as the movie frames are created at WMS side for each time slice for the same data layer, WMS publishes them as streams to specific Real Time Protocol (RTP) [32] sessions. RTP sessions are represented as <IP Address, Port Number> pairs. A video stream published to a RTP session can be visualized by any video client connecting to the same RTP session. RTP session can be configured at properties file. Map images are dynamically generated from raw geographic data and those images are transcoded into video streams. The supported video stream formats are H.261 and H.263, which are mostly used formats in AccessGrid sessions. Map video stream can be played in collaborative environments such as AccessGrid and GlobalMMCS sessions.

If the client oriented approaches is used, the client waits the movie to return and once it returns client either starts to play the movie or archive or both. Archived movies can be replayed without making another request to WMS. As it is shown in this scenario there are some tradeoffs between these two approaches.

To solve these tradeoffs we have developed a third approach. This is the best way for the performance issues that we will be using in the near future for the movie mapping. By this approach, creating and publishing movies to collaboration sessions will be achieved at the client side. Client will not make request for the movie. In other words, client will never set "format" variable to movie/<movietype>. It will make threaded "get map" requests for the defined periodicity between starting date and ending date. All the returned static maps in images will be stored locally based on their time intervals and periodicities. When client needs to replay some or all parts of stored images then, he does not need to make a "get map" request for these layers. If there is no match for some time intervals then clients make the "get map" requests just for these unmatched frames.

The map video stream has several parameters that can be adjusted. These parameters affect the quality of the produced map video stream. Among these configurable parameters are frame rate and video format of the stream, update rate of the map images in the video stream. In our experiments we updated map images for every 0.5 seconds while we kept the video frame rate at 10 frames per second (fps). This provides a high quality of the video stream at the receiving side. This is necessary because some clients might not be capable of visualizing video streams with low frame rate or can visualize them with very low quality.

Map video streams produced are published to RTP sessions, whether unicast or multicast session. AccessGrid clients use multicast sessions to send/receive video. Once a video is published to a multicast session, it can be received by any client listening that multicast session as long as the underlying network lets client receive multicast packets. GlobalMMCS can also provide this map video stream to its clients as unicast video stream, as discussed in Sec. 5.

4.2 Map Service Research Issues

We will continue to integrate the streaming map server with GlobalMMCS's archiving capabilities and will examine higher performance encodings. This will enable useful functionality, such as allowing users to select a movie from the archive. To make the WMS available for the collaborative conferences or online education, administrative users will be able to update map streams on the fly while they are playing.

We will be creating movies at the client side even in case of collaborative movie creating environment. There will be significant performance gain when we use this approach. Clients can archive both previously created frames and movies. If a client needs the same type of frames for the same matching time intervals then it does not need to go back to WMS and spend time getting the movie frames.

5. e-Annotation Collaborative Architecture

In this section we describe video archiving and annotation services that we are adding to the GlobalMMCS collaboration system. GlobalMMCS (Global Multimedia Collaboration System) is described in more detail in Ref. [4]. In brief summary, it is a scalable, robust, service-oriented collaboration system. GlobalMMCS integrates various services including videoconferencing, instant messaging and streaming, and is interoperable with multiple videoconferencing technologies. This collaboration system is developed based on the eXtensible Generic Session Protocol (XGSP) collaboration framework and NaradaBrokering messaging middleware. Such a service-oriented collaboration environment greatly improves the scalability of traditional videoconferencing system, benefits users with diverse multimedia terminals through different network connections, and simplifies the further extension and interoperability.

We use a component based design on top of NaradaBrokering shared event collaboration model to make e-Annotation system scalable and extensible for new functional plug-ins. e-Annotation is designed to run in grid computing environment, which spans different organizations across different countries. The architecture is shown in Fig. 2.

