Community Driven Data Integration for Emergency Response

Naveen Ashish
Calit2, University of California-Irvine
ashish@ics.uci.edu

Sharad Mehrotra
ICS, University of California-Irvine
sharad@ics.uci.edu

ABSTRACT

This paper describes our work in progress on an approach and technology for providing integrated data access in situational awareness applications – particularly for disaster and emergency response. The key new aspect of our work is an approach where information aggregation, processing, and integration capabilities are offered as a *service* to any new application builder. Further, we provide a framework for possibly reusing prior information integration knowledge, the development of which demands the major fraction of time and complexity in a new application, in a *customized* fashion for new application. Our overall goal is to provide a framework where integrated access to critical data in an emergency response situation can be enabled very rapidly and by personnel with basic IT and data handling expertise. Our approach, while general purpose, is currently motivated by and grounded in the context of situational awareness systems for incident commander decision support in the fire response domain.

Keywords

Information integration, situational-awareness, software-as-service, community driven approach.

INTRODUCTION

We are building an information integration system called the "Software EBox" (Ashish et al., 2009) for providing integrated access to multiple different sources of information in the context of situational awareness, specifically for disaster and emergency response. The EBox is a general purpose data integration engine that drives applications such as situational awareness systems that first responders and other emergency decision makers can use for decision support. Consider the following scenario and sequence of events to illustrate the role and value of the EBox, taking a context of fire response. Several calls are received at the 9-1-1 call center reporting a fire in a chemistry lab on the university campus. While fire and hazmat resources are enroute to the scene, the dispatchers check for and activate EBox resources pertaining to the particular location and type of incident (Fig 1). The unlocked EBox resources are available to incident commander and others via networked terminals installed in the fire apparatus and command vehicles (Fig 1). In this case, the resources available via the EBox could include a) detailed floor plans of the relevant building, b) an up-to-date inventory of hazardous materials obtained live via campus chemical inventory database and cross referenced to the floor plan, c) up-to-date contact phone numbers for the lab managers for the building so that the fire department can reach someone for confirmation of information, d) connection to the building surveillance cameras allowing video feeds to be observed from inside the building, and e) connections into the building alarm system. Even as the fire-fighters are arriving on the scene and within the first few minutes, the incident commander may already have been able to make several important determinations. The alarm panel indicates the particular lab where the fire is located, and using the phone contact information the incident commander is able to speak with the lab manager. Without this contact information it might have taken much longer to track down the responsible party. The incident commander is able to confirm the presence of a water reactive chemical stored in the lab which is indicated via the EBox hazmat inventory information. This precludes the use of water to extinguish the fire.

Fire-fighters and other emergency response personnel in general today lack such general purpose tools and capabilities that can provide them with vital information at the right time to help in critical decisions.

Reviewing Statement: This paper represents work in progress, an issue for discussion, a case study, best practice or other matters of interest and has been reviewed for clarity, relevance and significance.

