FFI-rapport 2012/01543

Information integration experiment at NATO CWIX 2012

Bjørn Jervell Hansen, Jonas Halvorsen and Audun Stolpe

Norwegian Defence Research Establishment (FFI)

5. September 2012

FFI-rapport 2012/01543

1176

P: ISBN 978-82-464-2163-6
E: ISBN 978-82-464-2164-3

Keywords

Nettverksbasert forsvar

Informasjonsintegrasjon

Ontologi (databehandling)

Semantisk web

Approved by

Rolf Rasmussen	Project manager
Anders Eggen	Director

English summary

An essential part of the NATO Network Enabled Capability (NNEC) vision is sharing of information in order to enhance shared situational awareness. This is anticipated to be an important contributor to building the decision superiority that in the end is expected to lead to increased mission effectiveness when put to use by decision makers.

However, in order for the information to lead to improved situational awareness, it also needs to be integrated. Traditionally this integration is done manually by the decision makers, but the increasing amount of information available when conducting operations according to NNEC means that there is a need for automated means to assist the decision makers in this integration. So far, the emphasis of the technical work on NNEC has been on how to make information available to other systems in the environment, but in order for NNEC to be of use to decision makers the challenge of integrating this information ultimately also needs to be addressed.

In this report, an approach for automated information integration suited to NNEC is described. The main idea is to give a military decision maker the possibility to request information using a vocabulary he or she is comfortable with, and have an information system taking care of harvesting and integrating the information from the appropriate sources. In such a scenario, the user does not need to know anything about the available sources or how they represent their information.

The approach rests upon:

1. Query Rewriting: How to rewrite a query (in this case the user's information request) according to ontologies defining the user's vocabulary and the different information source vocabularies
2. Information Discovery: How to discover information sources relevant to the information request
3. Federated Query Processing: How to aggregate the relevant information from the different sources found through information discovery

The approach was demonstrated successfully at NATO CWIX 2012 in cooperation with NATO C3 Agency (NC3A). This successful demonstration further strengthens our belief that information systems built using semantic technologies can offer the level of flexibility that is needed to support the essential information integration in the increasingly dynamic NNEC environment.

Sammendrag

NATO Network Enabled Capability (NNEC) bygger i stor grad på at utstrakt informasjonsdeling vil lede til økt felles situasjonsbevissthet, noe man mener beslutningstakere vil kunne utnytte til å skape økt effektivitet i utførelsen av operasjoner.

For at den delte informasjonen skal kunne lede til forbedret situasjonsbevissthet, må den imidlertid integreres. Denne integrasjonen gjøres tradisjonelt manuelt av beslutningstakere, men den store økningen i informasjonstilgangen, som blir et resultat av å gjennomføre operasjoner i henhold til NNEC, betyr at det er et behov for automatisert integrasjonsstøtte til beslutningstakerne. Så langt har teknologifokuset rundt NNEC dreid seg om hvordan man kan gjøre mer informasjon tilgjengelig for beslutningstakerne. Men ønsker man å omsette dette i økt situasjonsbevissthet, må man også håndtere utfordringen det er å integrere denne informasjonen.

I denne rapporten beskriver vi en tilnærming til automatisert informasjonsintegrasjon som er tilpasset behovene i NNEC. Hovedidéen er å gi en militær beslutningstaker muligheten til å spørre etter informasjon med et vokabular han eller hun kjenner, mens et informasjonssystem håndterer å samle inn og integrere informasjonen fra relevante kilder. I et slikt scenario trenger brukeren verken å vite noe om de tilgjengelige kildene eller hvordan informasjonen er representert.

Tilnærmingen bygger på:

1. Omskriving av spørringer: Hvordan skrive om spørringer (i dette tilfellet brukerens informasjonsbehov) i henhold til ontologier som definerer vokabularene til brukeren og de forskjellige informasjonskildene
2. Informasjonsoppdaging: Hvordan finne informasjonskilder som er relevante for informasjonsbehovet
3. Føderert prosessering av spørringer: Hvordan samle relevant informasjon fra de forskjellige kildene.

Dette ble demonstrert med suksess på NATO CWIX 2012 i samarbeid med NATO C3 Agency (NC3A), og dette styrker oss i vår oppfatning om at informasjonssystemer bygget ved hjelp av semantiske teknologier kan gis den fleksibiliteten som trengs for å støtte den viktige informasjonsintegrasjonen i NNEC.

Contents

1 Introduction

The primary objective when conducting operations according to NATO Network Enabled Capability (NNEC), and also Network Based Defence (NBD)[1], is attaining a high degree of shared situational awareness in order to obtain increased mission effectiveness. Information sharing is a prerequisite to this: Information systems should make their information available and understandable in order for decision makers to retrieve and utilise it when needed according to their role. (Buckman 2005).

So far, the emphasis of the technical work on NNEC has been on how to make information available to other systems in the environment. However, in order for NNEC to be of use to decision makers, the challenge of integrating this information ultimately also needs to be addressed. This challenge is further described in Section 2.

In the FFI projects 1085 - Semantic Services in the Information Infrastructure and 1176 - Service Orientation and Semantic Interoperability, we have explored semantic technologies (cf. Section 3) and their potential usefulness in the military domain with increasing emphasis on information integration (Hansen et al. 2007, Hansen et al. 2008, Hansen et al. 2010, Halvorsen & Hansen 2011). In this report, we continue to explore how semantic technologies can contribute to deal with the information integration challenges of NNEC. We focus on an approach that takes advantage of formal models (ontologies) in order to split a decision maker's information request into queries that can be processed by each relevant source (cf. Section 4). The more technical details of the approach, written with a technical audience in mind, are covered in Sections 5 - 7.

In order to explore this approach, and also get a feel for the potential operational benefit of such a solution, an experiment was conducted at NATO CWIX 2012 in collaboration with NATO C3 Agency (NC3A). More details on the setup of the experiment, as well as the results, are given in Section 8.

Our main experiences and observations from the CWIX experiment are captured in Section 9, while Section 10 concludes the report with a summary of the main results.

2 Information Integration in NNEC

One of the basic tenets of NATO Network Enabled Capability (NNEC) is improved information sharing among military units in order to enhance information quality and, in turn, shared situational awareness. This is anticipated to be an important contributor to building the decision superiority that in the end is expected to lead to increased mission effectiveness when put to use by decision makers (Buckman 2005).

In order to fulfill this vision, the available information sources has to share their information and this information have to be properly integrated.

