



Jul 1st, 12:00 AM

Sharing emergency information between Emergency Control Centres: the Project REACT

U. Delprato

M. Cristaldi

A. Gambardella

Follow this and additional works at: <https://scholarsarchive.byu.edu/iemssconference>

Delprato, U.; Cristaldi, M.; and Gambardella, A., "Sharing emergency information between Emergency Control Centres: the Project REACT" (2008). *International Congress on Environmental Modelling and Software*. 180.

<https://scholarsarchive.byu.edu/iemssconference/2008/all/180>

This Event is brought to you for free and open access by the Civil and Environmental Engineering at BYU ScholarsArchive. It has been accepted for inclusion in International Congress on Environmental Modelling and Software by an authorized administrator of BYU ScholarsArchive. For more information, please contact scholarsarchive@byu.edu, ellen_amatangelo@byu.edu.

Sharing emergency information between Emergency Control Centres: the Project REACT

U. Delprato^a, M. Cristaldi^a, A. Gambardella^b

^a*IES Solutions Srl, Italy, www.i4es.it, www.react-ist.net, info@i4es.it*

^b*National Corp of Firefighters, Italy, antonio.gambardella@vigilfuoco.it*

Abstract: Every year, thousands of emergency call-centres in the EU receive some 200 million calls from citizens in distress. In response to them, Public Safety Answering Points (PSAPs) dispatch ambulances, fire-fighter teams or police squads to help the callers. In Europe, response to the calls and interaction between different agencies are often uncoordinated. This holds true also between different departments of the same: information sharing between them is still typically based on faxes and phone calls. The European Project REACT (Reaction to Emergency Alerts using voice and Clustering Technologies) aims at creating a seamless way to allow Command and Control center of different agencies (or of the same agency in different locations) to share data in electronic format. Key technical drivers of the project are the CAP protocol, the TSO data dictionary, a distributed web service based architecture and the AtomPub protocol. REACT started circa one year ago and has currently completed the design phase.

Keywords: Common alerting protocol (CAP); TSO; PSAP; E112; semantic clustering; GIS; voice recognition; Emergency management.

1. INTRODUCTION

Fire brigades, emergency medical services and police are confronted with an increasing number of natural disasters and incidents on the one hand, and the trend towards bigger Public safety answering points (PSAPs) serving larger areas or integrating different emergency operators (typically Fire fighters and ambulances) on the other hand. An increasing importance is given to systems that assist call takers and dispatchers in getting a fast overview on incidents, either answered locally or coming from a neighbour agency. The overall situation in Europe shows a large number of emergency services using different command and control systems and not sharing information in electronic format, yet. Figure 1 synthesizes the current situation, where different PSAP or Emergency Control Centers (ECC) get different “inputs” from citizens and use phone calls of faxes as interoperability means.

This produces negative effects both in the response time and in the possibility that the right recipient of the piece of information is actually addressed and informed about an event. As an example, during the murderous Sarno (Italy) mudflow, several notification calls of minor precursory flows were addressed from on-spot citizens to police stations or municipalities: however, such precious and precise information was not addressed to the right decision maker. Despite the inherent local geological conditions and extreme rainfall, there is evidence that the high number of casualties was partly due to a lack of a unified information repository, available to emergency crews as well as emergency “intelligence” managers.

1.1 The European project REACT

The project REACT (Reaction to Emergency Alerts using voice and Clustering Technologies), funded by the European Commission under the Sixth Framework

Programme, addresses these problems by designing and implementing a working prototype that would allow:

- A seamless share of information between different agencies in electronic format
- A reliable voice (and language) recognition for capturing more information from caller/call takers conversations
- An intelligent “clustering” of apparently not related incidents into a dynamic scenario, being, this way, a decision support tool based on a large(r) group of calls.

