Information Management for Crisis Response in WORKPAD

Alessandro Faraotti
IBM Center for Advanced Studies of Rome
alessandro.faraotti@it.ibm.com

Antonella Poggi
Università di Roma "La Sapienza"
Dipartimento di Informatica e Sistemistica "Antonio Ruberti"
antonella.poggi@dis.uniroma1.it

Berardino Salvatore
IBM Rome Solutions Lab
salvatore@it.ibm.com

Guido Vetere
IBM Center for Advanced Studies of Rome
gvetere@it.ibm.com

ABSTRACT

WORKPAD (EU STREP project FP6-2005-IST-5-034749) is an experimental platform for Crisis Response which adopts a decentralized, event-driven approach to overcome problems and limitations of centralized systems. The flexibility of P2P networking is relevant when different organizations must get rapidly integrated the one another, without resorting on standardized ontologies and centralized middleware components. This paper illustrates the main features of the Information Integration platform we've designed. A number of relevant technical and theoretical issues related to decentralized platforms are discussed in the light of specific needs of Crisis Response.

Keywords

INFORMATION MANAGEMENT, DATA INTEGRATION, KNOWLEDGE REPRESENTATION, AUTOMATIC REASONING, ONTOLOGIES, EVENT-DRIVEN ARCHITECTURES, SERVICE ORIENTED ARCHITECTURES

INTRODUCTION

The European Project WORKPAD aims at developing a software infrastructure that allows human operators of Civil Defense organizations to work collaboratively in emergency scenarios. Such extreme situations require government agencies, civil protection forces, and volunteer organizations to reach a high degree of coordination and to effectively and timely exchange information. Data gathered from operators on the field has to be made readily available to all decision-makers that participate to the rescue operations, both in the disaster area and in the operation centers apart. Another important aspect to consider is local and central coordination of teams, either belonging to nationwide Civil Protection organization, or local administrations, or non-governmental forces, even if their information systems have never met before. Civil Protection teams operate on the field by pursuing common goals, but at the same time they have to manage unpredictable situations, re-adjust priorities, collect information from multiple sources and evaluate them, report conditions, receive and transmit orders, mind their own and other's safety and so on. In order to efficiently and effectively support teams in these complex situations, mission critical IT systems must enable timely sharing of integrated information and streamline processes, while keeping the needed flexibility with respect to different organizations and rapidly changing conditions. These requirements result in very hard challenges for information systems. The idea at the basis of WORKPAD is to approach such a complex scenario by means of a decentralized, event-driven information integration environment. In essence, information flows and integration patterns are not established beforehand by a centralized body in a centralized system, but are distributed among virtually any organization that takes part in operations, thus being dynamically extended and adapted to new organizational needs and unforeseen situations. To the sake of flexibility and robustness, WORKPAD's platform is designed to work with minimal requirements and even with limited connectivity. A plain Web Services support is required to enable back-end systems wrapping their existing databases, answering queries, and notifying updates, which provides the basis for making them acting as a decentralized, and yet integrated network. When some peers on the network are not reachable, the other peers still continue to provide their functionalities by temporary locking out the unreachable peers until these become available again.

If the concept behind WORKPAD is straightforward, the realization requires a deep insight into a number of non trivial problems, including event handling in time-bound systems, semantic data integration, multi-agent systems, knowledge representation and reasoning. In fact, while centralized control-room based coordination can be developed basing on traditional, industry-strength technologies, decentralized internetworking raises technical and theoretical issues that are still open. First, there is the problem of modeling formal semantics of peer-to-peer information integration. On one hand, WORKPAD can leverage state-of-the-art integration platforms based on classic model-theoretic semantics for databases, where the notion of certain answer is extended to encompass distributed and heterogeneous sources. On the other hand, it is not immediately clear what an answer would represent when mappings among information systems are established on a P2P basis, and neither semantic mediators nor global ontologies are available. Such a formal semantics has therefore been the first objective of our research. With formal semantics in place, another relevant issue is that of balancing tractability and expressiveness. It is known that powerful Semantic Web languages such as OWL, when used to manage large amounts of data, causes query answering to be intractable. This aspect clearly hinders their use in data-intensive, time-bound platforms such as WORKPAD. For the sake of both simplicity and computability, WORKPAD's platform adopts concept languages of reduced, and yet non-trivial, expressiveness.

