FFI-notat 2011/01851

Integrating Military Systems using Semantic Web Technologies and Lightweight Agents

Jonas Halvorsen and Bjørn Jervell Hansen

Norwegian Defence Research Establishment (FFI)

15 October 2011

FFI-notat 2011/01851

1176

Keywords

Nettverksbasert forsvar

Semantisk web

Ontologi (databehandling)

Informasjonsintegrasjon

Approved by

Rolf Rasmussen	Project manager
Anders Eggen	Director

English summary

The primary objective when conducting operations according to NATO Network Enabled Capability (NNEC) is to attain information superiority in order to gain increased mission effectiveness. NNEC is based on an idea of a common information space where the information is visible, available, and understandable, and through which the participating information systems supply information for others to utilize and retrieve when needed according to their role. In order to realize this idea, the challenge of integrating information from heterogeneous sources in a highly dynamic environment such as NNEC needs to be addressed.

In order to facilitate the necessary information integration in a NNEC setting, we propose a system of lightweight cooperative hybrid agents that rely on using the Semantic Web technology stack as far as possible and are built on top of an efficient peer-to-peer communication layer.

The contributions of the work presented in this report are:

- A holistic approach for information integration in a highly dynamic setting.
- A use case and demonstration of how lightweight hybrid agents utilizing Semantic Web technologies can facilitate on-the-fly, unanticipated information integration from heterogeneous sources with different formats/vocabularies.

The approach has been preliminarily tested in the context of the 2011 SOA pilot conducted by FFI.

Sammendrag

Når man gjennomfører militæroperasjoner i henhold til prinsippene i NATO Network Enabled Capability (NNEC), er målet å oppnå informasjonsoverlegenhet for å kunne gjennomføre mer effektive operasjoner. NNEC baserer seg på ideen om et informasjonsrom der informasjon er synlig, tilgjengelig og forståelig for deltakerne, og der deltakerne kan utveksle informasjon. For å virkeliggjøre denne ideen må deltakerne være istand til å integrere den tilgjengelige informasjonen. Dette er en stor utfordring, spesielt fordi informasjonskildene er heterogene og NNEC-omgivelsene er meget dynamiske.

I dette notatet foreslår vi å legge til rette for den nødvendige informasjonsintegrasjonen gjennom et system som består av lettvekts, hybride agenter basert på semantisk web-teknologier. Agentene samarbeider gjennom et effektivt kommunikasjonslag som utnytter peer-to-peer-teknologi.

Gjennom dette arbeidet søker vi å bidrar til å løse informasjonintegrasjonsproblemet gjennom å:

- foreslå en helhetlig tilnærming for informasjonsintegrasjon egnet for dynamiske situasjoner og
- demonstrere hvordan lettvekts, hybride agenter bygget på semantisk web-teknologier can legge til rette for integrasjon av informasjon fra uforutsette, oppdukkende kilder som tilbyr informasjon med forskjellige formater.

Vi har gjennomført forberedende tester av tilnærmingen som en del av SOA-piloten som ble gjennomført av FFI i 2011.

Contents

1 Introduction

The primary objective when conducting operations according to NATO Network Enabled Capability (NNEC), and also Network Based Defence (NBD)[1], is to attain information superiority in order to gain increased mission effectiveness. NNEC is based on an idea of a common information space where the information is visible, available, and understandable, and through which the participating information systems supply information for others to utilize and retrieve when needed according to their role (Buckman 2005). In order to realize this idea, the challenge of integrating information from heterogeneous sources in a highly dynamic environment such as NNEC needs to be addressed (see Section 2).

In order to facilitate the necessary information integration in a NNEC setting with its unique environmental requirements (see Section 2), we propose a system of lightweight cooperative hybrid agents that rely on using the Semantic Web technology stack as far as possible (see Sections 3 and 4). The multi-agent paradigm coupled with Semantic Web technologies (see Section 3.1) and an efficient peer-to-peer communication layer (see Section 4.1), provides the ability for software agents to cooperate in terms of obtaining relevant information for the task at hand, draw automatic conclusions, and disseminate the results to peers. The approach has been preliminarily tested in the context of the 2011 SOA pilot conducted by FFI (Rasmussen & Hansen 2011), and this report documents this experiment. The use case for the approach within the SOA pilot is described in Section 5

The contributions of this work are:

- A holistic approach for information integration in a highly dynamic setting.
- A use case and demonstration of how lightweight hybrid agents utilizing Semantic Web technologies can facilitate on-the-fly, unanticipated information integration from heterogeneous sources with different formats/vocabularies.