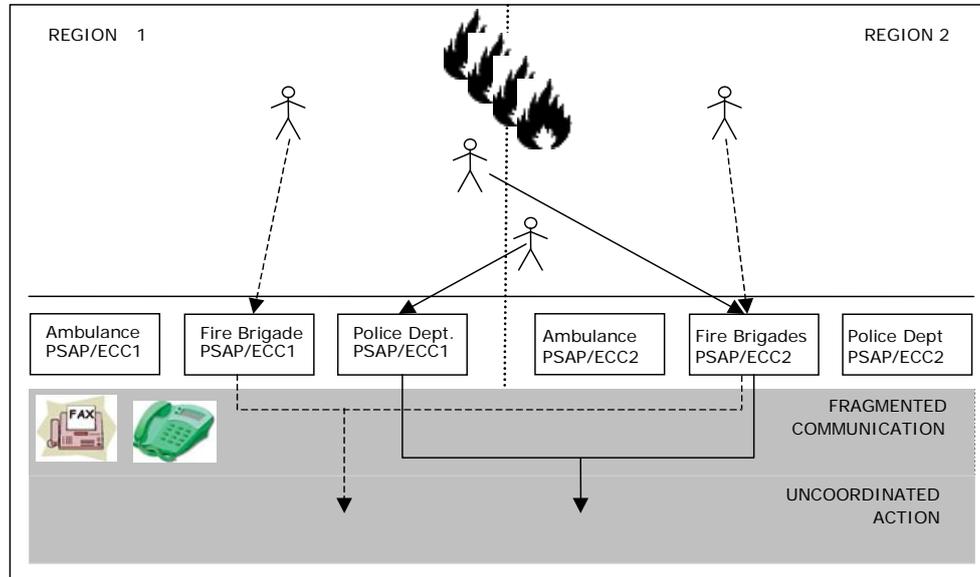


Figure 1. Communication between ECCs: today

The emergency services participating in the project have set the target for a system that will increase both efficiency and efficacy of work if, particularly when a lot of calls are received for the same event (potentially using different emergency numbers) and small incidents may go underestimated and escalate in scale. Figure 2 shows how the response to an event may be improved once REACT will be implemented and available.

The project partners are developing semantic technologies for call clustering and prioritisation, based on sensors (weather, air pollution, etc.), time and location of emergency calls, as well as keywords identified during the emergency call taking. Interoperability is implemented by using secure XML GIS-based data exchange between PSAPs. Beside the active participation of emergency services and experiences of other partners in this field, an active interaction with further emergency services all over Europe is actively sought to develop a system that would accommodate user needs, current processes, and infrastructures used for handling emergency calls at European level. It is worth to emphasize that REACT is designed as a system layer additional to existing computer aided dispatch and call taking systems, and it is not intending to replace them.

Being a shared effort of 11 partners spanning over two and half years, REACT is quite ambitious. This paper is focussed on the information sharing aspects, with the aim of describing how an event received and first managed by PSAP can be shared and displayed on a different PSAP system.

2. REACT ARCHITECTURE

The core architecture for REACT is based on a distributed model that supports efficient and performance optimised data exchange, coupled with platform independence. This type of architecture was chosen to satisfy requirements for deployment and assist integration with components from many different partners.