Real time notification on critical updates is clearly a key requisite for any rescue-support platform. This is needed to manage alerts, as well as to get a certain level of information synchronization. In fact, databases managed by WORKPAD's peers cannot support mutual synchronization with standard means, since most of them they are not even aware of each other at design time. To address notification, the platform supports a publishing-subscription mechanism that allows peers to get notified when the extension of relevant concepts in a single peer changes.
What follows summarizes approach we've adopted to provide WORKPAD with the most appropriate features to fulfill the requirements of Information Management for Crisis Response as they were analyzed in the first phase of the project [5]. We recall that the project also includes the integration of back-end information systems with workflow management for mobile-ad hoc networks [9], which is not addressed in the present paper.

DECENTRALIZED INFORMATION INTEGRATION

Due to the recent increase of safety threats such as environmental disasters or terrorist attacks, Crisis Response has become a relevant application field for Information Integration technologies, where applications are characterized by data-intensive activities requiring the transparent access to many different autonomous data sources. Analysts say that the biggest challenge when undertaking effective disaster response efforts is interoperability and access to data. As a matter of fact, the lack of information integration in an emergency scenario inhibits government agencies and volunteer organizations to successfully communicate and act in a coordinated way. An information integration solution would provide the capability to assemble teams and integrate information from existing sources, respectively using each source usual protocol, format and scheme. Nowadays, the most common approach to information integration for Crisis Response consists in centralization, meaning that heterogeneous sources and systems are consolidated or mediated by a single point of access. However, to be effective, this approach requires most of the integration work to be done before the crisis occurs, and little attention is paid to support integration during the crisis management. On the contrary, decentralized information integration appears to be a much more promising approach. In particular, here we focus on adopting *peer-to-peer* (P2P) architectures, which have been considered as a more flexible way for human operators to work collaboratively with minimal adaptation efforts, by exploiting a sort of *collective intelligence*[1], which can hardly be framed in rigid centralized systems.

Basically, P2P systems are based on the idea of removing the distinction between servers and clients. The advantages of these systems, compared to the client-server traditional ones, spread essentially from the pooling of storage space, bandwidth and computing power of the network components, thereby the capacity of the network increases as more members join, which is in contrast to client-server architectures, where adding more clients means slower data transfer for all users. In a P2P architecture, peers act as equal, each one merging the roles of client and server. Note that this makes P2P architectures fundamentally different from federated or distributed databases architectures [2]. Indeed, both in a distributed and in a federated database [7], there is one logical schema, that for the former is defined a priori, and for the latter is the result of an agreement between the participant databases. On the contrary, P2P architectures do not assume the presence of any global mediated schema. In addition, in a federated database, any of the component databases can itself be a federated database, while in a P2P database, the smallest granularity is represented by a peer system.

More specifically, the WORKPAD approach to Crisis Response consists in letting each organization play the role of an information peer in a P2P architecture, where systems can simultaneously provide information to and receive information from other organizations, without relying on any centralized feature. Moreover, each peer plays the role of integrator, in that it integrates the data provided by its local sources with the data provided by

1 This has been pointed out by Tim O' Reilly in the online article "Peer to Peer Information During Disasters" http://radar.oreilly.com/archives/2007/02/peer_to_peer_in.html

other organizations. In fact, a peer can be accessed either by other peers or by front-end applications in order to support various processes at front-end for Crisis Response. In particular, the latter feature allows operators on the field to use front-end applications, e.g. GIS information retrieval or Context Monitoring, to get or set information that is relevant to the situation they are facing or the action they are taking. Notably, such information is not necessarily contained into a single back-office system, but is potentially spread over the P2P information integration system network, that transparently collects, reconciles and delivers it through an integrated platform.