Information integration is a fundamental challenge in any environment where several systems need to exchange information unless the systems in question have been explicitly designed to interoperate,

[1] In this report, NBD and NNEC are treated as equivalent

and the NNEC environment is no exception to this general rule. NNEC constitutes a complex environment with a wide variety of information sources whose information can be important even to users who didn't anticipate to use this particular information source. This is in particular true in coalition missions and missions involving non-military participants. Adding to this challenge is the fact that these systems tend to expose their information using different formats and models. Further, the NNEC environment is highly dynamic with regards to the participating information systems. It has to be expected that information sources with vital information can appear (or disappear) at any time.

While the challenge of making the needed information available has received a lot of attention, the NNEC information integration challenge has so far failed to attract the same level of attention.

3 Semantic Technologies

Semantic technologies is a family of information technologies that utilize formal models (ontologies) in order to capture the meaning (the semantics) of the information at hand. Formalizing the semantics using ontologies paves the way for computers to manage and process it. This makes it possible to build flexible systems, more adapted to handling a changing information environment than systems built using traditional information technologies. This matches well with the NNEC information integration challenge described in Section 2, thus semantic technologies is a promising family of technologies for realizing solutions in this area.

A particularly interesting group of semantic technologies are those related to the vision of the Semantic Web. The Semantic Web (Berners-Lee et al. 2001) refers to an enhancement of the current World Wide Web where the contents of the Web is made accessible to computers as well as to humans.

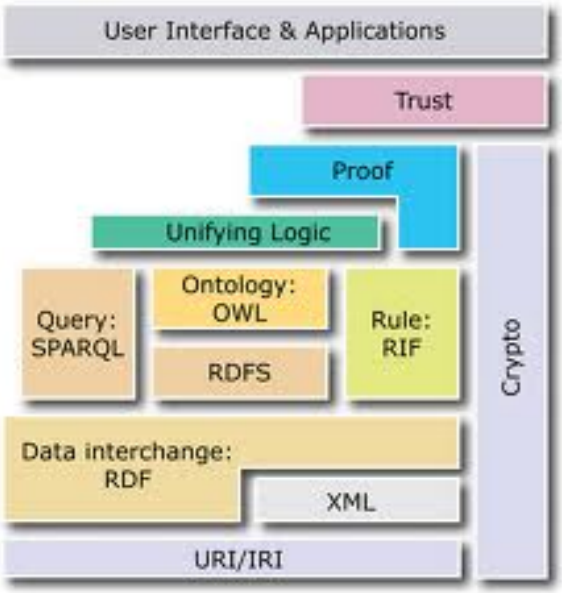

Figure 3.1 Semantic Web Layer Cake

Although the Semantic Web concentrates on the Web, the technology standards brought forward by the World Wide Web Consortium (W3C) in this area are also of interest in a more constraint domain like the military domain. We are thus focusing on these standards, which are shown in Figure 3.1. The most important Semantic Web standards in our work so far, has been:

RDF: Resource Description Framework (W3C 2004). A language for representing structured information in a graph.

OWL: Web Ontology Language (W3C 2009). A formally defined language for representing ontologies on the web. It is based on Description Logic, which is a family of logic-based knowledge representation formalisms with well-understood computational properties.

SPARQL: SPARQL Protocol and RDF Query Language (W3C 2012). A query language designed to allow querying on RDF graphs, much like SQL is used to query relational databases.

Further details on semantic technologies and the Semantic Web standards can be found in Hansen et al. (2007) and Hansen et al. (2010), and also at the W3C Semantic Web webpage [2].

4 Overview of the Information Integration Approach

The main idea in the information integration approach presented in this report, is to give a military decision maker the ability to request information using a vocabulary he or she is comfortable with and have an automated system taking care of harvesting and integrating the information from the appropriate sources. In such a scenario, the user does not need to know anything about the available sources or how they represent their information.

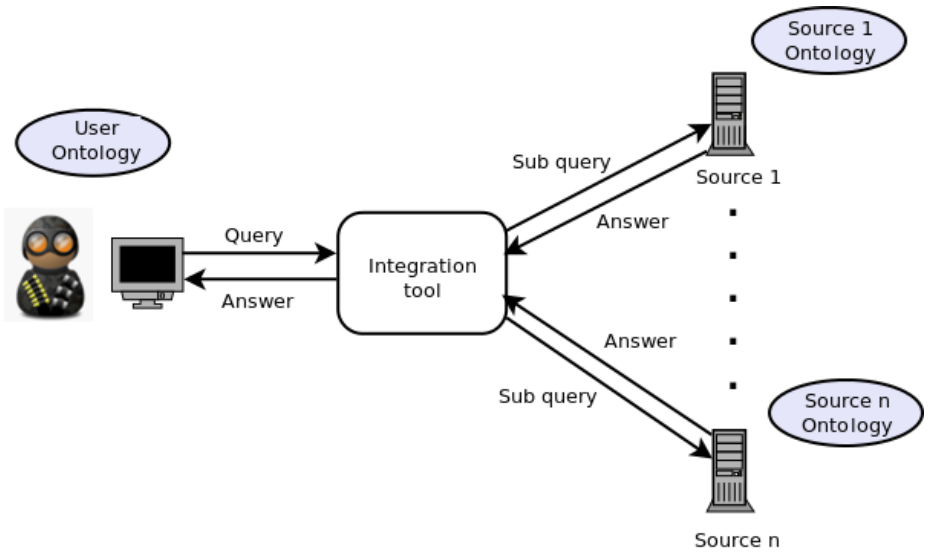

Figure 4.1 Information Integration Approach

The approach is illustrated in Figure 4.1 and in the following sections, three technical aspects needed to realize this idea are further described:

[2] http://www.w3.org/standards/semanticweb/

1. Query Rewriting (Section 5): How to rewrite a query (in this case the user's information request) according to ontologies defining the user's vocabulary and the different information source vocabularies.

2. Information Discovery (Section 6): How to discover information sources relevant to the information request.

3. Federated Query Processing (Section 7): How to aggregate the relevant information from the different sources found through information discovery.

5 Query Rewriting

Query Rewriting in the context of this report can roughly be described as the process of taking a SPARQL query and a set of aligned ontologies, and compiling the ontological reasoning into new queries (i.e. rewritten to source ontologies) that can be delegated and evaluated at the individual information sources. In this way, the answers to the original query will include implicit answers (i.e. terms in the query don't explicitly appear in the sources, but are logical consequences of ontology definitions that are defined over the source terms) as well as the explicit answers.

The query can thus be expressed at a conceptual level in a vocabulary independent of source-specific structure (e.g. without knowing source formats) nor the source location. This means that the query can be written without prior knowledge as to what sources exist at execution time. This approach is well suited for a distributed setting with highly dynamic sources and heterogeneous data, which characterizes the NNEC domain.

Query rewriting as a method of reasoning is based on the general approach of backward chaining, that is, working backwards from the query. The approach involves finding rules that imply the individual components of the query, and then trying to solve these as new goals. This process is repeated until the rules are satisfied by concrete data (facts) or no more rewrites are possible.