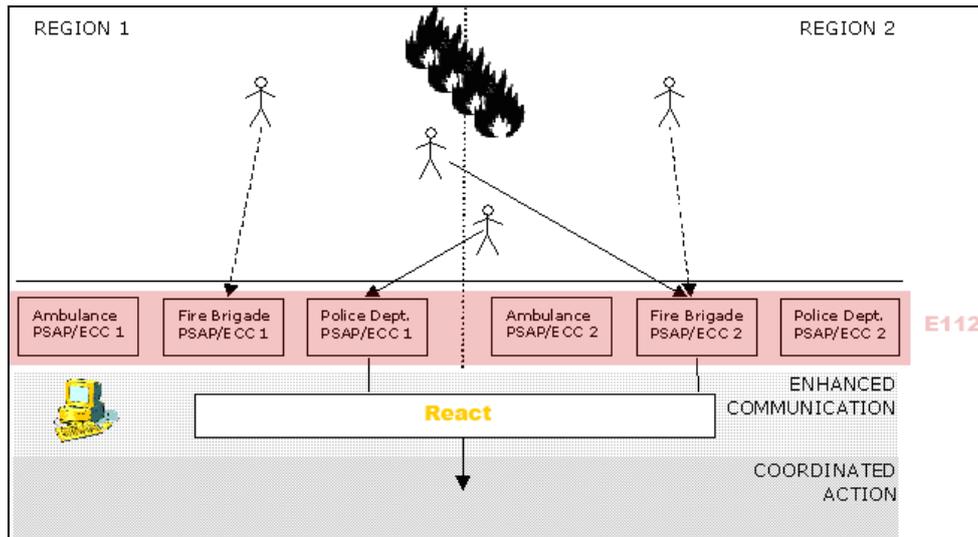


Figure 2. Communication between ECCs: with REACT

The communications backbone is provided by XML-based Web Services, to facilitate the interconnection of components using clearly defined APIs. Web Services were chosen because they are an industry standard way for disparate components to interoperate, independently from the used operating system or programming language.

Security is of paramount importance to protect the information transmitted between components or between REACT systems. Encryption is applied to the messages passed through Web Services for integrity and protection.

Because of the existing limitations in the modifications that can be brought to existing legacy infrastructure at each user site, a customised interface will be employed to plug their current Command and Control software into the REACT system.

Key elements of the REACT architecture are:

- Loose coupling between components
- Uses standard TCP/IP based protocols and links
 - XML
 - Web Services
 - Atom feeds
- Common Alerting Protocol as Incident Message format
- TSO as Incident Dictionary

Figure 3 shows the complete REACT architecture, with basic roles and responsibility of each components clearly depicted. The next sections of the paper will be focused on the Routing component.

2.1 Main Interfaces to REACT

Command and Control Legacy Interfaces (x3)

They represent the REACT interfaces to the Legacy Command and Control systems (i.e. the software currently used by REACT end users). The REACT consortium includes three end users, representing three typical examples of European Emergency Agencies:

- (i) Italian National Corp of Firefighters (a single organization covering the entire Italian country)
- (ii) Fire Brigades of Aachen, Germany (a brigade serving the municipality of Aachen in Germany, located close to the boundary with The Netherlands and Belgium)
- (iii) Sussex Police Authority, United Kingdom (serving an English County)

The Interfaces are built on users specific needs and the REACT users will continue using their current existing Command and Control systems. To minimize the changes to existing working procedures, specific interfaces will be created, where the core functionality will be to create, update and delete events.

Routing Component

Receives requests from local or remote REACT components. Filters outgoing messages based on recipient, provides message logging, and dispatches messages to defined recipients.

UDDI Web Service

Web Service that stores and provides information about the location of other REACT components.

Local / Remote Data Stores

Stores incident data and other data received from the Routing Component, and makes the data available to other modules. Notifies interested modules when the data changes.

Semantic Clustering Subsystem (Final product)

Provides all the REACT searching and clustering functionality, notifying PSAPs of potential clusters and providing search results when requested to do so.

Auditing

Audits all the messages that pass through REACT, and all configuration changes.

Monitoring

Monitors the status of REACT components.

Management / User Security

Manages the REACT system. Applies filters, priorities and security settings.

3. REACT ARCHITECTURE COMPONENTS ALLOWING DATA SHARING

Figure 4 shows the flow of information in REACT for when a new incident is created. The diagram shows the flow of information both during the emergency call and after the call has finished.

