An IT Appliance for Remote Collaborative Review of Mechanisms of Injury to Children in Motor Vehicle Crashes

M. Kumar, R. Newman and J. Fortes
Center for Autonomic Computing
University of Florida
Gainesville, FL USA
{makumar, nemo}@cise.ufl.edu, fortes@ufl.edu

D. Durbin and F. Winston
Center for Child Injury Prevention Studies
The Children's Hospital of Philadelphia
Philadelphia, PA USA
durbind@email.chop.edu, flaura@mail.med.upenn.edu

Abstract*— This paper describes the architecture and implementation of a Java-based appliance for collaborative review of crashes involving injured children in order to determine mechanisms of injury. The multidisciplinary expertise needed for such reviews is not available at any one institution, resulting in the need for remote collaboration, while the sensitive nature of the information requires secure transmission and controlled access of data. The intended users of the appliance are researchers, engineers, medical doctors, government regulators, automobile and restraint manufacturers, insurance company representatives, and others who are interested in understanding the types and causes of injuries to children involved in motor vehicle crashes. The ultimate goal is to devise engineering solutions that prevent similar injuries from occurring in the future. The collaboration appliance (called Telecenter) enables the following activities: (1) the distributed asynchronous collection of digital content needed for each crash case review under a scheme that consistently organizes content across multiple cases; (2) the secure, Web-based remote participation of users in case-review meetings that involve viewing of case-specific content, live communication (written or verbal), multimedia access and sharing (slide presentations/ images), and use of Web resources; and (3) archival and post-review access of case reviews for follow-up activities and other functions (e.g., statistics, search, and networking). The Telecenter design supports audio conferencing, remote delivery and viewing of slide presentations, and other collaboration features also available in commercial and public-domain collaboration middleware products. However, it goes beyond existing solutions by also embedding a specific workflow and content organization suited for traffic injury reviews, supporting spatio-temporal role-based access control, distributed management of content and seamless integration of existing services. The current status and experience from using an early prototype of the Telecenter in actual case reviews are discussed, along with planned extensions to its functionality.

Keywords-CSCW, spatio-temporal access control, collaborative computing, IT appliance, Telecenter.

I. INTRODUCTION

It is increasingly common for the solutions of complex engineering and societal problems to require contributions and collaborations of experts from different domains of science, business, industry and government. Information Technology (IT) can play an important enabling role in these collaborative processes. In particular, it can provide the necessary tools for (1) collecting, sharing, storing and monitoring information, (2) enabling communication among experts, and (3) providing controlled access to necessary data and resources. However, the use of these IT capabilities can be hindered by the need for every expert to master and use different IT tools and for cumbersome user management solutions that enforce access control policies and coordinate multiple participant contributions. To address the above goals and challenge, this paper describes an IT appliance that enables in an integrated user friendly manner all the activities needed by a diverse community of experts to collaboratively address an important public-health problem.

With advances in audio and video communication, Web interfaces and Internet speeds, Computer Supported Cooperative Work (CSCW) has emerged as a potential enabler of activities otherwise requiring traditional meetings and conferences. Collaborative IT appliances are set to replace the traditional notion of groupware, which mainly incorporates asynchronous communications such as email, discussion lists, and data sharing. More importantly, collaborative environments can enable functionalities and efficiencies that exceed those possible through conventional in-person meetings. For example, they can support both synchronous and asynchronous remote communication, relaxing the requirement for spatio-temporal collocation of participants and potentially leading to greater efficiency and cost savings. As another example, CSCW can allow participants to have different levels of access to shared information and activities. This paper describes an IT appliance that seeks to exploit these and other advantages of CSCW to enable all the activities needed for multiple experts to prepare, analyze, and archive joint reviews of injuries to children due to traffic crashes.

Commercial and open-source products are already available to support voice and video teleconferencing, document sharing, screen sharing, and other general purpose collaboration tasks. Commercial examples include *Citrix's Goto Meeting, Cisco's Webex, Skype* and *GoogleTalk* [4, 5, 6]. Typically these services provide some free capabilities (e.g.,

Digital Object Identifier: 10.4108/ICST.COLLABORATECOM2009.8367
http://dx.doi.org/10.4108/ICST.COLLABORATECOM2009.8367

free computer-to-computer calls, teleconference with limited number of participants, and document sharing) as well as paid premium services (e.g., certain phone calls, many-participant teleconferencing, and slide-show management). Open-source collaboration and free services include Spicebird, Zimbra, Telepathy and Mindtouch [12, 13, 14, 15]. Audio conferencing using softphones is one example of real-time synchronous communication in a collaborative appliance. Some other common features may include shared editors, whiteboards, PowerPoint presentation and access control. However, there aren't existing solutions that integrate all these components together with a process to enter and organize information into a single easy-to-manage IT appliance. Our aim in this paper is to address this issue of component integration and to design a collaborative appliance that enables participants to share data, conduct audio and video conferencing, and show PowerPoint presentations with constraints based on the access policies defined in the system.

We describe our design of a collaborative appliance for the analysis and review of traffic-related accidents and injuries, and how could they possibly be avoided. This is an important problem because traffic-related crashes and injuries are one of the major causes of deaths in US and involve billions of dollars in losses. In the US alone, the average annual medical expenditures involving traffic accidents exceed 164 billion dollars [8]. The ultimate goal and challenge is to devise engineering solutions that prevent similar injuries from occurring in the future. However, the multidisciplinary expertise needed to review the circumstances of each crash is not available at any one institution, resulting in the need for remote collaboration. This is the motivation for our appliance for which the intended users are researchers, engineers, medical doctors, government regulators, automobile and restraint manufacturers, insurance company representatives, and others who are interested in understanding the types and causes of injuries to children involved in motor vehicle crashes. An example of an intended use of the appliance is as follows.