Specifically, in WORKPAD, each peer is an information system that (i) manifests an *ontology*, providing a formal description of the data it makes available, (ii) allows a rapid integration of a variety of data sources, both internal and external, (iii) is capable of answering conjunctive queries expressed in terms of the ontology, and (iv) notifies information updates to those who are interested in them. Hence, each peer provides data according with its own ontology, encodes its acquaintances with other peers by establishing the relationship between their ontologies and its own by means of a set of assertions, called *mappings*, replies to queries posed against its schema by integrating its local data with data coming from the peer network, and proactively notifies updates.

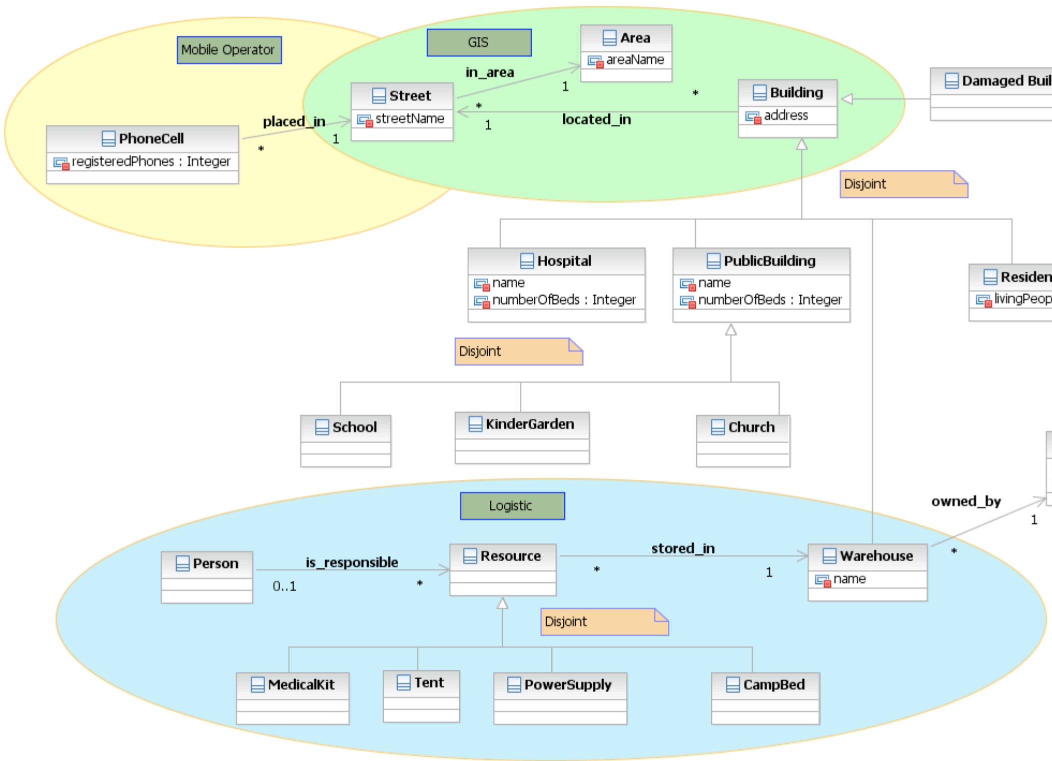

Figure 1 an example of ontology

While the approach described above is rather simple and intuitive, there are relevant issues that need to be addressed to make it working. First, given the amount of data at each peer, query answering should be as efficient as possible. This requires a language for the peer ontology and mappings that offers the right trade-off between expressiveness and efficiency in query answering. As already mentioned in the introduction, OWL is not suitable for our purposes, since query answering is intractable. Hence, in WORKPAD, we rely on the results by the SAPIENZA University of Rome, showing that in a traditional hierarchical data integration system, i.e. a single peer that provides access to different local data sources through its ontology and is not connected to any other peer, query answering is as efficient as the evaluation of an SQL query over a Relational Database Management System (RDBMS) under the following assumptions: (i) the ontology is expressed using a specific Description Logic named DL-Lite [10], and (ii) *local* mappings have the form:

