CAP Alerting
For Sahana Messaging Module

Real Time Biosurveillance Program
Software Requirements Specifications

Version 1.0

Authors

Gordon Gow & Nuwan Waidyanatha

Revision History

Date	Version	Description	Author
20. Feb. 2009	0.1	Contribution from RTBP alerting guideline	Gordon Gow
10. Mar. 2009	1.0	Include SRS specific literature	Nuwan Waidyanatha

TABLE OF CONTENTS

TABLE OF FIGURES

LIST OF TABLES

1. Introduction

The Real-Time Biosurveillance Program (RTBP) is a multi-partner research initiative that will study the potential for new Information and Communication Technologies (ICTs) to improve early detection and notification of disease outbreaks in Sri Lanka and India.

Experts in the field of biosurveillance and health informatics have argued that improvements in disease detection and notification can be achieved by introducing more efficient means of gathering, analyzing, and reporting on data from multiple locations. New information and communication technologies (ICTs) are regarded as a central means to achieve these efficiency gains. The primary research objective of the Real-time Biosurveillance Program (RTBP) is to examine these claims more closely by producing evidence to indicate in what ways and to what extent the introduction of new ICTs might achieve efficiency gains when integrated with existing disease surveillance and detection systems.

The RTBP research design includes the development of a testbed (pilot project) that will incorporate multiple methods to increase the efficiency of data collection and analysis. Under the current system in both Sri Lanka and India, patient data from regional and community health centres is gathered using paper-based forms and procedures. These forms are then sent to regional health officials where data analysis is carried out by qualified staff to identify potential disease outbreaks. Notifications are then issued from the regional health administrations to local authorities again using paper-based reporting methods.

The RTBP testbed will substitute each of these existing procedures with ICT-based components. Patient data will be gathered using software application implemented on handheld electronic devices and transmitted to a central server using a wireless data link. Data will be drawn from the central server and analysis will be carried out using advanced software developed by Carnegie Mellon University Auton Lab. Results will be made available to regional and local health officials as electronic notifications accessible through a variety of devices, including mobile phones.

These guidelines deal exclusively with the notification component of the RTBP testbed. Integration with data gathering and analysis will be essential for the project but each component has specific functional and operational considerations. It is the specific considerations for *alerting and notification* that these guidelines are intended to address.

Considerations related to integration of system components will be addressed in other documents.

In addition to the testbed component, the RTBP will also consider interoperability issues associated with national and international health-related organizations in the region of Sri Lanka and southern India. It is anticipated that implementation of the alerting and

notification component will provide important evidence regarding the opportunities and challenges associated with inter-jurisdictional alerting and notification for e-Health systems in the region.

The RTBP Alerting and Notification Guide is based on the US Center for Disease Control's Public Health Information Network (PHIN) Communication and Alerting Guide (PCA). The PCA Guide has been identified as useful model on which to base the RTBP Guide because it addresses the problem of inter-jurisdictional alerting, provides a comprehensive set of alerting attributes identified through extensive consultation by public health experts and expresses these attributes using Common Alerting Protocol (CAP) and Emergency Data Exchange Language (EDXL). RTBP will incorporate both CAP and EDXL as data interchange standards for use in the testbed in order to serve the primary objective of the project but also to take into account other objectives related to system growth and regional interoperability.

1.1. Objectives

The objective of this specification document is to provide a comprehensive description of the functional aspects of alerting and notification for the RTBP testbed. RTBP-compliant alert and notification messages must process, and manage alerts in a manner consistent with the requirements of this guide.

This implementation guide defines functional and technical specifications for any system intended to support RTBP Alerting and Notification. Readers are advised that RTBP is a research project and does not constitute an operational alerting system. As such, not all elements or functions of the RTBP testbed may be active or implemented at any specific time.

A primary contribution of this document is to define alerting attributes to be used in the RTBP testbed and how those attributes will be expressed using Common Alerting Protocol (CAP) Version 1.1, and the Emergency Data Exchange Language (EDXL) Version 1.0 Distribution Element.

Readers should have a basic conceptual understanding of CAP, EDXL, and XML in order to use this document.

2. RTBP Alerting and Notification Subsystems

The RTBP alerting and notification functionality will consist of five interconnected subsystems:

1. Message creation and validation
2. Message distribution
3. Message delivery
4. Message acknowledgement
5. Message system administration

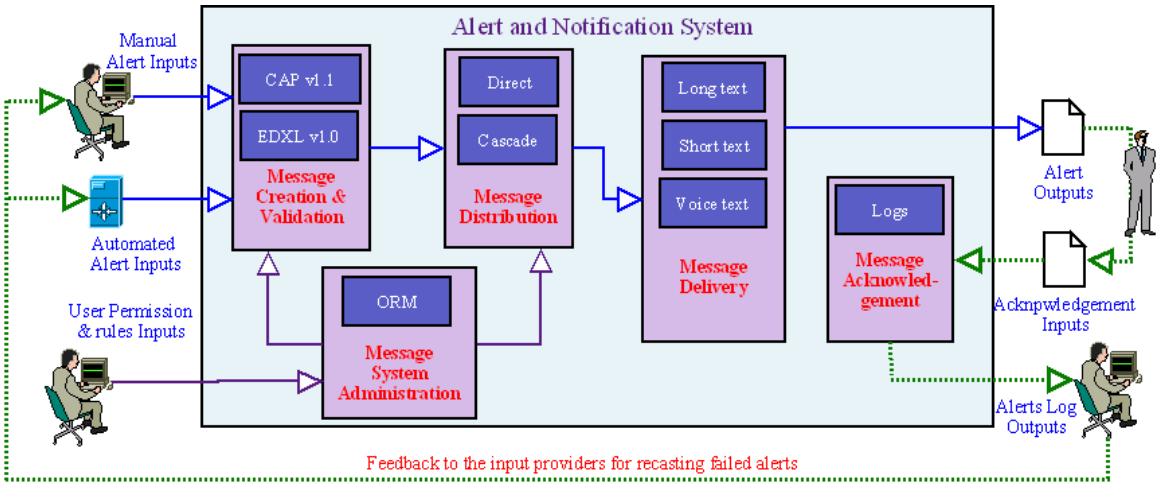

Figure 1 Alert and Notification subsystems with inputs and outputs

2.1. Message creation and validation

Message creation can be done manually or can be automated through middleware linked to the detection/decision component. Manual creation of messages requires one or more authorized users trained in policy and procedures for alerting and notification. A software interface and associated application(s) to support manual message creation will need to be implemented for the RTBP. The foundation for such an application is currently available with the open source Sahana Disaster Management System Messaging Module.[1] At present the Messaging Module provides a set of generic menus and inputs based on Common Alerting Protocol v1.1. A degree of customization of the module will be necessary for it to conform to RTBP alerting and notification attributes and vocabulary

[1] Sahana is a Free and Open Source Software platform that was created in the wake of the 2004 Indian Ocean Tsunami by the Lanka Software Foundation to address interoperability problems with existing disaster management systems. The CAP-based Messaging Module was later developed for the LIRNEasia HazInfo Project, but further development work is needed for its application in a biosurveillance context. For more information on Sahana see: http://www.sahana.lk/. For access to a demo version of the Messaging Module see: http://demo.sahana.lk/index.php?act=login.

as defined in this SRS.

Automated message creation will require a software application that is capable of formatting data provided by the detection and decision component and merging it with distribution and other data. A software application to support automated message creation will need to be identified or, more likely, custom designed for the project. This would entail opening a set of Messaging Module APIs for the external applications to directly use for issuing automated alerts. For example, the Auton Lab software components will run the detection algorithms as a services periodically (once a day), may detect a possible adverse event (i.e. disease outbreak). Then the Auton Lab software should be able to identify certain elements of the adverse event and use the Messaging Module APIs to communicate the necessary information related to the detected adverse event to health officials who engage in analysis and decision making. The aim of the alert is to get the analysis and decision making health officials to investigate the situation to take further action such as communicating the event to other health officials to notify them of the predicted threat.

RTBP research design at present does not include a provision for automated message creation. However, a generic API that provides a set of inputs to the Messaging Module can easily accommodate the automated alerting process; where the input would be similar to a CAP XML file and predefined EDXL Distribution Elements.

In either case, alerting and notification messages will be formed using pre-established attributes and vocabulary implemented in conformance with the Common Alerting Protocol (CAP) v1.1 standard and Emergency Data Exchange Language (EDXL) Distribution Element v1.0. Section 3 of this document provides detailed explanation of those elements and their implementation.