Example 1 The National Highway Traffic Safety Administration (NHTSA) under the US Department of Transportation has several research facilities which collectively form the Crash Injury Research & Engineering Network (CIREN) [7]. The goal of CIREN is to improve the prevention, treatment and rehabilitation of motor vehicle-related crash injuries and to minimize the associated number of deaths and overall economic cost. CIREN involves a collaboration of clinicians, engineers and other traffic safety researchers from academia, industry and government inside and outside the USA. CIREN-members often need to hold monthly meetings, ideally with the members of NHTSA, in order to study, analyze and discuss the injury cases. In addition to this, members from the restraint design and vehicle safety industry are also invited as this can help them better understand the performance of, and improve upon, the existing restraint and safety systems. But, often due to the need for simultaneous presence of everyone in a specific meeting location, participation is often limited and inconsistent. There is a clear need for a collaborative platform that could enable individuals to participate and/or contribute to the meeting activities remotely, possibly at different times, in a secure manner. Different participants should not be allowed to see certain pieces of information as they may contain private personal data about car crash victims or legal data that cannot be divulged. This adds dynamic access control requirements that also need to be supported by a collaborative tool. Moreover, it is desirable for CIREN to leverage itself as a network, i.e. making it possible for accumulated experience from multiple crash reviews and expert input from many participants to be used in future reviews. A desirable collaborative tool should therefore provide easy-to-search case-review archives that can be used to extract expert networks and content that is relevant to future case reviews. Each case requires a preparation phase during which information is culled from several databases and paper records from hospitals, police and other sources. A useful collaborative platform should therefore enable different people to access, store, organize and archive the necessary materials for review during meetings and potential retrieval at later times. The ideal collaborative appliance would be one where all the work needed to prepare, conduct, participate, save and revisit case reviews could be done by any legitimate participant from anywhere at any time.

The rest of the paper is organized as follows. In Section II, we provide some background and related work on collaborative appliances and access control. Section III explains the basic requirements of a good collaborative appliance for the purpose of child injury case reviews. Section IV describes the design and implementation of the Telecenter. Finally, Section V concludes the paper and comments on future work in this area.

II. Background & Related Work

A. Computer Supported Cooperative Work

Since the term CSCW was first coined by Irene Greif and Paul M. Cashman in 1984 [16], much work has been done and extensive agreement has been reached on the desirability of the following features of a CSCW system: support for controlled information sharing, message exchange, bi-directional synchronous and asynchronous communication, transparency in using existing services, and easy content management. It should also be intuitive and easy by its intended users. The spatiotemporal dimensions of a CSCW system are well captured in Figure 1 from [19]. The figure illustrates the multiple possible modes of cooperation and the variety of tools that could be used to support them.

The existence of a growing number of collaborative tools and services makes it possible to consider the development of domain-specific collaborative IT appliances that re-use such tools. However, one of the challenges to be faced is to hide from the end users the details and idiosyncrasies of individual tools. Ideally, the IT appliance should appear as a single entity with a homogenous interface and a consistent "look-and-feel" for all the capabilities offered to its users. When this is not possible in practice, the right balance must be achieved between tool transparency and tool re-use flexibility.

Digital Object Identifier: 10.4108/ICST.COLLABORATECOM2009.8367
http://dx.doi.org/10.4108/ICST.COLLABORATECOM2009.8367

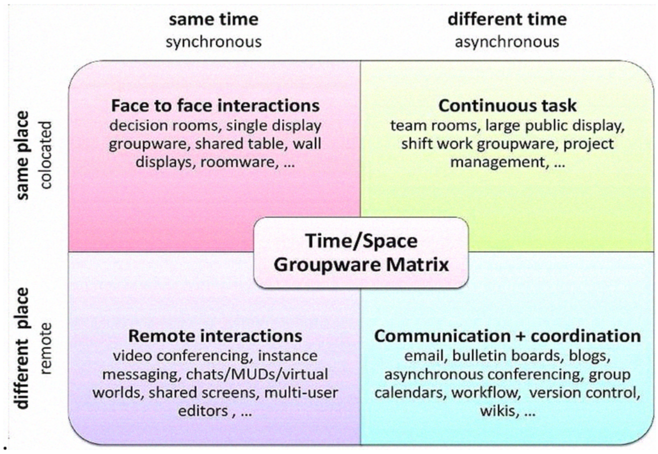

Figure 1. CSCW matrix [19].

B. *Role-Based Access Control(RBAC)*

An important instance of the transparency vs. flexibility tradeoff is access control in a CSCW system. Typically, individual tools require some form of user authentication (i.e. "sign-on") in order to control access to resources, data or other users. When multiple tools are integrated into a single CSCW system, it is desirable to provide a single sign-on capability; i.e., after initial authentication by a user, the system takes care of additional authentication steps when different tools are used. A possible approach to the provisioning of a single sign-on capability is discussed in [20] in the context of role-based access control (RBAC) [2].

