The Next Step in Information Sharing: The Distributed Alerts Dissemination Backbone

Derik Pack
SPAWAR Systems Center, Charleston
United States Navy
derik.pack@navy.mil

Clayton Coleman
SPAWAR Systems Center, Charleston
United States Navy
russell.c.coleman@navy.mil

John Osborne
SPAWAR Systems Center, Charleston
United States Navy
john.f.osborne@navy.mil

Abstract

As new national security threats have emerged, the major requirement for information management has changed from a "need to know" to a "need to share". This has led to numerous government initiatives to fuse data from heterogeneous sources to provide a robust operational picture for end users. These efforts are hindered by a lack of integration mechanisms between participating agencies. As a result, the Department of Homeland Security (DHS) and Department of Justice (DOJ) have proposed standards for capturing and distributing information. This paper presents factors driving this environment and discusses an architecture for it, the Distributed Alerts Dissemination Backbone (DADB). DADB is based on open standards and is designed to disseminate information to a geographically diverse Community of Interest (COI). It also provides mechanisms for information translation while on route to its final destination. This paper also examines the COI policy needed for information distribution.

1. Introduction

A combination of man-made and natural disasters within the past decade has begun a change of the information management environment within all levels of the United States government. The best examples of this change are seen with the formation of DHS and the Department of Defense's requirement for a more agile force. Both agencies are transitioning from their traditional views of information as "need to know" to "need to share".

This change has initially been felt in the organizations as they work to field better information management and dissemination solutions for their own personnel. The best example of this can be found in [1], where the National Institute of Justice (NIJ) provided a comparison of 10 critical incident management software (CIMS) products. Along with the comparison, the NIJ provided overall findings for the CIMS industry. One of the major findings in [1] was that all the vendors supported standards of operation in their software, but there was no significant effort spent in exchanging information with other systems in a standard manner. For a category of software used by multiple states and agencies, this would require multiple integration efforts and information sharing agreements to automate the policy for inter and intra-agency cooperation in an emergency situation. The report provided two major recommendations given this finding: standards must be developed for emergency management, and agencies must encourage industry to develop to standards by stating these needs and reinforcing it through their acquisitions.

As a result of these interoperability problems, DHS began developing a set of emergency management standards through the OASIS standards body. These standards include the Common Alerting Protocol (CAP) [2] and the Emergency Data Exchange Language (EDXL) [3].

The rest of this paper will describe current standard and system development within the emergency management community and present an architecture based upon some of these emerging standards. This architecture will be designed to address the interoperability needs of the first responder community and any community that depends on alerts as part of the execution strategy. Section 2 will provide a background for the evolving emergency management standards and the tools being used by that community. Section 3 will introduce the design principles behind the DADB architecture. Section 4 will provide an overview of major components within the architecture. Section 5 will provide lessons learned from implementing the architecture, and Section 6 will provide conclusions of the work.

2. Background

The events of September 11th had a major effect on information sharing policy within the United States. One of the recommendations of the 9-11 Commission [6] was to unify the knowledge of participating counter-terrorism agencies. The Homeland Security Act of 2002 [7] formed the Department of Homeland Security and gave the President the responsibility to set guidelines for information sharing policy and processes between local, state and federal agencies. The President delegated that responsibility to the Secretary of DHS, but moved it to a

978-0-7695-3261-5 2008

U.S. Government Work Not Protected by U.S. Copyright
DOI 10.1109/ICWE.2008.28
IEEE computer society

separate program manager in 2005 charged with creating an Information Sharing Environment [8].

While these policy efforts were being made at the legislative and executive branches of the federal government, individual federal agencies and states began or continued to maintain their own information sharing efforts. An example of effort being maintained is the National Law Enforcement Telecommunications System (NLETS). NLETS is a nationwide network for the states and federal government to exchange certain types of criminal justice information. Some of the available information on this network includes vehicle registration and criminal history records [9]. While NLETS is the oldest system of its kind at almost 4 decades, it is not the only network that contains criminal justice or emergency responder information. A more recent example is the Homeland Security Information Network (HSIN) [10]. HSIN was formed by DHS shortly after the department's creation. There is still an ongoing effort within the federal government to unify the various information sharing networks across agency and policy boundaries.