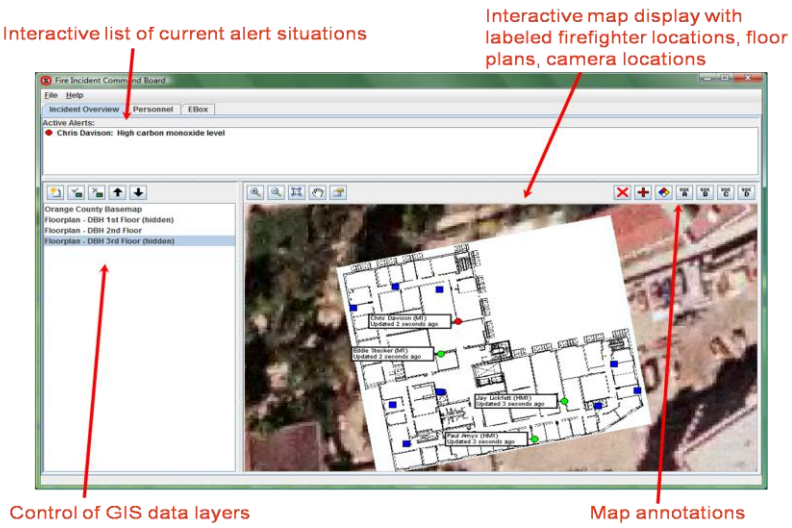

Fig 1. Ebox Interface

There are companies however (Tactical Survey Group, 2009) that provide highly customized integrated situational awareness information packages created for specific locations and organizations and for use by emergency response personnel such as fire fighters, police, etc. Information provided by such packages includes key emergency information such as names and contact information of key personnel, locations of exits and entrances, presence of hazardous materials and chemicals, maps and plans of the location and/or buildings, etc. Such customized aggregated information solutions, while of course very useful in emergency situations, require a significant amount of time and effort in assembling each new application. Consequently the costs are also quite high, typically in the > $1M range per new installation. Besides, commercial locations and facilities today are often instrumented with sensors of different kinds, such as surveillance cameras, people counters or detectors, etc., which can and must be exploited for situational awareness.

The EBox technology is aimed at addressing the limitations of the current available options in terms of functionality, as well as the high cost and effort per application. The EBox is built using a Software-as-Service (SaaS) architecture where organizations can pre-load data in advance to a central EBox server. Clients can then access integrated data from the EBox server at situation time, including real-time sources as well if available and necessary. Our introductory paper on the EBox (Ashish et al., 2009) provides a high level overview of the architecture, implementation and preliminary evaluation of such a system. We have also highlighted new research challenges posed by the EBox, such as being able to integrate information from real-time streaming data sources such as cameras, challenges in geospatially data alignment and integration from different sources, and finally being able to develop new applications rapidly and without much user expertise. This paper focuses on the last of the above mentioned issues as we see that the time, effort, and expertise required to facilitate such capabilities with the current state of the art is prohibitive to their realization. In the next section we briefly review the applicability of current data integration technology to this task followed by a summary of our approach and work in progress.

DATA PROCESSING AND INTEGRATION CAPABILITIES

For assembling a new instance of an EBox application (for an organization or location) we need to be able to (i) *Process* some of this information to make it amenable to sharing and integration. For instance we may need to extract particular pieces of information from data originally in text (such as in "MSDS" i.e., Materials Safety Data sheets providing Hazmat information) or extract information from maps or floor plans. (ii) *Integrate* such data so that it can be accessed seamlessly by clients in the future. Many tools are available today for these tasks. For instance we have software tools for automated information from text (Ashish and Mehrotra, 2009), data extraction from maps (Knoblock, Shahabi, and Chen, 2009) etc. We also have general purpose software for data integration from multiple kinds of data sources (Halevy et al., 2005). This includes information "mediators", commercial data integration software from the "EII" (Enterprise Information Integration) industry, as well as some open-source data integration engines (XAware, 2010) that have appeared recently. Typically most such data integration systems are based on a mediator architecture (Wiederhold, 1992) where the mediator is a software component that is aware of the information in all the various different sources and plays the role of a broker between a user and the various sources, abstracting the user from the fact that information is coming from various different sources. This overall architecture is largely common to various data integration systems,

albeit they may differ in particular aspects such as the choice of data modeling representation, query language over the data etc. From the EBox perspective what is important is what is entailed in developing a *new* application with such technologies. Regardless of the flavor of the particular data integration system being used, the following key tasks are required in assembling any new application: (i) **Data modeling**: We need to have a representation of the information in each of the individual information sources as well a "global" integrated view of the information as a whole. This is done with the defining of formal data models. (ii) Defining **links**: Related information across various sources needs to be syntactically, schematically, and semantically linked so as to provide an integrated view of the information. This is done by defining schema correspondences, and explicit links between source and global models. (iii) **Wrapper** and **adapter** development: These are software components that provide translation between the mediator query language and the query language native to each information source.

It is these very tasks that are time consuming in any new application and typically require months of effort from data integration consultants with a sufficiently high level of expertise. The data integration community has developed solutions to at least partially alleviating some of these tasks – examples include tools for semi-automated wrapper generation, automated schema matching tools, Web "mash-up" technologies (Yahoo Pipes, 2009), etc., all of which are aimed at making developing new information integration applications easier (Halevy et al., 2005, Saha, Stanoi and Clarkson 2010). However developing new data integration applications even with general purpose data integration software and associated tools such as the above remains a task that is complex, time consuming, and requires specialized expertise.