The report is concluded with a brief review of related work in Section 6 and a conclusion in Section 7.

The ideas described in this report have also been published in a NATO RTO symposium paper (Halvorsen & Hansen 2011).

2 The Information Integration Problem 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 (Buckman 2005).

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

In order to fulfill this vision, the information from the various information sources in the environment needs to be 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, and the NNEC environment is no exception to this general rule.

The NNEC environment is a complex environment with a wide variety of information systems whose information can be important even to unanticipated users. 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 unanticipated information sources with vital information can appear at any time.

Thus, information integration is particularly challenging in the NNEC environment.

3 Integration using Semantic Web Technologies and Lightweight Agents

We propose the use of Semantic Web technologies and lightweight, hybrid agents in order to address the problem of integrating heterogeneous systems in a NNEC setting.

Semantic Web technologies have been shown to be useful for information integration (Demir et al. 2010, Kobilarov et al. 2009, IO Informatics 2010). Agent-based systems, on the other hand, are able to operate autonomously in highly dynamic settings, making it possible to address complex, unforeseen and rapidly changing tasks. Together they facilitate dynamic problem solving by providing agents with the ability to collaborate also across the traditional military domains. The resulting system is tolerant to frequent changes in network topology and that of changing information sources (including utilizing new, unknown types of information sources), supporting unanticipated uses.

Our approach adheres to the W3C's[2] Semantic Web technology stack standards as far as possible. However, where there are shortcomings (such as no standardized solutions, or issues not yet fully addressed), we have taken the liberty of using solutions that we see best fit for the task.

3.1 Semantic Web Technologies

As noted earlier, Semantic Web technologies have shown to be well suited for information integration tasks, utilizing the generic knowledge representation of RDF[3] (Carroll & Klyne 2004) coupled with interlinking of differing terminology through the (re-)use of well-defined OWL-DL[4] ontologies. However, a basic assumption on the Semantic Web has been that information sources

[2]World Wide Web Consortium
[3]Resource Description Framework
[4]Description Logic subset of the Web Ontology Language

have near permanent presence (Tamma et al. 2005), an assumption that is not realistic in a NNEC environment. Furthermore, real-time data has until quite recently not been an important focus area. In a military setting, however, temporal data and streams of data are of high importance. Recent W3C focus on Semantic Sensor Networks and active research into stream reasoning has resulted in increased focus in this area. It is still, however, in an early stage and no standards or best-practice solutions exist yet.

3.2 Hybrid Agents

As previously stated, agent systems are appropriate for complex problems that can only be solved through the (automatic) cooperation and collaboration of several (loosely coupled) software components with differing specialty. Traditional multi-agent systems architectures can roughly be divided into two categories; deliberative and reactive (Nwana 1996), each with different strengths and weaknesses.

Deliberative agents are agents that commit to means-end-reasoning. Means-end reasoning can be described as a two step process: (1) deliberating between options of what to do and deciding on one to be set as the current goal, and (2) deciding how to achieve it (Walton 2006, p. 83). Individually, deliberative agents typically show a level of intelligence, and are able to decide for themselves how to bring about their goals (deliberative agents take the initiative to bring about changes).

Reactive agents, on the other hand, merely react on sensory input rather than taking the initiative themselves. They (typically) do not perform any reasoning regarding the environment. Furthermore, they are usually small and simple applications that perform a single tasks. As a result, individually they show little sign of intelligence. The main assumption for reactive agents is that intelligent behaviour emerges from the dynamic interaction between many small, specialized agents rather than the individual agents being intelligent (Wooldridge 2009, p. 85).

There are, however, issues with both these agent architectures. Deliberative agents can end up spending too much time on deliberation instead of acting due to the complexity of the deliberation task (planning, theorem proving, etc.), thus reducing the appropriateness in real-time settings (Wooldridge 2009, p. 85). Reactive agents, on the other hand, can be difficult to design in terms of actually providing emergent intelligence, especially when the number of different agents grow. They also have a weakness in that the reactive agents do not reason over the environment it works in, and as a result each agent will take short-term decisions (Wooldridge 2009, p. 92).

The current trend is thus to combine both reactive and deliberative agents in a pragmatic approach, that of hybrid agent systems (Walton 2006, pp. 108-109). In this architecture, agents have a varying degree of deliberative and reactive behaviour. Some are purely reactive, others are purely deliberative, but most are somewhere in between. This allows for the use of means-end-reasoning for certain agents, where intelligence can be useful (e.g. where there are several choices as to how to accomplish a task and there are costs associated with the different, possibly limited, resources. At the same time, other agents need to act fast to sensory input and do not need deliberation

(intelligence can emerge from the network of simple sensory agents feeding each other). Thus one can mix deliberation and reactive behaviour in a pragmatic approach that can be used in a time-critical setting.