Message creation can be centralized or distributed depending on the alerting and notification architecture adopted for the RTBP testbed. In the context of manual message creation, a centralized system provides a degree of simplicity but may place limits on the ability of the system to simulate real-time 24/7/365 alerting and notification because it may place limits on the number of authorized users that are reasonably able to access the system.

A distributed architecture is more complicated but will provide more flexibility in terms of assigning authorized users and simulating a real-time 24/7/365 alerting and notification context. Distributed message creation will likely require access to the application interface through an Internet-enabled desktop PC; however, it may be possible to provide access using a wireless mobile handheld device—although this would add to the complexity of the project and possible delays during implementation and testing. Therefore, the primary phase of the RTBP will only implement the PC desktop and internet enabled alerting version.

A third possibility is a system based on hybrid manual/automated message creation. Using this approach, a set of threshold variables could be identified within the analysis component to prompt a software application to create and issue an alert message under certain conditions. For instance, an automated system might send a message only to authorized users for a pre-determined condition that requires their immediate attention. If upon closer inspection by an authorized user the condition were then to warrant a wide-area alert or notification to local health officials, then this could be created and issued manually by that authorized user or his/her designate.

In other instances, if a certain pre-determined critical threshold were to be exceeded at the outset then the automated system might be authorized to issue wide-area alerts directly to health officials without the manual intervention of an authorized user.

A hybrid system could be based on a combination of centralized and distributed architecture for message creation. An important variation on the manual/automated hybrid is the concept of *cascade alerting*, which will be described in more detail in the message distribution subsection below. Although cascade alerting is, strictly speaking, a distribution method it involves a process of automated message generation.

The hybrid system design presents important opportunities from a research standpoint but it adds complexity on several dimensions and would require close attention to policy and procedural issues.

Table 1 Summary of message creation options

Manual message creation	centralized architecture
	distributed architecture
Automated message creation	centralized architecture
	cascade alerting*
Manual/automated hybrid	centralized architecture
	distributed architecture
	cascade alerting*

2.2. Message distribution

Once an alert or notification has been created it must then be distributed to designated recipients. The PHIN PCA Guidelines describe two primary methods for message distribution:

Direct alerting *is the normal process in which an alerting system delivers an alert*

to a human recipient. This is the normal mode of alerting when the recipient works within the organization or its jurisdiction. However, direct alerting can also be used to accomplish cross-jurisdictional alerting: an alerting system in one jurisdiction sending messages to recipients within another jurisdiction.

***Cascade alerting** is a process in which an alert is sent as a system-to-system message from one jurisdiction to another; the receiving system then distributes the alert to the appropriate recipients within the receiving jurisdiction. The message contains the alert along with parameters describing how and to whom the message should be delivered. Cascade alerting is the preferred method for sending cross-jurisdictional alerts, but it requires greater technical sophistication to implement.*[2]

For the purpose of the RTBP testbed, direct alerting will likely be the primary focus for the initial work. However, considerations for cascade alerting should be included in design and development efforts with a view to possible testing of an implementation during a later stage in the project. Cascade alerting will likely have more relevance as cross-jurisdictional consideration come into play, and this may be an important consideration as the project evolves.

EDXL Distribution Element is an important consideration in message distribution and several attributes have been identified in this doc pertaining to it. While its full implementation may not be immediately relevant to the project, it is advisable to include provisions for it in the event that the project seeks to attempt more advanced testing with distribution and, in particular, cascade alerting.

2.3. Message delivery

Messages must reach their destination through an appropriate receiving device, be it a mobile phone, desktop PC, or other means. The contents of the CAP XML message must then be rendered into human readable form while taking into account the limits of bandwidth, processor capabilities, and message display constraints inherent in any particular device. A software application is implemented on the receiving device to carry out this function.

Message delivery options can be categorized into three types based on PHIN PCA designations:

- *Long text—content rendered in a form appropriate for email, fax, or web presentation;*
- *Short text—content rendered in a form appropriate for SMS and pagers;*
- *Voice text—content rendered in a form appropriate for voice delivery or*

[2] <United States Centers for Disease Control and Prevention, 2008 #3, p. 14>

automated voice delivery by telephone.[3]

At present RTBP make provisions for message delivery as Long text and Short text. Sahana Messaging Module currently has an Email and SMS push function but would require a web post function but compliant with CAP and EDXL.

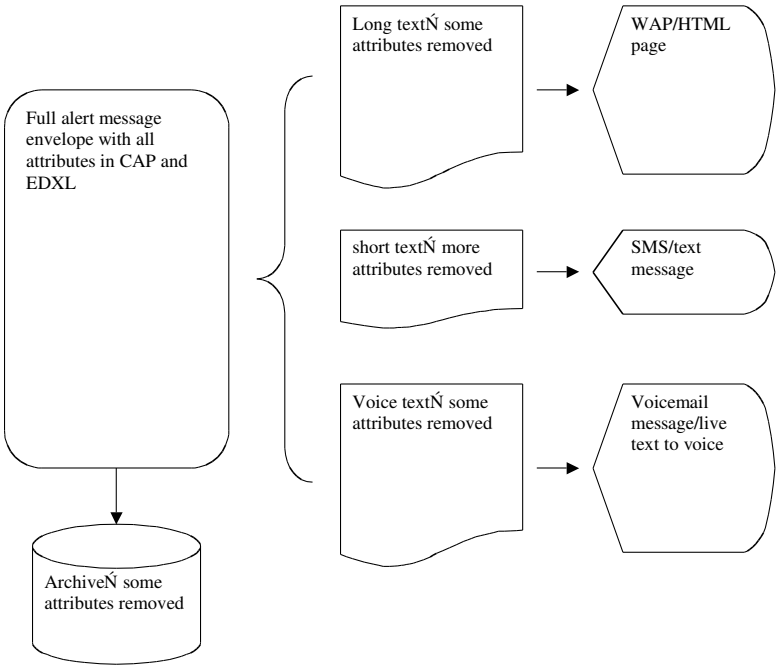

Figure 2: Rendering a CAP-XML message for end-user devices

Table 2 Suggested CAP elements and exmple for the delivery types

Message delivery type	***Terminal Device Display Attributes***	***Example***
Long Text	All elements of CAP	
	<*info.headline*>	Cholera outbreak is in effect
	<*area.areaDesc*>	Kurunegala District
	<*info.category*>	Health
	<*info.event*>	Disease outbreak
	<*info.senderName*>	Sarvodaya Suwadana Center
	<*alert.sender*>	suwacevo@sarvodaya.org
	<*alert.incident*>	200907240001
	<*alert.identifier*>	suwacevo-200807240001001
	<*alert.sent*>	2009-07-24T16:49:00+05:30
	<*alert.msgType*>	Alert
	<*alert.status*>	Exercise
	<*alert.scope*>	Restricted
	< *alert.restriction*>	This message is for registered health care workers
	<*info.language*>	English

[3] <United States Centers for Disease Control and Prevention, 2008 #3, p. 9>

	<*info.value*> <*info.valueName*> <*info.urgency*> <*info.severity*> <*info.certainty*> <*info.description*> <*info.web*> <*info.contact*>	Priority High Expected Moderate Observed This message is for an *Exercise* event, repeat this message is for an *Exercise* event - A *high priority Cholera outbreak is in affect* for the *Kurunegala District*. The *Health* alert for the *disease outbreak* was issued at *4:49pm on 24th day of July 2009* by the *Sarvodaya Suwadana Center*. For further instructions visit the website - http://www.sarvodaya.org/healthalert/ or call *+9411255566* - This message is for an *Exercise*, repeat this message is for an *Exercise* event. http://www.sarvodaya.org/healthalert/ +9411255566
Short Text	<*info.headline*> <*area.areaDesc*> <*info.value*> <*info.valueName*> <*info.category*> <*info.event*> <*info.senderName*> <*alert.identifier*> <*alert.sent*> < *alert.msgType* > < *alert.status*> <*info.web*> <*info.contact*>	Cholera outbreak is in effect Kurunegala District Priority High Health Disease outbreak Sarvodaya Suwadana Center sarvodaya-rtbp-2255 2009-07-24T16:49:00+05:30 Alert Exercise http://www.sarvodaya.org/healthalert/ +9411255566
Voice Text	<*info.headline*> <*info.description*>	Cholera outbreak is in effect This message is for an *Exercise* event, repeat this message is for an *Exercise* event - A *high priority Cholera outbreak is in affect* for the *Kurunegala District*. The *Health* alert for the *disease outbreak* was issued by the *Sarvodaya Suwadana Center*. For further instructions visit the *Suwadana Center Health Alert* website or call *their hotline number +9411255566* - This message is for an *Exercise*, repeat this message is for an *Exercise* event.