We now exemplify the process of query rewriting with the following information requirement: a human decision maker needs to find information as to what armored units can operate at night.

With respect to the example ontology fragment in Table 5.1 and instance data in Table 5.2, we rephrase the query to ask for *something Armored and NightCapable*. The query rewriting process will, based on the initial query and the ontology axioms as input, produce several new queries, where one interesting illustratory rewrite will be to ask for *APCs that have IRCameras*.

The other main method of reasoning apart from backward chaining (which query rewriting is an instance of) is that of forward chaining, which is data driven. Forward chaining works forward from known facts (data) and add new conclusions exhaustively, inferring all that can be inferred. Query evaluation thus boils down to merely matching the query pattern to the completed data. We have explored the potential use of forward changing in a previous experiment (Halvorsen & Hansen 2011).

The two approaches to reasoning, that of forward and backwards chaining, have differing strengths

User Ontology Concept	Definition
SensorPlatform	Vehicle that *has some* Sensor
NightCapable	SensorPlatform that *has some* IRCamera
IRCamera	A type of Sensor
APC	A type of Vehicle that is Armored

Table 5.1 Example ontology

Facts
apc-1 is a APC
apc-1 is a SensorPlatform
apc-1 has ir-cam-1
ir-cam-1 is an IRCamera

Table 5.2 Example data

and weaknesses that make them suitable for different application areas. Backwards chaining can be said to be more resilient to frequently changing data as inferences are made at query time. This contrasts with the forward chaining approach that involves pre-computing inferences before evaluating the query. Furthermore, another weakness with forward chaining in such a setting is that it involves the need for truth maintenance (i.e. invalidating facts and inferences).

There are also problematic issues with backwards chaining, especially in terms of execution time. Backwards chaining is more expensive at query-time than forward chaining as the latter approach will have performed the necessary inferences before query evaluation while the former will need to do this as part of the query evaluation step. In general, however, backwards chaining is the most appropriate approach to a query evaluation setting where data is distributed over several sources and/or is frequently changing. It is also worth to note that there is a significant amount of research into hybrid combinations of the two approaches which might be of practical use (Urbani et al. 2011).

Returning to the concrete problem of query rewriting, one can observe that most query rewriting approaches focus on keeping the algorithmic complexity low by controlling expressivity of the language. In practice, this means that queries will be evaluated within reasonable time but at the expense of what structures are possible to express in the language.[3] The CWIX trial presented in Section 8 was used to evaluate the suitability of certain fragments of the language OWL 2. See Pérez-Urbina et al. (2009*b*)) for addressing information integration in situational awareness setting. Furthermore, results from this experiment (cf. Sections 8 and 9) has led us to start looking at slightly more expressive languages.

[3] A more formal and theoretical description of query rewriting can be found in Pérez-Urbina et al. (2010)

6 Information Discovery

As stated earlier, the NNEC environment is highly dynamic with regards to the participating information systems. It has to be expected that unanticipated information sources with vital information can appear at any time. Thus we need to be able to facilitate on-the-fly, unanticipated information integration from heterogeneous sources with different formats/vocabularies.

More concretely, an information integration solution needs to be tolerant to frequent changes in network topology and that of changing information sources (including utilizing new, unknown types of information sources), supporting unanticipated uses. However, a basic assumption on the Semantic Web so far has been that information sources have near permanent presence (Tamma et al. 2005), an assumption that is not realistic in a NNEC environment.

Our solution to this is based on using

1. mDNS[4] for broadcasting and discovering the presence of information sources,
2. DNS-SD[5] for high-level description of the source in terms of pointers to query endpoint location and content description location, and
3. the VoID[6] vocabulary for actually describing the source content.

In addition to addressing the NNEC needs outlined above, the approach has the further strength that it is not dependent on a centralized registry, thus eliminating the issue of network fragmentation.

The approach is quite similar to the TIDE Transformation Baseline 3.0 specification of the Information Discovery protocol (ACT 2009, Annex C). Both approaches are based on the use of RDF, OWL ontologies and DNS-SD. The main difference is in choice of vocabulary/ontology for describing information sources. The TIDE approach uses a TIDE-specific ontology, while our approach is based on using VoID, a general-purpose ontology for describing data sources (specified in a W3C technical report) that has become somewhat of a de facto standard within the Semantic Web community.

7 Federated Query Processing

Data federation technology is software that provides a user with the ability to aggregate data from different sources without having to worry about the physical location of the data. Usually, as in the case of ontology-based data integration, a federation system also aims to provide declarative access to the data, thus abstracting away details associated with the representation of the data and/or its manner of storage. The idea is to enable the user to manipulate and analyze the data in terms of its conceptual content (or meaning) rather than in terms of, say, concrete data types and/or access protocols.

[4] http://www.multicastdns.org/
[5] http://www.dns-sd.org/
[6] http://www.w3.org/TR/void/

Approaches to federation can be sorted into two main paradigms. The *data warehousing* approach pulls data into a local repository and executes queries against the local copy (see e.g. Bishop et al. (2011)). *Distributed query answering*, on the other hand, executes sub-queries against each of a selection of remote sources, and builds the answer to the original query from the intermediate results returned by each of these (Schwarte et al. 2011, Quilitz & Leser 2008, Görlitz & Staab 2011). Both paradigms have their strengths and weaknesses.

A data warehouse is usually implemented in a conventional relational database system. It therefore benefits from the extremely well-studied body of query optimization techniques that are built into modern databases. On the negative side, the overhead associated with copying large amounts of data is usually computationally significant. The local copy tends to be relatively static, therefore, and, in particular, does not usually adapt to individual queries. In other words, it is an inherent assumption of the warehousing approach that the computational environment in which a query is executed does not have to be updated frequently.

This contrasts instructively with the distributed query processing paradigm, whose main strength is precisely that it is compatible with a query-dependent selection of sources with a rapid rate of change. In distributed query processing, the integration of data is only virtual insofar as the query posed to the system is broken up into sub-queries that are executed *directly* against one or more remote source. The sources need not be copied into local storage, so source selection at query-time becomes feasible and the problem of maintaining an updated repository simply does not arise.

Unfortunately, the adaptivity and dynamism of the distributed query processing paradigm comes at the expense of robustness and the likelihood of obtaining an answer. This is due to the fact that the partial results returned by the remote sources may depend on each other in intricate ways, because of the join attributes in the original query. That is, answering one sub-query may presuppose the bindings returned by another, whence the ordering and granularity of the sub-queries become important factors in the overall query answering process. Sometimes, even for rather small queries, these constraints may cause the number of HTTP requests to range in the tens of thousands (Schwarte et al. 2011) in effect exposing the system to network latency and denial of service.