3.3 Outlined Approach

In our approach, an agent (representing a user or a dedicated task) first provides its information need formulated as a SPARQL query (Prud'hommeaux & Seaborne 2008) to the multi-agent system. Information needs can, for example, be requiring information about a certain named individual or information about all units of a certain category within a specified geographic area. The agent then broadcasts this requirement to the other agents in the network. Receiving agents then decide if they are *a*) capable of answering the information request, and *b*) willing to answer the request. If it is not in the interest of the receiving agent to answer the request, may it be due to other higher priority work, being a trust issue, or agents having incompatible goals, then it may refrain from answering. This approach does thus not require a common register and, as a result, sensitive sources can be more obfuscated.

Information requests in our system can be either a one-off query or a request for a stream (continuous flow of answers). Furthermore, more than one agent can answer a query, as the agents can have complimentary information.

We claim that our approach can be described as *lightweight* in terms of it consisting of a simple overlay protocol with only two types of messages (query and response). Advanced functionality is built merely out of use of these two message types. Furthermore, there are very few compulsory functionalities that the agent has to support as the approach is opportunistic. A third aspect of our approach that can be said to contribute to it being lightweight is that the architecture does not rely on centralized components.

Our contribution can be summarized as follows:

1. A holistic approach for information integration in a highly dynamic setting. This includes an ontology and method for querying for streams and one-off queries in a situation where the source endpoints are not known.
2. A use case and demonstration of how the use of lightweight agents utilizing ontologies and other Semantic Web technologies can be used to perform on-the-fly, unanticipated information integration from sources with different formats/vocabularies. We exemplify this with a use case where the above approach is used for situational awareness and threat detection.

4 Details

We will now outline the details of our approach.

4.1 Inter-Agent Communication

Our agent-based approach assumes a topic-centered peer-to-peer network architecture for inter-agent communication. In our implementation we have utilized a java-based implementation of the Mist protocol (Skjegstad et al. 2010), yet our solution is not prescriptive to the choice of peer-to-peer solution other than that it has to provide the notion of topics.

4.2 Agent Requirements

An agent must have a single, unique identifier which it uses to identify itself in message exchanges. It is preferable that the identifier is a URI, but it is not compulsory.

An agent must also have a message topic that works as its private "mailbox" that other agents can post to. This message topic must be unique for each agent. For simplicity, it is recommended that the message topic is set to the same as the agent identifier. However, this is not compulsory, and participating agents should not assume this. An agent is, however, allowed to listen to more than one topic. Furthermore several agents can listen to and post to shared topics. The only topic that is not allowed to be shared is the "mailbox".

Our approach assumes a single common built-in topic called the `Query` topic. This topic is used by agents to pose information requests to the agent environment. All agents should subscribe to this topic if they wish to be able to cooperate and collaborate with other agents. As the approach does not rely on a registry of agents and what they provide, this topic acts as the main method of discovering and retrieving information. It can also be used to discover agents that exist in the environment by posing an information request asking agents to describe themselves.

Agents can choose to be selective in terms of what queries it decides to answer and from whom it accepts queries. This includes not replying to queries from other agents asking for agents to describe themselves. An agent can choose to either not listen to the `Query` topic, or be strict as to the source of the query whether it reacts to it or not. The agent can also be passively watching the `Query` topic and associated replies in order to build up an internal map of agents and what information they can provide. This map can then be used to contact agents directly for queries if more discretion is required (e.g. they do not want to show information request to the wider agent network) or they do not want to be discovered.

4.3 Information Representation & Serialization

All communication between the agents is in RDF. More specifically, N-Quads (Cyganiak et al. 2009) is used as the concrete serialization format in order to allow the use of Named Graphs (Carroll et al. 2005). The reason for mandating support for Named Graphs is that it provides the notion of contexts, which makes it possible to assert statements about the RDF graph itself.

Examples of useful assertions about a graph are creation timestamps, time restrictions on validity of the data in the graph, as well as attribution as to who created/asserted the graph (for trust and provenance). These features are necessary in a setting with real-time, dynamic data and situations where trust plays a vital role.