The government has also invested in a parallel effort to develop emergency management standards as many of the emergency management information sharing networks are being unified. Standards development is progressing internally in government agencies and in coordination with standards bodies. Examples of internal efforts are the National Information Exchange Model (NIEM) and its' predecessor, Global Justice XML Data Model (GJXDM) [13, 14]. NIEM is a program to support the development of information sharing standards across all branches of governments. It expanded on the effort of GJXDM which supported standards for public safety and justice. NIEM also serves as a standards warehouse for federal agencies. Many of the standards developed in conjunction with standards bodies and industry are referenced in NIEM. Two such standards, CAP and EDXL-DE, were mentioned early in this paper. Both are XML standards which were developed and ratified by the OASIS Emergency Management Technical Committee. CAP is a standard format to exchange emergency alerts and public warnings over various systems [2]. EDXL-DE as a standard provides a formal message distribution framework and can distribute CAP messages as XML content [3].

OASIS has two additional draft standards that may eventually be ratified as distribution payloads for EDXL-DE messages. The first is the Emergency Data Exchange Language Resource Messaging (EDXL-RM). This draft provides a standard format for XML emergency response messages and is designed to be a XML payload for EDXL-DE [11]. The second is the Emergency data Exchange Language Hospital Availability Exchange. It is a potential EDXL-DE XML payload, and it specifies a format for the status of a hospital, and its services and resources [12].

After considering this background information, the authors chose to design a system around EDXL-DE for the following reasons:

- Standards for distribution and routing will be critical for the adoption and maintenance of information sharing networks.
- EDXL-DE can serve as the distribution mechanism for the other three OASIS emergency management standards. An understanding of its limitations is critical to the use of those standards.
- At the time the research began, there were no public examples of EDXL usage that could be used to highlight the pros and cons of the standard.

A graphical representation of the schema for this standard can be found in Figure 1.

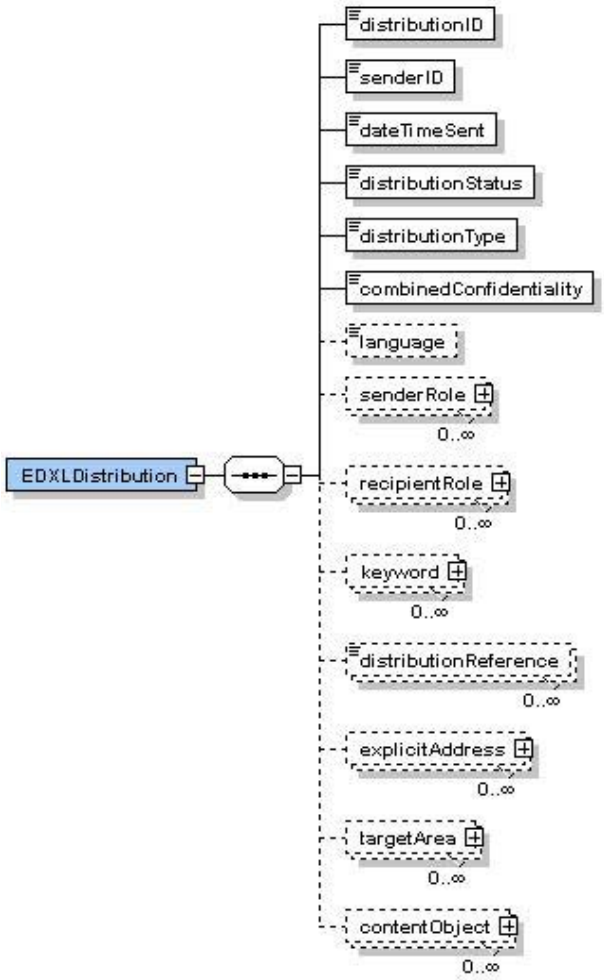

Figure 1. EDXL-DE schema

The prior works that influence the design principles and EDXL-DE's support of the design principles are discussed in the next section.