2.4. Message acknowledgement

Finally, in some cases it may be advisable or desirable to include a backchannel for message acknowledgement from recipients. This would require a communication link back to the message delivery system to collect acknowledgement receipts and present these as a report. The software must be capable of handling this function and a database needs to be established to store this data. Features and data capture associated with message acknowledgement would need to be defined. The PHIN PCA Guidelines include certain attributes for message acknowledgement, which could be adapted to support this function.

2.5. Message system administration

Message creation subsystem must also take into account security provisions such as user access control and originator rights management. This will require a database of user names, passwords, and ORM - Originator Rights Management (set of privileges and rules) profiles linked to the user interface software. The user can primarily be categorized by the domain expertise in relation to the predefine <*info.category*> and <*info.status*> values; where a "health" alert communication expert has the privilege only to create "Health" category and "Draft" message status templates. Similarly, only health workers with permissions to issue health alerts can issue "Health" category alerts but within that category of authorized persons the administrative subsystem can further restrict to issuing "Actual" or "Exercise" amerts.

For the purpose of the RTBP pilot the ORM profiles are not essential because only a handful of the health workers will have the user privileges to create templates and issue alert. Therefore, the developers should have some sort of an authentication mechanism with the full message system administration functions in mind.

3. Message Attributes

RTBP alerting and notification should adopt a design approach based on a fundamental principle of extensibility, meaning that it should be easy to integrate additional functionality into the system over time. Functionality could include the ability to support a range of new end-user devices not included in the initial stages of the project, to enable the introduction of advanced distribution capabilities (e.g., precision geo-targeting and routing of messages), and the ability to support cross-jurisdictional and cascade alerting and notification as a future enhancement to the system.

In support of extensible design, it is advised that RTBP follow the CDC PHIN PCA approach and adopt a set of standardized alerting attributes to support semantic compatibility across systems. These attributes provide a framework for shared vocabulary, predictable system response, and more broadly for identifying policy and procedural issues of interest for the research project.

Following the PHIN PCA approach, 'Alert Attributes' are semantic descriptors that are associated with specific functional elements and defined precisely using Common Alerting Protocol (CAP) and the Emergency Data Exchange Language (EDXL) Distribution Element.

3.1. Alert format

According to PHIN PCA Guidelines, 'a degree of standardization of alert format helps to ensure that public health organizations can communicate effectively within their jurisdictions and with other jurisdictions, especially during emergencies.'

Each alert should address a single issue or health event, rather than combining multiple issues and events into one alert.

For the initial stages of the project, alerts will be created using CAP v1.1. Message contents will be contained in the CAP envelope that will be transported using one or more transport protocols appropriate to the message distribution sub-system. Later stages of the RTBP project may include the addition of an EDXL envelope that adds an additional layer of information using the EDXL Distribution Element.

Each RTBP alert message entering the message distribution subsystem must include the following nine attributes:

1. Identity of the agency that issued the alert {*agencyIdentifier*}
2. Message identifier for tracking purposes {*alertIdentifier*}
3. Time and date that the message was sent from the issuing agency {*sendTime*}
4. Indication of whether it is an actual alert, exercise, or test {*status*}
5. Indication of whether it is an original alert, update, or cancellation of a previous alert {*msgType*}
6. Indication of the scope of distribution for the alert (i.e., public, restricted, private) {*scope*}
7. The priority of the message (i.e., urgent, high, low) {*priority*}
8. Indication of the event or incident type {*event*}
9. Contents of the alert message {*message*}

3.1.1. Message attribute {*agencyIdentifier*}

Each message must identify the agency that issued the alert. PHIN PCA Guide v1.0 refers to an "Object Identifer (OID)" of the originating agency and the future creation of an OID and ebXML registry for PHIN. However, OIDs are currently managed by PCA at CDC.

CAP v1.1 specifies this as a required sub-element within the alert element as <*alert.sender*>. CAP v1.1 further specifies that it must identify "the originator of this alert. Guaranteed by assigner to be unique globally; e.g., may be based on an Internet domain name" and "MUST NOT include spaces, commas or restricted characters (< and &)."

EDXL Distribution Element specifies this attribute as a required sub-element *senderID* and that it must be used only "once and only once" per message. It further specifies that it must be a "unique identifier of the sender" and indicates it must take the form "actor@domain-name", where "domain-name" refers to a valid address using the Internet Domain Name System. It must be a properly formed—escaped if necessary—XML string.

3.1.1.1. It is recommended that persons, organizations, and agencies authorized to issue alerts within the RTBP project be assigned unique identifiers based on a valid and appropriate Internet domain name (e.g., RTBP@lireasia.org).

3.1.1.2. There is a need to approve an acceptable and reliable naming convention with participants and to establish a registry of persons, organizations, and agencies that are authorized to issue alerts for the RTBP project. Further investigation into PCA's proposed approach to OID and ebXML registry may be useful for latter stages of the project including a cross-jurisdictional workshop.

3.1.2. Message attribute {*alertIdentifier*}

Each RTBP alert message must include a unique identifier.

PHIN PCA Guide v1.0 does not specify an encoding requirement for this attribute; however, it notes that "every alerting program must have a unique namespace and its own protocol for generating unique alert identifiers."

CAP v1.1 specifies this as a required sub-element within the alert element as <*alert.identifer*>. CAP v1.1 further specifies that it must be "a number or string uniquely identifying this message, assigned by the sender" and "MUST NOT include spaces, commas or restricted characters (< and &)."

EDXL Distribution Element specifies this attribute as a required sub-element *distributionID* and that it must be used only "once and only once" per message. It further specifies that it must be a unique identifier, wherein "that uniqueness is assigned by the sender to be unique for that sender." It must be a properly formed—escaped if necessary —XML string.

3.1.2.1. It is recommended that RTBP establish a convention for generating and assigning the attribute {*alertIdentifier*}. This must conform to CAP v1.1 and EDXL Distribution Element standards. Participants and authorized issuers should be encouraged to adopt that convention when issuing alerts over the system.

3.1.2.2. There is a need to approve an acceptable and reliable message identity convention within the RTBP system.

3.1.3. Message attribute {*sendTime*}

Each RTBP message must include the time and date that it was first issued. PHIN PCA Guide v1.0 specifies that this attribute is to be encoded using ISO 8601 format, which corresponds with CAP v1.1 requirement (see below).

CAP v1.1 specifies this as a required sub-element within the alert element as <*alert.sent*>. CAP v1.1 further specifies that it must be "represented in [dateTime] format (e.g., "2002-05-24T16:49:00-07:00" for 24 May 2002 at 16:49 PDT)" and that "Alphabetic timezone indicators such as 'Z' MUST NOT be used. The timezone indicator for UTC MUST be represented as '-00:00' or '+00:00."

EDXL Distribution Element specifies this attribute as required sub-element *dateTimeSent* and that it must be used only "once and only once" per message. It further specifies that it must include the offset for the time zone from where it was sent and "must be in the W3C form for the XML [dateTime] data type."

3.1.3.1. It is recommended that RTBP adopt the ISO 8601 dateTime standard format, taking into account any other considerations related to the W3C form for XML dateTime. This must conform to CAP v1.1 and EDXL Distribution Element standards.

3.1.3.2. It is recommended that the ISO 8601 format be embedded in the message creation sub-system software to eliminate need for individuals to enter this data themselves.

3.1.3.3. It is recommended that assignment of the {*sendTime*} attribute should be done automatically by the message creation sub-system at the moment the message is sent to the distribution subsystem. The {*sendTime*} attribute should NOT be assigned by the message creation sub-system in order to avoid confusion in situations where a message is created as a preliminary template or draft.

3.1.3.4. There is a need to examine potential issues with time zone differences and identify a reliable source for the dateTime data feed.

3.1.4. Message attribute {*status*}

Each RTBP alert message must indicate whether it is an actual alert, exercise, or test.

PHIN PCA specifies enumeration values of "Actual" (referring to a live event), "Exercise" (indicates that designated recipients must respond to the alert as part of an exercise), "Test" (indicates that the message is related to a technical system test and should be disregarded by recipients).

