INTERNATIONAL JOURNAL OF INTELLIGENT CONTROL AND SYSTEMS
VOL. 11, NO. 2, JUNE 2006, 114-123

WHISPER - Service Integrated Incident Management System

S. RAMASWAMY, M. ROGERS, A. D. CROCKETT, D. FEAKER, and M. CARTER

ABSTRACT: This paper presents a cohesive summary of existing emergency response systems. We investigate and integrate principles, theories, and practices from four diverse, yet related, fields of knowledge with respect to information representation and decision support capability requirements for emergency planning and response (EPR) systems. This enables the cooperation between constituent agencies (e.g., fire, police and medical) and surrounding municipalities which operate using assorted decision support protocols, system architectures, networking strategies and along different levels of data security needs. Based upon our investigation, we have built a service architectural framework for providing and disseminating an integrated platform of knowledge capable of being used as intelligent interconnects between distributed EPR systems. Such a framework can support affordable integration for municipalities of all sizes, in particular smaller municipalities that often cannot afford costly off-the-shelf software solutions consisting of proprietary logic and requiring extensive customization and support cost. We also present a prototype web service based implementation and summarize the limitations of such an approach.

Index: Emergency response system, emergency planning and response, emergency management, decision support, web service.

I. INTRODUCTION

Infrastructure elements for Emergency Response (ER) have had limited success in the area of coordinated response due to inadequate, insecure, and incompatible training tools and information technologies. Despite the advances in technologies, emergency response agencies have been unable to achieve a seamless integration of local, state, and/or federal information management systems to create a coherent interconnected network. The intent of WHISPER (Web-inspired, Hierarchical Integration of Secure and Private Emergency Resource (WHISPER) Knowledge Networks) is to develop and demonstrate a prototype framework for the fusion and dissemination of both real-time and stored information; development of knowledge-based system technologies; and integrating these within planning, scheduling and resource management strategies. Our research effort is an attempt to explore, evaluate and design mechanisms to add intelligence to create knowledge and support decision makers in planning for, and responding to, emergency situations. Most systems specification and design techniques available today do not effectively contend with the needs of such large-scale distributed systems. Hence WHISPER attempts to investigate different facets of information integration and fusion phenomena with the goal of fusing together information from a wide variety of heterogeneous sources in a way that unifies and distills the combined information. The following five core principles highlight the design philosophies that drive the architectural development of WHISPER: *balanced, yet open; collaborative, yet modular; controlled, yet evolutionary; synchronized, yet scalable; and user-centered, yet responsive.* These core principles provide a scalable solution that allows the emergency response agencies to respond effectively.

This paper is organized as follows: Section II provides an overview of various ERP systems and available techniques in various areas (planning and scheduling, artificial intelligence, distributed computing and web services) that can be leveraged for developing effective integration mechanisms. Section III provides a brief analysis of the problem scope and various functional needs. Section IV presents the design and development of the prototype implementation. Section V concludes the paper.

II. LITERATURE SURVEY

A. EPR Systems

A vast amount of research on planning, scheduling, dispatching, human/computer interfaces, distributed computing and artificial intelligence has been conducted over the past few decades. It would be very difficult to summarize all of this literature within the span of several pages. Consequently, we will focus primarily upon research related to our work and not delve into comparing and contrasting specific research works and/or solution frameworks.

Within the areas of planning and scheduling, Dai, et al. studies the combinatorial characteristics of the scheduling problem within emergency systems [12]. Goldberg, et al. investigate the problem of optimizing the location of emergency response vehicles subject to stochastic travel times, unequal vehicle utilizations, various call types, and service times that depend on call location [24]. Hatono, et al. proposes a cooperative scheduling protocol and dispatching algorithm for distributed real-time scheduling of emergency jobs [22]. Kitchen discusses urban terrorism and, in particular, the planning and response strategies adopted by the city of Manchester, England following a terrorist (bomb) attack in 1996 [63]. In [64-66], the authors address workflow design issues for EPR systems.

Received on March 1, 2006; revised on June 1, 2006.

S. Ramaswamy is with the Knowledge Enterprises for Scalable Resilient Infrastructures Laboratory in Department of Computer Science at the University of Arkansas at Little Rock. Email: srini@acm.org / srini@ieee.org. M. Rogers is with the Advanced Computing Experiences Laboratory in the Department of Computer Science at Tennessee Tech. Email: mrogers@csc.tntech.edu. A. D. Crockett is with the Angelo and Jennette Volpe Library at Tennessee Tech. David Feaker and Monty Carter are with Computer Science Department at Tennessee Tech.

Within the areas of artificial intelligence and information/data representation, Wilson, et al. use artificial intelligence, in the form of an expert system, to improve the quality of dispatch information in emergency situations involving dangerous materials [10]. Shively employs an ecological task analysis (ETA) approach to develop an improved GUI-based display for a police dispatch center [27]. Zografos, et al. study on-board computer technology and mobile data terminals and find them to be technically feasible and operationally effective [34].

Within the area of emergency dispatching, Weintraub, et al. study the problem of assigning and routing emergency repair service vehicles for a Chilean electric company using heuristic algorithms [2]. Cameron and Schultz evaluate GPS (global positioning system) and GIS (geographic information system) technologies as applied to emergency dispatching and location [41]. Rhodenizer, et al. utilizes data from real emergencies to study the breakdowns in communication between dispatchers and policies to develop improved strategies [40]. Nathanail proposes a methodology to evaluate alternative dispatching policies for emergency vehicles [54]. Shibuya, et al. studies an emergency vehicle preemption system using two-way communication functionality based on an infrared beacon to re-route vehicles [52].

Several EPR software systems are commercially available, such as Blue292 (www.blue292.com), EM2000 (www.cx2tech.com), Incident Master (www.essential-technologies.com), LEADERS, WebEOC (www.esi911.com) and RESPONSE (www.earenfroe.com). A government study [1] finds that there is no “best” system and no “perfect” fit; rather the best product for a particular agency is based upon considerations such as budget, system environment, scale and sophistication of operation, and discipline to implement. However, when multiple agencies and municipalities are involved, the problem becomes much more complex due to interoperability issues between disparate system architectures. Extensive customization would be necessary to solve such interoperability problems. In addition, off-the-shelf solutions rely on proprietary intelligence that, in turn, renders their underlying logic as secretive and hence not readily “inter-operable” for addressing needs at another agency.

B. Literature from Other Related Applications

In addition to the above papers that deal specifically with EPR system applications of theory, many useful papers focus on general-purpose applications. These come from research in the areas of distributed computing and artificial intelligence.