3. DADB Design Principles

Before discussing the design principles for DADB, it must be noted that the authors chose to constrain the design problem by looking at past work in interoperability and information sharing for government organizations and COIs. Key works include the Net-Centric Enterprise Services (NCES) being developed by DISA and the NCES early adopter portfolio, Horizontal Fusion [4], [5]. NCES provides a set of core services including service discovery and security services. Horizontal Fusion portfolio programs provided a mechanism for content discovery called Federated Search which used web services and the NCES Core Services. This mechanism allowed a user to query several heterogeneous data silos through one interface.

While NCES and Horizontal Fusion have developed mechanisms to support users querying for information, they have not presented a capability that provides the best fit for alerts. Current content discovery mechanisms implement a pull-based architecture where a user query initiates the information sharing process. For information sharing to become a reality for first-responders and in the DoD, a push-based architecture for an alert generator to initiate the distribution of a message to interested users is required.

Given the use case of an interoperable push-based alerting application for a given COI, the authors believe the architecture must be created with the following goals in mind: modularity, translation-capability, geographical awareness, and role agnosticism.

The architecture must be modular to include COI dependent components at low or no cost to the COI. An appropriate example of such a component would be a dissemination policy module. The implementation of dissemination policy could vary depending on the COI or even the specific alert that is being disseminated. Much of the modularity needed for the architecture can be accomplished through effective interfaces and the use of *dynamic* or *run-time reflection* to provide new implementations of those interfaces at run-time.

When discussing the translation capability of the architecture, the authors must note that the translation is from data format to data format, as opposed to language to language. Translation is necessary because end-users of an alerts dissemination mechanism will not have the same end-device or capability to view the alert. Translation mechanisms will need to at least make a best attempt to convert a message's data format to a format appropriate for a given user. In addition, the translation process should be executed at the end of distribution to reduce the performance cost of translations when users are not interest or authorized for a message.

The next goal of the architecture is to be geographically aware. A common element for decision making when disseminating alerts is the user role. However, in a first responder situation spanning multiple agencies and states, geographic location can be a greater contributing factor than the specific role of the user.

The last goal of the architecture is to be role agnostic. While role is a major factor in authorization and dissemination policy, roles may or may not be meaningful across COI boundaries. The semantic understanding of role is completely COI dependant. For example, a system administrator of one DoD system in Florida is not going to be a system administrator of a DoJ system in Washington. The DADB architecture must thus be agnostic to role at the enterprise but support the COI and its policy.

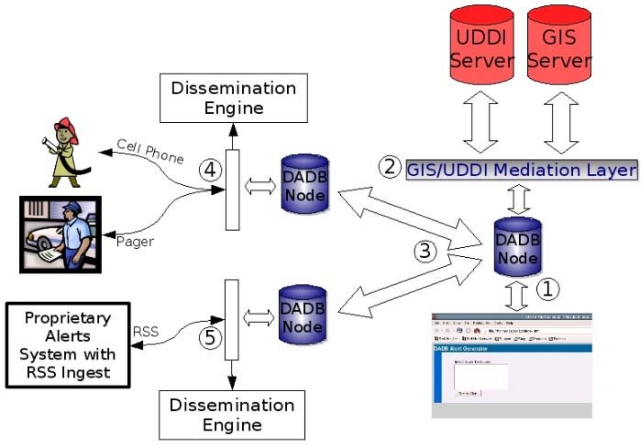

Figure 2. Architecture overview

4. Components of DADB

Given the design goals for DADB, the implementation takes a unique approach. Instead of creating web services to query an authoritative data source, DADB uses web services to disseminate alerts messages. Figure 2 represents the notional DADB architecture. Each web service in the architecture is designated as a node (Figure 2.1). Modular interfaces are implemented within the DADB nodes. Geographic awareness is established within each node from UDDI and GIS servers as appropriate (Figure 2.2). The final dissemination engines are responsible for enforcing dissemination policy to the final users; thereby allowing the inter-node communication to remain role agnostic (Figure 2.3, 2.4, 2.5).