The actual information requests are described as SPARQL queries. We have limited our approach to queries of the forms DESCRIBE and CONSTRUCT in order to be compliant with our assumption that all data transferred in messages between agents is in RDF. Allowing for SPARQL ASK and SELECT queries would break this assumption as per the standard they return answers in the SPARQL Query Results XML Format (Beckett & Broekstra 2008), not as RDF.

4.4 Agent Communication Protocol

As implied above, the agent messages themselves are represented as RDF graphs[5]. A message envelope is represented as a URI resource, and typed either as `http://sem.ffi/ont/agent#Query` for initializing a query, or `http://sem.ffi/ont/agent#Inform` for a query reply.

For a query-message, the following assertions are mandatory[6]:

conversation-id	A conversation-id, which is unique for the conversation and will be used in a reply to relate answers to the specific information request.
sender	The ID of the sender. This will be the agents unique identifier.
reply-topic	The topic that a reply is to be posted to. Usually the agent's "mailbox", but could also be a shared topic.
query	The SPARQL-query itself, serialized as a literal value.

One minor detail one should note is that the SPARQL query itself is serialized as a literal value in the RDF graph that constitutes the query message. Ideally, the SPARQL query would also be serialized as RDF but at the current time there is no standard for this. There exists a couple of candidate vocabularies for doing this, where SPIN[7] seems to be the most mature solution.

For a reply-message, the following assertions are mandatory:

conversation-id	The ID of the conversation that the message belongs to (same as for the query-message).
sender	The ID of the sender (same as for the query-message).
content	Link to a named graph containing the answer.

[5] An RDF graph is a set of RDF triples.

[6] In the namespace `http://sem.ffi/ont/agent`

[7] `http://www.w3.org/Submission/spin-sparql/`

Listing 1 Regular query

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>

DESCRIBE ?subj
WHERE
{
    ?subj foaf:knows [].
}
```

Listing 2 Stream query

```
PREFIX stream: <http://sem.ffi/ont/stream#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>