$$Q(x) \rightarrow F$$

where Q is an SQL query over the local data sources, x is a variable (or couple of variables) and F is an assertion over the peer ontology, built on top of x. Intuitively, the mapping above states that for each tuple t satisfying Q over the ontology, there exists an object (or a couple of objects), built on top of t, that make F true in the peer ontology. Hence, in WORKPAD, we follow the same assumptions and further assume that for each peer, a set of *global* mappings among the peer and other network peers have the same form as local mappings, with the provision that Q is restricted to be a conjunctive query over the other peer ontology. The second issue that need to be addressed concerns the formal semantics to adopt for query answering in the case of peer-to-peer integration systems. From the standpoint of a peer, the problem is how to make use of information gathered by other peers. In particular, it was shown that under arbitrary mappings, query answering under the first-order semantics is undecidable [3]. In such semantics if two peers P and P' make an inconsistent assertion, i.e. for a given individual x, P asserts Q(x) and P' asserts $\neg$Q(x), then the knowledge base is in fact invalidated. To overcome this problem, an *epistemic semantics* was first proposed.[3]. Intuitively, under an epistemic semantics, a query to a peer belonging to a P2P information integration system has to take into account the *epistemic* difference between what the peer *knows* (i.e. local data) and what the peer *knows the others know* (i.e. information collected on the network). Then, it is assumed that only the knowledge from other peers that is *certain* is collected into one peer. Peer-to-peer information integration systems using the epistemic semantics have been studied and experimented in a joint research initiative of Sapienza Università di Roma and IBM Italia, named *Hyper* [4]. The experiments results were promising. However, by applying such approach to WORKPAD, it rapidly emerged that in order to better exploit the overall system features, it would be suitable to be able to discriminate inside a single peer between its own original knowledge and imported knowledge. In fact, when the originators of assertions can be tracked down, it is possible to remove conflicting assertions in a selective way, by using selection policies that may be based, for example, on the degree of trust assigned to each of the peers in the network. This intuition has lead to an approach that we called *doxastic*, where for each peer, knowledge from another peer is kept separated from its own knowledge, so that a peer, when asked, only refers to other peers' known facts, without taking responsibility for them.. This approach is well-suited to model situations in which inconsistencies may easily arise. Intuitively, a fact known by a peer cannot be inconsistent with a fact known by another peer, since they are on two different *epistemic contexts*, which is good for systems like WORKPAD, where each organization is autonomous and has its own stand-alone local data. A detailed formalization of this approach has been presented in [8].

Another key issue that emerged regards named entities, and the way they are referred networkwide, which can also be seen as related to the so-called *object-relational impedance mismatch*. In a globally integrated P2P network of loosely-coupled knowledge bases such as WORKPAD, it is not a realistic assumption that all knowledge bases hold the same identifiers for the same entities of the world. For example, a peer P may identify a person with her/his name, surname and address while a peer Q may identify persons with her/his social security number. The solutions to this problem cannot be realistically solved at the 'intensional level', by enforcing a synthetic functions fPQ for each pair of peers (P,Q) of the network such that the identifier of an object in Q is mapped into a corresponding identifier in P (fPQ(idQ) = idP). On the other hand, addressing the problem at the 'extensional level', where fPQ enumerates explicitly the identifiers mappings between Q and P for all the objects in the world, would require a centralized registry of named entities, which would break one of the starting assumptions of our project. Currently, the problem of individual mappings is left to peers to be handled case by case, along with conceptual level mappings, but we recognize this as one of the major open issues of our approach.

DECENTRALIZED EVENT MANAGEMENT

One of the main goals of WORKPAD is to allow mission critical information to be circulated between peers and applications in an appropriate (relevant to both sender and receivers), quick and efficient (i.e. resource-aware) manner. In fact, in rescue scenarios such as those addressed in WORKPAD, a single bit of information delivered to the right people at the right time can make the difference when people's life and health are at risk. Notification services, which are part of the set of WORKPAD back-end services, are designed to meet these purposes. More specifically, their responsibility is to allow peer keeping their data synchronized, and have external applications alerted when relevant updates are available. In order to ensure efficiency and loose-coupling between senders and receivers, a publish-subscribe design pattern has been adopted: when an update event is produced in one of the data sources in the network, a suitable notification message is sent to each parties that have declared their interest in receiving news about topics related to the event[2]. To handle this, a specific abstract component provides *notification management* capabilities. The market offers several specialized middleware solutions that provide notification management functionalities such as ESB and queues, often based on standards such as JMS for Java messages and WS-Notification for Web Service messaging.