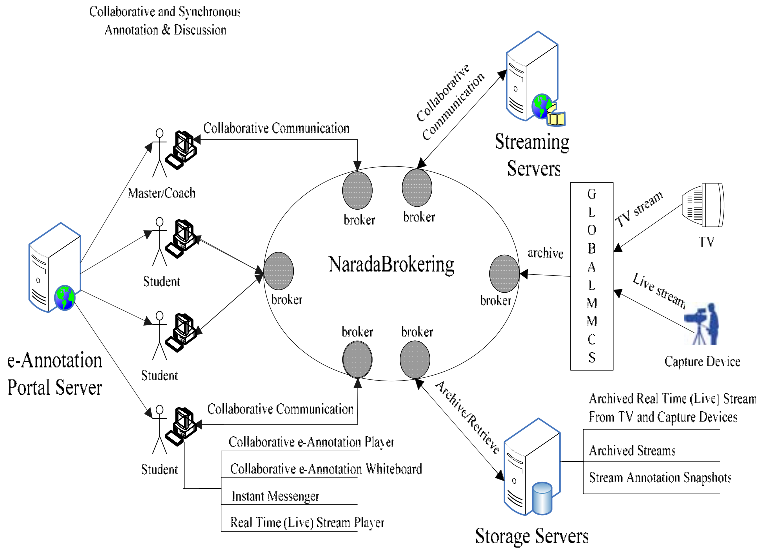
Collaborative and Synchronous
Annotation & Discussion
Collaborative Communication
Collaborative Communication
Streaming Servers
Master/Coach
broker
broker
NaradaBrokering
broker
broker
archive
G L O B A L M M C S
TV stream
TV
Live stream
Capture Device
Student
Student
broker
broker
e-Annotation Portal Server
Collaborative Communication
Archive/Retrieve
Archived Real Time (Live) Stream From TV and Capture Devices
Archived Streams
Stream Annotation Snapshots
Student
Collaborative e-Annotation Player
Collaborative e-Annotation Whiteboard
Instant Messenger
Real Time (Live) Stream Player
Storage Servers

Fig. 2 e-Annotation Architecture

5. 1 Streaming Server

The Real Time Streaming Protocol (RTSP) [33] is a protocol specified in the Internet Engineering Task Force's RFC 2326 for control over the delivery of live or stored data with real-time properties. RTSP is similar in syntax and operation to HTTP/1.1, which allows it to be extended by HTTP, but as opposed to HTTP, RTSP maintains state by default. States involved in RTSP are init, ready, playing/recording. Basic RTSP control functionalities are play, pause, seeking (absolute positioning to a specific point in the stream) etc.

Our collaboration system provides a Web Services-based streaming server which provides RTSP semantics with control functionalities mentioned above. In addition, the streaming service also provides seeking of live streams, which is not supported by the current RTSP implementations such as RealNetworks Helix Server [34] and QuickTime Streaming Server [35].

The streaming service uses NaradaBrokering middleware to transport data across network. Streaming clients (e-Annotation client) receives and sends data through NaradaBrokering nodes. Data sent across the network is wrapped inside native NaradaBrokering events. Any type of data, audio, video, still images or text messages can be wrapped inside events and sent across the broker network. Based on the stream metadata information, client processes accordingly whether it is video, audio or any other data type.

5.2 Streaming Client Support

In order to support e-Annotation client with RTSP semantics, a client side service called RTSP Client Engine has been developed. RTSP Client Engine interacts with the streaming services to ensure the RTSP functionalities required by the e-Annotation client for control over the delivery of the stream. The e-Annotation client can initiate as many RTSP sessions as possible using this service. In addition to replay of the stream, this also enables e-Annotation client to announce and record a new stream.

Seeking in live streams can be done until the point where it is last recorded. In order to enable the e-Annotation client to absolute position in the live stream, it requires timing information regarding the part of the live stream being recorded. The e-Annotation client subscribes to this service to retrieve timestamp information of the last data stored.

5. 3 e-Annotation Collaborative Tools

The e-Annotation collaborative architecture is a peer-to-peer collaboration system based on NaradaBrokering. The e-Annotation users are identical peers in the system, they communicate with each other to collaboratively annotate a real time or archived video stream. Each peer shares the same collaborative applications which include e-Annotation Player and e-Annotation Whiteboard.