DESCRIBE ?agent-id ?stream-topic
WHERE
{
  ?stream a stream:StreamingGraph;
          stream:broadcastTopic ?stream-topic;
          stream:agentId ?agent-id.
  GRAPH ?stream
  {
    [] foaf:knows [].
  }
}
```

4.5 Streams

As noted earlier, our approach supports both one-off queries and continuous, streaming queries.

A one-off query is represented as a standard SPARQL query as represented in Listing 1. In the example shown, we want to find information (descriptions) about subjects that know someone (`foaf:knows`).

A stream-query differs somewhat from the regular query in terms of it relying on using a domain ontology (vocabulary) for describing streams. Listing 2 represents an example of a stream query. Note however that it is still a valid SPARQL query (the SPARQL language has not been extended).

In the example shown in Listing 2, we wish to find streams that broadcast information about subjects that know someone. Note the difference from Listing 1 in that here we ask not for the resources that know someone, but rather we ask about the IDs and stream-topics of the sources that broadcast that type of information. In order to get the information about the subjects that know someone, one then has to subscribe to the stream-topic.

The answer to the stream-topic will be a regular “Inform” message (see Subsection 4.4). The content of the message will be an RDF graph describing the agent and the stream-topic. The agent

issuing the query can then use this information to subscribe to this topic and receive continuous information.

4.6 Translation

When querying for information, there will often be situations where another source has the data that one seeks, yet it is expressed in another ontology than the query was formulated in. Thus there is a need for evaluating queries in terms of utilizing ontology alignments. In our approach we take an opportunistic stance to this problem:

1. Clients translate the original query into a query that encompasses equivalent terms from other ontologies through the use of alignments. The resulting query will involve use of the SPARQL UNION operator.
2. Receivers try to translate the received query into a query that it can understand, using available alignments. If no translation is possible, no answer will be generated.

Our solution is opportunistic in the way that no one solution is mandated. The software developers that create the agents are free to choose if the agent is to have translation capabilities. Furthermore, if both the requester and the responder translate there should not be a problem as long as the ontologies with alignments are not inconsistent.

4.7 Ontology Alignment Distribution

The agent will typically keep a set of ontologies necessary to describe the information it can provide, and each agent will formulate its queries according to the ontologies it holds. For other agents to be able to respond to the query, they either need to have the same classes and properties in their ontology collection or there has to be links between the ontologies of the querying agent and the replying agent. These links are called *alignments* (Euzenat & Shvaiko 2007, Definition 2.10, p. 47).

In order to retrieve relevant alignments, each agent first creates a *signature* containing the class and property URIs that are used in assertions in its local knowledge base. SPARQL queries are then generated based on this signature, asking for definitions (e.g. subclass and subproperty statements) of these resources. These are sent out as regular queries in the network, and listening agents can return an ontology module of relevant concept definitions and alignments from its combined ontologies. The concrete method used for calculating relevant alignments is up to the agents. This approach is based on what is outlined in Doran et al. (2008). Note that an agent is not forced to answer or have this functionality, as the approach is opportunistic.

The motivation behind this process is that we assume the ontologies used by the different agents to reuse other ontologies. As a result, concepts in the ontology collections of two different agents can be linked through a common concept in another ontology. Using the process described here, these links will be exchanged so that they will be kept by both agents internally. Furthermore,

one can envision that in a normal military setting, where all the agents are able to communicate with each other, there are agents in the network capable of providing new alignments. These alignments will typically come from manual or semiautomatic processes. In a setting like this, new alignments added through these agents will eventually be received by the relevant agents and then be assimilated. If the network then becomes partitioned (e.g. a squad leaves HQ for a patrol mission), the agent network can still utilize the alignments that have been distributed to communicate with others, even though the alignment-providing agent is no longer reachable.