---

2 Alternative approaches based on data polling have been excluded because they are more resource consuming and thus less efficient

However, to fulfill WORKPAD requirements, no special middleware is supposed to exist beforehand, nor it is acceptable to plan and execute a complex deployment scenario (like the installation of distributed middleware) right after the onset of an emergency. For this reason, WORKPAD provides own point-to-point notification capabilities, with basic features such as message persistency, delivery and retry. It should be noted however that this approach does not prevent the possibility to provide a different packaging of WORKPAD services that relies instead on external messaging services, in case the existing technical and organizational environment would permit and require it.

Next sections will illustrate how updates are propagated to interested parties that may be either other peers or external applications, that are responsible to alert rescuers deployed in the emergency scenario by providing all new information that is produced everywhere in the network that is relevant to them. More formally, an update in a peer data source consists in one of the following facts: a new tuple is inserted in one of the data source's persistent structures (i.e. tables, in case of RDBMS), an existing tuple is changed in the non-identifying part and an existing tuple is deleted[3]. Any of the updates described above, may cause the peer knowledge base to change, which triggers one or more notification messages to be posted to the appropriate recipients. Hereby after with *topic* or *subscription topic* we mean an abstract recipient where messages of a certain category are posted by one or more *message producers* and to which any interested parties (*message consumers*) can subscribe in order to receive them. The publish-subscribe pattern we have designed requires any WORKPAD peer P to support a specific subscription service. Any client C (either a WORKPAD peer P' or an external application A) that is interested in receiving news from a peer P, must implement a specific notification endpoint, which P will use when updating. Subscription topics are defined with respect to the publisher's ontology: each concept, role, and attribute name (i.e. any symbol in the predicate alphabet of peers' ontologies), is eligible as a topic for update subscriptions.

It is important to recall that subscribers will be notified with knowledge items, i.e. concepts, roles, and attribute instances, rather than database values. Being the individual matching problem basically unsolved at a systematic level in the current state of the art, and considering that nothing prevents peers to map equivalent concepts each other, it is easy to see the risk of falling in never-ending notification loops caused by different and inconsistent recognition of individual entities. WORKPAD provides a mechanism that prevents this effect, by supplementing notification message with the identifiers of peers that have already handled it. The general contract is that each peer that receives a notification message should first check if its own identifier is already present in the list. If this condition occurs, the received notification should not be taken into account. Otherwise, the first time the peer receives a notification message concerning a "thread" or chain of changes, it must add its own identifier to the list and use that list for any outbound notifications that may be produced as a consequence of local updates.

In the following sections are described notification services, in particular how clients (both peers and external applications) express their "interest" about some topic during subscription, how notifications are produced, propagated and handled. Moreover, as event-driven architectures may pose problems because of high network traffic when updates are frequent, and being this the case for WORKPAD, some query aggregation, caching and priority-based mechanisms have been studied to reduce these network effects. These mechanisms are described hereafter too.

CURRENT WORKPAD DESIGN AND IMPLEMENTATION

WORKPAD P2P Data Integration System is implemented by providing each back-office information system with a set of specific Web Services. These systems therefore qualify as WORKPAD Knowledge Peers (KPs), which are ready to support integrated information exchanges. It is important to note that KPs are not requested to sign into any centralized component such as registries or brokers. KPs are therefore the only structural elements of WORKPAD's Data Integration. Each KP makes the network aware of itself by publishing an XML document called *manifest*. KPs can be configured by a specific environment used by both knowledge engineers and a database engineers. A KP takes in input an ontology, a set of sources and some sets of mappings and provides services to perform queries to the whole data integration system and to manage notifications. KP clients can be either external application (front-end peers) or other KPs.

Figure 1 outlines the current KP implementation, based on (i) a visual interface layer, (ii) a service layer, (iii) an implementation layer, and (iv) a resource layer.

[3] Notice that if an existing tuple is changed in its identifying part (i.e. the primary key, in case of RDBMS), this fact can be expressed as a sequence of pairs of delete\insert updates..