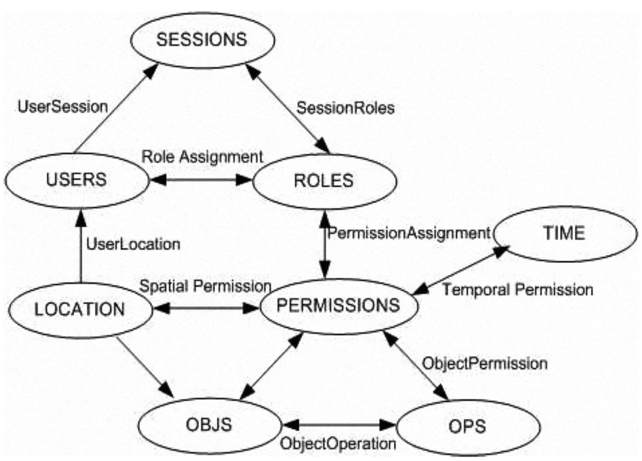

Figure 2. Spatio-temporal role-based access control model.

RBAC has emerged as a promising alternative to traditional mandatory access control [1] and discretionary access control lists. Instead of giving permission to individual users, permissions are assigned to roles, and in turn users are assigned to roles. A user acquires the permissions associated with the role he/she is currently assigned. Therefore, the resource access policies in the system are easy to define, manage, and extend in an RBAC system. Spatio-Temporal RBAC (STRBAC) [3], as the name suggests, is an extension of traditional RBAC. In STRBAC, the permissions are not only dependent on the object and the associated operation, but also on time of access and the current location of the user. STRBAC defines a hierarchical model for logical location and time in which it is easy to define most temporal or spatial constraints. Figure 2 shows the relationship between different components of a STRBAC system. STRBAC is an example of a feature that is not available in existing CSCW components.

III. Requirements of a Collaborative Appliance for Case Reviews

The term "Collaborative Appliance" is used in this paper to denote an IT application that is self-contained to the extent that it can run anywhere and has all the functionality needed to enable a user to engage in a collaborative activity over the Internet. In practice, in our implementation, it is a client-side Java application which can execute on any Java-enabled platform. In general, an IT appliance can also leverage and integrate services or applications in addition to providing unique capabilities independently or "on top" of the existing ones. The goal of the Telecenter is not to design everything from scratch, but to facilitate users to use these products and services in a transparent and intuitive way. Some of the basic requirements of our appliance are discussed in the following subsections.

A. *Communication Capabilities*

Communication is the most important feature of any collaborative appliance. People or team members need to communicate with each other and discuss their ideas to reach a consensus or come up with a solution to an existing problem. In a collaborative environment, communication can be divided into two major categories: *synchronous* and *asynchronous* communication. Synchronous communication refers to any real-time communication taking place between two parties. Examples include audio and video conferencing and chat rooms. Audio conferencing is particularly important since it enables members to conduct online meetings and share their ideas. In our appliance, researchers from automobile companies, doctors and highway designers attend case review meeting where they discuss one case using slide-show and radiology images. Audio conferencing provides them a medium to communicate to each other and discuss those cases to reach a conclusion. Video conferencing is still catching up with other technologies since it demands high network bandwidth and CPU processing time. It may not be as important as audio conferencing since members do not need to see each other in order to conduct a meeting. It may be of use in certain cases such as giving a Webinar or showing a video clip related to the accident or injury.

A good collaborative appliance should also be able to support common asynchronous communication tools like calendaring, emails, voicemail, discussion lists, etc. Asynchronous communication is equally important as compared to synchronous communication since audio and video data have limited capability in sharing precise technical information. Calendaring support is useful in keeping track of the case review meeting time and important dates. Also, to keep a track of useful information and discussions, a

Digital Object Identifier: 10.4108/ICST.COLLABORATECOM2009.8367
http://dx.doi.org/10.4108/ICST.COLLABORATECOM2009.8367

collaborative appliance needs a scalable and easy to use discussion list and email services. SMS is another way to communicate with members since cell phones have become increasingly common these days. This may be of use in scenarios where the required member is not present in the online meeting, or to send an emergency message to one or more members.

B. *Presentation & Slide Show*

Audio conferencing is a great tool to conduct online meetings, but audio information is often imprecise and sometimes difficult to understand. Presentation and slide show capabilities enable users to share any common textual and visual data with high precision. In a usual case review meeting, the members share many radiology images to identify the damage and impact of injury on the body. Slide shows provide a way to share these images with other members and at the same time control what images can be shown to the members because of privacy issues. The Telecenter needs to provide two separate slide show screens to give more flexibility to share text and images. One screen might be used to control the textual presentation while the other can be used to control the shared radiology and accident related images.

Sometimes, the viewers might perceive a delay in transition of slides because of low bandwidth or high resolution images. In such a case, it is desirable to reduce the quality of images using compression to have a better viewing experience and less transitional delay.

C. *Content Management*

In most collaborative environments, the system needs to keep track of the information shared during the meeting. In other cases, every collaborative appliance has a particular application associated with it. The appliance should be able to archive and manage data in an organized way so that it can be accessed from anywhere by users across the globe having proper rights. Content management systems should be able to interact with the access control module to restrict access to information only to authorized users. This can be done in two ways: (1) each object has a label associated with it that corresponds to an access control list, and (2) every time any object is accessed, the system should check if the user is allowed to access it or not. The advantage of the first one is that objects can have restricted access even when the user is offline.

Databases are one of the best ways to store and manage small size textual and binary data. Bigger objects should not be stored in databases as it can reduce the performance of the database server. Bigger objects can be stored on the file system and the database can be used to store the path to the file. This helps in faster queries and less overhead on the database server. The cases are organized in different tables. The records across different tables are interdependent. For example, the damaged vehicle information for a case can be stored in a table named "Vehicle" while the occupant information can be stored in another table named "Occupant". Hence, one vehicle record and occupant record correspond to the same case. A careful database schema design with foreign key constraints is required to reduce size of records and maintain consistency for each case.