B.1. Related Distributed Computing Research

Within the general area of distributed computing, various papers focus on agent-based techniques for resolving resource-based conflicts [8,14,15,25,26,28,29, 30,31,32,47,51,55]. Such conflicts would be expected to occur in complex EPR systems. Additionally, various papers focus on distributed databases [21,44] which have similar data transportation and consistency needs in comparison to our framework. Additional distributed computing papers on data transport and sharing include message passing schemes [18,19], Remote Procedure Calls and Remote Method Invocation [5,20,21,38,42,56], and distributed shared memory [3,11,17,36,48,49,50,53]. Likewise, the work on distributed file systems [6,7,35,37,43] mirror the needs in our framework for distributed cache management. Also, papers on systems that use the Object Exchange Model (OEM) [57-62] are similar in that we are using the OEM format to combine heterogeneous information sources. OEM serves as the basic data model in numerous projects of the Stanford University Database Group, including Tsimmis, Lore, and C^3.

B.2. Related Artificial Intelligence Research

Within the general area of artificial intelligence, Yen discusses distributed artificial intelligence as a new approach to solve scheduling problems by using a set of scheduling systems that mutually interact using agents [4]. Other papers focus on adaptive learning concepts [9,13, 16,23,33,39], several of which are discussed in conjunction with agent-based techniques. This knowledge may be useful to embed as logic within EPR systems. Within the general area of scheduling and planning, Richard, et al. employ Petri nets as an approach to generalize the formulation of the scheduling problem [46]. Prosser and Buchanan survey the evolution of intelligent scheduling systems and comment upon future directions [45].

C. Literature Survey Summary

In conclusion, various research papers have applied theoretical concepts from the fields of scheduling, distributed computing and artificial intelligence to emergency planning/response systems and homeland security issues. Most of these efforts have taken the form of applying a specific concept (e.g., Petri nets) to a specific situation (e.g., chemical/biological weapon attack). However, no research was found that integrated knowledge across multiple fields into a single integrated platform capable of intelligently planning and dispatching emergency resources across multiple agencies and municipalities on a proactive and reactive basis. Thus, it would be unwieldy to use theory in its current format to facilitate the development of a broad-based EPR system capable of tying together disparate systems from various agencies and municipalities into a unified system for intelligent decision making. Moreover, existing off-the-shelf solutions for similar software systems require extensive customization to accommodate different agency and municipality system architectures and data security needs. To resolve these shortcomings, it is necessary to merge existing theory into a unified knowledge framework that can be used to design and develop a system "from the ground up" in conjunction with EPR agencies and third-party software systems companies.

Thus, WHISPER deals with planning, scheduling, dispatching, human/computer interfaces, distributed computing, artificial intelligence, and data integration. WHISPER is an effort to deal with the issues of data consistency and integration in a distributed computing environment.

III. PROBLEM SCOPE OF INVESTIGATION

It was noted that commercially available packages suffered from a common problem – lack of interoperability. Individual agencies have data stored in various formats on dissimilar architectures, and no mechanisms were in existence currently to link these diverse databases. That is, a software system that works for the police department may not work for the fire or ambulance center. Consequently, agencies have little initiative to either plan or to operate collaboratively. Another key finding was that most commercially available emergency management software (e.g., Blue292, WebEOC, etc.) is expensive, often prohibitively expensive for smaller counties and towns. It was observed that:

1. Existing research, principles, and practices in distributed computing can be leveraged to address how current systems involving multiple agencies and multiple counties can be integrated. Ideally, such a solution would permit retaining existing systems and tying their disparate architectures and security needs together using a common, higher layer of structure.
2. Existing research, principles, and practices in scheduling and its existing algorithms and heuristics can be applied to optimize the efficiency and effectiveness of emergency planning and response activities. After all, the problem an be reduced to scheduling in manufacturing environments: i.e. finding the best way to assign resources (view machines as emergency responders and equipment) to jobs (here - emergency events) on either a proactive or reactive basis, given that resource availability can be a hard or soft constraint (e.g., soft implies that a resource borrowing from adjacent municipalities becomes possible).
3. Existing research, principles, and practices in artificial intelligence can be applied to the intelligence required within (inter-agency and inter-municipality) emergency

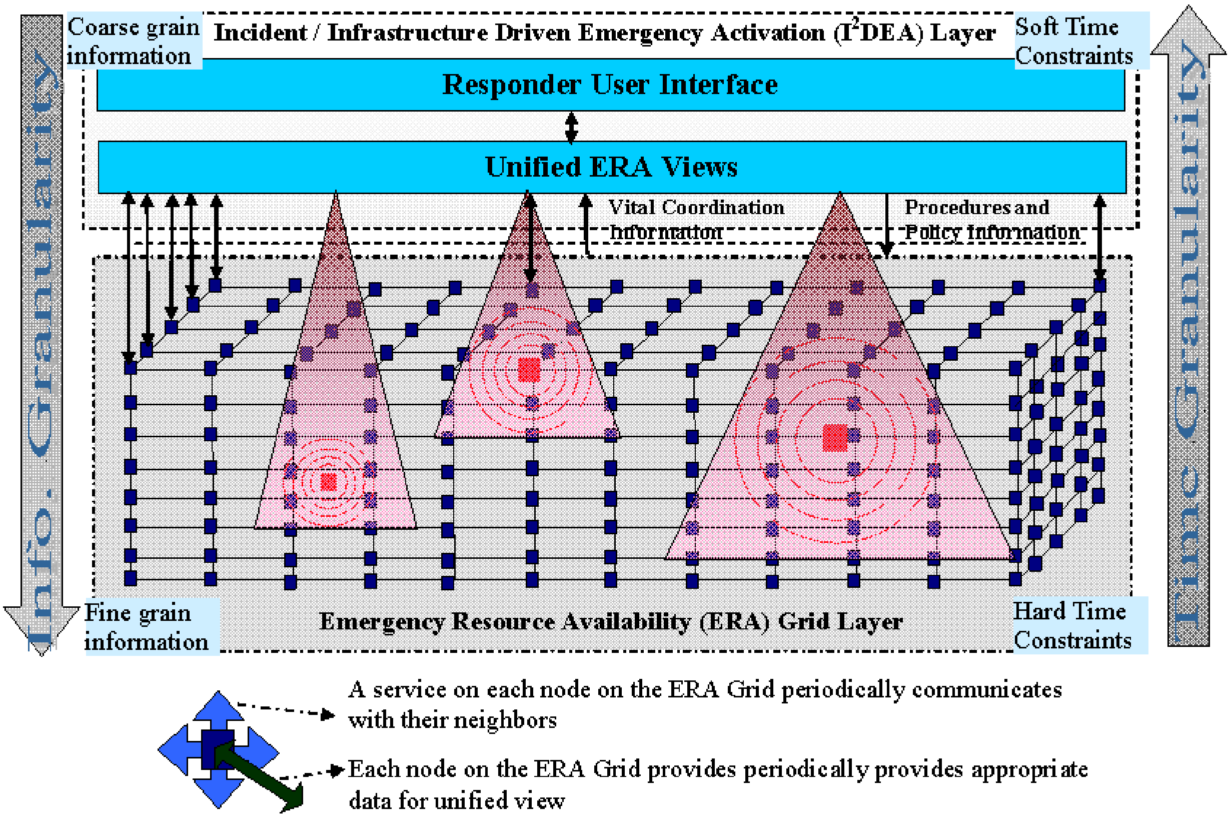


Figure 1. WHISPER Architecture

planning/response systems. In particular, these systems can be made to learn and to adapt while dynamically modifying their rule sets and logic along the way.