5. 4 e-Annotation Player

As shown in Fig. 3, the e-Annotation player is composed of four components:

1. Stream list panel
2. Real time live video panel
3. Streaming player panel
4. Video annotation snapshot player panel.

The stream list panel contains the real time live video stream list, archived video stream list and the composite annotation stream list. All of the stream lists information is gotten from streaming server by subscribing the streaming control info topic from a broker. Real time video play panel plays the real time video that is selected by user in the real time video list. It subscribes the real time video stream data topic to get real time video data published by GlobalMMCS [globalmmcs]. Streaming player panel plays the archived video stream. Streaming player supports pausing, forwarding, rewinding a video stream with dynamic length (live video) or fixed length (archived one). It can also take snapshots on a video stream whiling playing. When taking a snapshot, the timestamp is associated with that snapshot. These snapshots are loaded to whiteboard to be annotated collaboratively. When playing back the composite annotation stream, the original video stream is played in the streaming player panel, at the same time, video annotation snapshot player play the annotation snapshots synchronized with the original video stream by the timestamps.

The e-Annotation Player is collaborative: each participant will view the same content in his/her e-Annotation Player, the only difference is that only the participant with the 'coach' role can control the play of a video stream(to pause, rewind, take snapshot).

5. 5 e-Annotation Whiteboard

The whiteboard works collaboratively in peer to peer mode as the e-Annotation Player does, each peer has the same view of the current whiteboard content. One user's draw on the whiteboard can be seen immediately in all other users' whiteboard. Each action in one peer's whiteboard will generate a NaradaBrokering event that will be broadcasted using whiteboard communication topic through NaradaBrokering to all other peers in this session, so all the peers get a consistent and synchronized view of the shared whiteboard. The user with the 'leader' role can control the save and erasure of whiteboard content, other 'student' users can only add comments (text, any shape, pictures etc) to the whiteboard, the whiteboard is where all the users do the annotation on a snapshot of a video stream taken from e-Annotation player, annotated snapshots will be saved as JPEG images with timestamps to generate a new composite annotation stream, this composite annotation stream is composed of the video stream plus the annotation snapshots, they are synchronized using the timestamps.

The e-Annotation Player and Whiteboard collaborative tools user interfaces are shown in Fig. 3.

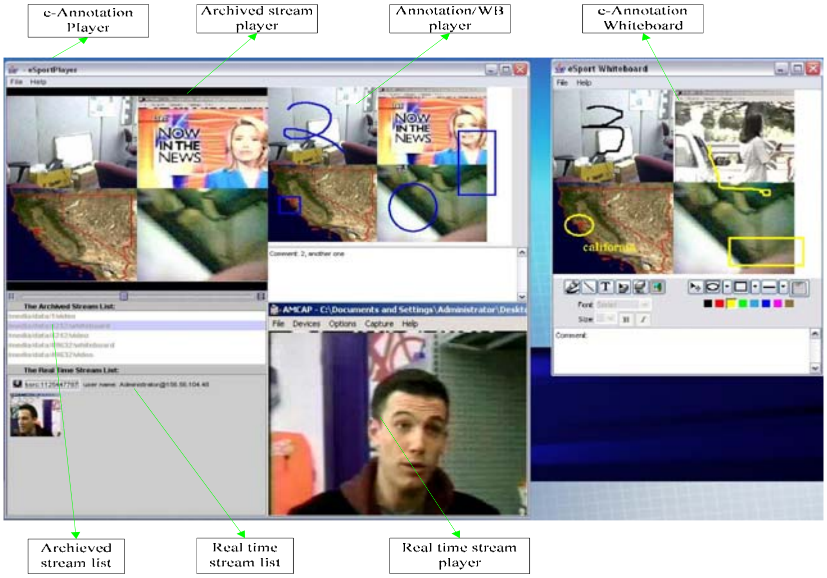

Fig. 3 e-Annotation Player and e-Annotation Whiteboard User Interface

5. 7 e-Annotation Status and Future Work

e-Annotation system supports synchronous collaborative video annotation. It supports annotation about a live real time stream from capturing devices or TVs. The annotation of video stream can be played back synchronously with the original stream by creating a new composite stream. The e-Annotation system provides a framework for integrating different multimedia collaborative tools for e-Coaching and e-Education.