As mentioned earlier, each case consists of several radiology and accident scene images. Hence, the overall size of a particular case can become large. In such a scenario, client side persistent caching should be provided to minimize the delay each time a case is accessed. However, it should also make sure that the data can no longer be accessed from the local cache if the permission has expired for that object. There are two ways to prevent this; (1) all cached objects on the client side are encrypted and the keys to decrypt them are only available through the server. (2) Steganographically embed spatio-temporal access control information in the object itself so that every time the object is accessed, the permissions are checked using the stego data inside the object. For the second option, we have to make sure that the object cannot be opened using any generic application and the algorithm to decode the object is unknown to the user. The second option is harder to implement and requires more CPU processing.

D. *Access Control*

In order to provide a mechanism for finer granularity to controlled access to the resources, a collaborative appliance needs an access control module, which would help in defining policies, roles, and permissions to objects in the systems and provide restricted access to them. For example, before viewing a particular resource such as radiology image related to an occupant injury, the system should check if the role associated with the user viewing the image has proper access rights or not. Since the Telecenter spans across multiple domains, a role-based access control system would be desirable to provide flexibility. Each of the users can be assigned roles based on their profession and expertise.

Usually, in a case review meeting, some experts are invited as guests who require access to particular resources during the meeting. To solve this issue, the Telecenter provides a user-friendly system that enables users with required privileges to create guests and assign roles for a short period of time. These roles have limited time access and expire within a certain interval. To have a time constraint, the Telecenter requires permission assignment with location and temporal constraints in addition to the generic access rights. The permission tuple consists of (object, operation, start time, end-time, location). The access control system should also incorporate dynamic role activation and temporary permission delegation to other users.

E. *Graphical User Interface (GUI)*

GUI plays an important role in the usability of any appliance. Since case reviews involve collaborators from diverse areas, which are often other than computer engineering; they may not be comfortable using the system if it is not user-friendly and intuitive. An appliance should be designed in such a way that any person with a decent knowledge of computers can use it with little or no help.

Digital Object Identifier: 10.4108/ICST.COLLABORATECOM2009.8367
http://dx.doi.org/10.4108/ICST.COLLABORATECOM2009.8367

As mentioned earlier, a case can contain several radiology images and other vehicle and scene images. Since the space in the GUI is limited and can only show the smaller version of these images, the GUI must provide the users to view them at their original size when needed. Users should be able to scroll over, or zoom in and zoom out these images at their convenience. The GUI should also place all the case related data in an organized way so that users can access them without any trouble. The GUI should be able to dynamically adjust its size according to the user's computer screen size.

IV. PROPOSED DESIGN & IMPLEMENTATION

The Telecenter is a purely Java-based platform-independent appliance. It is based on a typical client-server model. The client side is a fat client, meaning that it can perform many functions without being dependent on the server. It has an embedded Java SoftPhone, encryption/decryption modules, caches for case-data retrieval, two VNC clients and all the GUI windows needed by all the functions supported by the Telecenter. The advantage of using this fat client model includes (1) reduced dependence on the server, (2) low network bandwidth requirements, (3) faster response time, and (4) more flexibility and independence regardless of server design. Since case data are also downloaded to the user machine, the use of a thick client also enables asynchronous activities, some of which can take place even if the client is disconnected from the server.

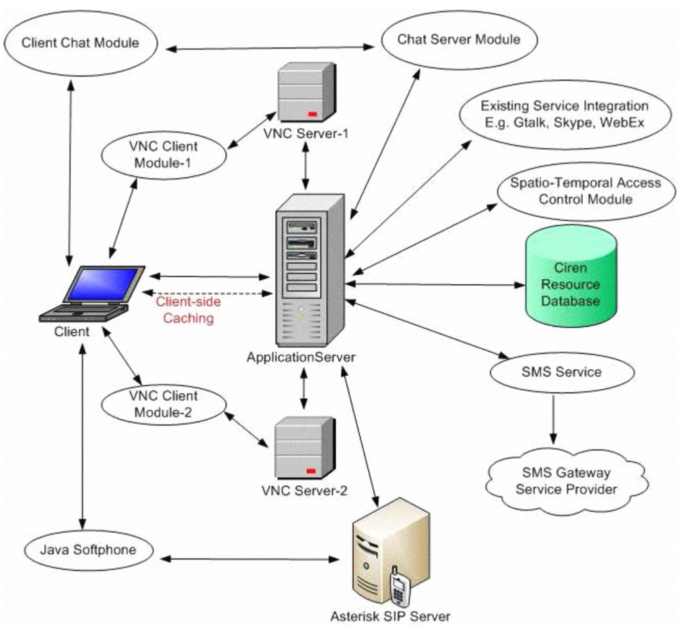

Figure 3. System-oriented view of the Telecenter .

The client is downloaded to the user's computer for its first usage and remains cached for the future uses. We use Java Web Start (JWS), which caches the application on to the client's local disk. Also, JWS-based clients do not require any installation or setup and can run with a single click of the mouse. The users only need to go to the required webpage and click on a link to launch the application. The only requirement is that Java has to be installed on to the local computer.

The server side of the Telecenter is web-service oriented, meaning that any client request or server response uses a standard HTTP packet. The server provides JAX-WS API (Java API for XML Web Services), which allows clients to communicate with the server using standard XML-based message. The advantage of using JAX-WS is that it is very easy to implement and developers do not have to worry about creating threads, preventing deadlocks and maintaining synchronization. Another advantage is that it offers flexibility to the developers to write message-oriented as well as RPC-oriented web services. To deploy these web services, the Telecenter uses the Glassfish application server.