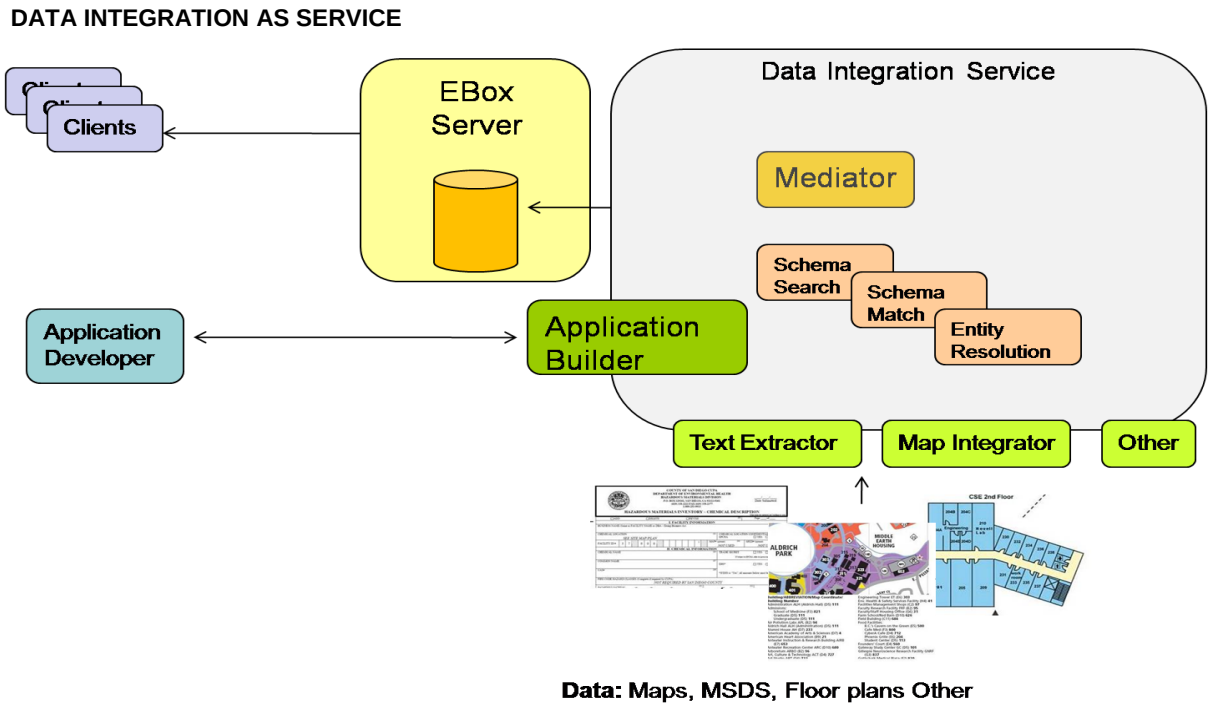

Fig 2. DVD Interface

Our approach is to offer data processing and integration capabilities as a *service* as illustrated in Fig 2. New application developers then do not have to deal with the complexity and overhead of installing and managing tools for these tasks themselves, rather they can use the existing tools for data synthesis and data integration offered as a service. While the idea of offering software capabilities as a service is per se not new, there is a key aspect about the EBox kind of data integration applications which make it a special class of applications more attractive for such a framework. Every instantiation of the EBox at any location or organization is *similar* across instances i.e., each instantiation is about integrating useful local information relevant to emergency response (at that facility or location). Moreover the kinds of data sources the information is gathered from are similar as well, for instance collections of Hazmat reports are likely similar across different universities (certainly an important subclass of organizations that will benefit from the EBox). This implies that (i) Instantiations of data processing tools used in previous applications are likely useful for new applications as well, for instance a text extractor instantiated for extracting data from MSDS sheets is likely applicable to the MSDS sheets in a new application. (ii) Considering data integration, any prior integration knowledge such as domain or data models, semantic

linkages, and even components such as wrappers for an application, developed by the community in previous application instances can be reused in a new application. This can significantly reduce the time and complexity in assembling a new application as the application developer does not have to assemble this integration knowledge from scratch. For such an approach to be successful however just offering previous integration knowledge for (re)use only "as-is" is too rigid. For instance a new application might find *most* of an existing domain model useful for his particular application, but ideally like a domain model that is better customized to his needs.