A second and associated weakness of the distributed query processing paradigm is that the process of building an answer to a query from the partial results returned from the remote sources, interleaves federation and optimization in a way that makes it infeasible to calculate and/or preserve the logical properties of the original query through the process of rewriting and distributing it. That is, the answer to a query is typically defined in a *procedural* manner in terms of the join-order heuristics. Thus, even if we were to apply a reasoner to the query, we would have no guarantee that the process of executing the query on the remote sources would return a complete answer set, or indeed one that is correct, according to the ontology that the reasoner is inferring from.

Based on a requirement analysis for the CWIX-case, we set ourselves the following goal: To define a federation regime that

A) guarantees completeness of query answering in order to yield correct and complete result when coupled with a rewriter,

B) is compatible with a query-dependent selection of sources,

C) is suitable for a computational environment where the data changes rapidly, and

D) does not exceed an easily tractable number of HTTP requests to the remote sources.

We realized that this would land us somewhere in the middle of the data warehousing approach and the distributed query approach since A) and D) typically characterize the former, whereas B) and C) typically characterize the latter. Our solution is described next.

7.1 System description

Based on the desiderata mentioned above, we designed our federation engine to steer a course between the aforementioned paradigms that retains the dynamic and query-dependent nature of distributed querying, whilst, like warehousing, querying against a local copy of the data. Unlike the warehousing approach, however, our local copy is not persisted, but exists only in-memory for the duration of the query execution process. It is essentially a snapshot of that part of the remote sources which is relevant for answering the particular query at hand, and it is adapted to each individual query based on an analysis of the vocabulary in which that query is expressed.

This per-query approach to local storage preserves the dynamic and query-dependent nature of the distributed query processing paradigm. Sources can be selected based on the content of the query, and changes in the selected sources will be captured by the local mirror each time a new query is issued. But also, since the original query is eventually executed against the local mirror, and *not* against the remote sources directly, there is no need for join-order heuristics and pipe-lining of sub-queries. The in-memory representation of the remote sources is, one might say, a static snapshot of the relevant content residing in the remote sources. Because of this, it becomes possible to define the local representation declaratively and make it adhere to a predefined logical form that provably preserves the correctness and completeness of the reasoning procedure. This strategy of using a per-query in-memory representation of the relevant content of the remote sources, therefore achieves the right blend of adaptiveness and predictability, as measured by the aforementioned desiderata.

An overview of the resulting system, is shown in Figure 7.1: the system takes as input a SPARQL query Q, and a collection of aligned ontologies Σ, which are used by the query rewriter to produce the rewritten query Q_Σ. The rewritten query is then handed to the federator component which performs service discovery at run-time to identify live and relevant sources (cf. Section 6). Relevance here means signature overlap, where by a signature we understand a set of RDF properties. In other words, a source is deemed relevant to the query if it contains data that is described in a signature, according to the VoID description of the endpoint in question, that overlaps that of the query. Stated differently, relevant source are those that will be able to answer parts of the original query. Signature overlap is further used for routing the relevant parts of the original query to the matching remote

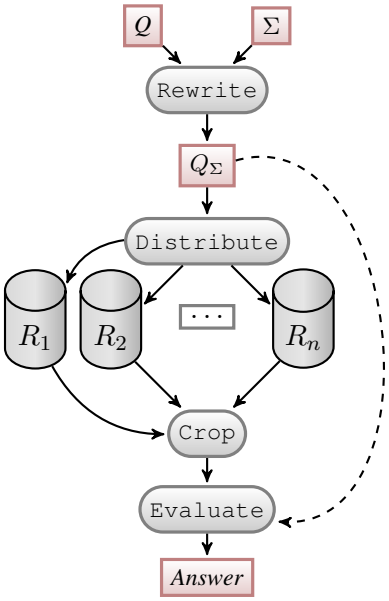

Figure 7.1 System overview

sources. Each of the routed sub-queries takes the form of SPARQL `CONSTRUCT` query. A SPARQL `CONSTRUCT` query is essentially a blueprint for building a new RDF graph from an old one, in this case for building a snapshot of relevant information based on the remote source to which this `CONSTRUCT` query is routed. The rewritten query Q_Σ is finally evaluated over the union of the RDF graphs returned by each of the sources, producing the answer to the original query Q.

In a bit more technical detail, we defined for each SPARQL query Q and an associated set of sources $\mathscr{R} = \bigcup_{i \in I} R_i$, where R_i is an RDF graph, the *cropping* $\mathscr{A}_Q^{\mathscr{R}}$ of $\mathscr{R}$. The cropping of $\mathscr{R}$ determined by Q is a new RDF graph that provably contains all the information from $\mathscr{R}$ of potential relevance for answering Q over Σ. It is built afresh for every Q_Σ by generating one SPARQL `CONSTRUCT` query for each of the sources R_i and taking the union of the results (modulo renaming of blank nodes). Since these `CONSTRUCT` queries are designed to adhere to a predefined logical form, they are sufficiently structured to allow us to calculate the properties of the resulting data set (the details are given in Appendix A). The cropping can therefore be used to answer the original query without the use of join-order heuristics. Not only does this enable us to guarantee the soundness and completeness of query answering, but it also keeps the number of HTTP request minimal. In fact, only one HTTP-request per source is needed. We conclude that this approach meets all our desiderata A) to D).

8 CWIX 2012 Experiment

NATO Coalition Warrior Interoperability eXploration, eXperimentation, eXamination, eXercise (CWIX) is an annual NATO event for NATO and partnering nations to test interoperability. CWIX 2012 took place at the NATO Joint Forces Training Center in Bydgoszsc, Poland, and gathered military personnel and technical experts from 20 nations plus NATO personnel.

8.1 Experiment Outline

In order to test our information integration approach in an environment closer to an operational environment than what is available at FFI, we decided to bring our solution to CWIX for experimentation.

The experiment was build upon the following scenario: A military analyst is monitoring planned medical evacuation flight missions, and is on the lookout for missions that might be threatened by enemy activity. Normally, this means she has to keep an eye on several systems, as information about evacuation flights and information about enemy activity is not kept in the same system. With the aid of a system as outlined so far in this report, however, the analyst can be given the opportunity to pose one query formulating what she is looking for and let an information integration system take care of the rest:

1. discover what information sources are available,
2. split the query in subqueries that can be sent to each relevant information system,
3. collect the relevant information from each of these systems, and
4. pose the original query against the collected information.

8.2 Experiment Setup

To create an environment as close as possible to the dynamic and multinational environment of NNEC, the experiment was conducted in cooperation with NATO C3 Agency (NC3A). They prepared two operational information systems for us to test against:

- JOCWatch - a web application that allows the Combined Joint Operations Center (CJOC) staff to manage, analyze and publish information on incidents of relevance to the command in an electronic event log
- MEDWatch - a web-based medical mission tracking tool to support the planning, logging and monitoring of medical evacuation missions

These two systems resided in a different partition of the CWIX network than the user application, as illustrated in Figure 8.1.