Figure 3 shows the main components of the Telecenter system while Figure 4 shows its service-oriented view. The application server is the main entry point for any incoming client request or outgoing server response. It also interacts with the database server in order to store case-review records and other user-related information. The database server can either run on the same machine or on a different one. We use MySql server as our database since it is freely available and has a record of high reliability and good performance. The system-oriented view shows the interaction between different modules while the service-oriented view shows the web services available to the clients. A single sign-on capability is needed for the users to access these services. These services along with their dependent modules are discussed in the following subsections.

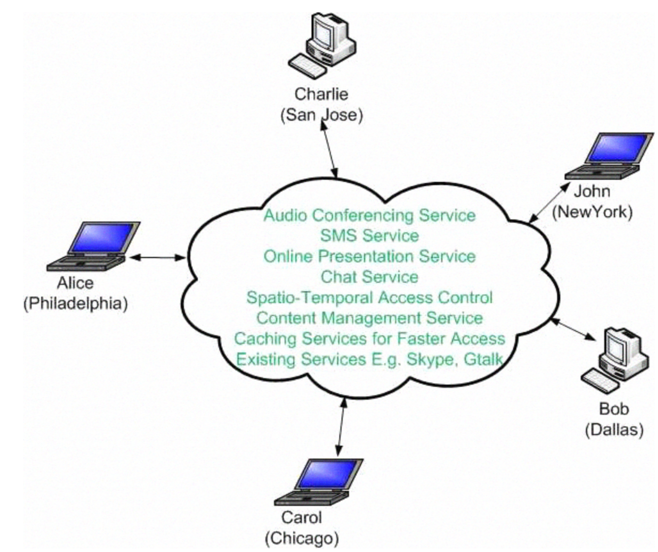

Figure 4. Service-oriented view of the Telecenter.

A. *Audio Conferencing Service*

In order to discuss injury cases during case review meetings, the Telecenter provides an audio conferencing platform where users can join the conference with a single click and present their ideas and views. Client-server audio communication is done using standard SIP and RTP protocols. Our appliance integrates a highly professional and powerful open source PBX telephony engine called Asterisk [9], which acts as a SIP server for all the clients connecting to it using the inbuilt SoftPhone.

Digital Object Identifier: 10.4108/ICST.COLLABORATECOM2009.8367
http://dx.doi.org/10.4108/ICST.COLLABORATECOM2009.8367

Since Asterisk doesn't have built-in support for conferencing, we use an external library to enable voice mixing and conferencing support. The client side uses SipCommunicator, which is a Java based softphone [10]. SipCommunicator can be integrated with any client-side Java program after minor modifications, and can be used to communicate with any SIP server to make calls or join conferences. Since both the client and server are behind a NAT (Network Address Translator) in most cases, we enable support for the softphone and Asterisk server to work across machines behind a NAT.

Figure 3 shows the relationship between an Asterisk server and a SoftPhone. There is a two way arrow to indicate that they exchange voice and control information with each other. There is also a relationship between the Asterisk server and the application server. The application server accepts a request from the client and validates it before passing it on to the Asterisk server. The application server also sends periodic updates about the status of the client to Asterisk. If the client is logging off, the application server sends a message to the Asterisk server to remove it from the list of active conference users. Also, during the conference joining process, the application server is responsible for sending the request to the Asterisk server for details about which conference to join.

B. *SMS Service*

The SMS capability was included in the list of services so that a member could send important messages to another member's cell phone if they cannot be reached via the Telecenter. A typical scenario would be that a case review meeting is scheduled to take place in a few minutes and one of the members required to be present is not logged into the system. In such a case, any member can send an SMS notification from the Telecenter to the missing member reminding him/her about the meeting time and schedule. Ad-hoc invitations and asynchronous messaging would also be possible. In a nutshell, this capability enables users to send emergency notifications or messages to one or more users. To use this feature, the recipient must have a valid phone number in his/her profile. The SMS module uses *Clickatell SMS* API [17] to deliver the message to the recipient's phone. A confirmation message will be delivered to the sender indicating whether the message was successfully delivered or not. Figure 3 shows the relationship between the SMS components. The details about the message including the sender, receiver, date, time and the message contents are stored on the database for logging purposes.

C. *Online Presentation Service*

Usually, at a case-review meeting, a summary of the case is presented by the crash-investigation team in a PowerPoint format with textual summaries, as well as the photographs of the crash sites and the related radiology images of the injured patients. The Telecenter provides a VNC client-server platform through which a member can control a PowerPoint presentation for simultaneous viewing by other participants on their computer screens. The integrated audio-conferencing service can be used by the presenter to explain the slides and by other members to ask questions.

The Telecenter uses TightVNC, a freeware Java-based VNC server and client [11]. The VNC client is embedded in the client application itself, so users don't have to download any separate software to make it work. Figure 3 shows two VNC servers namely "VNC Server-1" and "VNC Server-2". Each of these VNC servers can be used to start a presentation or to display images related to an injury. The two way arrow between the VNC server and the application server in figure 3 indicates the exchange of control information between the two servers. The control information can include the name of the user, password and the PowerPoint filename. It also checks if any other presentation is currently in progress. For example, if user A has started a presentation on VNC server-1, other users can only view the presentation on VNC server-1. They will not be allowed to start another presentation on the same machine unless user A terminates the presentation or logs off. However, users can still start another presentation on VNC server-2 if there is no current presentation. This ensures consistency and prevents users from accidentally terminating another user's ongoing presentation.