CAP v1.1 specifies this as a required sub-element within the alert element as <*alert.status*>. CAP v1.1 further specifies that it be represented as one of five designated code values, each with specific meaning and intent:

"Actual"—actionable by all targeted recipients
"Exercise"—actionable only by designated exercise participants
"System"—for messages that support alert network internal functions
"Test"—technical testing only, all recipients disregard
"Draft"—a preliminary template or draft, not actionable in its current form

CAP v1.1 recommends that <*alert.note*> sub-element be used to provide an exercise identifier when message is assigned "Exercise" status.

EDXL Distribution Element specifies this attribute as required sub-element *distributionStatus* and that it must be used only "once and only once" per message. This attribute conveys the actionability of the message with one of the following values:

Actual—"Real-world" information for action
Exercise—simulated information for exercise participants
System—message regarding or supporting network functions
Test—discardable messages for technical testing only

The status MUST be a properly formed—escaped if necessary—XML string.

3.1.4.1. It is recommended that RTBP adopt the CAP v1.1 code values and definitions for the {status} attribute.

3.1.4.2. It is recommended that message creation software provide a menu choice "Draft" in addition to the other status values to enable the creation of preliminary templates.

3.1.4.3. It is recommended that message creation software be designed to prevent messages with "Draft" status from being sent to the distribution sub-system.

3.1.4.4. There is a need to consider establishing unique identifiers for exercises and simulations that are distinct from actual alerts.

3.1.4.5. There is a need to establish clear specifications and rules for using the values "System" and "Test" within the scope of the RTBP project.

3.1.5. Message attribute {*msgType*}

Each RTBP message must indicate whether it is an original alert, update, or cancellation of a previous alert.

PHIN PCA specifies enumeration values "Alert" (to indicate an original alert), "Update" (to indicate that a prior alert has been update and superseded), "Cancel" (to indicate that a prior alert has been cancelled), "Error" (to indicate that a prior alert has been retracted).

If {*msgType*} is "Update", "Cancel" or "Error" then the message attribute {*reference*} must be included in the message to provide a unique identifier of the message being updated, cancelled, or issued in error.

CAP v1.1 specifies this as a required sub-element within the alert element as <*alert.msgType*>. CAP v1.1 further specifies that is be represented as one of five designated code values, each with specific meaning and intent:

"Alert"—initial information requiring attention by targeted recipients
"Update"—updates and supersedes the earlier message(s) identified in <references>

"Cancel"—cancels the earlier message(s) identified in <references>
"Ack"—acknowledges receipt and acceptance of the message(s) identified in <references>
"Error"—indicates rejection of the message(s) identified in <references>

CAP v1.1 requires that <*alert.references*> subelement be used to provide an unique message identifier when message type is "Update", "Cancel", "Ack", or "Error".

CAP v1.1 suggests that <*alert.note*> sub-element be used to provide an explanation when message type is "Error".

EDXL Distribution Element v1.0 specifies this attribute as required sub-element *distributionType* and that it must be used only "once and only once" per message. This attribute conveys the function of the message with one of the following values:

Report—new information regarding an incident or activity
Update—updated information superseding a previous message
Cancel—a cancellation or revocation of a previous message
Request—a request for resources, information or action
Response—a response to a previous request
Dispatch—a commitment of resources or assistance
Ack—acknowledgment of receipt of an earlier message
Error—rejection of an earlier message (for technical reasons)

EDXL Distribution Element v1.0 also specifies that "the distribution type applies to the function of the content objects as a set. Those cases where payloads have different distribution types should be clustered in different distribution elements."

EDXL Distribution Element v1.0 requires that *distributionReference* sub-element be used to provide an unique message identifier when message type is "Update", "Cancel", "Ack", or "Error".

The *distributionType* MUST be a properly formed—escaped if necessary—XML string.

3.1.5.1. It is recommended that RTBP adopt the CAP v1.1 code values and definitions for <*alert.msgType*> within the CAP envelope.

3.1.5.2. It is suggested that RTBP adopt EDXL Distribution Element v1.0 code values and definitions for message distribution, mapped appropriately to the CAP v1.1 values for the EDXL envelope (e.g., "Alert" is equivalent to "Report"; "Update is equivalent to "Update).

3.1.5.3. It is suggested that it is not necessary for RTBP to implement the code values "Request", "Response" and "Dispatch" at this time.

3.1.5.4. There is a need to establish a procedure and rules for issuing various message types, with particular guidelines for updates, cancellations, and errors.

3.1.5.5. There is a need to establish a method for generating and assigning <*alert.reference*> (CAP) and *distributionReference* (EDXL) code values when required.

3.1.6. Message attribute {*scope*}

Each RTBP message must indicate the scope of distribution for the alert (i.e., public, restricted, private).

PHIN PCA Guide v1.0 specifies that "PHIN alerting systems should always use the value 'Restricted', meaning 'for dissemination only to users with a known operational requirement.'" This is not a required attribute in PCA Guide v1.0 but it is acknowledged that the attribute must be included to produce valid XML messages conforming to CAP/EDXL.

CAP v1.1 specifies this as a required sub-element within the alert element as <alert.scope>. CAP v1.1 further specifies that is be represented as one of three designated code values, each with specific meaning and intent:

"Public"—for general dissemination to unrestricted audiences
"Restricted"—for dissemination only to users with a known operational requirement
"Private"—for dissemination only to specific addresses

CAP v1.1 requires that sub-element <*alert.restriction*> be used when the scope value is "Restricted." The <*alert.restriction*> sub-element is therefore conditional and contains "text describing the rule for limiting distribution of the restricted alert message."

CAP v1.1 requires that sub-element <*alert.addresses*> be used when the scope value is "Private." The <*alert.addresses*> element is therefore conditional and contains "the

group listing of intended recipients of the private alert message." CAP v1.1 specifies certain rules for this sub-element: "each recipient SHALL be identified by an identifier or address", "multiple space-delimited addresses MAY be included. Addresses including whitespace MUST be enclosed in double-quotes."

There is no corresponding EDXL Distribution Element identified in PHIN PCA Guide v1.0 for the message attribute {*scope*}. However, the required sub-element *combinedConfidentiality* might serve to provide similar a semantic equivalent for the EDXL envelope. PHIN PCA Guide v1.0 in fact specifies this EDXL sub-element for the required message attribute {*sensitive*} with enumeration values of "Sensitive" and "NotSensitive".

3.1.6.1. It is recommended that RTBP adopt the CAP v1.1 code values and definitions for <*alert.scope*> within the CAP envelope.

3.1.6.2. It is recommended that RTBP adopt a rule whereby all messages issued within the scope of the project be designated as "Restricted" or "Private".

3.1.6.3. It is recommended that RTBP message creation software provide menu options only for "Restricted" or "Private" messages, with future provision for "Public" messages.

3.1.6.4. There is a need to create text to describe the rule for limiting distribution of "Restricted" alert messages conforming to CAP v1.1 sub-element <*alert.restricted*>.

3.1.6.5. There is a need to establish a registry of addresses for intended recipients of messages designated as "Private" conforming to CAP v1.1 sub-element <*alert.addressess*>.

3.1.6.6. There is a need to consider the use of EDXL sub-element combinedConfidentiality as an equivalent to <*alert.scope*>. If adopted, then it is recommended that the code value "Sensitive" be assigned to all messages.

3.1.6.7. There is a need to establish a procedure and rules for assigning RTBP messages as "Restricted" or "Private".

3.1.7. Message attribute {*priority*}

Each RTBP message must indicate the priority of the alert.

PHIN PCA Guide v1.0 does not specify an equivalent message attribute {*priority*} but includes three related message attributes: *severity*, *urgency*, *certainty*. Of these, *severity* is the only required attribute. Code values for these attributes are to follow CAP v1.1 enumeration values for corresponding CAP sub-elements.