The information in MEDWatch and JOCWatch were made available through SPARQL endpoints by using D2R (Bizer & Cyganiak 2009). In addition, both sources were supplied with a service description according to the SPARQL 1.1 specification (W3C 2012), and each source made available its ontology at an URL described in the service description.

As this was a purely technical experiment, no GUI was created for the user application.

The experiment included three ontologies:

- a JOCWatch ontology,

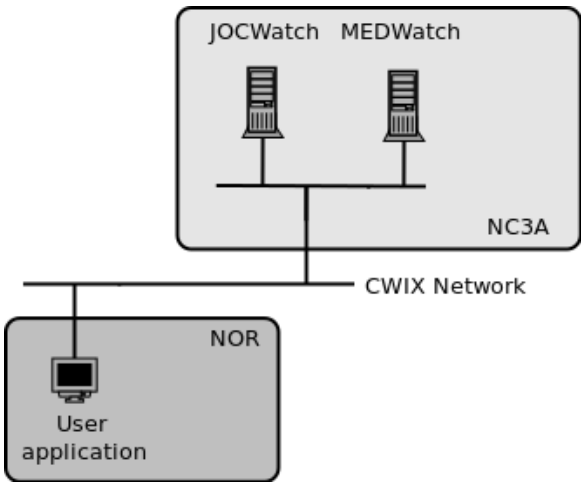

Figure 8.1 CWIX network

- a MEDWatch ontology, and
- an ontology containing the concepts used by the user of the system.

The relationship between these three ontologies is illustrated in Figure 8.2. The two information source ontologies were independent, while in the user ontology (`user`), the concepts needed by the user in order to query for evacuation missions threatened by enemy units were defined in terms of concepts and relations from both the information source ontologies. Thus, the user ontology can be considered a mapping ontology tying the JOCWatch and MEDWatch ontologies together. The definitions of the user ontology are described in Table 8.1.

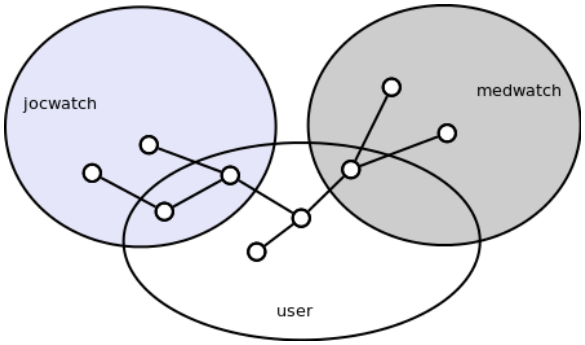

Figure 8.2 The ontologies in the experiment. The user ontology defines the concepts the user needs in terms of concepts and relations from the two other ontologies

8.3 Experiment Results

The main motivation behind this experiment, was testing whether it is feasible to provide a decision maker with the means to request information without prior knowledge about available information sources and their vocabularies. I.e. creating a looser coupling between the requesting information system and the information sources, something that should be highly relevant in the dynamic NNEC

User Ontology Concept	Definition
`ThreatenedMission`	All MEDWatch missions that are related to a `ThreateningIncident`
`ThreateningIncident`	All JOCWatch incidents that are both a MilitaryOperation (from the JOCWatch ontology) and a `HostileIncident`
`HostileIncident`	All incidents that has a `HostileInstigator`
`HostileInstigator`	All incident participants that are classified as being hostile.

Table 8.1 The definitions in the user ontology

environment. This proved to be possible with the approach described in Sections 5 - 7 and with the ontologies described in Section 8.2.

The user query was represented as a SPARQL query *Return all medical evacuation missions that can be classified as being threatened by enemy units (ThreatenedMission)*:

```
SELECT ?mission
WHERE
{
    ?mission a ThreatenedMission.
}
```

Here `ThreatenedMission` is a term specific to the user's vocabulary - posing this query to any of the information sources would not give any answers.

The query was decomposed and distributed as illustrated in Figure 8.3. The decomposition was guided by the user ontology, and the defined relations between the user ontology and the source ontologies (see Table 8.1).

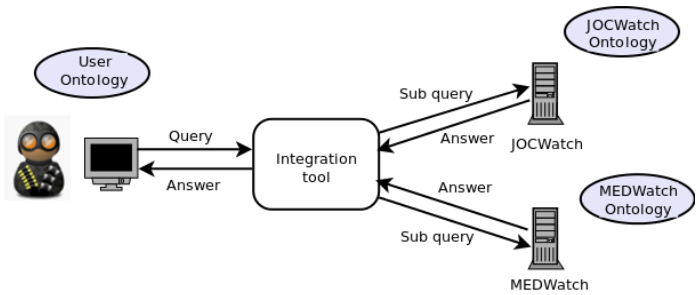

Figure 8.3 The user query is decomposed into source-specific sub queries

Based on these definitions, the original query was distributed as shown in Table 8.2.

We consider the CWIX test as being successful, as we were able to demonstrate that the user query, containing no information source specific vocabulary, resulted in a list of threatened missions.

Source	Description
JOCWatch	Incidents that are military operations AND has a hostile participant
MEDWatch	Missions with links to JOCWatch incidents

Table 8.2 Distribution of the user query

9 Experiences and Observations

The combination of once-per-query federation and rewriting, appears to be well-suited for data integration in a dynamic network topology such as the NNEC environment (cf. Section 2). Unlike the strategy of maintaining a data warehouse, our approach does not require any kind of content curation apart from the design of the ontology. This ontology acts as conceptual schema and query interface for the eligible data sources, yet the technique of compiling the ontology into the query makes the reasoning process completely independent of the federation step. Therefore, the ontology and the reasoning process are not essentially tied to any particular data source, but can be made to adapt to the environment at any given point in time. Stated differently, the rewriter simply unfolds the query (in accordance with the ontology) in total oblivion of the underlying sources, which in the NNEC environment may come and go. The availability of sources is checked regularly, whence the system is equipped to deliver optimal situational awareness as measured by availability of information in the network. There are trade-offs of course, and we record some pros and cons in the following.

9.1 Expressiveness of the Modeling Languages

We decided to model the user ontology in a restricted fragment of OWL 2, more specifically in one of the logics of the DL-Lite (Calvanese et al. 2007) or $\mathcal{EL}$ (Baader, Brandt & Lutz 2005) families. These families of languages form the basis for respectively the OWL 2 QL and OWL 2 EL language profiles, and have been carefully crafted to balance expressiveness against the complexity of reasoning (Baader, Brand & Lutz 2005, Calvanese et al. 2009). More specifically, these languages were selected because they are all *first-order rewritable*. First-order rewritability is a very desirable property that ensures that the reasoning process can largely be decoupled from data-access, meaning that the process of rewriting a query does not become harder as the amount of data increases (Calvanese et al. 2009). These languages are therefore particularly suitable for a backwards chaining approach to data integration in general, and for our case study in particular.