The server side of VNC can either be installed on a Linux or a Windows platform. We chose to install the VNC server on a Windows platform because Linux only supports OpenOffice for PowerPoint presentation. Therefore, some slides cannot be displayed in the correct format using linux-based systems, especially the ones in the .pptx format. Each of the Windows machines running the VNC server is equipped with Microsoft Office suite and a Glassfish application server. This application server is responsible for the exchange of control information with the central application server as mentioned above. It also deals with the task of starting and terminating the slideshow on the VNC server and downloading the PowerPoint file from the client machine.

Figure 3 also shows arrows between the VNC server and the VNC client module. After the exchange of control information, the VNC client module can initialize and connect to the VNC server directly to receive the screen updates and pass keyboard and mouse activity back to the VNC server. The presentation is shown in a separate window and can be resized if needed. It can be started in the "view mode" or "edit mode" depending on whether the client is the owner or not. By default, the presenter or the one who uploads the file is the owner and therefore can control the slide transition on the VNC server. However ownership can also be delegated to other users dynamically.

D. *Spatio-Temporal Role-Based Access Control(STRBAC)*

Access control is one of the most important modules in the Telecenter. At the time of this writing, it has been designed and implemented but not yet deployed in the Telecenter prototype. Traffic related accident and injury information is highly sensitive and, therefore a secure and careful access control design is needed in order to prevent unauthorized access to the system. In addition to the generic role-based access control mechanism, the Telecenter supports spatial and temporal restrictions, which limit a particular role to access a given resource only from specified locations and during

Digital Object Identifier: 10.4108/ICST.COLLABORATECOM2009.8367
http://dx.doi.org/10.4108/ICST.COLLABORATECOM2009.8367

specified time intervals. Any command or task going to the application server has to pass through the STRBAC module to authorize itself before going further. A user requesting a task has to send its tuple {role, password, object, action} to the STRBAC module using a standard encryption protocol.

The STRBAC module supports logical time and location definition, which can be customized according to a particular organization. For example, one company can define office hours to be from 8am to 5pm while another can define them to be from 7am to 4pm. Figure 5 shows the user interface to define logical time constraints. Temporal constraints can be complicated and long because they can involve multiple ANDs, ORs and NOTs. Hence, each of these expressions is given a name so that it is easy to remember and can be reused again if needed. The logical time definition panel guides the administrator through various steps to define a temporal constraint. Logical location can also be defined in a similar way. Currently, we have not included support for location determination but plan to do it in our future work. Approximate location of a user can be determined using the IP address of his/her local computer. There are other methods to determine location but these require extra hardware and authentication systems. Once the logical time and location have been defined, we can formulate spatio-temporal permissions and assign them to a given role. The Telecenter provides a friendly easy-to-use graphical interface to assign permissions as shown in Figure 6 for the case of users at the University of Florida.

The access control data and policies are stored in MySql database as objects with individual tables for users, roles, permissions, objects, etc. Each of the tables has foreign key constraints to maintain consistency and reduce redundant rows in the tables. The module also supports dynamic activation and revocation of permission if permission expires after a specified period of time. This is done by periodic checks of all the permissions held by a logged on user. The Telecenter client has a *keepAlive* thread that sends a request to the server every five seconds to confirm its user's identity and the permissions associated with him/her.

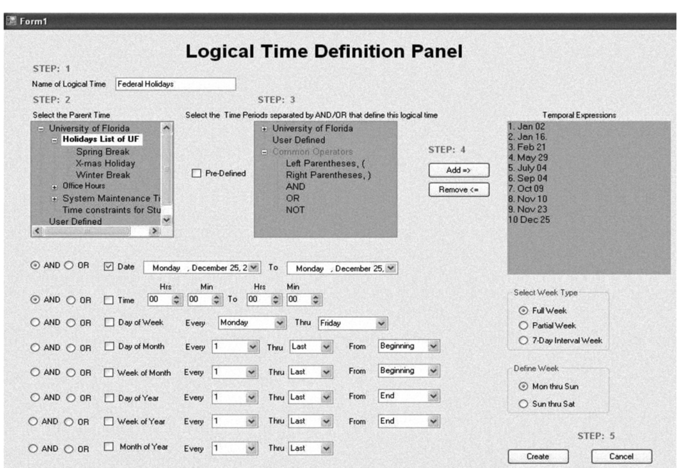

Figure 5. Temporal Constraint Definition Panel.

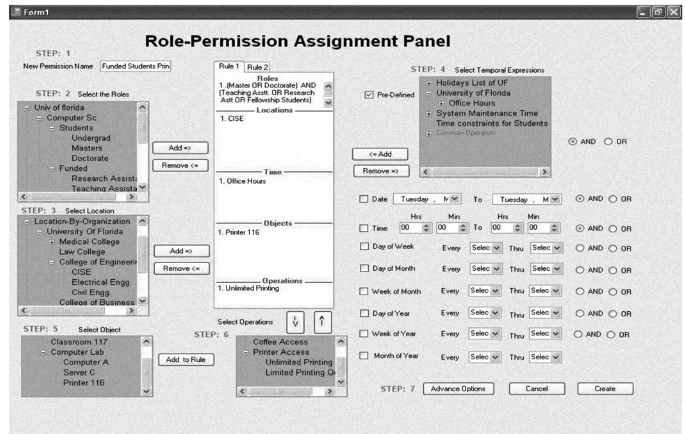

Figure 6. Role-Permission Assignment Panel.