The rest of this section will further define these components, their interfaces, and the major design decisions made in each component to achieve the system goals.

4.1. External Interface

For the initial implementation of DADB, the authors decided to adopt the OASIS Emergency Data Exchange Language-Distribution Element (EDXL-DE) [3] as the primary external interface for the system. This specification and its accompanying XML Schema outlined a generic data encapsulation format. This standard is

robust enough to mandate a unique identifier for each message and encapsulate both XML and binary content.

EDXL-DE was chosen because the structure of this standard supported the architecture's goals to be translation-capable, geographically aware, and role agnostic. Translation capability is supported by specifying encapsulated content as either XML content or binary content. Binary content is defined by a mime type and an optional URL. The XML content could be passed directly from the EDXL-DE message itself. The use of this supporting metadata simplifies the problem of determining the content's data format in order to perform a translation.

EDXL-DE supported geographic awareness through the use of its targetArea element. This element specifies the geographic location where the message is to be sent. It gives the option for several methods to represent a geographic area. These methods and their corresponding standards are shown in Table 1.

Table 1. TargetArea elements

TargetArea Element	Standard	Example
polygon	WGS84	42,-124 42,-120 39,-120 35,-114 34, -120
circle	WGS84	25, -110 10
country	ISO-3166-1	US
subdivision	ISO-3166-2	US-SC
locCodeUN	UN/LOCODE 2006-2	USCHS

EDXL-DE supports the role agnostic goal by providing a recipientRole element but making the use of that element optional. Since the standard does not come with a set of enterprise roles, the choice for those roles is left to the given COI.

4.2. Geographic Discovery

By its nature, the discovery of a service based upon the geographic area it supports is a natural extension of service discovery research. While it was not in the scope of this paper to support explicit geographic queries about services by extending the UDDI specification, it is a natural area of inquiry for future work. The authors constrained the work to create a discovery mechanism using categories and classifications retrieved from UDDI. These classifications were used to describe DADB nodes based upon the same geographic representations defined for an EDXL-DE message. This was achieved by creating Technical Models (tModels) or taking advantage of existing tModels that provide geographic information. A tModel already existed for ISO 3166, and the authors derived a checked tModel for UN/LOCODE. Unchecked tModels were created for WGS84 polygons and circles. A server side geographic query engine was developed to interact with the UDDI server and retrieve services with these attributes from the specified tModels. This query engine cached service information from the UDDI server to offset the cost of repeated query for every user request. A set of intersection algorithms was used to determine whether the targetArea of an EDXL-DE message overlapped the geographic area associated with a given service node in the cache. If there was an intersection, the message was flagged so it would be routed to the given node.

Beyond obviously supporting geographic awareness, this piece of the architecture was developed using dynamic reflection. The authors recognized the needs of a COI to replace or update the intersection algorithms or the mechanism to access the UDDI server.

4.3. Dissemination Engine

Dissemination is the transition for a message which has been moving between DADB nodes to an end user or system. Within DADB, there are several criteria to disseminate a message to the end user. First the user must have an interest in the content the message contains. This implies the user also has a mechanism through which they can view the message content, or a translation is available from the message content to a content the user is able to view. Once interest is established, the dissemination engine must determine the authorization of the user to view the content. Once authorization is established, any necessary translations take place and the message is disseminated.