The future work includes design and employs quality-of-service control mechanism to improve the performance and develop a new metadata framework for collaborative multimedia annotation system using XML as the metadata exchange and description

6. Summary and Future Work

We have reviewed in this paper several applications that build upon the Community Grid Laboratory's general Web Services messaging infrastructure (NaradaBrokering) and general collaboration services (GlobalMMCS). Our generalized metadata and information management research concentrates on extending static metadata services (UDDI) to support richer descriptions and discovery capabilities. We also distinguish between these session-independent services and session-dependent metadata services needed to support dynamic groups of services and users. We next reviewed our work on streaming data for Geographical Information Systems, which are described by the information services. GIS applications include both streaming data (GPS data streams and filters) for data analysis codes as well as video streams for human users. In the latter case, we build upon video streaming standards, which allow us to integrate collaborative mapping tools into more general audio/video systems. This in turn enables us to combine streaming map tools with general purpose video stream playback and annotation tools.

We have identified many areas of ongoing research at the end of each section. Of particular general interest to us is the investigation of high performance Web Services that take advantage of efficient message representations and higher performance transport protocols. This research will impact all of the services that we have described.

References

1. Shrideep Pallickara and Geoffrey Fox. NaradaBrokering: A Middleware Framework and Architecture for Enabling Durable Peer-to-Peer Grids. Proceedings of ACM/IFIP/USENIX International Middleware Conference Middleware-2003.
2. Shrideep Pallickara and Geoffrey Fox. A Scheme for Reliable Delivery of Events in Distributed Middleware Systems. Proceeding sof the IEEE International Conference on Autonomic Computing. 2004.
3. Shrideep Pallickara, Geoffrey Fox and Harshawardhan Gadgil. On the Creation & Discovery of Topics in Distributed Publish/Subscribe systems. (To appear) Proceedings of the IEEE/ACM GRID 2005. Seattle, WA.
4. Wenjun Wu, Geoffrey Fox, Hasan Bulut, Ahmet Uyar, Harun Altay "Design and Implementation of A Collaboration Web-services system", Journal of Neural, Parallel & Scientific Computations (NPSC), Volume 12, 2004. See also http://www.globalmmcs.org.
5. GIS Research at Community Grids Lab, web site: www.crisisgrid.org
6. Geoffrey Fox, Galip Aydin, Harshawardhan Gadgil, Shrideep Pallickara, Marlon Pierce, and Wenjun Wu Management of Real-Time Streaming Data Grid Services Invited talk at Fourth International Conference on Grid and Cooperative Computing (GCC2005), held in Beijing, China, during Nov 30-Dec 3, 2005.
7. Geoffrey Fox The Rule of the Millisecond in CISE Magazine Vol 6, No 2, pp 93-96 (2004).
8. Berman, F., Fox, G., and Hey, T., (eds.). Grid Computing: Making the Global Infrastructure a Reality, John Wiley & Sons, Chichester, England, ISBN 0-470-85319-0 (2003). http://www.grid2002.org.
9. Geoffrey Fox, Shrideep Pallickara, and Marlon Pierce Building a Grid of Grids: Messaging Substrates and Information Management to appear as chapter in book "Grid Computational Methods" Edited by M.P. Bekakos, G.A. Gravvanis and H.R. Arabnia
10. Galip Aydin, Mehmet S. Aktas, Geoffrey C. Fox, Harshawardhan Gadgil, Marlon Pierce, Ahmet Sayar SERVOGrid Complexity Computational Environments (CCE) Integrated

Performance Analysis Technical report June 2005 accepted as poster and short paper in Grid2005 Workshop, 2005.