CAP v1.1 establishes message priority with the info element using three required sub-

elements: <*info.urgency*>, <*info.severity*>, <*info.certainty*>. All three elements must be included to produce a valid CAP-XML document.
CAP v1.1 specifies the following code values for the sub-element <*info.urgency*>:

"Immediate"—responsive action should be taken immediately
"Expected"—responsive action should be taken soon (within next hour)
"Future"—responsive action should be taken in the near future
"Past"—responsive action is no longer required
"Unknown"—urgency not known

CAP v1.1 specifies the following code values for the sub-element <*info.severity*>:

"Extreme"—extraordinary threat to life or property
"Severe"—significant threat to life or property
"Moderate"—possible threat to life or property
"Minor"—minimal threat to life or property
"Unknown"—severity unknown

CAP v1.1 specifies the following code values for the sub-element <*info.certainty*>:

"Observed"—determined to have occurred or to be ongoing
"Likely"—likely (p > ~50%)
"Possible"—possible but not likely (p <= ~50%)
"Unlikely"—not expected to occur (p ~ 0)
"Unknown"—certainty unknown

There is no corresponding EDXL Distribution Element identified in PHIN PCA Guide v1.0 for the message attribute {*priority*} or for the corresponding CAP sub-elements noted above.

The LIRNE*asia* HazInfo[4] project established message priority by adopting a bundled approach that used pre-defined code values for each of the CAP sub-elements noted above. Details of this approach are provided in section 5 of this document.

3.1.7.1. It is recommended that RTBP adapt the message prioritization scheme developed for the LIRNEasia HazInfo project.

3.1.7.2. It is recommended that RTBP message creation software provide users with a limited menu of choices based on this message prioritization scheme to enhance reliability and simplicity.

3.1.7.3. There is a need to establish guidelines for assigning priority levels to RTBP alerts.

[4] HazInfo technical report can be found here – http://www.lirneasia.net/

3.1.8. Message attribute {*event*}

Each RTBP message must indicate the event or incident type.

PHIN PCA Guide v1.0 does not specify a message attribute {event} but includes two related message attributes: *alertProgram* and *category*. Of these, only *alertProgram* is a required message attribute and is specified using CAP v1.1 required sub-element <*info.event*>. Enumeration values for this attribute refer to specific PHIN alerting programs (e.g., HAN, Epi-X). The attribute *category* is specified using the CAP v1.1 required sub-element <*info.category*> and is always enumerated as "Health." However, for the general design of a CAP enabled messaging module this element can give the user the option to select from any predefined <*info.category*> element values such as *Geo, Met, Safety, Security, Rescue, Fire, etc.* An approach would be to maintain a configuration file that specifies whether it is a general implementation (i.e. *config.info.category* = "default") or a domain specific implementation (i.e. *config.info.category* = "Health"). If configuration value is set to default then the template creation or message creation form control will display all predefined <*info.caegory*> values; else if set to "Health", for example, would display only that value blocking the template or message creator from editing it.

CAP v1.1 specifies that all messages contain sub-elements <*info.category*> and <*info.event*>. Sub-element <*info.category*> denotes the general category of the subject event of the alert message and must correspond to a range code values specified in CAP v1.1 standard. For the RTBP project, the code value "Health" is appropriate.

The code value for sub-element <*info.event*> is to provide "the text denoting the type of the subject event of the alert message" and is intended to be more specific than the <*info.category*> sub-element. CAP v1.1 does not provide specific code values.

There is no corresponding EDXL Distribution Element identified in PHIN PCA Guide v1.0 for the message attribute {*event*} or for the corresponding CAP sub-elements noted above.

3.1.8.1. It is recommended that CAP v1.1 sub-element <*info.category*> be specified as "Health" for all RTBP alert messages.

3.1.8.2. It is recommended that RTBP message creation software automatically assign all RTBP alerts as "Health" messages using CAP v1.1 <*info.category*>.

3.1.8.3. It is recommended that CAP v1.1 sub-element <*info.event*> be included in all RTBP alert messages to ensure CAP-XML compliance.

3.1.8.4. It is recommended that RTBP message creation software provide a list of one or more RTBP-designated events corresponding to the foreseeable subject events of potential alert messages.

3.1.8.5. There is a need to develop an event list (containing one or more events) suited to the purpose of the RTBP project. There is a need to consider management of an event list registry.

3.1.9. Message attribute {*message*}

Each RTBP alert message must include a description of the alert.

PHIN PCA Guide v1.0 refers to this as "the main message text" and specifies CAP v1.1 required sub-element <*info.description*> to convey this information. It is a required attribute in PHIN PCA Guide v1.0.

CAP v1.1 does NOT require messages to include the info sub-element <*info.description*>. The element is specified as "an extended human readable description of the hazard or event that occasioned this message."

CAP v1.1 also includes an optional info sub-element <*info.headline*> that provides "a brief human-readable headline … that SHOULD be made as direct and actionable as possible while remaining short. 160 characters MAY be a useful target for headline length."

In addition, CAP v1.1 includes an optional info sub-element <*info.instructions*> that provides "extended human readable instructions to targeted recipients" that describes "recommended action to be taken by recipients of the alert message." PHIN PCA Guide v1.0 specifies this sub-element for an optional message attribute *dissemination* intended to provide instructions for sharing message information beyond the initial intended recipient.

3.1.9.1. It is recommended that RTBP adopt CAP v1.1 info sub-element <*info.description*> to convey a human readable description of the event that occasioned the alert message.

3.1.9.2. It is recommended that RTBP adopt CAP v1.1 info sub-element <headline> to convey a brief human readable message under 160 characters describing the event that occasioned the alert message.

3.1.9.3. It is recommended that RTBP include consideration of CAP v1.1 info sub-element <*info.instructions*> for future implementation.

3.1.9.4. There is a need to develop procedures and guidelines for message texts pertaining to various alerts that will be issued during the RTBP project.

3.1.9.5. There is a need to ensure that message delivery software will correctly and reliably render message contents from <*info.description*> and <*info.headline*> sub-elements to correspond with long text, short text, and voice messages.

Note – there are two options to implementing the logic for short and long text messages. The CAP messaging module would provide a feature for the implementers to assign various CAP elements to the various fields of the transport and display technologies. To give an example – the implementers can assign the <*info.headline*> human readable element restricted to 160 characters, to the message box of an SMS text. While a HTML website post could carry both the <*info.headline*> and <*info.description*> element values.

4. Message Prioritization with the HazInfo Project

The following section is an excerpt taken from *Guidelines for HIH Procedures, System Activation, and Testing* v1.2.1 (July 12, 2006)[5] developed for the LIRNE*asia* HazInfo Project.

The excerpt describes how the CAP sub-elements are pre-assigned code values and bundled to create a reliable and simplified message prioritization scheme.

4.1. Message Priority

When reporting an event of interest (EOI) an authorized user may request the HIH Executive to issue an **Urgent Priority** warning message when one or more of the following threat conditions are present:

- The life or safety of groups, communities or villages is at immediate risk.

[5] Guidelines for HazInfo can be found here – http://www.lirneasia.net/cap-guidelines-hazinfo

- The danger to the community is impending and widespread.
- The potential impact to the community is catastrophic.
- Local first responders needs to be informed of critical, life saving information and be advised to activate their local response plans.

The following table contains recommended CAP values for <urgency>, <severity>, and <certainty> elements in an urgent priority message.

Table 3 CAP values for an urgent priority message

CAP <info> element	Value (recommended)	Interpretation
Urgency	"Immediate"	Immediate responsive action should be taken
Severity	"Extreme"	Hazard presents an extraordinary threat to life or property
Certainty	"Observed"	The hazard event has occurred or is ongoing (or, > 50%).

Alternately, an authorized user may request the HIH Executive to issue a **High Priority** warning message when one or more of the following threat conditions are present:

- The life or safety of communities or villages is possibly at risk.
- Neighbouring communities or villages have been issued an urgent priority warning.
- Residents of the community may see/hear/smell (detect) signs of the hazard and may perceive a danger or health risk.
- Local first responders need to be informed of the hazard situation to provide information to community members.
- Local first responders must be advised to standby to activate their local response plans.

The following table contains recommended CAP values for <urgency>, <severity>, and <certainty> elements in a high priority message.

Table 4 CAP values for a high priority message

CAP <info> element	Value (recommended)	Interpretation
Urgency	"Expected"	Responsive action might need to be taken in near future.
Severity	"Severe"	Hazard presents a significant threat to life or property.
Certainty	"Observed"	The hazard event has occurred or is ongoing (or, > 50%).

An authorized user may request the HIH Executive to issue a **Low Priority** warning message when one or more of the following conditions are present:

- The life or safety of a community might be at risk due to a developing hazard.
- A neighbouring community has been issued a high priority warning.
- Residents of the community may see/hear/smell (detect) signs of a hazard or nearby response effort and may be curious.
- Local first responders need to be informed of the hazard situation to provide information to community members.
- Local first responders must be advised to standby for further information.