4. Existing research, principles, and practices in human-computer interaction can be applied to information visualization, interaction, and communication needs of emergency planning and response activities. In other words, personnel should be able to use the computer systems via an efficient user interface to monitor and evaluate the emergency response activities, and use the interface to communicate emergency planning decisions among agencies.

With the above observations, we have prototyped a service-integrated system with an emerging technology - Web Services. The key focus is the design of an efficient and expandable system architecture. Efficient system architecture comprises two design issues; *(i)* database design and, *(ii)* optimal extraction of unified data from the databases. Prototype data necessary for three emergency services (fire, police and E911) have been collected from their respective sources.

IV. DESIGN CHALLENGES

The general system structure is presented in Figure 1. The challenge was to integrate the best-of-bread technologies into a culpable framework that can be effectively distributed across agencies. The proposed software architectural framework is represented in Figure 2. As shown, the local service layer consists of the following four sub-layers (not including the data repository) which interact with the local data repository of a participating agency. It also interfaces with the unified view layers to transport the information in a robust, secure, efficient, consistent, and timely manner. The technologies that we will use to build this layer of the WHISPER framework primarily come from the discipline of distributed computing. Table 1 summarizes the differences between two possible implementation options (Java servlets versus .NET) identified for the implementation.

A. Functional Requirements

WHSIPER is an ongoing research and development effort. Salient research and development challenges that WHISPER aspires to address include the following:

1. **Information sharing:** Current approaches to coordination between different agencies suffer from the problem that information on resource availability is neither current nor accurate; and critical issues that are known to one agency are not easily known to others involved in emergency incidents. Sharing information is the key component of the coordination needed to make life-saving decisions during emergency response activities. Shared information must be readily available, accurate, and timely. WHISPER will therefore develop technologies that enable the robust and efficient transportation of information from distributed sources that is necessary for sharing information among cooperating emergency response

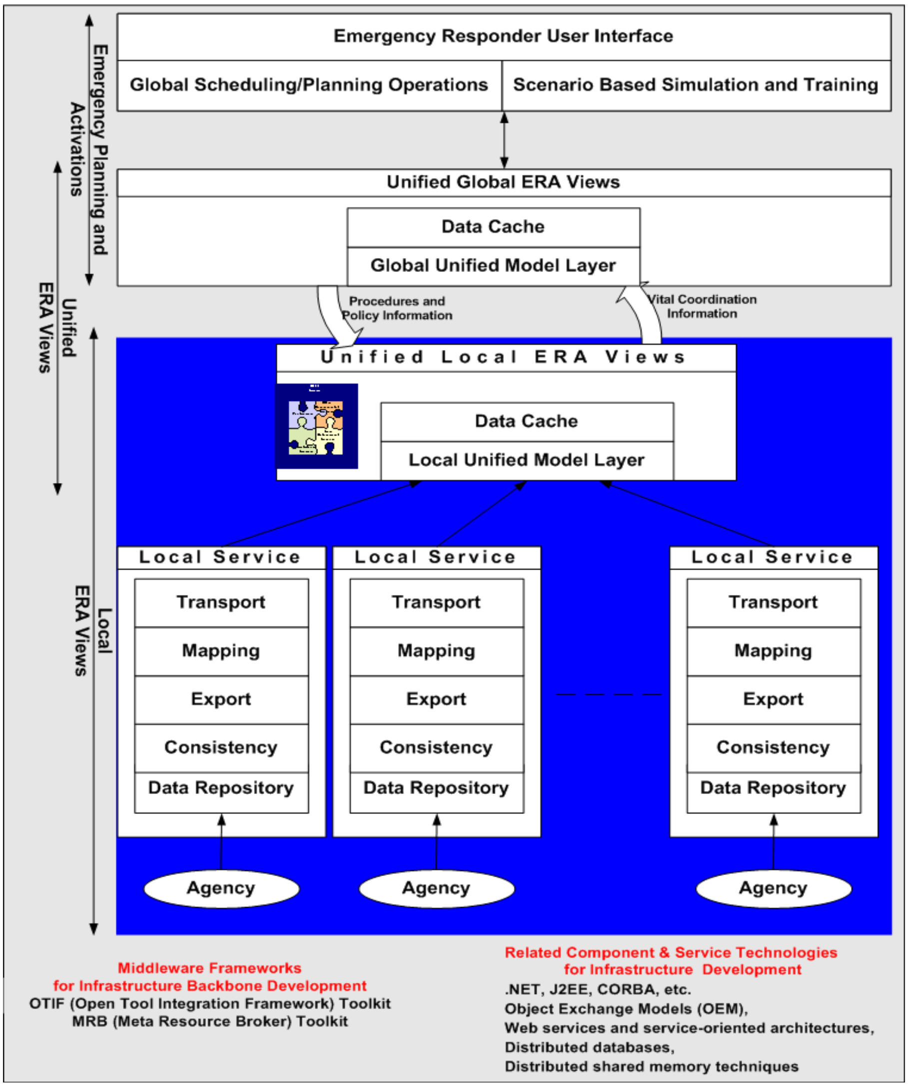


Figure 2. Interagency Coordination

agencies so that they may make the best possible decisions.

2. **Information security.** The need for security in distributed EPR systems is obvious, particularly in light of the sensitive nature of emergency response information. The security needs for each agency may vary widely, however, and determining the security capabilities based upon the interaction between agencies can be complex. This complexity may therefore cause agencies to implement security policies that are either too restrictive or too weak. Moreover, this complexity may affect the overall performance of the underlying software infrastructure since elaborate and intricate security checks must be done when accessing information. Our prior work in emergency response systems indicates that each individual agency will have unique informational requirements based upon when that agency needs to see the information and at what detail. Based on these requirements, we will devise automated and dynamic generation and publication of customized security proxies based upon the informational needs and dependencies among agencies to provide added security and improved performance of our security mechanisms, and the proxies will be able to provide the appropriate security and improve performance by taking advantage of relaxed consistency policies when managing the unified views of information. We will investigate automated methods of disseminating information among the agencies based upon the agencies' informational requirements, including the detail of information and the temporal constraints needed for timely delivery.