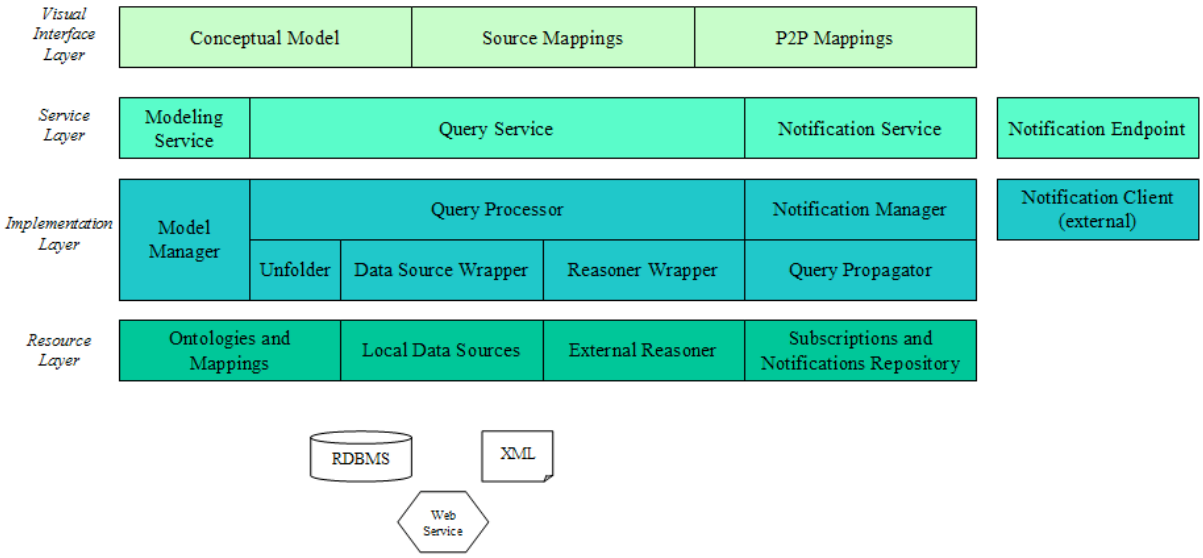

Figure 2 WORKPAD KP layers

The visual interface layer consists in a configuration environment implemented by an Eclipse based rich client platform that can be deployed in a separate node. This tool assists the user, through wizards and other tools, to fully define a configuration and to upload it on the KP.

The first step to configure a KP is to define its unique identifier and the URLs of services that it is going to provide.. Then the local ontology has to be set. The configuration environment allows importing any OWL ontology which can be designed with an existing modeling tool (e.g. Protégé). OWL ontologies are validated and translated in DL-Lite, which may cause some of the formal semantics of the source ontology to be lost (notice, however, that this is unavoidable for keeping efficiency in query answering). Then, modeling tool automatically generates the manifest of the new peer.

The next step is to define sources for the KP. The tool provides a wizard to automatically import the schema of a DBMS in terms of source relations. Once both the sources and the ontology are defined, the tool assists the user in defining mappings between them. Moreover, the modeling environment allows importing other KPs' manifests together with their ontologies and to define P2P mappings to that peers.

Once defined, the configuration can be uploaded into the KP through a specific service. Usually, each KP belongs to a distinct organization, and interacts with the other KPs of different organization. In order preserve inter organization security the service layer of the KP provides two different interfaces: one accessible from applications of the same organization, and the other accessible form other KPs. Both interfaces include the query answering and the notification services.

WORKPAD services are implemented in a layer which actually contains the logic of the KP. The implementation layer uses the underlying resource layer to access peer resources. In particular the resource layer wraps an external reasoner and allows to access to the local sources. In the current implementation the used reasoner is QuOnto [1] and only DBMS are supported as local sources.

The KP data integration logic supports the query answering and the notification handling services. Through a specific "Query Answering" service, a KP is able to answer to conjunctive queries (CQs) expressed over the local ontology's alphabet. A Query Processor component handles the requests and uses QuOnto to perfectly reformulate the query into a union of conjunctive queries. The reformulated CQ is evaluated on the local ABox and the answer is built and returned. The local Assertional Box (ABox) contains the facts known by the KPs with metadata about provenience to support the doxastic approach described in [8].