11. B. Plale, P. Dinda, and G. Von Laszewski., Key Concepts and Services of a Grid Information Service. In Proceedings of the 15th International Conference on Parallel and Distributed Computing Systems (PDCS 2002), 2002.
12. Mehmet S. Aktas, Geoffrey Fox, Marlon Pierce Managing Dynamic Metadata as Context Istanbul International Computational Science and Engineering Conference (ICCSE2005) June 2005 also please see: http://www.opengrids.org/fthpis.
13. Bellwood, T., Clement, L., and von Riegen, C. (eds) (2003), UDDI Version 3.0.1: UDDI Spec Technical Committee Specification. Available from http://uddi.org/pubs/uddi-v3.0.1-20031014.htm.
14. Bunting, B., Chapman, M., Hurlery, O., Little M., Mischinkinky, J., Newcomer, E., Webber J., and Swenson, K., Web Services Context (WS-Context), available from http://www.arjuna.com/library/specs/ws_caf_1-0/WS-CTX.pdf.
15. Mehmet S. Aktas, Galip Aydin, Geoffrey C. Fox, Harshawardhan Gadgil, Marlon Pierce, Ahmet Sayar, Information Services for Grid/Web Service Oriented Architecture (SOA) Based Geospatial Applications, Technical Report, June, 2005.
16. Mehmet S. Aktas, Geoffrey C. Fox, Marlon Pierce, "An Architecture for Supporting Information in Dynamically Assembled Semantic Grids", Technical report August 2005. Available at http://grids.ucs.indiana.edu/ptliupages/publications/SKG2005_Aktas.pdf.
17. D. Estrin, R. Govindan, J. Heidemann and S. Kumar, "Next Century Challenges: Scalable Coordination in Sensor Networks," In Proceedings of the Fifth Annual International Conference on Mobile Computing and Networks (MobiCOM '99), August 1999, Seattle, Washington.
18. Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., and Cayirci, E., "A Survey on Sensor Networks" IEEE Communications Magazine, August 2002.
19. Archana Bharathidasan, Vijay Anand Sai Ponduru, "Sensor Networks: An Overview".
20. Bock, Y., Prawirodirdjo, L, Melbourne, T I. : "Detection of arbitrarily large dynamic ground motions with a dense high-rate GPS network" GEOPHYSICAL RESEARCH LETTERS, VOL. 31, 2004.
21. Southern California Integrated GPS Network web site: http://www.scign.org/.
22. Scripps Orbit and Permanent Array Center web site: http://sopac.ucsd.edu/.
23. The Geography Markup Language described in an Encoding Specification from Open Geospatial Consortium available from http://portal.opengeospatial.org/files/?artifact_id=4700.
24. Open Geospatial Consortium Discussion Paper, Editor Simon Cox: "Observations and Measurements". OGC Document Number: OGC 03-022r3.
25. The Open Geospatial Consortium, Inc. web site: http://www.opengeospatial.org/.
26. Chiu, K., Govindaraju, M., and Bramley, R.: Investigating the Limits of SOAP Performance for Scientific Computing, Proc. of 11th IEEE International Symposium on High Performance Distributed Computing HPDC-11 (2002) 256.
27. Oh, S., Bulut, H., Uyar, A., Wu, W., Fox G., Optimized Communication using the SOAP Infoset For Mobile Multimedia Collaboration Applications. In proceedings of the International Symposium on Collaborative Technologies and Systems CTS05 (2005).
28. R. Berjon, chair, "XML Binary Characterization Working Group Public Page." Available from http://www.w3c.org/XML/Binary/
29. Paul Sandoz, Alessando Triglia, and Santiago Pericas-Geertsen, "Fast Infoset." Avaialble from http://java.sun.com/developer/technicalArticles/xml/fastinfoset/.

30. Jeff De La Beaujardiere, OpenGIS Consortium Web Mapping Server Implementation Specification 1.3, OGC Document #04-024, August 2002.
31. Ahmet Sayar, Marlon Pierce, Geoffrey Fox OGC Compatible Geographical Information Services Technical Report (Mar 2005), Indiana Computer Science Report TR610.
32. H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, RTP: A Transport Protocol for Real-Time Applications. Internet Engineering Task Force Request for Comments 3550 (2003). http://www.ietf.org/rfc/rfc3550.txt
33. H. Schulzrinne, A. Rao, and R. Lanphier, Real Time Streaming Protocol (RTSP), Internet Engineering Task Force Request For Comments 2326 (1998). http://www.ietf.org/rfc/rfc2326.txt
34. RealNetworks Helix Server http://www.realnetworks.com/
35. QuickTime Streaming Server http://www.apple.com/quicktime/streamingserver/