While defining the overall logic for the dissemination engine, the authors noted the COI-specific nature of many of the decisions made within it. As Figure 3 shows, a dissemination engine will need to deal with node specific, user specific, and translation policy.

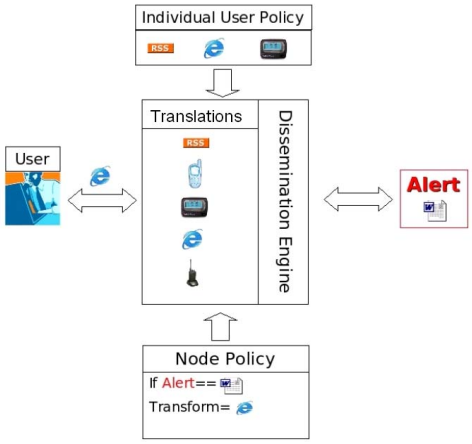

Figure 3. Dissemination engine.

Since this part of the architecture was extremely dependent on COI or node specific choices, the authors

chose to implement both the Dissemination API and the Translator API using interfaces and dynamic reflection. For dissemination, this means that that a COI can specify enterprise roles and each participating organization can specify in their implementation of the dissemination engine how these roles map to their identity infrastructure. For translation, it means that an organization can maintain a set of translators for their specific needs on their node, independent of the business logic of the rest of the enterprise. In addition, these design decisions help the system to fulfill its modular, translation-capable, and role agnostic design goals.

```
if (intersectsNodeAOR &&
messageHasLocalSender)
{
    route && disseminate
}
else if (intersectsNodeAOR &&
!messageHasLocalSender)
{
    disseminate
}
else if (!intersectsNodeAOR &&
!messageHasLocalSender)
{
    route
}
else if (!intersectsNodeAOR &&
messageHasLocalSender)
{
    route
}
```

Figure 4. Pseudo code for message dissemination and routing

4.4 Inter-Node Routing

The last area to discuss in the development of DADB is the inter-node routing of messages. The current logic to route a message is based on the assumption that a node will be able to distinguish between a message coming from a sender within its geographic area versus outside its geographic area. Figure 4 shows the pseudo-code for the conditional statements that determine how a message is handled.

This simple piece of logic shows that the only case where a message is routed and disseminated is where its targetArea intersects the node's area of responsibility and the message's sender is local to the node. If the sender is remote to the node, and the message's targetArea intersects, then the message is disseminated. This logic is used to keep nodes from entering a spanning tree loop as messages are routed around the network. Obviously, this logic is rudimentary and does not cover the full complexity of routing within a distributed system. The authors discuss these requirements and the best routing protocol in the next section of the paper.

5. Lessons Learned

By developing DADB, the authors have derived lessons learned that are applicable to the alerts community and many other fields of study. This section is ordered by the major components of the architecture.

5.1. External Interface Lessons Learned

While it is outside the scope of this paper to discuss all the lessons learned while using EDXL-DE as an external interface for a DADB node, the authors feel it is imperative to share some of the implications of using layered standards. As stated in Section 3.1, EDXL-DE makes use of several public standards for geographic representation. Implementers must take note of the effects these standards will have when designing their own architectures. The best example of this came from EDXL-DE's use of UN locator codes. The UN locator codes were grouped in key/value pairs where the key was the UN/LOCODE and the value was city or region name. While it was expected to find the same value (e.g., a city named Broughton) for multiple UN/LOCODE keys, the authors were surprised to find the same key attached to multiple values. An example would be the use of the UN/LOCODE, IT VRC, with the values 'Verucchio' and 'Villa Verucchio'. While the UN/LOCODE referred to the same geographic area, this system made it hard to map the specification to standard data structures and algorithms in the development process. Information like this should be included in the errata of the EDXL-DE specification to expose what could be troublesome implementation details to adopters.