As expected, though, we quickly ran up against the expressivity limitations inherent in these languages. Some of them we were able to work around and some, with the means at hand at the time, we were not. We list three of the more interesting findings below:

The need for constants in the ontology. Initially we wanted to use the QUEST rewriter (Rodriguez-Muro & Calvanese 2012) for compiling the ontology into the query. QUEST is a reasoner that has been researched and developed by *Dipartimento di Informatica e Sistemistica SAPIENZA Università di Roma* which focuses on efficiency in the presence of very large volumes of data and very large ontologies. Its key service is SPARQL query answering under the OWL 2 QL entailment regime.

However, QUEST turned out not to be suitable for our purposes, because the QL fragment of OWL 2 does not allow constants—that is proper names or singular terms—to occur in the ontology. Constants occured with a certain frequency, though, in the military sources we were considering, usually in the form of code lists. For instance the JOCWatch database contains the codes `hostile` and `friendly`, which are used to classify the instigator of an event. Thus, in order to formalize the classes given in Table 8.1, say for instance the class `HostileInstigator`, we had to refer to these codes in the ontology.

We were able to overcome this problem by swapping QUEST with REQUIEM. Unlike quest, the REQUIEM rewriter (Pérez-Urbina et al. 2009*b*)—researched and developed by the *Oxford University Computing Laboratory*—can handle fragments of OWL 2 that go beyond the QL profile. In fact, the algorithm supports ontologies expressed in a Description Logic that captures most of the EL profile of OWL 2, and that allows constants in the ontology.

Note that the reason we were able to do this, is that we have designed our federation framework in such a way that it is not essentially tied to any particular reasoner. Rather the framework offers a modular plug-in architecture, that makes it easy to connect and test different reasoners for different purposes.

The lack of rules in the ontology language. Recall the scenario of our CWIX experiment: A military analyst is monitoring planned medical evacuation flight missions, and is on the lookout for missions that might be threatened by enemy activity. In such a situation, a critical factor for success could be to determine the presence of friendly units in the vicinity of the evacuation point. This is feasible, since each friendly unit can be expected to report its own position and type. Depending on the military capability of these units, it might be possible secure the mission. In order to support the decisions of the commanding officer, the integration system should therefore ideally provide the military analyst with a survey of the friendly units (within a certain radius from the evacuation point) that are equipped to handle the relevant types of enemy activity (given that the type of activity can be ascertained). From a modeling point of view, the most natural way to do this is to include an ontology statement for each type of friendly unit, saying which types of enemy activity it can cope with, for instance 'any `Artillery` unit `canHandle` any `SAFFIRE` event' (where `SAFFIRE` stands for 'surface-to-air-fire'). Unfortunately, due to the second occurrence of the universal quantifier 'any', this is not expressible in Description Logic without a rule-language extension. Note that this particular expressivity limitation is not a structure that applies only to OWL 2 QL and OWL 2 EL but to Description Logic in general. To make matters worse, none of the Description Logic rewriters currently on offer include rule-languages.

There are other semantic representation languages, though, that do not suffer from this limitation. One notable such is the so-called *sticky-join* fragment of $Datalog^{\pm}$ (Gottlob et al. 2011). This language is expressive enough to include rules, contains DL-Lite and $\mathcal{EL}$ as proper sub-languages, and manages to retain the aforementioned highly desirable property of first-order rewritability. We therefore regard it as a principal candidate for further research. At the time of writing we are in the process of implementing a rewriter for this language, which will eventually be plugged in to our

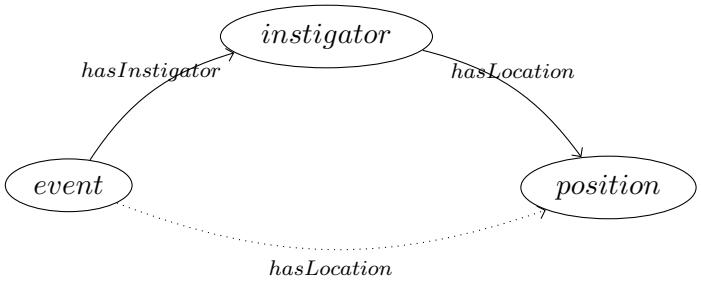

Figure 9.1 Expressing the property chain `hasInstigator AND hasLocation → hasPosition`

federation architecture instead of REQUIEM.

The lack of property chains. Another limitation of the DL fragment we originally wanted to restrict ourselves to, was that of defining new properties in terms of links of other properties (known in OWL as property chains). E.g. expressing that if a JOCWatch event has an instigator whose position is X, then the position of the event is also X. This is illustrated in Figure 9.1.

We originally wanted to test queries where we not only were searching for threatened missions, but also for own units in the vicinity that could handle the threat. We found that we were not able to model this without property chains, as the concept of positions were modeled differently in the different source ontologies.

9.2 Scalability

The scalability of our system may vary along two dimensions; the size of the in-memory local representation of the remote sources, and the number of clauses (that is, the number of `UNION` blocks each consisting of a conjunctive query pattern) in the rewritten query.

In the general case, if a predicate is used in a query in conjunction with an unbound subject and object, and the predicate in question is a piece of logical vocabulary such as `rdf:type` (or some other general-purpose and widely used property), then our approach may lead to downloading large parts of one or more of the remote sources. This is not only a problem on the receiver side, where it may be solved by streaming the results or by building the graph incrementally, but also on the side of the sources, who may become unresponsive if they are to regularly send large portions of their data sets in response to queries.

As regards the size of the rewritten query, some care and skill is required in designing the ontology in order to avoid a combinatorial explosion of clauses. Typical sources of complexity in this respect are excessive use of qualified existential restrictions (that is, ontology statements of the form 'the class of all objects that are so-and-so related to some object of type such-and-such') (Pérez-Urbina et al. 2009*a*), and role sub-typing (that is, ontology statements of the form 'if x is y's brother then x is y's relative'). Experience shows that ontologies in which these constructs are used with caution, produces significantly smaller rewritings.

We have found the following combination of rules of thumb to work well in practice:

1. Avoid logical vocabulary such as `rdf:type` or be explicit about the type, e. g. `?unit rdf:type jocwatch:Unit`,

2. practice rule 1 wrt. all general-purpose properties in widespread use,
3. use qualified existential restrictions sparingly,
4. keep the connectedness of qualified existential restrictions low,
5. avoid deep nesting levels wrt. role sub-typing.