The "Notification Service" allows the KP synchronizing its knowledge. This service is used both for the synchronization of the KP Knowledge and to provide a notification service to external clients. Through this service KP receives notifications about changes in the facts known by other peers. Each notification asserts that facts about one ore more predicates, belongings to the ontology alphabet of the sender, are changed. When a KP receives a notification invalidates the related facts known about the sender.

Each time a fragment of the local knowledge is invalidated the KP decides to update it according to its own update policy. During the update process ad hoc queries are generated by the KP and sent to the local source or to other peers by means of the appropriate set of mappings. An 'unfolding' component takes the role of using mappings in order to reformulate the conjunctive queries. It uses P2P mappings to reformulate conjunctive queries into the other KP's ontology and the local mappings to reformulate conjunctive queries in terms of the source relations. Note that unfolding process is also in charge of dealing with the impedance mismatch between values of the source and individuals of the ontology, by using transformation functions (a.k.a. Skolem functions) which are part of local sources' mappings.

The update policy adopted in the current implementation is tailored to optimize the trade-off between the need of having up-to-date knowledge and the need of reducing network operations. This is done by assigning a priority to the subscriptions. Notifications, regarding sensitive information, are taken into account by the KP that immediately updates its knowledge. Less significant knowledge is periodically updated through a back ground process.

The KP knowledge synchronization mechanism leverage publish-subscribe notification services described in the previous section. The KP uses mappings to set-up the synchronization process with the other KPs. In fact, KPs use P2P mappings from the local ontology to ontologies of other peers to find out which predicates of foreign ontologies are needed to maintain its knowledge. Notification of changes regarding mapped concepts will be subscribed accordingly. Knowledge engineers may also set priority flags on those concepts whose updates are most critical for the system to be responsive. In WORKPAD, a subscription request is done by giving some information to the KP acting a publisher. This information contains the endpoint URI on which the subscriber wants to be notified, a predicate indicating the topic and three parameters: (i) the priority of the notification, (ii) the interval in which at most one notification has to be sent to the subscriber and (iii) the time to live indicating notification expiration interval. When an update occurs, the KP create a notification instance and sent it to the subscriber taking into account the priority of the notification and calculating the intervals according to the subscription parameters. In the current implementation, the notification is considered to be sent in the interval obtained adding the minimum interval and the time to live interval to the current time. Periodically, a scheduler checks if there are notifications to be sent and in that case prepare them for the delivery.

Through mappings, notifications of updates are reflected on facts about some predicates of the KP local ontology. As explained in the previous section, when a notification is received by a KP, the correspondent local update (if any) is established according to an arbitrary update policy. If other peers have subscribed to the local predicates influenced by the notification, then the receiving KP propagates notifications of changes to the other KPs. In order to avoid notification loops, messages contain the list of peer that have either generated or received them. If a KP generates a new notification because of a change caused by some notification received from another peer, then the KP copies the list of the peers' identifiers in the new message. The KP does not generate updates based on notifications containing its own identifier, thus preventing the system to fall in recursion loops. In the current implementation, a similar notification based service allows the KP to be notified about changes in the data stored in the local sources. Every update on the data source has to be notified to the peer. The responsibility to notify updates is up to the application that performs them or directly up to the source databases through triggers. These notifications have to be sent through the service "Notify Update" provided by the KP. Finally, note that, in order to let the system easy to install without particular dependencies, the notification services provided by a KP are implemented without relying on any particular middleware. Anyway, a more sophisticated implementation would be possible basing on existing middleware, e.g. those implementing the WS-Notification specification[4].

CONCLUSION AND FUTURE WORK

WORKPAD is an experimental platform for Crisis Response which overcomes problems and limitations of centralized systems and provides innovative features by adopting a decentralized, event-driven approach. The main advantage of adopting P2P networking is flexibility, which is relevant when different organizations must find a way to quickly set-up cooperation patterns without previously defined semantic agreements (i.e. globally standardized ontologies) and without reliance on centralized middleware components. On the way of designing such a platform, a number of relevant issues have to be addressed, such as formal semantics of knowledge integration (both at intensional and extensional level), tractability of query answering, time-bound notification of information updates. Some of these issues, such as that of distributed ontology-based data integration, are not addressed by industry-strength solutions; they are very open research fields for both technological and

[4] An example of existing middleware implementing the WS-Notification specification is OASIS, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsn

theoretical aspects. This paper has provided an outline of the main options we've adopted, based on state-of-the art in research and development, which can be useful for bringing light on key aspects of Crisis Response information management.