The following table contains recommended CAP values for <urgency>, <severity>, and <certainty> elements in a low priority message.

Table 5 CAP values for a low priority message

CAP <info> element	Value (recommended)	Interpretation
Urgency	"Expected"	Responsive action might need to be taken in near future.
Severity	"Moderate"	Hazard presents a minimal threat to life or property.
Certainty	"Observed"	The hazard event has occurred or is ongoing (p > 50%).

4.2. Simplifying entry of priority for the user

One of the configurations would defining the <*info.urgency*>, <*info.severity*>, and <*info.certainty*> mapping to urgent, high, and low priority elements.

5. Auto Generate the Message Attribute {Description}

When creating the CAP template the user can be given the choice to auto generate the information for the <*info.description*> through a template designed for the <*info.description*> element; where incident specific values are populated from other elements of the CAP message.

"A ***<info.event>*** alert has been issued for ***<area.areaDesc>*** by ***<info.senderName>***. Persons in this area are encouraged to ***<responseType>***, and ***<instruction>*** (if fields . This event is rated as ***<info.severity>***, and is ***<info.certainty>***. Responsive action should be taken ***<info.urgency>***. For more information about this event, visit ***<info.web>*** or call ***<info.contact>***."

6. Required Software Components

This section outlines the required software components to achieve the objectives of the alerting and notification logic discussed in section 3 and to provide the functionality for the subsystems discussed in section 2. The architecture of the software complies with the layered architecture.

The main software components comprise:

1. Message Editor
2. Delivery Configuration
3. Transport Gateway

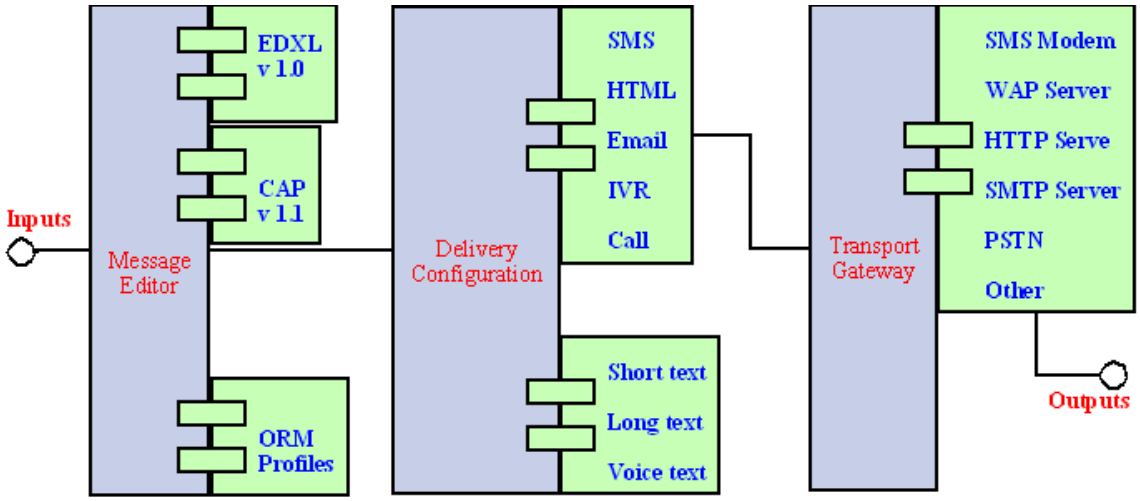

Figure 3 Software components of the EDXL/CAP multi-transport sub-module

6.1. Message Editor

The message editor is the main entry point for constructing message templates and issuing alerts. The underlying information data structure and logic for creating message templates and issuing messages are governed by standard CAP v1.1 and EDXL 1.0 XML schemas. The schema rules and control rules for each of the CAP elements are discussed in section 3. When a message or template is created they are stored as XML files.

The ORM profiles contain the set rules for controlling the various functional aspects of the message editor based on the rules and the privileges assigned to the authorized user. Section 2.5 describes the message administration subsystem for which the ORM profile component provides the functions.

The ORM further acts as the component that provides the controls for issuing either cascade or direct messages. Thus the ORM contains the records of the recipient lists. In the context of the software a cascade alert is simply the transmitting of a Long text

message to another system either via email or web services. Therefore, the mechanics is no different that a direct alert.

The users will be provided with a series of form controls to develop the templates and issue messages. The use cases for message template creation and issuing is explained in section 7.

6.2. Delivery Configuration

It is in the delivery configuration component that defines the delivery type and mapping to the various technologies as described in section 2.3 and section 10. The configuration is done at the system implementation stage before users can even start creating message templates. This requires developing an object relational mapping between the CAP elements and the three types of delivery mechanisms: long text, short text, and voice text. This object relational mapping is fixed for a particular deployment that is government by the CAP Profile implementers and domain experts. The users are not given the flexibility to alter these during run time. If the policies are to be changed then it must be changed at the system level.

First the system implementers and experts would define the object relational mapping between the CAP elements and the long text, short text, and voice text as defined in Table 3 in section 2.3. This component would be purely and XML code based and would not require an RDBMS.

Section 9 discusses available XML schemas for each of the transport and display technologies. The object relational mapping would take place between the CAP XML elements and the transport technology XML schemas. Hence, the transport technology XML schemas must be established standards. While all elements of the delivery types would map to the transport technology in most cases only selected elements of that mapped set would be displayed while other elements may be used for control logic.

6.3. Transport Gateway

The transport gateway plug-in is a software object that provides the necessary Application Programming Interface (API) functionality for different technologies to couple with the Messaging Module for carrying the alert messages over different technologies. For the purpose of the project we will use SMS, WAP, HTML, and Email as transports. Each of the technologies must identify whether they carry short text, long text, or voice messages. The transport gateway plug-in contains the underlying bottom layers of the network stack components to carry the messages through the individual technology networks. The outputs will be the short-text, long-text, and voice messages conforming to the message structure of the individual technologies. For example, an SMS

will be configured to carry a short text message; where the header of the SMS will contain the sender, recipient, and date information while the body of the SMS would contain the <*info.priority*>, <*info.headline*>, <*info.web*>, <*info.sent*>, and <*info.effective*> values of the CAP message. The predefined mapping instructions are provided in section 10.

7. Use cases of the Alert and Notification system

The required software components, discussed in section 6, are expected to provide functionality to address the necessary use cases. CAP/EDXL sub-module, which is part of the greater scheme of the SHN Messaging Module, is intended for communicating messages between the actors of the system and shall be used to communicate periodic national health status reports as well as instant alerts to communicate adverse health events.