Table 1. Implementation Options

Criteria	Electronic Data Interchange (EDI) & Middleware	EDI & Web Services
Compatibility	Compatible only in developed environment	Highly compatible as accessed through browsers
Cost	Network cost, Connection cost, Middleware cost, Integration cost, Labor Cost	Internet connectivity cost, Labor Cost
Data Description	Low, because of high network cost incurred	High. Self descriptive
Distributive	Not highly distributive. Map times increases	Highly Distributive
Encryption	Requires high encryption	Requires less encryption

3. **Global consistency and interoperability:** To coordinate the emergency response effort, agencies must be able to agree on many details that make up the state of an emergency, including what the emergency is, how severe it is, where it is occurring, what personnel are available and where they are located, which personnel have the necessary skills for this emergency, what equipment is available, where the equipment is located, etc. Constructing such globally consistent views of the emergency event is hard for two reasons: (a) the state of the emergency is constantly changing and (b) the information is distributed piecemeal across geographically distant agencies having incompatible software systems. WHISPER will therefore develop technologies that provide globally consistent unified views of an emergency response activity.
4. **Visualizing Information presentation and interface uniformity:** Although decisions must be made quickly during emergency response activities, quick decision making can be hampered by the sheer volume of information that EPR personnel must consider. Simplified readouts with graphical representations and information represented as sound can help reduce the amount of time it takes for personnel to evaluate pertinent information. WHISPER will therefore integrate advances in HCI to create simple, yet informative, computer visualizations of emergency information.
5. **Decision support:** The large amount of information that emergency response personnel must process, aggregate, review, and filter to make timely decisions can be overwhelming, and such a flood of information can introduce errors in the decision making process. Work in other fields where computers assist personnel by processing information and suggesting courses of action has had significant success in speeding up decision time and reducing errors. WHISPER will therefore allow the development of decision-support software to aid EPR personnel in their decision making by suggesting possible courses of action or warning personnel of possibly dangerous situations that may be beyond the scope of the particular agency's purview. The main functional requirement is to create an information integration tool. This should include emergency agencies such as fire stations and unified agencies that combine all relevant emergency agencies.

B. Design Architecture

The architecture of WHISPER can be described in three major component layers that are designed to meet the functional requirements: the emergency planning and activations layer, the unified ERA view layer, and the local service layer.

B.1. Emergency Planning and Activations

The goals of the emergency planning and activations layer is to present ERA data in a coherent manner and to maximize the benefit of sharing ERA data among cooperating entities. To those ends, we have identified the following sublayers. *(i)* Emergency responder user interface. *(ii)* Global scheduling and planning operations manager, and, *(iii)* Scenario-based simulation and training. Presently, the last two sublayers do not exist in the current prototype. The user interface provides real-time details on status and availability various emergency resources.

B.2. Unified ERA view

The goal of the unified ERA view layer is to present a unified data model to the upper layer components. This layer consists of the following sublayers for constructing the model.

Unified global ERA views. This sublayer contains the data processing components that present a *global unified data model* to the components in the emergency planning and activations layer. Components in this sublayer collect the data from local unified ERA views, and map the data in a standard schema. We use a derivation of the RDF markup and RDFS schema languages to describe our mappings, and a pattern matching language developed in-house to automate the mapping process.

Unified Local ERA views. This sublayer contains the data processing components that present a *local unified data model* that represents regional data to the components in the unified global ERA views. Components in this sublayer collect the data from local service layers of individual agencies and map the data in a standard schema using definition from the mapping sublayer below. As with the unified global ERA views, we use a derivation of the RDF markup and RDFS schema languages to describe our mappings, and a pattern matching language developed in-house to automate the mapping process.

B.3. Local Service Layer

The local service layer transports data from the local agencies to the local unified ERA view. To this end, we have implemented the following sublayers.

B.4. Transport sublayer

The transport sublayer reliably transports the information from the local repository to unified views in a reliable, efficient, and secure manner, and authenticates the source of the information. We are currently investigating the use of web services for data transport. Web services seems particularly promising for this project because of its ability to integrate heterogeneous software and hardware systems via standardized web-based protocols, and for its ability to locate services in distributed environments. To implement the framework at this sublayer, we use lessons learned from distributed databases, distributed shared memory, networking, and network security, as well as web services.

This module parses updates received from the DataManager to create an Object Exchange Model (OEM) object.

B.5. Mapping sublayer

The mapping sublayer (Data Manager) provides the tools for new agencies to define how the information from each agency maps into unified views. Additionally, the unified view layer can query the mapping sublayer to obtain these definitions when the unified view layer needs to map the agencies' information into the unified view information cache. To implement the framework at this sublayer, we use lessons learned from web computing, semi-structured data management, and techniques and models for unifying web data.

This module is notified whenever there is a change in the emergency service database. It serves as the control center of the emergency service part of the system. It first retrieves the updates from the emergency service database. Then, this module sends the updates to the Transport module. It also gets the updates back from the Transport module in the form of OEM Objects. Then, it sends the OEM Objects to the Consistency module and gets the objects back when the consistency module decides it is time to send the updates. The DataManager sends the OEM object updates to the differences class when the updates are ready to be sent, which in turn inspects each OEM object and decides whether the unified view needs this information. The reason the unified view might not need the information is because either it is unimportant in the unified view, or more likely, the unified view already has this information and does not require any updates. If required, it sends the OEM object differences back to the DataManager to be exported. The Data Manager finally sends the updates in the form of OEM Objects to the Unified View.