We have illustrated the main features of our approach to information integration and event management, and motivated the benefit of adopting a decentralized architecture for developing flexible and effective IT systems for Crisis Management. Entering a decentralized services network such as WORKPAD, however, does not come for free. Information managers who want to plug existing systems into the network are requested to either provide high level conceptual models of their data (i.e. local ontologies) or to adopt standard conceptual models (i.e. shared ontologies), to map them to their concrete databases as well as to other peers, to support publish-subscribe protocols both to notify updates and to get notified. Even if integration can be incremental and not disruptive with respect to the normal working of legacy systems, the task of putting systems together in a P2P way is far from trivial, and more research is needed to understand how to facilitate it. A number of issues including security, data provenance and quality, consistency management, also require more investigation and are part of our future work. Moreover, there are problems such as individual mapping whose solution can hardly be given in a decentralized manner at a merely technological level, but unavoidably require inter-organizational processes and a certain degree of central control. Also, the 'relativistic' approach to semantic integration we've adopted, which consists in the possibility for each peer to adopt its own, possibly idiosyncratic ontology, could be enhanced by supporting network-wide semantic standards as soon as they are delivered by standardization bodies. In fact, a uniform way for referring to foundational notions (e.g. spatial-temporal ones) would significantly boost Crisis Response systems interoperability, hence facilitating the adoption of decentralized platforms like the one we are researching.

As a project, WORKPAD has entered the final stage, and a prototypical implementation of the platform will be made available for experimental purposes by September this year. The implementation of some of the features we've illustrated in this paper is still prototypical; moving to an industrial strength, generally available solution will require more work.

REFERENCES

1. Andrea Acciarri, Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, Mattia Palmieri, and Riccardo Rosati QuOnto: Querying Ontologies. *In Proc. of AAAI 2005*, 2005.
2. A. Bonifati, P. K. Chrysanthis, A. M. Ouksel, and K. Sattler. Distributed databases and peer-to-peer databases: past and present. SIGMOD Rec. 37, 2008.
3. Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, and Riccardo Rosati. Logical Foundations of Peer-To-Peer Data Integration. *In Proc. of PODS 2004*, 2004.
4. Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, Riccardo Rosati, and Guido Vetere. Hyper: A Framework for Peer-to-Peer Data Integration on Grids. *In Proc. of ICSNW 2004*, 2004.
5. Massimiliano de Leoni, Fabio De Rosa, Andrea Marrella, Massimo Mecella, Antonella Poggi, Alenka Krek and Francesco Manti. Emergency Management:from User Requirements to a Flexible P2P Architecture. *In Proceedings of ISCRAM 2007,* 2007.
6. Antonella Poggi, Domenico Lembo, Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, and Riccardo Rosati. Linking Ontologies to Data. *Journal on Data Semantics.* X: 133-173, 2008.
7. A.P. Sheth and J.A.Larson. Federated database systems for managing distributed, heterogeneous, and autonomous databases. ACM Comput. Surv. 22, 3, 1990.
8. Guido Vetere, Fabrizio Venditti, and Alessandro Faraotti. Semantic Integration of Peer to Peer Systems: A Doxastic Approach. *In Proc. of ICSC 2008*, 2008.
9. Catarci, T., de Leoni, M., Marrella, A., Mecella, M., Vetere, G., Salvatore, B., Dustdar, S., Juszczyk, L., Manzoor, A., Truong, H.-L. (2008). Pervasive and Peer-to-Peer Software Environments for Supporting Disaster Responses, IEEE Internet Computing, January
10. Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, Riccardo Rosati, and Guido Vetere, DL-Lite: Practical Reasoning for Rich DLs, Proc. of the 2004 Description Logic Workshop (DL 2004)