5.2 Geographic Discovery Lessons Learned

Section 3.2 discussed the use of UDDI for geographic discovery. The authors chose this method to maintain one repository where the web services and the geographic information those services use could be discovered. There are shortcomings to this method. The most noticeable one is the processing power required on a geographic cache miss. Since UDDI will not allow for wildcard searches on unchecked taxonomies, the geographic query service must retrieve all DADB Nodes from UDDI server and then determine geographic intersection for each node. This leads to a processing requirement, P, with the following big O notation: $P \equiv O(n^2)$.

While there are many potential paths to solve this problem, the authors submit the following two to invoke further discussion. The first is for OASIS community to formally adopt a change to the UDDI specification. This

would provide an optional geographic area of responsibility for a service and operations for intersection mechanisms. The second possible path is to create a separate geographic discovery specification which implements a web service interface. Information for this service would be discoverable in UDDI.

5.3. Dissemination Engine Lessons Learned

The major lesson learned from implementing the dissemination engine was the need for even greater modularity and extensibility. While the implementation used reflection, the authors realized every COI or node owner would not want to implement their own dissemination engine. At best they would want to specify their own business logic into a generic module. To fulfill this requirement, a future implementation of the dissemination engine would need to implement a rules engine. Such an engine would allow a business process expert to express his COI's business rules in a simple and easy to understand grammar.

5.4. Inter-Node Routing Lessons Learned

Inter-node routing contained the most rudimentary implementation in this first iteration of the architecture. As such, its next steps and lessons learned are the most involved.

DADB is designed to route messages based on message level details instead of a direct target. Each node within DADB is responsible for evaluating the dissemination and routing of a message. A system or user publishing a message may or may not know all of the recipients; moreover, the system or user sending the message should not have to know all recipients in order to send a message. Message routing and node distribution are inherently overlapping system design parameters. The mechanism by which nodes are distributed determines the means by which a message is routed. Likewise, the routing requirements impact the node distribution mechanism.

The first iteration of DADB focused on routing based on geographic parameters inherent to the message. Future iterations must support a hierarchical structure where a geographic region is the primary unit to group nodes within the architecture. Government agencies at the federal, state, and local levels must communicate across geographic regions. Using a hierarchal distribution of nodes, messages can be distributed across geographic regions without having to be aware of every node within the region. Regions should be defined within UDDI for each node and should optionally be dynamically generated. Similarly, the dynamic nature of COI and ad-hoc efforts across agencies can be facilitated with the dynamic regions and groupings.

The authors provide Figure 5 as an example of this point. In this figure, N represents a top level regional node, such as a state. N(x), N(y), and N(p) are each subordinate nodes of N, counties for example. If a user/system from a county N(x) sends a message for its own county, then N(x) sends the message back down to the dissemination points (DP) within the county. Should N(x) receive a message from a local user that is destined for a region outside of its county then N(x) knows to route that message to N, the higher level node at the state level. Should N(x) receive a message from a user not local to its county but destined for its county, it will only disseminate the message downward. The final routing results in messages being hierarchically routed, reducing the overall burden of any given node. Regions can be defined dynamically and arbitrarily deep; that is to say, for any N node, it may have a higher level node H, dissemination point P, or node Y. The resulting hierarchical structure can be dynamically generated or reconfigured without changing the underlying logic within any node. Likewise, any node can be both a higher level node and a subordinate node, facilitating both coarse and fine grained regional definitions. The authors feel that the dynamic nature of government agencies requires a configurable routing methodology that does not burden the networks with duplicate messages.