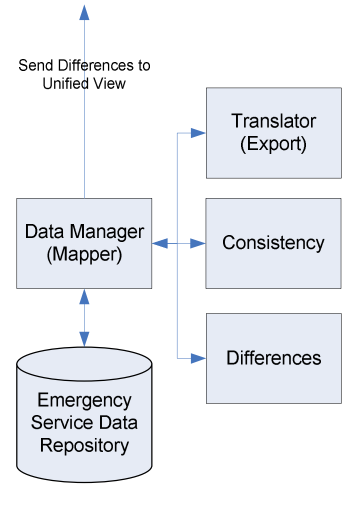


Figure 3. Prototype Structure

B.6. Export sublayer

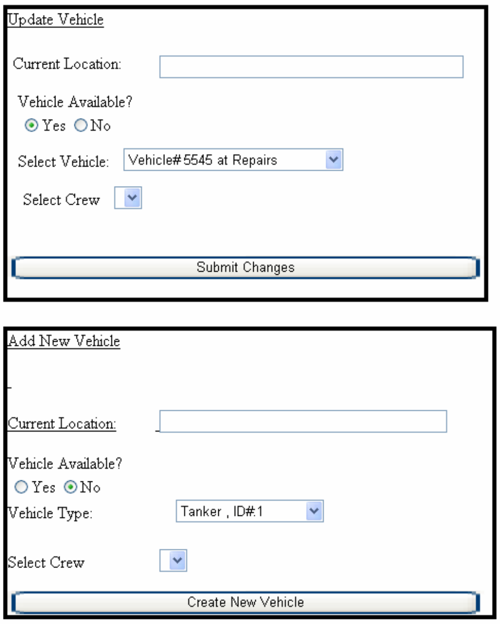
The export sublayer provides the software to retrieve updated data from the local agency's data repository and provide it in a canonical form to the upper layers. Additionally, this sublayer may interact with the consistency sublayer (described below), to provide copies of local information for consistency management. In the current prototype implementation, due to the relatively small nature of the prototype application, we have merged the export layer with the transport layer for the implementation. To implement the framework at this sublayer, we use lessons learned from web computing, semi-structured data management, and techniques for modeling web data. This is illustrated in Figure 3.

In the future, we plan to leverage the adapter and translator capabilities similar to the Open Tool Integration Framework to automate the export and import of data between layers and tools.

B.7. Consistency sublayer

The consistency sublayer provides the software to manage what data has been updated, who needs that information, and how that data should be transported to the unified view layer (via the transport sublayer). To implement the framework at this sublayer, we use lessons learned from distributed databases, distributed shared memory, and web caching.

This module tells the system where and when to send the OEM Object updates. The current implementation uses the "eager" consistency mechanism, which means that updates are sent immediately. Also, since WHISPER has only one unified view, so there is only one place to send the updates. The consistency module is simple for this implementation, but the framework is provided to make it more complex by adding multiple consistency mechanisms or unified views. There are three reasons to use a consistency module. This project will eventually be expanded to having the emergency services and global view separated by a Wide Area Network (WAN). One goal of a Consistency model is to reduce the number of messages. The consistency module can confirm that the updates are not being duplicated and sent more than once. The second reason is to add authorization security measures. When we are sending updates to several unified views, we need to make sure that they are the only ones that get the updates and that they only get the information they are authorized for. The third reason is robustness. We will be able to recover gracefully when messages and data are not received or sent properly with a consistency module.


Update Vehicle
Current Location:
Vehicle Available? ● Yes ○ No
Select Vehicle: Vehicle#5545 at Repairs
Select Crew

Add New Vehicle
Current Location:
Vehicle Available? ○ Yes ● No
Vehicle Type: Tanker , ID#1
Select Crew

Figure 4. Fire Station Interface

V. PROTOTYPE DESIGN

The goal of the WHISPER is to allow each emergency agency to efficiently integrate their databases at minimal cost and effort. It allows the integration of an agency's existing data repository that has its own statistical data, including resource availability, resource description, and other emergency information, into one or more *unified views.* Each unified view contains a data repository holding combined information from each emergency agency. WHISPER ensures that the combined information is consistent, readily available, and updated in a timely manner. Additionally, given that WHISPER will be built on top of the agencies' existing databases that store their information in various types of data repositories such as relational databases, spreadsheets, and text files, that have varying database schemas, even for similar information, WHISPER resolves discrepancies between the different data sources in order to combine them for the unified view.

The interface for each emergency resource (entity) serves two purposes. The first purpose is for the user to be able to change a resource's availability and current location. The second is for the user to be able to add a new resource to their database. A sample screen shot for web service interfaces for the fire service is shown in Figure 4.

A. Unified Views

WHISPER users can choose the type of service they want to monitor. Depending on the selection, they are redirected to the corresponding data view. When selected, all the information that is stored in the unified cache appears. The displayed information in the unified view contains all the details that correspond to the type of service it provides. To enable this feature a ***Timer*** variable was set which will update the information regularly. The timer is initially set to a default value of '5 minutes'. A user can change the timer variable as needed. At the end of each interval an updated Unified view is displayed. For example, the unified view for Fire Service (Figure 5) contains all the necessary information that is required by a higher command. This data helps the command to keep track of the fire truck and crew attributes. The attributes include crew, availability of vehicle, availability of the crew on the truck, current location of the fire truck and fire station details. Users can find the details of the truck type by clicking on the Truck_Type_ID. To look into the station details of that particular truck, a user can click on Station_ID. Crew information on the specified truck at a specific point of time can be tracked by clicking on Crew_ID.

VI. IMPLEMENTATION CONSIDERATIONS

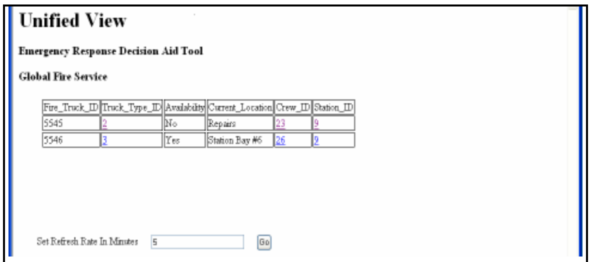
Several implementation considerations need to be accounted for. Specifically, the overall performance of Web Services depends on application logic, network, and most importantly on underlying messaging and transport protocols, such as SOAP and HTTP. Some observed limitations include:

1. The SOAP protocol is still maturing and has many of performance and scalability problems.
2. The SOAP protocol uses a multi-step process to complete a communication cycle.
3. The Web Services currently rely on transport protocols such as HTTP, which are inherently stateless and follow a best-effort delivery mechanism. It does not guarantee whether the message is delivered to the destination. Application performance might suffer or might appear unreliable.
4. The SOAP request begins with the business logic of your application learning the method and parameter to call from a Web Services Description Language (WSDL) document. This whole process is a time-consuming one, which requires various levels of XML parsing and XML validation and hence hits the performance of the Web Service.
5. From a stability perspective, Web Services are still nascent and not necessarily good for scalable parallel architectures.
6. For interoperability, both platforms should have similar class hierarchy.
7. Building scalable systems is expensive, and may cause smaller companies to defer this requirement. Also, this becomes an infrastructure issue for companies that deploy Web Services within their enterprise.
8. Building fault-tolerant systems for highly available Web Services is expensive.
9. Inability to understand and translate the semantics of information being exchanged.