5 Case Study

In order to perform a preliminary assessment concerning whether the approach described is viable, a case study focusing on threat detection was performed in connection with the SOA pilot described in Rasmussen & Hansen (2011). We will now describe the case study as well as a sketch of an alternate case.

5.1 Threat Detection Case

In the SOA pilot scenario, a system incorporating the ideas presented in this report was said to reside at a tactical HQ (TU A HQ) in a coalition mission. In this particular case, the information need of the decision maker at the HQ was to retrieve information on all potential threats to friendly units.

The involved systems were:

- NORCCIS II, a Norwegian command and control information system providing track information.
- JocWatch, a NATO incident system providing incident information.
- DBPedia[8] providing background information.
- A wiki acting as a fictitious intelligence wiki also providing background information.

All sources were exposed to the agent environment by having wrapping agents working on their behalf. The agents of the two military systems provided real-time streams of data from the relational databases of the systems using D2R (see Bizer & Cyganiak (2009) and also Hansen et al. (2010, Section 6.2.3)), while the others provided query/reply-functionality.

In the use case, which is illustrated in Figure 5.1, the decision maker has an agent dedicated to finding threats to friendly units (*Threat Detection Agent*). The agent could find these threats using automated reasoning by querying the system for all known military units, and applying rules that formalize how one unit can be a threat to another.

[8] `http://dbpedia.org/`

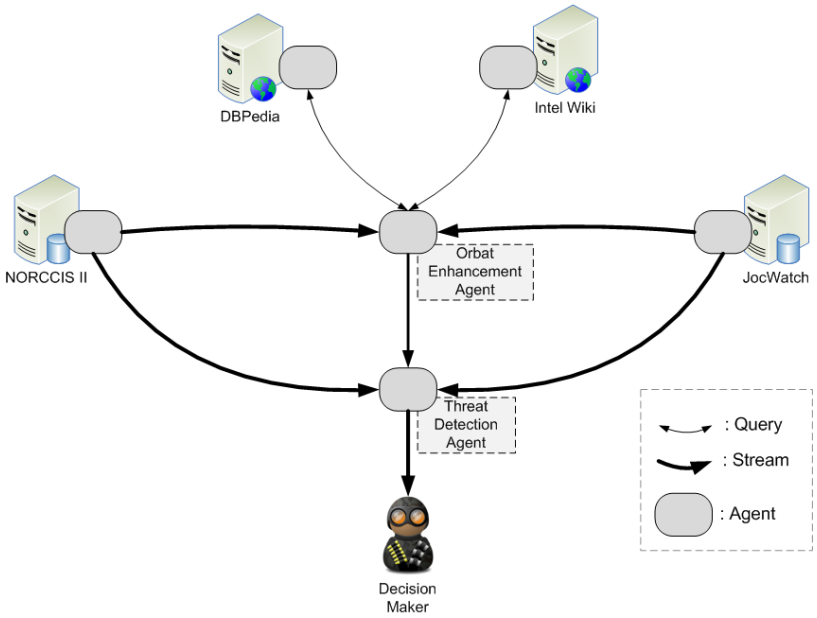

Figure 5.1 The agents comprising the system in the threat detection case

The main actor in the system is the *Orbat Enhancement Agent* whose task is to monitor the available information, look for information about military units, and try to fetch more information about them from other sources. The units in focus in this case were the friendly unit *GTA1* (reported from NORCCIS II) and an enemy unit *16BLA* (reported in an incident from JocWatch). The Orbat Enhancement Agent then fetched more information regarding *16BLA*, in particular what organization it belonged to (from the intelligence wiki), what weapons units from that organization typically were equipped with (from the intelligence wiki), and the known capability and range of those weapons (from DBPedia).

Collecting the information from NORCCIS II, JocWatch, and the Orbat Enhancement Agent, the Threat Detection Agent was able to conclude that the available information indicated that the enemy unit *16BLA* represented a possible threat to friendly unit *GTA1*, which was out on a reconnaissance mission. The agent also provided the information indicating this possible threat, which was presented to the decision maker with an explanation to this conclusion (also shown in Figure 5.2):

1. GTA1 is a Vehicle
2. 16BLA has Anti-Vehicle Capability (due to having equipment RPG-7)
3. Anti-Vehicle Capabilities can represent a threat to Vehicles
4. GTA1 can be within shooting range of 16BLA (because the distance between them is less than the typical range of a RPG-7)

In this particular case, the warning and the explanation convinced the decision maker to call for reinforcements for the friendly unit.

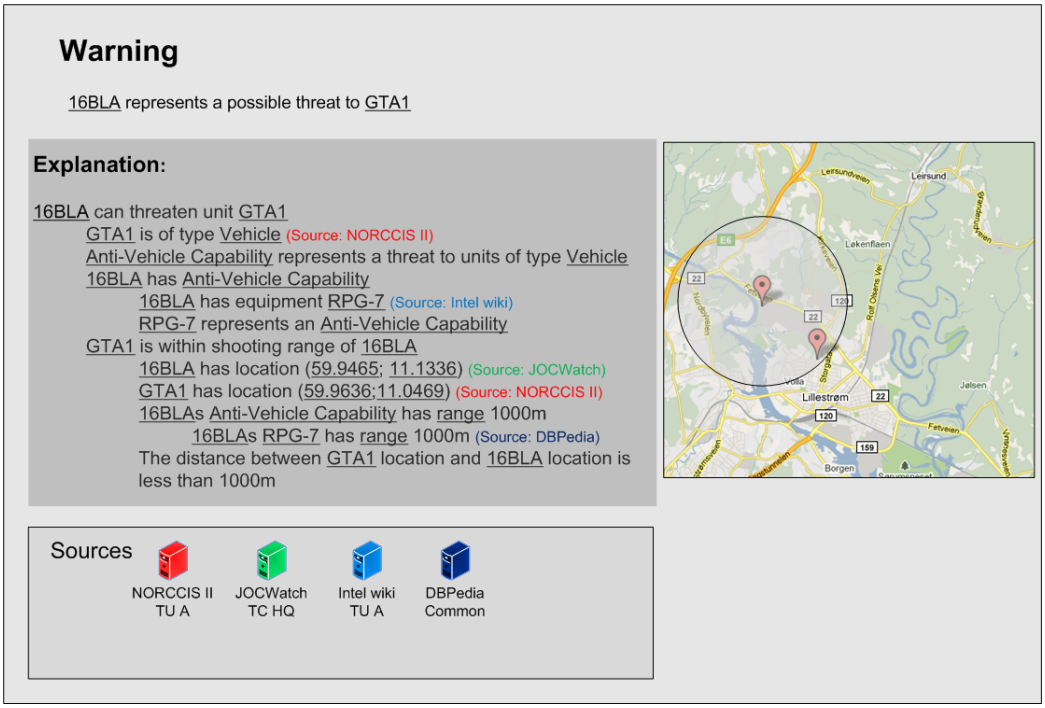

Figure 5.2 The explanation shown to the decision maker in the threat detection case

5.2 Alternate Case

In order to illustrate that the approach outlined in this paper can be used in a variety of cases, we will here shortly outline a possible new case that the approach can solve by simply changing the agents and ontologies compared to the case described above.

In this case the task to be supported by the agent-based system is generating alerts for plans that, due to circumstances indicated by information available in the network, can not be applied: A commander has made a plan for a route a convoy is to follow through a particular area. The commander will in this case consult the system by providing it with the plan via a user agent.

The system will have to be provided with rules describing when a plan cannot be applied. Reasons can e.g. be roads not designed to carry the vehicles included in the plan, damaged infrastructure on the route (e.g. a damaged bridge), observed enemy activity in the area, etc.

The agents needed in this case can be:

- A user agent,
- a reasoning agent with access to the plan applicability rules,
- an agent offering information from sources regarding infrastructure quality (e.g. roads. Can be an non-military information source),