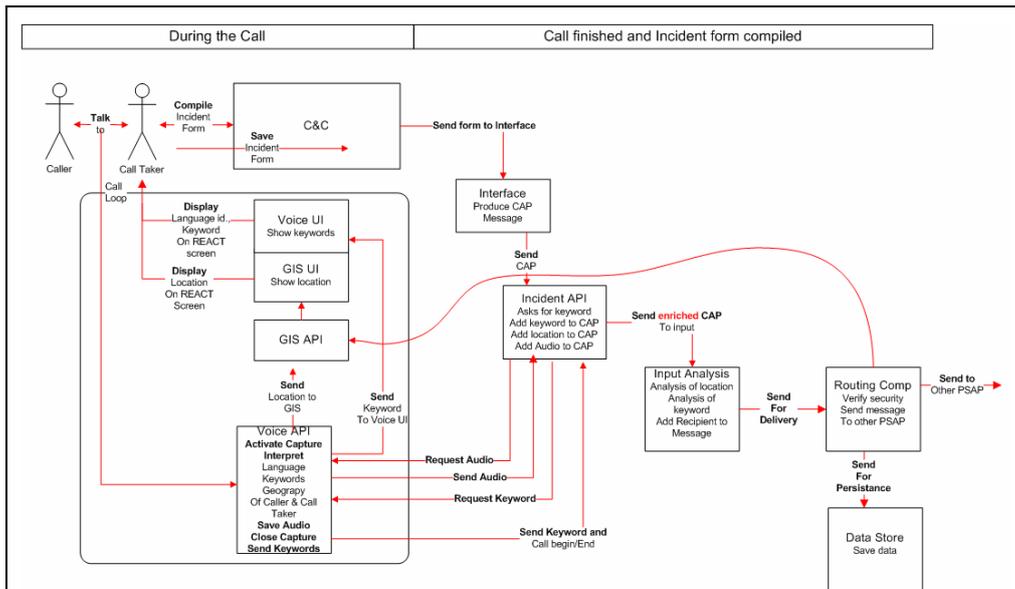


Figure 4. REACT Conceptual Sequence Diagram

During the Call:

- The emergency phone at the PSAP rings. The Call Taker and Caller talk to each other about the incident.
- The Call Taker enters data into an Incident form.
- The Voice API activates and captures language, keywords and geographical location of the caller and Call Taker.
- The Voice API sends keywords to the Voice UI, which displays LanguageId, keywords to the Call Taker on the REACT screen.
- The Voice API sends the location to the GIS API, which sends details to the GIS Client to display the location on the REACT screen

Call finished and Incident Form completed:

- A completed Incident form is sent to the C&C Legacy Interface, which converts it to CAP message format and sends it to the Incident API.
- The Incident API receives audio from the Voice API after requesting it.
- The Incident API receives keywords from the Voice API after requesting them.
- The Incident API sends an enriched CAP message to Input Analysis.
- Input Analysis analyses the location and keywords. It determines potentially interested recipients and forwards the analysed data to the Routing component.
- The Routing component verifies security, and sends the incident details to other PSAPs and the Data Store for persistence.
- The Data Store saves the received incident data.

Figure 5 shows the flow of information for incoming incident details from an external REACT system.

- The Routing component receives incident details from an external REACT system.
- The Routing component verifies security. It sends it to other PSAPs and the Data Store for persistence.
- The GIS API sends incident details to the GIS Client User Interface to be displayed on the Call Taker's and Dispatcher's REACT screen. The GIS API will be able to dispatch information to two different GIS User Interfaces: a thick client, provided by one of the Consortium partners and a thin client (a web browser based on Google Maps that will exploit KML and, potentially, GEORss data formats).
- The Data Store saves the received incident data.
- The Data Store sends the incident details to the Semantic Clustering component to update the cache. The Semantic Clustering component runs Geo and Semantic algorithms.

3.1 Focus on the Routing Component

From the REACT perspective, every system that is able to produce messages in CAP format can make use of REACT services. The real entry point for REACT is the IncidentAPI component: ECC legacy interfaces are offered to the final users as a set of API that can be used to convert legacy data into CAP format. This will allow a simple exploitation of REACT and will make the whole system attractive for other potential users. The sequence diagrams above are reported to focus on the component that allow interoperability between two different REACT system: the Routing Component. It has been conceived as an Atom feed built upon Apache Abdera. The Atom Publishing Protocol, known also as the AtomPub protocol, is a way to publish and manage collections of resources using the basic HTTP GET, POST, PUT, and DELETE operations. While originally designed as a way to post new entries to weblog software, the Atompub protocol is well suited as a way to manage nearly any kind of Web-based content.