E. Content Management

The Content Management module deals with the creation, archiving, updating and deletion of crash injury records. Any kind of access to this module requires proper authorization from the access control module as discussed above. The Telecenter uses the MySql database server to store and manage contents associated with the case. This database is also responsible for storing the access control data as discussed in the previous section. Since the Telecenter is mainly suitable for crash injury review, it can contain many radiology and crash scene images, each of which might be large in size. Therefore, we do not store the images in the database. Instead, we save the files on disk and store their locations in the database.

Since the data associated with a case can be large, we have implemented client-side persistent caching to reduce the access time to view any particular case. The Telecenter uses Java Caching System (JCS) which significantly reduces the latency time for accessing large data or objects from the server [18]. JCS can be implemented both on client as well as the server. We have implemented JCS on client side only since the number of clients is not large and hence the server is not overloaded. Without caching, the client has to wait several seconds or even minutes for the case-related data and images to be downloaded from the server to his or her local machine. This can be frustrating, especially when the clients want to switch back and forth between different cases. Caching stores the data on the client's machine when the case is accessed for the first time. Any future access to the same case will be minimal because of caching. Caching system also makes sure that the data is not stale on the local machine. Hence, before displaying the cached data, the client sends a query to the server to see if any update has been made since the last download. If there is an update to the case, the updated objects will be downloaded (and not the entire case) from the server and cached data will be replaced. Caching system also queries the access control module to see if the client still has proper permissions to access the resources that have been cached. If permission has been revoked or has expired, the client should

Digital Object Identifier: 10.4108/ICST.COLLABORATECOM2009.8367
http://dx.doi.org/10.4108/ICST.COLLABORATECOM2009.8367

not be able to access that resource. To implement this, each of the cached resources on the client machine is encrypted with a key known only to the access control module on the server. The resource is decrypted only when the access control system provides the key to the client machine.

F. User GUI

Figure 7 shows the implemented design of the user GUI of the Telecenter. It is mainly divided into three parts. The right side of the GUI deals with the list of active users and conference members as well as some useful resources. These resources include SMS capability, access to Google and CIREN websites, and access to "Expert Network." The expert network shows the relationship between different Telecenter members and gives information on whom to contact to address issues related to a case.

The middle part of the screen deals with the case review and its content management. In case-review mode, the user will typically magnify the case-review window so that it takes the full available screen (see Figure 8). Authorized users can add, edit, or delete cases in this section of the screen. Each case contains five panels, namely (1) Crash/Scene, (2) Vehicle, (3) Occupant/Injuries, (4) Injury Analysis, and (5) Presentation. The Crash/Scene panel displays the summary of the crash and the scene associated with the crash. It also shows some additional details such as date and time of crash, Injury Severity Score (ISS), Abbreviated Injury Scale (AIS), and the type of restraint system used. The Vehicle panel shows the details of the occupant's vehicle involved in the crash and the images associated with the damaged vehicle from different angles. It shows the year, make, and model of the vehicle, speed during crash, Collision Deformation Classification (CDC), and Principal Direction of Force (PDOF). The Occupant/Injuries panel shows information about the injured patient involved in the crash. It shows the age, sex, height, weight, position, posture, and type of restraint system used. It also shows the radiology images related to the injury. The medical researchers use these images to study the injuries and discuss with engineers ways to prevent these types of injuries. Injury Analysis panel deals with the analysis and final report associated with the injuries for a single occupant. Presentation panel is divided into two separate windows, each capable of starting and viewing one presentation. Users can view the ongoing presentations in the small windows or expand one of them to view it in full screen mode.

The left side deals with the communication capabilities of the Telecenter. It deals with audio conferencing and chat. Right now, users can only join one single conference. Multiple simultaneous conferences will be supported in our future work. The conference window also provides the capability to choose the right network interface and adjust microphone and speaker volume.

Pulldown menu selections provide several capabilities. These include ability to (1) add/delete users, (2) define and assign access control policies, (3) update profile and password, and (4) upload documents or presentations to the server.

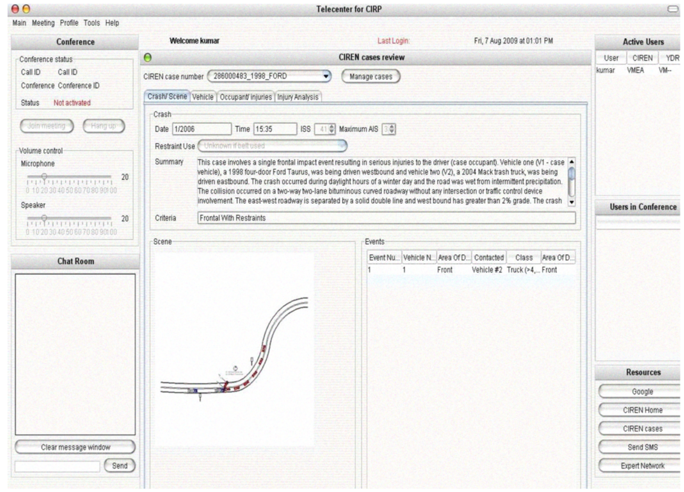

Figure 7. Client-side user interface for the Telecenter.