VII. CONCLUSIONS AND FUTURE WORK

While service oriented architectures hold vast potential for helping create diversified, agile programs to synergistically function together to solve complex problems, implementing effective communication mechanisms is still a difficult problem. Incorporating detailed semantic descriptions is a key factor in locating and retrieving related objects / services on the Web. Normally this is addressed by developing ontologies. However, the appropriate level of granularity, coarse-grained versus fine-grained ontologies is often what distinguishes the success of a retrieval process. While coarse-grained ontology descriptions are easily created, their vague descriptions make it difficult for ‘exact’ searches. On the other hand, fine-grained ontologies are expensive and much harder to create. However, their more detailed information elements make it easier for a service discovery process. Current research in this direction is attempting to find a trade-off between necessary detail versus performance cost trade-offs. One possible approach is taken by the INFRAWEBS project [67], which attempts to use similarity-based reasoning to locate possible matching services based on a vague ontology that covers a broad area of general functionality, while applying logic-based reasoning to elaborate logical definitions using highly precise ontologies that cover a smaller number of precise definitions thereby supporting more detailed discovery needs.

The intent of this paper is to develop and prototype a scalable architecture for the integration of several disjoint incident management systems and implement a prototype implementation based on web services. As identified there are several drawbacks in using service-oriented architectures, but it is a very promising avenue for large-scale implementation which is also supported by sustained efforts by various corporations and government sponsored projects.



Unified View

Emergency Response Decision Aid Tool

Global Fire Service

Fire_Truck_ID	Truck_Type_ID	Availability	Current_Location	Crew_ID	Station_ID
5545	2	No	Repairs	23	9
5546	3	Yes	Station Bay #6	26	9

Set Refresh Rate In Minutes 5

Figure 5. Unified View of Fire Services

VIII. ACKNOWLEDGEMENTS

The authors wish to thank Dr. Doug Talbert at TTU and Dr. Gary Black of University of Southern Indiana for facilitating discussions with local emergency officials during the early phases of this project. They also wish to thank Drs. Doug Schmidt and Andy Gokhale from Vanderbilt University for some engaging discussions on middleware support. They also thank graduate students Subash Sudireddy and Sajane Narre from TTU for support during the prototype development.

IX. REFERENCES

1. Crisis Information Management Software (CIMS) Feature Comparison Report, Special Report. Technical Report, U.S. DOJ, National Institute of Justice, Washington D.C., 2002.
2. A. Weintraub, J. Aboud, C. Fernandez, G. Laporte, and E. Ramirez. Emergency Vehicle Dispatching System for an Electric Utility in Chile. *J. of the Operational Research Society*, 50(7):690-696, 1999.
3. C. Amza, A. L. Cox, S. Dwarkadas, P. Keleher, H. Lu, R. Rajamony, W. Yu, and W. Zwaenepoel. TreadMarks: Shared Memory Computing on Networks of Workstations. *IEEE Comp.*, 29(2):18-28, Feb.1996.
4. B. Yen. Communication Infrastructure in Distributed Scheduling. *Computers and Industrial Engineering*, 42(2-4):149-161, 2002.
5. A. D. Birrell and B. J. Nelson. Implementing Remote Procedure Calls. 2(1):39-59, February 1984.
6. B. Callaghan, B. Pawlowski, and P. Staubach. NFS Version 3 Protocol Specification. RFC 1813, June 1995.