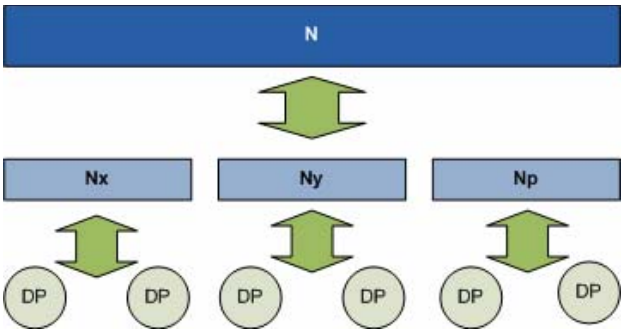

Figure 5. Hierarchical DADB nodes

6. Conclusions

Through the development and implementation of DADB, the authors have begun to investigate a new paradigm in alerting systems. It is our belief that architectures of this type will play an important part of providing interoperability for local, state, and federal agencies. By providing a modular, geographic aware, role agnostic, and translation capable system, DADB is flexible enough to adapt to the changing mission needs and COI requirements that abound within government organizations.

Beyond the specific benefits of the architecture as a whole, the modular design of components gives implementers a unique set of tools for both dissemination and translation of information.

As this work progresses, various user communities will find additional benefit in a refined geographic decision service either through the extension of UDDI or an altogether separate specification.

The development of DADB focused on the use of EDXL-DE as a key building block for the interface of the system. Through this publication, the authors believe we have shown some of the versatility of this specification. As the usefulness of the specification becomes better known, adoption of architectures like DADB will become more prevalent both for the first responder community and governments in general. To help facilitate the understanding of the standard, the authors provide additional evaluations of the standard in [15].

Finally, the background for this work showed the continuing and somewhat ad-hoc development of standards and systems by the US government to achieve interoperability across agency boundaries. Continuing industry and academic support will be needed to transition from information sharing policy to interoperable production systems.

7. References

[1] "Crisis Information Management Software (CIMS) Feature Comparison Report", National Institute of Justice, US, October, 2002.

[2] "Common Alerting Protocol", OASIS Standard, Version 1.1, http://www.oasis-open.org/committees/download.php/15135/emergency-CAPv1.1-Corrected_DOM.pdf.

[3] "Emergency Data Exchange Language (EDXL) Distribution Element", OASIS Standard, Version 1.0, http://docs.oasis-open.org/emergency/edxl-de/v1.0/EDXL-DE_Spec_v1.0.pdf,.

[4] Horizontal Fusion, http://horizontalfusion.dtic.mil/, US Department of Defense.

[5] G.S. Smith, "IT Architecture for Homeland Security", MILCOM 2005, October 2005.

[6] "9-11 Commission Report", http://www.gpoaccess.gov/911/Index.html, July 2004.

[7] "Homeland Security Act of 2002", http://www.dhs.gov/xlibrary/assets/hr_5005_enr.pdf, November 2002.

[8] "The Federal Government needs to Establish Policies and Processes for Sharing Terrorism-Related and Sensitive but Unclassified Information", Government Accountability Office, http://www.gao.gov/new.items/d06385.pdf, March 2006.

[9] James X. Dempsey, "Overview of Criminal Justice Information Systems," CFP, April 2000

[10] "Homeland Security Information Network Needs to Be Better Coordinated with Key State and Local Initiatives", Government Accountability Office, http://www.gao.gov/new.items/d07822t.pdf, May 2007.

[11] "Emergency Data Exchange Language Resource Messaging" OASIS Public Review Draft 02, Version 1.0, http://docs.oasis-open.org/emergency/edxl-rm/v1.0/pr02/EDXL-RM-SPEC-V1.0.pdf

[12] "Emergency Data Exchange Language Hospital AVailability Exchange", OASIS, Public Review Draft 05, Version 1.0, http://docs.oasis-open.org/emergency/edxl-have/pr05/emergency_edxl_have-1.0-spec-pr05.pdf

[13] "Introduction to the National Information Exchange Model", NIEM Program Management Office, http://www.niem.gov/files/NIEM_Introduction.pdf February 2007.

[14] "Global Justice XML Data Model", http://www.it.ojp.gov/jxdm/

[15] D. Pack, R. C. Coleman, "Assessing Interoperability in Emergency Management Standards", IEEE SouthEastCon 2008, April 2008