9.3 Usability

Ontology-based data access is useful in all scenarios in which accessing data in a unified and coherent way is difficult. This may happen for several reasons. The data sources may have been developed for different purposes by different agencies or institutions, may not have a coherent design, and may not record similar types of information in the same manner. A well-designed ontology gives a unified view of the domain in terms of the *concepts* that are of interest to the user.

An interesting aspect of this, is that there is in general more than one valid view over a data source. That is, the same data may often be conceptualized and perspectivized in quite different ways depending on the user and his or her needs. Ontology-based data access provides, one might say, the plug-and-play capability to swap one such view for another. One may consider an ontology a lens that is superimposed onto the data to serve the needs of a particular community of users in a terminology that is familiar to them. The approach we are advocating reduces the task of adapting the same data to different communities of users to that of expressing the frequently used terms and the relationships between them in the form of an ontology.

10 Summary and Conclusion

An essential part of the NATO Network Enabled Capability (NNEC) vision is sharing of information in order to enhance shared situational awareness. This is anticipated to be an important contributor to building the decision superiority that in the end is expected to lead to increased mission effectiveness, when put to use by decision makers.

However, in order for the information to lead to improved situational awareness, it also needs to be integrated. Traditionally this integration is done manually by the decision makers, but the increasing amount of information available when conducting operations according to NNEC, means that there is a need for automated means to assist the decision makers in this integration.

In this report, an approach for automated information integration suited to NNEC is described. The main idea is to give a military decision maker the possibility to request information using a vocabulary she is comfortable with, and have an information system taking care of harvesting and integrating the information from the appropriate sources. In such a scenario, the user does not need to know anything about the available sources or how they represent their information.

The approach was demonstrated successfully at NATO CWIX 2012 in cooperation with NATO C3 Agency (NC3A). Moreover, the experiment indicated that extending expressivity for practical use warrants further study.

The successful demonstration of this approach at CWIX 2012 further strengthens our belief that information systems built using semantic technologies can offer the level of flexibility that is needed to support the essential information integration in the increasingly dynamic NNEC environment.

Appendix A Properties of the Cropping

For each SPARQL query Q and an associated set of sources $\mathscr{R} = \bigcup_{i \in I} R_i$, where R_i is an RDF graph, we define the *cropping* $\mathscr{A}_Q^{\mathscr{R}}$ of $\mathscr{R}$. The cropping of $\mathscr{R}$ determined by Q is a new RDF graph that provably contains all the information from $\mathscr{R}$ of potential relevance for answering Q over Σ. It is built afresh for every Q_Σ by generating one SPARQL `CONSTRUCT` query for each of the sources R_i and taking the union of the results (modulo renaming of blank nodes). Since these `CONSTRUCT` queries are designed to adhere to a predefined logical form, they are sufficiently structured to allow us to calculate the properties of the resulting data set.

Notation. Henceforth, $\langle P, \vec{x} \rangle$ denotes a SPARQL `SELECT` query with P a graph pattern and $\vec{x}$ a vector of projected variables. $\langle C, S \rangle$ denotes a `CONSTRUCT` query where C is the `CONSTRUCT` block and S the `WHERE` block. If Q is a query, then $Q(G)$ is the evaluation of Q over the RDF graph G. Familiarity with SPARQL syntax and semantics is henceforth assumed, cf. Arenas et al. (2009).

In order to display our approach in the simplest manner, let $\langle P, \vec{x} \rangle$ be a conjunctive SPARQL query where P is a *basic* graph pattern. Then we define the SPARQL `WHERE` pattern $S_P^{R_i}$ relative to source R_i and P as the SPARQL `UNION` of the following sequence of basic graph patterns:

$$S_P^{R_i} := \mathscr{E}, \mathscr{M}_1 \ldots \mathscr{M}_m, \mathscr{C}_1 \ldots \mathscr{C}_n$$

Here $\mathscr{E}$ is an *exclusive group*, a notion we take over from SPLENDID (Görlitz & Staab 2011) and FedX (Schwarte et al. 2011). That is, $\mathscr{E}$ consists of all the triples whose signature belongs exclusively to that of R_i. No restrictions apply to the grouping of these triples. The sequence $\mathscr{M}_1 \ldots \mathscr{M}_m$, on the other hand, is a set of *maximally unconstrained* triples collected from the patterns of P that do not have a concrete value in subject or object position. By 'unconstrained' we mean that no two triples in $\mathscr{M}_i$ share a variable. For every $\mathscr{M}_i$, the existence of an answer in R_i is guaranteed by the signature check that determines the query. However, the grouping needs to be controlled in order to reduce the number of union clauses whilst not constraining the results so as to exclude joins with other sources: a clause `{?x p ?y. ?x q ?z}` may exclude a triple `s p o` from R_i for which there is a join `o p' o'` in a different source R_j. Hence these dependencies need to be broken up. Finally $\mathscr{C}_1 \ldots \mathscr{C}_n$ is a sequence of singleton clauses, one for each concrete (in the aforementioned sense) triple pattern of P whose predicate is contained in the signature of R_i. Unlike the $\mathscr{M}_i$ the signature check alone does not guarantee the existence of answers to any of these patterns, whence each needs a separate `UNION` clause in order not to constrain the cropping. Now, for any $S_P^{R_i}$ we define a corresponding `CONSTRUCT` query as follows:

$$C_P^{R_i} = \langle f(\bigcup S_P^{R_i}), f(S_P^{R_i}) \rangle$$

Here, the function f is a *separation function* which essentially standardizes apart variable names that are shared between two or more clauses of $S_P^{R_i}$. This is necessary in order to avoid mixing valuations and concrete values in the `CONSTRUCT` block in ways that create an unsound cropping. Consider

the following example:

Example. Let G be the RDF graph $\mathtt{c}_1\,\mathtt{p}\,\mathtt{d}_1.\mathtt{c}_2\,\mathtt{q}\,\mathtt{d}_2.$ and let the query be

```
CONSTRUCT {?s q ?o. ?s p ?o.}
WHERE     {{?s p ?o} UNION {?s q ?o}}
```

Let μ_1: $?\mathtt{s} \mapsto \mathtt{c}_1, ?\mathtt{o} \mapsto \mathtt{d}_1$. Then $\mu(?\mathtt{s}\,\mathtt{q}\,?\mathtt{o}.) = \mathtt{c}_1\,\mathtt{q}\,\mathtt{d}_1.$ is in the resulting RDF graph but not in G, whence the query is unsound.