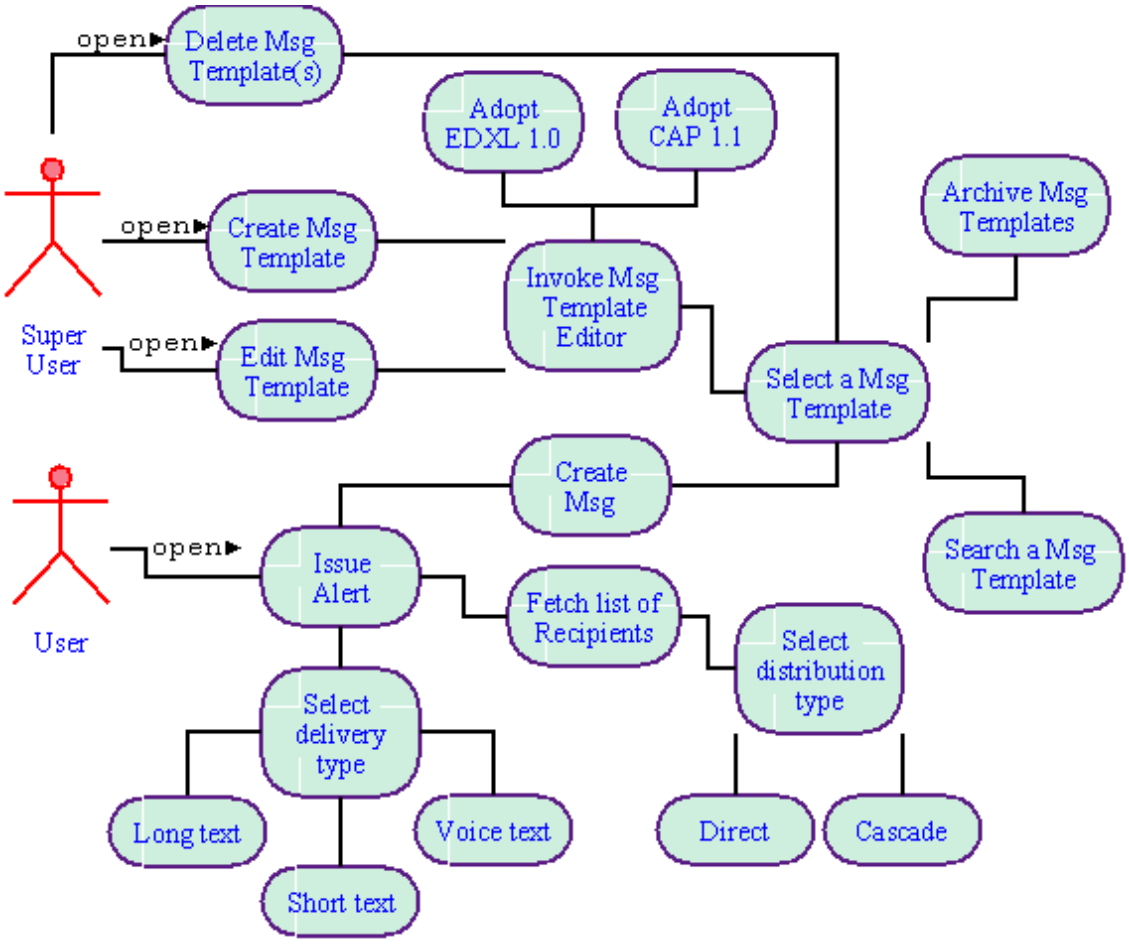

Figure 4 Use case diagram for EDXL/CAP enabled alert and notification

Table 6 Description of the individual elements of Figure 1

Search a template	A control for user to rapidly search a previously saved an

	EDXL/CAP template; another set of criteria will be defined within the health category to be able to retrieve the required template; user will select from presented drop down lists and enter partial information in the presented text controls to provide the filter criteria for the search. By default user is forced to search a template to avoid creating new messages and to ensure the user applies the appropriate language and structure that has been predetermined and not recreate ambiguous messages.
Archive template	The database of all super user created templates; most likely to be organized by the alert category; for the same of the RTBP these will be all labelled as <*info:category*> = "Health"
Select a message template	Based on the search criteria a set of results will be presented to the user to select one of the templates. If the required template is unavailable then the user may chose to create a new template from this point.
Invoke message template editor	Once a template is selected the EDXL/CAP message template editor is invoked for the super user to update the information. The same editor is used to create a new message template.
Adopt EDXL 1.0	If the super users decide to adopt EDXL then the system will provide the associated EDXL attributes and rules in generating the message templates. The template editor will use this knowledge.
Adopt CAP 1.1	If the super users decide to adopt CAP then the system will provide the associated CAP attributes and rules in generating the message templates. The template editor will use this knowledge.
Create a message template	Super user creating the template is forced to enter the mandatory elements and given the option to mark none mandatory elements as mandatory for the user creating the message to enter; the mandatory elements may be defined based on the country profile and implementers' policies.
Edit message template	The users are first forced to search for the desired template through the search a template use case. Then select the desired template for editing and saving upon completion.
Delete message template(s)	Only super users or (i.e. template creators) can remove templates. The super user must first search the desired template through the search template use case and then select one or more templates from the list and through the delete control remove them from the database.
Issue Alert	It is through the issue alert use case that the message senders access message templates, complete the message, select the delivery type (short text, long text, or voice), and distributions mode to issue the message.
Create Message	To create a message the user must first find a template through the search a template use case. Thereafter, complete the message by filling in the voids prior to issuing. The form control will apply the

	rules to validate the message completeness.
Fetch list of recipients	The recipients may be individuals or selected from a predefined group. The user will be displayed the list of recipients similar to an address book for user to select from. The selected recipients will be populated in the recipient's list control and be further separated by the delivery technology where email addresses are separated from phone numbers.
Select distribution type	The distribution type is either cascade or direct alerts as discussed in section 2.2. Through this use case the user will pick the distribution type. Based on this criteria the system will apply certain rules accordingly.
Direct	Direct alerting is when a message is sent from the system directly to the recipient, which is a person and the message is intended for the receiving person. The messaging protocol will contain the list of intended recipients. In the case of an SMS and email the recipient list would contain the mobile phone numbers and email addresses respectively. See section 2.3.
Cascade	Cascade alerting is when a message is sent to another system; i.e. a jurisdiction. The sending protocol would have a general address such as a phone number for SMS or email address for email of the receiving system. It is up to the receiving system to decide whether or not to propagate same of an edited version of the message to the distribution list associated with that system. E.g. the Irrigation department may issue an alert to warn the possibility of opening flood gates to release excess reservoir water to the
Select Publish mode(s)	The distribution types: short message, long message, and voice message will be presented to the user to select one or more of the choices.
Long Text	A full CAP message with all CAP elements in tact. Section 2.2 describes this concept.
Short Text	Parts of the CAP message that can fit in to a small display screen as well as constrained by the delivery capacity. The short message will be an incomplete message just to get the attention of the recipient with instruction or link to the full CAP message. Section 2.2 describes this concept.
Voice Text	Mainly the <*info.header*> and <*info.description*> elements of a CAP message intended for voicing out the alert via an IVR or simple phone call. This concept is discussed in section 2.2.

8. Entity Relationships

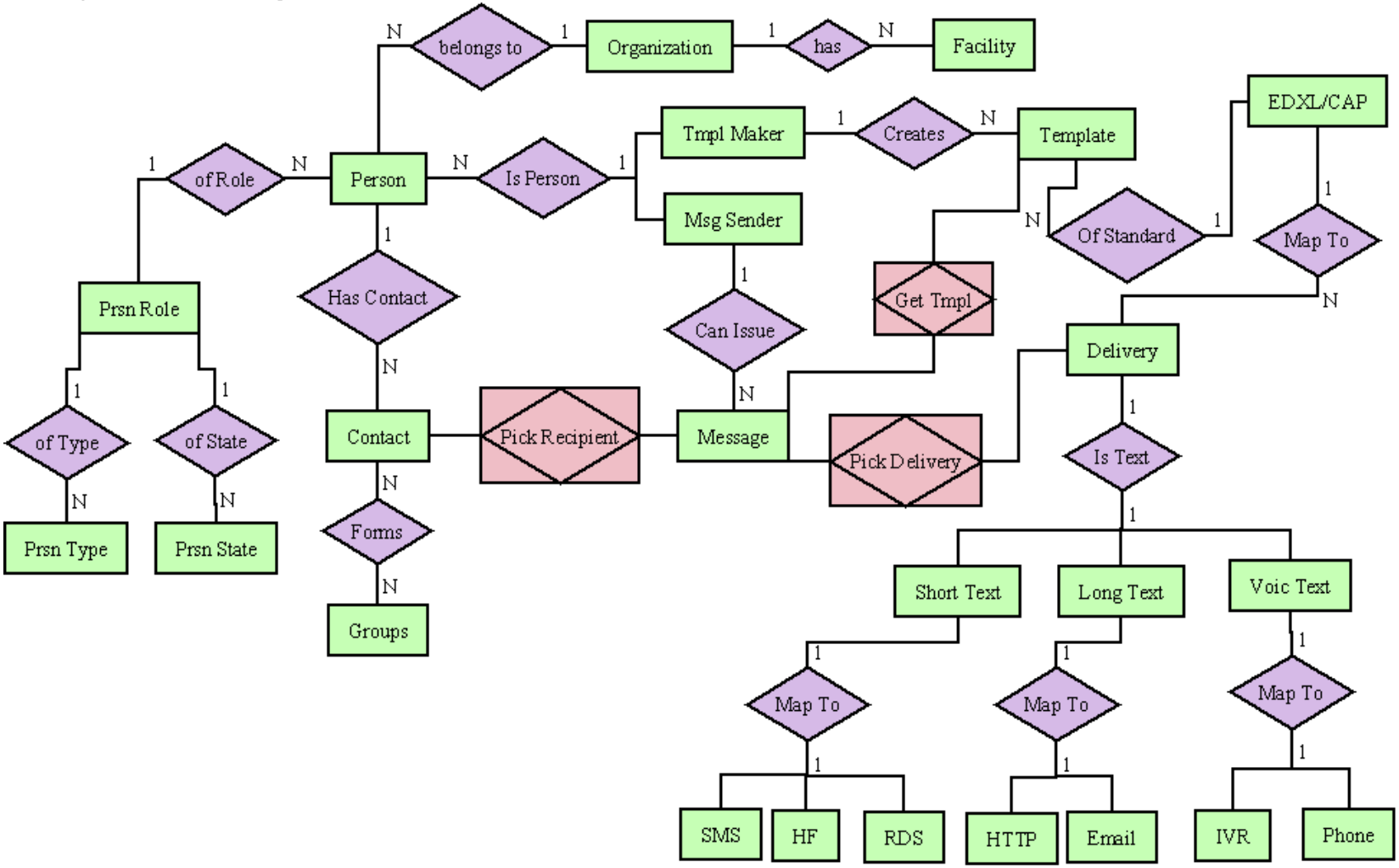

Figure 5 Entity relationship diagram for the CAP sub-module schema

This section describes the relationships between the entities that makeup the data structure for the alert and notification system. While some parts of this relational data structure is made of RDBMS tables the others are pure XML files.