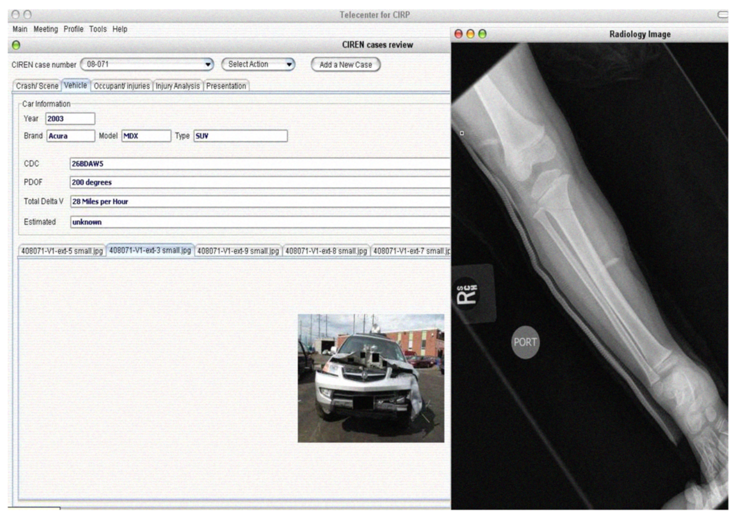

Figure 8. Telecenter screen in case-review mode showing textual and pictorial information about a case.

V. Conclusions & Future Work

The work in this paper describes the architecture and implementation of a collaborative appliance called Telecenter. In addition to the traditional and existing collaborative tools in the market, the Telecenter provides a customized solution for reviewing cases of injury in children due to vehicle crashes.

It incorporates many familiar tools so that users do not have to learn new tool interfaces, and combines the tools needed by the target users. These include: audio-conferencing, archival and management of injury cases, SMS capability, two simultaneous PowerPoint presentations, spatio-temporal role-based access control, controlled caching services, expert network overview and an inbuilt browser.

Digital Object Identifier: 10.4108/ICST.COLLABORATECOM2009.8367
http://dx.doi.org/10.4108/ICST.COLLABORATECOM2009.8367

The Telecenter prototype is already functional and has been used in experimental settings by a subset of its intended users. Meetings have been conducted using the Telecenter with participants in different states and in different continents. Based on initial feedback, the Telecenter is perceived as a significant improvement over current practices in terms of convenience, capabilities, efficiency and efficacy. Work is under way to increase the usability and robustness of the prototype in preparation for its deployment at several sites for routine use during future meetings.

In the future, we plan to extend the appliance by implementing a robust location-determination technique and integrating social networking services. We also plan to enable users to customize their view and choose components which they like to view in the Telecenter, enable whiteboards, discussion forums, shared text areas and support for multiple conferences. Finally, we plan on allowing an explicit representation of workflow so to enable the adaptation of our appliance to other contexts.

VI. ACKNOWLEDGEMENTS

The US National Science Foundation partially supports this work under grant numbers IIP 0925103, IIP 0940969 and IIP 0758596. We also acknowledge the support of the Bell-South Foundation and Shared University Research grants from IBM. We thank and credit Keyu Chen, a former graduate student of the ACIS lab, for contributions to the development of an early version of the interface of the Telecenter.

REFERENCES

[1] D. E. Bell and L. J. L. Padula, “Secure computer system:unified exposition and MULTICS”, Report ESD-TR-75-306, The MITRE Corporation, 1976.

[2] David F. Ferraiolo, Ravi Sandhu, Serban Gavrila, D. Richard Kuhn, and Ramaswamy Chandramouli, “Proposed NIST Standard for Role-Based Access Control”, ACM Transaction on Information and System Security, 4:224–274, 2001.

[3] Mahendra Kumar and R.E. Newman, “STRBAC-An approach towards spatio-temporal role-based access control”, red, 547(803):150.

[4] Cisco Webex http://www.webex.com/

[5]GoToMeeting https://www.gotomeeting.com

[6] B. Sat and B.W.Wah, “Analysis and evaluation of the Skype and Google-Talk VoIP systems”, In Proc. IEEE Int’l Conf. on Multimedia and Expo, July 2006.

[7] Jerome T Scally, Catherine A. McCullough, Louis J. Brown, and Rolf Eppinger, “Development of the crash injury research and engineering network”, International Journal of Trauma Nursing, Volume 5, Issue 4, October-December 1999.

[8] U.S. car accident cost http://money.cnn.com/2008/03/05/news/economy/AAA_study/

[9] Van Meggelen, J. Smith, and L. Madsen, “Asterisk™: the Future of Telephony”, 2nd Edition, Second. O'Reilly.

[10] SipCommunicator http://sip-communicator.org/

[11] TightVNC http://www.TightVNC.com/

[12]Spicebird http://www.spicebird.com/

[13]Zimbra http: http://www.zimbra.com/

[14]Telepathy http://telepathy.freedesktop.org/wiki/

[15]Mindtouch http://www.mindtouch.com/

[16] J. Grudin, “Computer-supported cooperative work: History and focus”, IEEE Computer, pp. 19–26, 1994.

[17] Clickatell Developer SMS API http://www.clickatell.com/developers/clickatell_api2.php

[18] Java Caching System (JCS) http://jakarta.apache.org/jcs/

[19] CSCW matrix http://commons.wikimedia.org/wiki/File:Cscwmatrix.jpg

[20] Sumalatha Adabala, Andréa Matsunaga, Maurício Tsugawa, Renato Figueiredo, José Fortes, “Single Sign-On in In-VIGO: Role-based Access via Delegation Mechanisms Using Short-lived User Identities”, In Proceedings of 18th International Parallel and Distributed Processing Symposium (IPDPS-2004), Santa Fe, New Mexico, p.22, 26-30 April, 2004

Digital Object Identifier: 10.4108/ICST.COLLABORATECOM2009.8367
http://dx.doi.org/10.4108/ICST.COLLABORATECOM2009.8367