What would be useful thus is a capability to *modify* existing integration knowledge so that it can be reused but better adapted and customized for a new application. A new application developer can start with an existing domain model for the application, existing models of particular information sources, and existing semantic links between the domain and source models. She can then modify any of these artifacts as required – as an example an existing model for a collection of MSDS sheets may be as a relation with two fields (MSDS Sheet, Catalog Numbers) and the user may decide to add a third field i.e., "First Aid Measures" to this relation. Or the user may decide to link a source model field of "First Aid Measures" to a *global* model field of "Treatments" instead of an existing link to a global model field of "Response", determining it to be a better semantic link than the one existing. Changes in one aspect of knowledge can affect other aspects, for instance a change to a global or source model affects the links involving the particular concepts and fields changes in these models. We are currently developing a formal framework (and associated tools with a graphical interface) to support such modifications. We provide a brief summary of a formal integration knowledge modification framework we are developing.

Modifying Integration Models and Linkages

We are constrained by lack of space to elaborate, but essentially we employ logic based languages where we use *assertions* to declare the contents of a source and *logical rules* to inter-link content from multiple sources (Wiederhold, 1992). Regardless of the particular representation, the three elements of (i) a global domain model, (ii) source models, and (iii) links between the domain and source models are generic to any integration approach. We thus define this combination of elements as an entity or first-class data element in its own right that can be manipulated in a principled fashion. We use the term *concept* to refer to the basic structured data representation formalism in any chosen representation, such as a concept or class in a description logic based model, a table in a relational model or an element in an XML model. We now define an *integration artifact set* I, as a 3-set I = <*D, S, A*> where *D* is a set of global concepts, *S* is a set of source concepts, and *A* is a set of relationships. The artifact set is generic in that global or source concepts could be in any chosen formalism as could be the set of relationships. In our implementation we are employing an open-source version of the Prometheus information mediator (Prometheus 2009) available to us for this project. We have thus employed the Prometheus representation where concepts are represented as logical Datalog predicates and relationships are represented as logical Datalog *axioms* relating global concepts (LHS) with one or more source concepts and conditions (RHS). We treat the integration artifact set itself as a first-class data element and provide below a principled approach – in the form of a set of algebraic operators for modifying such an integration model.

OPERATOR	EFFECT
1) MODIFY CONCEPT This operator, γ, take as parameters a source or global concept and an attribute and adds (or deletes) that attribute from the concept. In the case of modifying a source concept there is also the propagating effect of modifying any axioms referring to that source concept and in the case of source concept attribute deletion there is an additional effect of modifying domain concepts that are part of axioms involving that source concept.	$\gamma_{Si,anew}(I) \rightarrow I'$; I' = <D,S',A'> where a_{new} is the attribute to be added to the source concept Si ε S S' = S-{Si} U {Si(a_1,..,a_k,a_{new})} A' = A – A_{si} U A_{newsi}, where A_{si} is the set of all axioms that contain an instance of si, and A_{newsi} is the set of axioms obtained by replacing source concept si in each axiom in A_{si} with its new definition. $\gamma_{si,-ai}(I) \rightarrow I'$; I' = <D',S',A'> S' = S-{Si} U {Si(a_1,..,a_{i-1}, a_{i+1} a_k)} A' = A – A_{si} U A_{newsi}, where A_{si} is the set of all axioms that contain an instance of si, and A_{newsi} is the set of axioms obtained by replacing source concept si in each axiom in A_{si} with its new definition. D' = D – D_{si} U D_{newsi} where D_{si} is the set of all axioms that contain an instance of si, and D_{newsi} is the set of axioms obtained by replacing the domain concept in each axiom in D_{si} by the domain concept with attribute a_i eliminated.
2) MODIFY CONCEPT SETS D, S This operator, μ, takes as parameter a	$\mu_c(I) \rightarrow I'$; I = <D',S',A> where D' = D U {c} and D is unchanged (similarly for S' and S)