To begin with, the *person* entity holds the user information of the person's name and other information that are made available by the general person table of the SHN_BSM module. These persons are authorized with permissions to create templates and issue messages. The authorized persons are labelled as a "user" in the *person roles* table. Some users would be granted permissions to create templates while others will be granted permission to simply issue alerts. The restrictions can be further partitioned categorically such that a particular user may be given permission to issue alerts with respect to a subset of the hazard categories defined by the predefined values of the <*info.category*> CAP element.

The message template creators, message senders, and message recipients are further distinguished by *person types*. The *person* state denotes the active stature of the person; where the state can take on several transitional states; where a person may go from inactive, to probations, to active.

Through the *person status* the users can be identified as to whether they are active or inactive in the system. Although Figure 5 shows two entities: Tmpl Maker (Template Maker) and Msg Sender (Message Sender) they don't necessarily have to be implemented as two different tables but can be implemented as two separate person roles. These two roles are shown in the diagram for illustration purposes only.

Each person's (or users) *contact* information is store in the system. The contact information can be a telephone number, email address, website, or blog that is used to populate some of the CAP elements such as the <*alert.recipients*> list and <*alert.sender*> details. Contact *groups* (i.e. recipient lists) can be created in advance to minimize the time on retrieving sets of recipients.

Persons are also associated with an *organization*. This fulfils the ORM requirement when issuing direct and cascade alerts. Moreover, the <*alert.identifier*> would use the information on the organization registry to auto create the identifier based on a predefined formula. The <*alert.senderName*> would be assigned the name of the organization through the alert was issued; where the relationship is established through the person entity with role = user.

Message templates developed for future use are stored as XML files in the system. The template saved names along with information such as the <*info.category*> and the XML file name are stored in a table for template creators and alert issuing users to search when they want to retrieve a template to modify or remove it and message senders to search for the right template to generate the message to issue the alert.

While templates can be of various styles, the *CAP* templates use the predefined CAP elements and the respective knowledge. In order to accommodate three different delivery types: short text,

long text, and voice the implementers must decide the set of elements that must go in to each delivery type. For example, a short message sent over SMS does not have the space to carry a <*info.description*> CAP element value but will be assigned the <*info.headline*> CAP element value. The *delivery medium* and the CAP entity will provide the knowledge of the mapping on the set of elements that should be used in each of the delivery types. Each of the predefined delivery types will be stored as XML files on the server.

It is assumed that each of the technologies SMS, Email, HTTP, etc can be defined through a structured XML file. The object relational mapping between the delivery types and the various technologies will be stored in a file that can be retrieved to assign the CAP values at the time of issuing the message.

9. Recommendations for Transport Technology Data Structure

9.1. Short Message Service

A typical SMS message submit XML DTD which can be used as the data structure to map the CAP elements to the SMS message elements. It is possible to adopt a SMAP or MMAP message structures to define the SMS data structure.[6]

```
<?xml version="1.0" encoding="ISO-8859-1"?>
   <!DOCTYPE message SYSTEM "SMSmessage.dtd">
   <message>
      <submit>
         <da><anumber>+358991234567</anumber></da>
         <da><anumber>+358997654321</anumber></da>
         <ud> Merry Christmas !</ud>
         <timing>
            <time>
                <year> 2000 </year>
                <month> 12 </month>
                <day> 24 </day>
                <hour> 18 </hour>
                <minute> 0 </minute>
                <second> 0 </second>
                <timezone> -2 </timezone>
            </time>
         </timing>
         <messageID> 1 </messageID>
      </submit>
   </message>
```

9.2. Hyper Text Mark-up Language

XSLT and CSS are the more popular styling and mapping methods that can be used to map the XML tags in the CAP message to display in a web page; where the CAP message tags and values

[6] Use XML to Send SMS - http://www.ibm.com/developerworks/xml/library/x-tipsms1.html

can be displayed in a table with two columns. Also adding in some logic to eliminate CAP tags that are empty and converting the <*info.web*> values to hypertext links and values related to the <*resource*> tags also use appropriate logic to enhance the functionality.

9.3. Email

Email Templates includes support for XML. XML Documents can be searched for elements and attributes that can be inserted into your message. A simple example –

```
<email>
    <from>rafe@rafe.us</from>
    <to>someone@example.com</to>
    <cc>someoneelse@example.com</cc>
    <bcc>rafe@rafe.us</bcc>
    <subject>This is the subject</subject>
    <body>This is the body of an email message.</body>
    <attach>c:/doc/alert.xml</attach>
</email>
```

<from> element would be equivalent to the <*alert.sender*> and <subject> would be equivalent to the <*info.event*> or <*infor.headline*> elements and can be implemented either way. Given that <*info.event*> is a mandatory element it can be mapped to the email <subject>.

9.4. Interactive Voice Response

VoiceXML (VXML) is the W3C's standard XML format for specifying interactive voice dialogues between a human and a computer. Simple example –

Main.vml

```
<?xml version="1.0"?>
<vxml version="1.0" application="app-root.vxml">
 <form id="say_goodbye">
  <field name="answer" type="boolean">
    <prompt>Shall we say
      <value expr="application.bye"/>?
    </prompt>
    <filled>
      <if cond="answer">
      <exit/>
    </if>
    <clear namelist="answer"/>
    </filled>
  </field>
 </form>
```

</vxml>

app-root.vml

```
<?xml version="1.0"?>
<vxml version="1.0">
 <var name="bye" expr="'Ciao'"/>
 <link next="operator_xfer.vxml"> <grammar>
  operator </grammar> </link>
</vxml>
```

10. Mapping of Delivery to Technologies

The delivery types have been established as long text, short text and voice text.

Table 7 Mapping delivery types to technologies displays

Technology	***Delivery Type***	***CAP Attributes with Message format***
SMS	Short text	<*info.headline*>+" for "+ <*area.areaDesc*> +"…"+ <*info.valueName*>+" "+ <*info.value*>+" "+<*info.event*> +" issued by "+<*info.senderName*> +". Msg:"+<*alert.identifier*>+" sent "+<*alert.sent*>+" is a "+<*info.status*>+" "+<*info.type*>+". More info "+ <*info.web*>+" "+<*info.contact*>
HTTP/WAP	Long text	<*alert*> ….</*alert*> (full cap message with CAP tags in column 1 without the "<>" symbols and values in column 2.
Email	Short text	[subject] <*info.headline*>+" for "+ <*area.areaDesc*> [body]<*info.headline*>+" for "+ <*area.areaDesc*> +"…"+ <*info.valueName*>+" "+ <*info.value*>+" "+<*info.event*> +" issued by "+<*info.senderName*> +". Msg:"+<*alert.identifier*>+" sent "+<*alert.sent*>+" is a "+<*info.status*>+" "+<*info.type*>+". More info "+ <*info.web*>+" "+<*info.contact*>
	Long text (XML file attachment)	[Subject] <*info.headline*> [Attachment] alertmsg.xml [body]<*info.headline*>+" for "+ <*area.areaDesc*> +"…"+ <*info.valueName*>+" "+ <*info.value*>+" "+<*info.event*> +" issued by "+<*info.senderName*> +". Message ID:"+ <*alert.identifier*>+" sent "+<*alert.sent*>+" is a "+<*info.status*> +" "+<*info.type*> +". More info "+<*info.web*>+" "+ <*info.contact*>
IVR	Voice text	<*info.headline*> …<<pause>> … <*info.description*>