- an agent offering information from sources regarding enemy activity (C2 systems, incident systems. Typically military systems), and

- an agent offering information on the status on the infrastructure (e.g. damaged bridges. Typically military systems).

The reasoning agent can then perform automated reasoning based on the plan applicability rules, and issue a warning via the user agent if it can find information that makes it impossible to apply the plan.

6 Related Work

There exists plenty of research on each of the two topics multi-agent systems and semantic technologies in connection with the information integration challenge, see e.g. Faulkner et al. (2004), Panti et al. (2002), and Rahimi & Carver (2005) regarding multi-agent systems, and De Bruijn (2004), Stoutenburg et al. (2007), and Noy (2004) regarding semantic technologies. But to our knowledge, not many concrete solutions exist that combine the two areas in order to address this challenge (in particular when taking NNEC requirements into consideration).

In his work on RDFAgents (Shinavier 2011), Shinavier presents an idea similar to ours. RDF-Agents is a messaging protocol for real-time, peer-to-peer knowledge sharing using Semantic Web technologies, geared toward lightweight devices with variable network connectivity. The main difference compared to our approach is that while we use a common `Query` topic for agents to discover information requests, how to bootstrap the agent discovery, i.e. how to discover the first agent, is not specified in RDFAgents. Further, we handle streams by letting the streaming agents publish the information on a specific topic for all interested agents to listen to, while in RDFAgents each single stream element is sent directly to the receiving agent.

In García-Sánchez et al. (2008) the combination of Semantic Web technologies and agents is explored for information integration. Their approach, however, is based on the information sources being available as Semantic Web Services and the agents being an extra layer concerned with automatic handling of these services. In contrast, the agents in our approach are agnostic to how the information is fetched from the underlying systems, and thus is not dependent on the information sources being exposed as Semantic Web Services.

Semantic Routing System (SERSE) (Tamma et al. 2005) is a distributed multi-agent query-handling system built on peer-to-peer technology and ontologies. It handles queries by routing them from an agent to the next according to what concepts the different agents know. This differs from our approach, as we rather put the queries out to all agents via the `Query` topic and let the agents answer the query if they can. Furthermore, the SERSE approach relies on all agents knowing what information their neighboring agents can provide. It is in other words not possible for an agent to hide what kind of information it holds, something that can be desirable in a military scenario.

7 Conclusions and Further Work

In order to attain the coveted information superiority in NATO Network Enabled Capability, the challenge of integrating information from different sources in this highly dynamic environment needs to be solved.

In this report, we have proposed to address this challenge by using a system of lightweight cooperative hybrid agents that rely on Semantic Web technologies and an efficient peer-to-peer communication layer. We have also described a test of the approach on a use case involving military information systems.

The contributions of the work described in this report are:

- A holistic approach for information integration in a highly dynamic setting.
- A use case and demonstration of how lightweight hybrid agents utilizing Semantic Web technologies can facilitate on-the-fly, unanticipated information integration from heterogeneous sources with different formats/vocabularies.

The proposed approach is still at an early stage requiring more work to be done in order to assess its viability. Although the approach has been using military information system, the case was limited with regards to the number of agents. Thus it is necessary to test the scalability of the approach on a larger case. Further, we wish to perform experiments to understand what capabilities are needed in the agents to handle real-life data that often will contain errors and contradictions. We also wish to explore what kind of ontology matching is needed in order for the agents to answer queries formulated according to ontologies not known to the answering agent in advance. This exploration will be conducted according to the ideas presented in Hansen (2011).

Based on the testing done so far, we find the presented approach to be promising. We also feel that the requirements from NNEC with regards to information integration means that further work in this direction should be pursued.