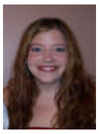
7. T. Cortes, S. Girona, and J. Labarta. Avoiding the Cache-Coherence Problem in a Parallel/Distributed File System. In *Proc. of the High-Performance Comp. and Networking*, pages 860-869, April 1997.
8. E. Doran, S. Franklin, N. Jennings, and T. Norman. On Cooperation in Multi-Agent Systems. *The Know. Engg Rev.*, 12(3):309-314, 1997.
9. E. Durfee. The Distributed Artificial Intelligence Melting Pot. *IEEE Trans. on Systems, Man, and Cybernetics*, 21(6):1301-1306, 1991.
10. F. Wilson, L. Cantin, and B. Bisson. Emergency Response System for Dangerous Goods Movements by Highways. *Journal of Transportation Engineering*, 116(6):789-794, 1990.
11. V. W. Freeh, D. K. Lowenthal, and G. R. Andrews. Distributed Filaments: Efficient Fine-Grain Parallelism on a Cluster of Workstations. In *Proc. of the 1st Symp. on Operating Systems Design and Implementation*, pages 201-213, November 1994.
12. G. Dai and Q. Da. Study of Combinatorial Scheduling Problem in Emergency Systems. *System Eng. Th.y and Prac.*, 20(9):52-55, 2000.
13. G. Nitschke. Cooperating Air Traffic Control Agents. *Applied Artificial Intelligence*, 15(2):209-235, 2001.
14. G. Pappas, C. Tomlin, and S. Sastry. Conflict Resolution for Multi-Agent Hybrid Systems. In *Proceedings of the IEEE Conference on Decision and Control v2*, pages 1184-1189, 1996.
15. G. Tan, C. Hayes, and M. Shaw. An Intelligent-Agent Framework for Concurrent Product Design and Planning. *IEEE Trans. on Engineering Management*, 43(3):297-306., 1996.
16. Garrido and F. Barber. Integrating Planning and Scheduling. *Applied Artificial Intelligence*, 15(3):471-491, 2001.
17. J. Griffioen, R. Yavatkar, and R. Finkel. Unify: a Scalable Approach to Multicomputer Design. *IEEE Comp.r Soc. Bulletin of the Tech.l Cmmt. on Operating Systems and Application Environment*, 1995.
18. W. Gropp and E. Lusk. Sowing MPICH: a Case Study in the Dissemination of a Portable Environment for Parallel Scientific Computing. *The International Journal of Supercomputer Applications and High Performance Computing*, 11(2):103-114, Summer 1997.
19. W. Gropp, E. Lusk, N. Doss, and A. Skjellum. A High-Performance, Portable, Implementation of the MPI Message Passing Interface Standard. *Parallel Computing*, 22(6):789-828, September 1996.
20. M. Henning. Binding, Migration, and Scalability in Corba. *Communications of the ACM*, 41(10), October 1998.
21. W. Hoschek. A unified peer-to-peer database framework and its application for scalable service discovery. In *the 3rd Int'l. IEEE/ACM Workshop on Grid Computing (Grid'2002).*, 2002.
22. I. Hatono, Y. Huang, M. Umano, and H. Tamura. Cooperative Scheduling Protocol and Dispatching Algorithm Taken Account of Emergency Jobs for Distributed Real-time Scheduling. In *Proc.s of the Japan/USA Symp. on Flexible Automation*, # 2, pp 1383-1389, 1996.
23. J. Ambite and C. Knoblock. Planning by Rewriting. *Journal of Artificial Intelligence Research*, 15:207-261, 2001.
24. J. Goldberg and L. Paz. Locating Emergency Vehicle Bases When Service Time Depends on Call Location. *Transportation Science*, 25(4):264-280, 1991.
25. J. Kuchar and L. Yang. Survey of Conflict Detection and Resolution Modeling Methods. In *AIAA-97-3732, AIAA Guidance, Navigation, and Control Conference*, pages 1388-1397., New Orleans, LA, 1997.
26. J. Kuchar and L. Yang. A Review of Conflict Detection and Resolution Modeling Methods. *IEEE Trans. on Intl. Trans. Sys.*, 1(4):179-189, 2000.
27. J. Shively. Dispatcher Decision Making: Ecological Display Development. In *Proc. of the Human Factors and Ergonomics Society*, number 1, pages 506-510, 1995.
28. K. Barber and C. Martin. Dynamic Reorganization of Decision-Making Groups. In *Proc. of the Fifth Intnl Conf on Autonomous Agents*, pp 513-520, Montreal, Quebec, Canada, 2001. AGENTS'01.
29. K. Barber, R. McKay, A. Goel, D. Han, J. Kim, T. Liu, and C. Martin. Sensible Agents: The Distributed Architecture and Testbed. *IEICE Transactions on Communications*, E83-B(5):951-960, 2000.
30. K. Barber, S. Ramaswamy. Conflict Detection During Plan Integration for Multi-Agent Systems. *IEEE Trans. on SMC*, 31(4):616-628, 2001.
31. K. Barber, T. Liu, A. Goel, and C. Martin. Conflict Representation and Classification in a Domain-Independent Conflict Management Framework. In *Proceedings of the Third International Conference on Autonomous Agents*, pages 346-347, 1999.
32. K. Barber, T. Liu, A. Goel, and S. Ramaswamy. Flexible Reasoning Using Sensible Agent-Based Systems: A Case Study in Job Flow Scheduling. *Jour. of Prod Plan and Control*, 10(7):606-615, 1999.
33. K. Biggers and T. Ioerger. Automatic Generation of Communication and Teamwork within Multi-Agent Teams *Applied Artificial Intelligence*, 15(10):875-916, 2001.
34. K. Zografos, C. Douligeris, J. Haupt, and J. Jordon. Methodological Framework for Evaluating On-Board Computer Technology in Emergency Dispatch Operations. In *Proceedings - Society of Automotive Engineers*, pages 103-112, 1991.
35. A. Kahol, S. Khurana, S. Gupta, and P. K. Srimani. A Strategy to Manage Cache Consistency in a Disconnected Distributed Environment. 12(7):686-700, 2001.
36. P. Keleher, A. L. Cox, S. Dwarkadas, and W. Zwaenepoel. An Evaluation of Software-Based Release Consistent Protocols. *Journal of Parallel and Distributed Computing*, 29(2):126-141, Sept. 1995.
37. James Kistler and M. Satyanarayanan. Disconnected Operation in the Coda File System. 10(1):3-25, February 1992.
38. I. Kumaran. *Jini Technology: An Overview*. Prentice Hall, Nov. 2001.
39. L. Kaelbling, M. Littman, and A. Moore. Reinforcement Learning: A Survey. *Jour. of Artificial Intelligence Research*, (4):237-285, 1996.
40. L. Rhodenizer, F. Jentsch, and C. Bowers. Emergency Service Dispatch. *Ergonomics in Design*, 8(3):24-28, 2000.
41. M. Cameron and M. Schultz. Use of Commercial GIS and GPS for Emergency Location and Dispatch. *Proc. Ann. Mtg. - Inst. of Navig., 50 Yrs of Navig. Progress from Art to Utility*, pp. 659-667, 1995.
42. Sun Microsystems. Java Remote Method Invocation - Distributed Computing for Java. *http://java.sun.com/marketing/collateral/javarmi.html*.
43. M. Nelson, B. Welch, , and J. Ousterhout. Caching in the Sprite Network File System. 6(1):134-154, February 1988.
44. M. Nicola and M. Jarke. Performance modeling of distributed and replicated databases. *IEEE Trans. on KDE*, 12(4):645-672, 2000.
45. P. Prosser and I. Buchanan. Intelligent Scheduling: Past, Present and Future. *Intelligent Systems Engineering*, 3(2):67-78, 1994.
46. P. Richard, N. Jacquet, C. Cavalier, and C. Proust. Solving Scheduling Problems Using Petri Nets and Constraint Logic Programming. In *Proceedings of 1995 INRIA/IEEE Symposium on Emerging Technologies and Factory Automation , Part 1 (of 3)*, 1995.
47. R. Rabelo and L. Camarinha-Matos. Negotiation in Multi-Agent Based Dynamic Scheduling. *Robotics & CIM*, 11(4):303-309, 1994.
48. M. Raynal and M. Mizuno. How to find his way in the jungle of consistency criteria for distributed shared memories (or how to escape from Minos' labyrinth). In *Proc. of the 4th IEEE Workshop on Future Trends of Distributed Computing Systems*, pages 340-346, 1993.
49. M. D. Rogers. *Reactive Shadows: A New Framework for Building Adaptive, Distributed Applications*. PhD thesis, Univ. of KY, 2002.
50. Michael D. Rogers, Christopher S. Diaz, Raphael Finkel, James Griffioen, and Jr. James E. Lumpp. BTMD: Small, Fast Diffs for WAN Based DSM. In *Workshop on Distibuted Shared Memory 2000 in conjunction with ICS'00*, May 2000.
51. S. Sen and G. Weiss. *Learning in Multiagent Systems*, pages 259-298. MIT Press, 1999.
52. S. Shibuya, et. al. Fast Emergency Vehicle/Preemption Systems. *Trans. Res. Record*, (1739):44-50, 2000.
53. E. Speight and J. K. Bennett. Brazos: A Third Generation DSM System. In *Proc. of the USENIX Windows NT Workshop*, Aug. 1997.
54. T. Nathanail. Methodology for Evaluating Alternative Policies for Dispatching Emergency Vehicles. In *Road and Vehicle Safety – Proc. of the 27th Intnl Symp on Automotive Tech and Atmn*, pp 347, 94.
55. V. Lesser. Cooperative Multiagent Systems: A Personal View of the

State of the Art. *IEEE Transactions on KDE*, 11(1):133-142., 1999.