Web feeds allow software programs to check for updates published on a web site. To provide a web feed, a site owner may use specialized software (such as a content management system) that publishes a list (or "feed") of recent articles or content in a standardized, machine-readable format. The feed can then be downloaded by web sites that syndicate content from the feed, or by feed reader programs that allow Internet users to subscribe to feeds and view their content. A feed contains entries, which may be headlines, full-text articles, excerpts, summaries, and/or links to content on a web site, along with various metadata.

The development of Atom was motivated by the existence of many incompatible versions of the RSS syndication format, all of which had shortcomings, and the poor interoperability of XML-RPC-based publishing protocols. The Atom syndication format was published as an IETF "proposed standard" in RFC 4287, and the Atom Publishing Protocol was published as RFC 5023.

In REACT, the AtomPUB protocol is used within the Routing component to create and diffuse CAP messages. Other components create CAP messages into the Routing component and any other subscribed components will be acknowledged of the presence of a new CAP message. The routing component offers also security and message encryption.

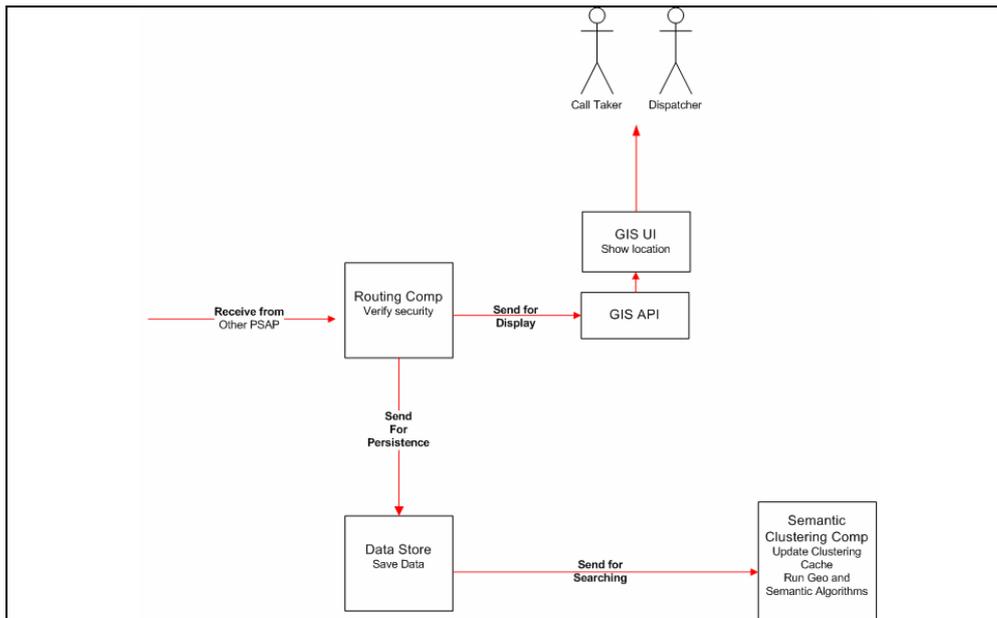


Figure 5. Flow diagram – Incoming incident

3.2 CAP protocol in REACT

The Common Alerting Protocol (CAP) is an XML data format used by many different organization in the to exchange information about a broad range of warnings and incidents. The CAP standard defines a type of document called an alert, which is used to exchange information about geological, meteorological, public health and safety, rescue, law-enforcement, environmental, transportation, infrastructure, and terrorist warnings and events. Such alerts can be generated either manually by incident responders or automatically by monitoring and sensing equipment, and they can be distributed using a variety of means.