References

Beckett, D. & Broekstra, J. (2008), SPARQL Query Results XML Format, W3C Recommendation, W3C.

`http://www.w3.org/TR/rdf-sparql-XMLres/`

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.

Carroll, J. J., Bizer, C., Hayes, P. & Stickler, P. (2005), Named graphs, provenance and trust, *in* 'Proceedings of the 14th international conference on World Wide Web', ACM Press, New York, New York, USA, p. 613.

Carroll, J. J. & Klyne, G. (2004), Resource Description Framework (RDF): Concepts and Abstract Syntax, W3C Recommendation, W3C.

`http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/`

Cyganiak, R., Harth, A. & Hogan, A. (2009), 'N-Quads: Extending N-Triples with Context'.

`http://sw.deri.org/2008/07/n-quads/`

De Bruijn, J. (2004), Semantic Information Integration Inside and Across Organizational Boundaries, Technical report, Digital Enterprise Research Institute.

Demir, E., Cary, M., Paley, S., Fukuda, K., Lemer, C., Vastrik, I., Wu, G., D'Eustachio, P., Schaefer, C., Luciano, J. et al. (2010), 'The BioPAX community standard for pathway data sharing', *Nature biotechnology* **28**(9), 935–942.

Doran, P., Palmisano, I. & Tamma, V. (2008), SOMET: Algorithm and Tool for SPARQL Based Ontology Module Extraction, *in* 'Proceedings of the 2008 ESWC International Workshop on Ontologies Reasoning and Modularity, WORM08, Tenerife, Spain', CEUR-WS.org.

Euzenat, J. & Shvaiko, P. (2007), *Ontology Matching*, Springer-Verlag, Heidelberg (DE).

Faulkner, S., Kolp, M., Nguyen, T., Coyette, A. & Do, T. (2004), Information integration architecture development: a Multi-Agent approach, *in* 'Proceedings of the 16th International Conference on Software Engineering and Knowledge Engineering (SEKE)'.

García-Sánchez, F., Fernández-Breis, J. T., Valencia-García, R., Gómez, J. M. & Martínez-Béjar, R. (2008), 'Combining Semantic Web technologies with Multi-Agent Systems for integrated access to biological resources', *Journal of Biomedical Informatics* **41**, 848–859.

Halvorsen, J. & Hansen, B. J. (2011), Integrating Military Systems using Semantic Web Technologies and Lightweight Agents, *in* 'NATO RTO Symposium RTO-IST-101 Semantic and Domain-Based Interoperability', NATO RTO. To appear.

Hansen, B. J. (2011), Towards Ontology Matching Suitable for Information Integration in Time-Critical Situations, *in* 'Proceedings of the 16th International Command and Control Research and Technology Symposium', The Command and Control Research Program (CCRP).

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).

IO Informatics (2010), Semantic Integration and Modeling of Scientific Data Sources, White paper, IO Informatics.

Kobilarov, G., Scott, T., Raimond, Y., Oliver, S., Sizemore, C., Smethurst, M., Bizer, C. & Lee, R. (2009), Media meets semantic web–how the bbc uses dbpedia and linked data to make connections, *in* L. Aroyo, P. Traverso, F. Ciravegna, P. Cimiano, T. Heath, E. Hyvönen, R. Mizoguchi, E. Oren, M. Sabou & E. Simperl, eds, 'The Semantic Web: Research and Applications', Vol. 554 of *Lecture Notes in Computer Science*, Springer, pp. 723–737.

Noy, N. (2004), 'Semantic integration: a survey of ontology-based approaches', *ACM Sigmod Record* **33**(4), 65–70.

Nwana, H. S. (1996), 'Software agents: An overview', *Knowledge Engineering Review* **11**, 205–244.

Panti, M., Penserini, L., Spalazzi, L. & Tacconi, S. (2002), A multi-agent system based on the p2p model to information integration, *in* 'Proceedings of the 1st International Conference on Autonomous Agents and MultiAgent Systems AAMAS'.

Prud'hommeaux, E. & Seaborne, A. (2008), SPARQL query language for RDF, W3C Recommendation, W3C.

`http://www.w3.org/TR/rdf-sparql-query/`

Rahimi, S. & Carver, N. (2005), A Multi-Agent Architecture for Distributed Domain-Specific Information Integration, *in* 'Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05)', IEEE.

Rasmussen, R. & Hansen, B. J. (2011), Experiment Report: SOA Pilot 2011, FFI-rapport 2011/02407, Norwegian Defence Research Establishment (FFI).

Shinavier, J. (2011), 'RDFAgents'. Site visited 28.09.2011.

`http://fortytwo.net/2011/rdfagents/spec`

Skjegstad, M., Johnsen, F., Hafsoe, T. & Lund, K. (2010), A protocol for robust and efficient service discovery in large, highly mobile radio networks, *in* 'MILITARY COMMUNICATIONS CONFERENCE, 2010 - MILCOM 2010', pp. 456 –463.

Stoutenburg, S., Obrst, L., McCandless, D., Nichols, D., Franklin, P., Prausa, M. & Sward, R. (2007), Ontologies for Rapid Integration of Heterogeneous Data for Command, Control, & Intelligence, *in* 'Proceedings of Ontology for the Intelligence Community (OIC-2007)'.

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.

Walton, C. (2006), *Agency and the Semantic Web*, Oxford University Press, Oxford.

Wooldridge, M. J. (2009), *An Introduction to MultiAgent Systems*, 2 edn, Wiley.