56. S. Vinoski. Corba: Integrating Diverse Apps within Distributed Heterogeneous Environments. *IEEE Comm. Mag*., 14(2), Feb 1997.
57. Yannis Papakonstantinou, Hector Garcia-Molina, Jeffrey Ullman. MedMaker: A Mediation System Based on Declarative Specifications. *Proc of the 12th Intnl Conf. on Data Engg*, 1995.
58. Yannis Papakonstantinou, Hector Garcia-Molina, Jennifer Widom. Object Exchange Across Heterogeneous Information Sources. *11th Conference on Data Engineering*, 1995.
59. R. Goldman, S. Chawathe, A. Crespo, J. McHugh. A Standard Textual Interchange Format for the Object Exchange Model (OEM). http://www.db.stanford.edu/~mchughj/oemsyntax/oemsyntax.html.
60. H. Garcia-Molina, Y. Papakonstantinou, D. Quass, A. Rajaraman, Y. Sagiv, J. Ullman, J. Widom. The TSIMMIS Approach to Mediation: Data Models and Languages (Extended Abstract). *J. of Int. Info. Sys*, 8(2):pp. 117-132, Mar-Apr 1997.
61. Jason McHugh, Serge Abiteboul, Roy Goldman, Dallan Quass, Jennifer Widom. Lore: A Database Management System for Semistructured Data. *SIGMOD Record*, 1997.
62. Roy Goldman, Jason McHugh, Jennifer Widom. From Semistructured Data to XML: Migrating the Lore Data Model and Query Language. *Workshop on the Web and Databases (WebDB '99)*, 1999.
63. Kitchen, T. “Planning in response to terrorism: the case of Manchester”, *J.. of Arch. and Plan. Res.,* 18 (4), 2001, pp. 325-340.
64. J. Wang, D. Rosca, W. Tepfenhart and A. Milewski, Incident Command System Workflow Modeling and Analysis: A Case Study, Proceedings of ISCRAM’06, Newark, NJ, April 2006.
65. Poulymenopoulou M, Malamateniou F and Vassilacopoulos G. Specifying Workflow Process Requirements for an Emergency Medical Service. *Journal of Medical Systems,* 27(4), 323-333, 2003.
66. M. Llavador, Precise Yet Flexible Specification of Emergency Resolution Procedures, Proc. of ISCRAM’06, Newark, NJ, April 2006
67. http://www.infrawebs-eu.org

Dr. Ramaswamy is currently the Chairperson of Department of Computer Science at University of Arkansas at Little Rock. Earlier he was the Chairperson of the Computer Science Department at Tennessee Tech University. He currently coordinates the development of the Knowledge Enterprises for Scalable Resilient Infrastructures Laboratory (KESRI) with a focus on integrating and bridging the gap between computing systems and social sciences, in a society that is increasingly bound by complex software systems deployed on heterogeneous distributed platforms. His research interests are on intelligent and flexible control systems, behavior modeling, analysis and simulation, software stability and scalability; particularly in the design and development of better software systems for real-time control issues in manufacturing and other distributed, real-time applications. He has actively consulted on the design, development and implementation of a knowledge capture, classification and mining project with eFutureKnowledge, Inc. and on a NIST sponsored ATP project with InRAD, LLC. During the summers of 2003 and 2004, he was a visiting research professor of Computer Science in the Institute of Software Integrated Systems (ISIS) at Vanderbilt University as part of a NSF ITR project - Foundations of Hybrid and Embedded Software Systems. In 1994-1995, and subsequently during the summer months of 1996 and 1998, he was a post-doctoral research fellow / visiting scientist in the Laboratory for Intelligent Processes and Systems (LIPS) at the University of Texas at Austin where he helped with research efforts on Sensible Agents. Dr. Ramaswamy earned his Ph.D. degree in Computer Science in 1994 from the Center for Advanced Computer Studies (CACS) at the University of Louisiana at Lafayette, LA, USA. He is member of the ACM, Society for Computer Simulation International, Computing Professionals for Social Responsibility and a senior member of the IEEE.

Dr. Michael Rogers has been actively involved in cluster and distributed computing research. He was awarded his Ph.D.. from the University of Kentucky where the he was part of a team that built a distributed shared memory operating system called Unify that implemented multiple consistency models. In particular, he implemented a novel differencing mechanism that increased the performance and scalability of Unify. Additionally, his thesis work entailed adding a monitoring system and an object-oriented library with scripted components to Unify that allowed programmers to build their own tailor made adaptive applications that could react to changes in the computing environment, and modify the application so that its computations remained efficient. While at the University of Kentucky, he was involved in the Digital Atheneum project. He was largely responsible for building a Digital Archive in a Networked Collaborative Environment, or DANCE. DANCE is a large scaled distributed system that allows literary scholars to collaborate, i.e. concurrently view and modify, and well as share editing techniques, in order to create digital libraries. DANCE included many interesting features, including a scalable image server, new tagging techniques, and the ability to advertise and share image and text manipulation algorithms. Currently, Dr. Rogers is involved in many projects at Tennessee Technological University (TTU). He was a Co-PI on TTU’s Internet2 grant proposal. Additionally, he advises several students involved in various activities. Among them are designing distributed neural networks, designing technologies for interoperability in distributed medical record sharing systems, system administration using ontologies, and a designing a performance testing suite for continuous query systems.



Ms. April Dean Crockett received a MS in Computer Science from Tennessee Tech University in Fall 2004 while working at the Putnam County Library as a Technical Coordinator. Her research interests are in scalability, modeling and simulation and Data integration. She currently works as the Website Development Manager at the Angelo and Jennette Volpe Library at Tennessee Tech.

Mr. David Feaker graduated with a B.S. in Computer Science from Tennessee Tech University in 2005. He is currently working with the Oak Ridge National Laboratory’s SensorNet project and is responsible for the development of sensor integration applications which will be deployed nationwide at multiple sea ports and truck weigh stations. He is also a part-time teacher of introductory programming classes at a local technical college. His current research interests include how to effectively detect radiated vehicles at high speeds, sensor integration in a web application environment, and extension of schemas developed for radiation devices

Mr. Monty Carter is a student at Tennessee Tech University pursuing a B.S. degree in Computer Science. During 2005-2006 he worked as a software developer with Milliken and Company in Spartanburg, SC, where he was responsible for development of software applications for internal company use, distribution of software packages throughout the company. He is currently working for the Center for Manufacturing Research at Tennessee Tech University on the development of an application for management of the centers projects and personnel. His electronic portfolio is available at:http://www.csc.tntech.edu/~mbcarter