In our case incident data are extracted from Command and control legacy system and converted into CAP format by the Command and Control legacy interfaces. They give the possibility to map each field in C&C system into a CAP field.

3.3 TSO in REACT

The Tactical Situation Object (TSO) has been adopted by the EU-OASIS project to increase the interoperability level between several agencies that are working jointly.

This object is one of the key means to reach a minimum level of interoperability between agencies during the disaster and emergency operations. This minimum level is defined for the purpose of the 1st version of the TSO, but in the future, the TSO could be extended progressively, allowing agencies to collaborate more efficiently during operations by sharing a timely and comprehensive common operating picture.

The TSO provides the capability to exchange pieces of information but it is not intended to provide all detailed information. There is a full flexibility in defining the mapping how each field in the ECC systems maps against the TSO data dictionary .

TSO in REACT has been selected ad a data dictionary to allow interoperability between agency. CAP will be used as the data format that will “carry” also TSO messages.

4. CONCLUSIONS

A key aspect of Emergency management is the communications between emergency services. Almost every emergency arising from everyday accidents requires the intervention of two or even three emergency services. Ambulances must help victims and police has to regulate traffic around the scene of the accident or must start investigations in

case of criminal acts. Fire-fighters are often required to liberate victims from wrecked cars or from debris. Of course they are the real protagonists in the case of fires – where the other emergency services are also required. All these interventions require communications between the emergency services involved. This becomes imperative in case of major incidents and disasters covering wide areas and necessitating the intervention of emergency services from different local or regional authorities or even (in the case of disasters with a European or international dimension) of multinational, multidiscipline teams.

Improvements are clearly needed for ensuring that the call to the 112 is “appropriately answered and handled”. This involves many issues on the front-end of the PSAP (particularly multilingual support, caller location and verbalisation of the verbal communication), but also interoperability and storing/retrieval of information is a key point REACT aims at creating a favourable environment for an efficient management of local emergencies, where still a timely and focused intervention saves lives, but the communication and command chain is shorter and more direct. This requires anyway the setting up of an interoperable system, allowing an efficient exchange of data and a common representation of the needs of the person in distress.

With REACT in place PSAPs will be able to share information in a seamless way, only provided that their current Command and Control softwares allow an “incident-per-incident” extraction of incident data in CAP format. The project will hence demonstrate that an integration between emergency actors is possible and, will pave the way for triggering discussions at coordination, management and political levels.

In this article, the REACT architecture has been described, along with the motivation brought in by the final users in terms of usability, interoperability and functionalities needed for a successful future integration in their current legacy systems.

The key contributions of REACT to the current emergency communication space are:

- Integrated use of both CAP and TSO as enhanced protocol for representing and sharing information about events
- Integration of the emergency caller location featured by the E112 and eCall – reducing response times.
- Call clustering and prioritisation through semantic, time, location analysis allowing emergency services to identify and consolidate incoming data – leading to clearer decision making and support during cross-border incidents (4D GIS).
- Automatic identification of keywords from the PSAP operator speaking supporting the filling of a job description forms (including multilingualism)
- The adoption of the Atom Publishing Protocol for ensuring interoperability across call centers and emergency services.

ACKNOWLEDGMENTS

REACT is co-funded by the European Commission, Framework Program 6, IST priority, under the contract IST-2005-033607 “REACT - Reaction to Emergency Alerts using voice and Clustering Technologies”. The project has started in September 2006 and will run for 30 months.

REFERENCES

- Snell, J., An overview of the Atom 1.0 Syndication , *developerWorks*, 2005
- Snell, J., The Getting to know the Atom Publishing Protocol series, *developerWorks*, 2006
- The CAP Cookbook:
http://www.incident.com/cookbook/index.php/Welcome_to_the_CAP_Cookbook
- The CAP V1.1 specification:
http://www.oasis-open.org/committees/download.php/15135/emergency-CAPv1.1-Corrected_DOM.pdf