concept and adds/deletes the concept to/from the set of global or source concepts.	$\mu_{-c}(I) \rightarrow I'$; I = <D',S',A'> where D' = D - {c} and D is unchanged (similarly for S' and S) A' = A – Ac, where Ac is the set of axioms that contain concept c.
3) MODIFY AXIOMS SET This operator, Θ, takes as parameter a new/existing axiom and adds/deletes that axiom to/from the set of axioms A.	$\Theta_{a}(I) \rightarrow I'$, I' = <D,S,A'> A' = A U {a} $\Theta_{-a}(I) \rightarrow I'$, I' = <D,S,A'> A' = A - {a}
4) RENAME This operator, Ψ, takes as parameters an existing concept name, a new concept name and renames an existing source domain concept with the new name	$\Psi_{s,sn}(I) = I' = <D',S',A>$ The effect of Ψ s,sn (I) is to simply rename s to sn in S or D in I resulting in a new S' or D'.
5) MERGE This operator, Ω, takes as parameters two domain concepts with the exact same attribute set and adds a domain concept that is the union of the two.	Given $d1(a_1,..a_k)$, $d2(a_1,..,a_k)$, the effect of $\Omega_{d1,d2}(I)$ where d1, and d2 are domain concept is as follows: A new domain concept, $dnew(a_1,,a_k,a_{k+1})$ is added to D resulting in D' The following new axioms are added to A resulting in A'. dnew(X1,X2,..,d1) ← d1(X1,..,Xk) dnew(X1,X2,..,d2) ← d2(X1,..,Xk)

Below we present a simple example where an integration model I (comprising of 1 domain concept, 2 source concepts and 1 integration rule) is modified by a sequence of operators (modify concept followed by modify concept set) to result in a new integration model I'. Note that the axioms are also impacted by this modification.

$\mu_{hazmateffects}(\gamma_{hazmat,-summary}(I)) \rightarrow I'$

I	I'
D: glob-hazmat(id, name, building, classification, summary) S: hazmat-locations(id, name, building) hazmat(id, classification, summary) A: glob-hazmat(id, name, building, classification, summary) ← hazmat-locations(id, name, building) ^ hazmat(id, classification, summary)	D': glob-hazmat(id, name, building, classification, summary) S': hazmat-locations(id, name, building) hazmat(id, classification) hazmateffects (id, effects) A': glob-hazmat(id, name, building, classification) ← hazmat-locations(id, name, building) ^ hazmat(id, classification)

DISCUSSION

We would like to discuss this work in progress to inform, seek feedback on, and ideally involve more members of the community in this community driven approach to data integration for emergency response. The research described here has been supported by a FEMA grant EMW-2007-FP-02535 and NSF grants 0331707 and 0403433.

REFERENCES

1. Ashish, N., Lickfett, J., Mehrotra, S., and Venkatasubramanian, N. (2009) The Software EBox: Information Integration for Situational Awareness, *IEEE ISI Conference*, Dallas TX.
2. Tactical Survey Group. (2009) Web: http://www.tacticalsurvey.com
3. Ashish, N. and Mehrotra, S. (2009) XAR: An Integrated Framework for Semantic Extraction and Annotation. Cases on Semantic Interoperability for Information Systems Integration: Practices and Applications. Yannis Kalfoglou (Editor), IGI Global, June 2009
4. Chiang, Y., Knoblock, C., Shahabi, C., and Chen, C. (2009). Automatic and Accurate Extraction of Road Intersections from Raster Maps. GeoInformatica 13(2): 121-157
5. Halevy, A.Y., Ashish, N., Bitton, D., Carey, M., Draper, D., Pollock, J., Rosenthal, A., and Sikka, V. (2005). "Enterprise information integration: successes, challenges and controversies". *SIGMOD 2005*: 778-787.
6. Berners-Lee, T., Hendler, J., and Lasilla, O. (2001) The Semantic Web Scientific American, 284(5): 34–43
7. Yahoo Pipes. (2009) Web: http://pipes.yahoo.com

8. OpenII. (2009) Web: http://openintegration.org
9. XAware. (2010). Web: http://www.xaware.org/
10. Gio Wiederhold. (1992) Mediators in the Architecture of Future Information Systems. IEEE Computer 25(3): 38-49
11. Prometheus (2009). Web: http://www.isi.edu/integration/Prometheus/
12. Saha, B., Stanoi, I., and Clarkson, K.L. (2010). Schema Covering: a Step Towards Enabling Reuse in Information Integration, *Proc of ICDE Conference*, Long Beach CA