The cropping itself may now be defined simply as the union of the result of evaluating each of these `CONSTRUCT` queries over its associated source, i.e. as $\mathscr{A}_P^{\mathscr{R}} := \bigcup_{i\in I} C_P^{R_i}(R_i)$. We have the following crucial result:

Soundness and Completeness: $\langle P, \vec{x}\rangle(\mathscr{A}_P^{\mathscr{R}}) = \langle P, \vec{x}\rangle(\mathscr{R})$

The SPARQL semantics is compositional, so this result straightforwardly extends to unions of conjunctive queries.

Returning to the issue of potential network latency and efficiency, it is a trivial property of our approach that only one `CONSTRUCT` query is generated for every source, so federation is obviously minimal wrt. the number of HTTP request. For the same reason, there are no semi-joins, and no pipe-lined execution, whence the federation process is completely parallelizable. This in turn makes it possible to set an upper bound on the time required to construct the cropping, for since $S_P^{R_i}$ is on union normal form, Arenas et al. (2009, Corollary 1) yields:

Tractability Every P-selection $S_P^{R_i}$ can be evaluated in time $O(|P| \times |R_i|)$

It follows that the cropping itself can be built in polynomial time.

As always there is a price to be paid, this time in the currency of space. Especially when evaluating non-selective and ubiquitous triple patterns such as e.g. `?s rdf:type ?o` the cropping may become large. We are currently looking into streaming and/or incremental construction of the cropping as a way to leverage this limitation.

References

ACT (2009), 'TIDE Transformational Baseline 3.0'.

Arenas, M., Gutierrez, C. & Pérez, J. (2009), Foundations of rdf databases, *in* 'Reasoning Web. Semantic Technologies for Information Systems', Springer.

Baader, F., Brand, S. & Lutz, C. (2005), Pushing the el envelope, *in* 'In Proc. of IJCAI 2005', Morgan-Kaufmann Publishers, pp. 364–369.

Baader, F., Brandt, S. & Lutz, C. (2005), Pushing the EL Envelope, *in* 'Proceedings of the 19th Joint International Conference on Artificial Intelligence (IJCAI 2005)'.

Berners-Lee, T., Hendler, J. & Lassila, O. (2001), 'The Semantic Web', *Scientific American* **284**(5), 28 – 37.

Bishop, B., Kiryakov, A., Ognyanov, D., Peikov, I., Tashev, Z. & Velkov, R. (2011), 'Factforge: a fast track to the web of data', *Semant. web* **2**(2), 157–166.

Bizer, C. & Cyganiak, R. (2009), 'Publishing Relational Databases on the Semantic Web', `http://www4.wiwiss.fu-berlin.de/bizer/d2r-server/`.

Buckman, T. (2005), NATO Network Enabled Capability Feasibility Study Executive Summary. Version 2.0, Technical report, NATO Consultation, Command and Control Agency.

Calvanese, D., Giacomo, G. D., Lembo, D., Lenzerini, M., Poggi, A., Rodriguez-Muro, M. & Rosati, R. (2009), Ontologies and databases: The dl-lite approach, *in* 'Semantic Technologies for Informations Systems', Springer.

Calvanese, D., Giacomo, G. D., Lembo, D., Lenzerini, M. & Rosati, R. (2007), 'Tractable Reasoning and Efficient Query Answering in Description Logics: The DL-Lite Family', *Journal of Automated Reasoning* **39**, 385 – 429.

Görlitz, O. & Staab, S. (2011), Splendid: Sparql endpoint federation exploiting void descriptions, *in* 'Proc. COLD 2011'.

Gottlob, G., Orsi, G. & Pieris, A. (2011), 'Ontological queries: Rewriting and optimization (extended version)', *CoRR* **abs/1112.0343**.

Halvorsen, J. & Hansen, B. J. (2011), Integrating Military Systems using Semantic Web Technologies and Lightweight Agents, FFI-notat 2011/01851, Norwegian Defence Research Establishment (FFI).

Hansen, B. J., Gagnes, T., Rasmussen, R. E., Rustad, M. & Sletten, G. (2007), Semantic Technologies, FFI-rapport 2007/02461, Norwegian Defence Research Establishment (FFI).

Hansen, B. J., Halvorsen, J., Kristiansen, S. I. & Rasmussen, R. E. (2008), Experiment report: Semantic SOA - NATO CWID 2008, FFI-rapport 2008/01557, Norwegian Defence Research Establishment (FFI).

Hansen, B. J., Halvorsen, J., Kristiansen, S. I., Rasmussen, R., Rustad, M. & Sletten, G. (2010), Recommended application areas for semantic technologies, FFI-rapport 2010/00015, Norwegian Defence Research Establishment (FFI).

Pérez-Urbina, H., Horrocks, I. & Motik, B. (2009*a*), Efficient query answering for owl 2, *in* 'Proceedings of the 8th International Semantic Web Conference', ISWC '09, Springer-Verlag, Berlin, Heidelberg, pp. 489–504.

Pérez-Urbina, H., Horrocks, I. & Motik, B. (2009*b*), Practical aspects of query rewriting for owl 2, *in* R. Hoekstra & P. F. Patel-Schneider, eds, 'OWLED', Vol. 529 of *CEUR Workshop Proceedings*, CEUR-WS.org.

Pérez-Urbina, H., Motik, B. & Horrocks, I. (2010), 'Tractable query answering and rewriting under description logic constraints', *Journal of Applied Logic* **8**(2), 186–209.

Quilitz, B. & Leser, U. (2008), Querying distributed rdf data sources with sparql, *in* 'Proc. ESWC' 08'.

Rodriguez-Muro, M. & Calvanese, D. (2012), Quest, an owl 2 ql reasoner for ontology-based data access, *in* 'Proc. of the 9th Int. Workshop on OWL: Experiences and Directions (OWLED 2012)', Vol. 849 of *CEUR Electronic Workshop Proceedings, http://ceur-ws.org/*.

Schwarte, A., Haase, P., Hose, K., Schenkel, R. & Schmidt, M. (2011), Fedx: Optimization techniques for federated query processing on linked data, *in* 'Proc. ISWC' 11'.

Tamma, V., Blacoe, I., Lithgow-Smith, B. & Wooldridge, M. (2005), Introducing autonomic behaviour in semantic web agents, *in* 'Proceedings of the International Semantic Web Conference (ISWC) 2005', Springer, pp. 653–667.

Urbani, J., van Harmelen, F., Schlobach, S. & Bal, H. (2011), QueryPIE: Backward reasoning for OWL Horst over very large knowledge bases, *in* 'Research Papers Track of the ISWC2011, Bonn, Germany'.

W3C (2004), 'RDF Primer', http://www.w3.org/TR/rdf-primer/.

W3C (2009), 'OWL 2 Web Ontology Language Primer', http://www.w3.org/TR/2009/REC-owl2-primer-20091027/.

W3C (2012), 'SPARQL 1.1 Query Language', http://www.w3.org/TR/sparql11-query/. Working Draft.