FFI-rapport 2008/01557

Experiment report: “Semantic SOA” – NATO CWID 2008

Bjørn Jervell Hansen, Jonas Halvorsen, Svein Ivar Kristiansen and Rolf E. Rasmussen

Norwegian Defence Research Establishment (FFI)

9 September 2008

FFI-rapport 2008/01557

1085

P: ISBN 978-82-464-1423-2

E: ISBN 978-82-464-1424-9

Keywords

Nettverksbasert forsvar

Tjenesteorientert arkitektur

Eksperimenter

Kunnskapsbaserte systemer

Approved by

Rolf E. Rasmussen	Project manager
Vidar S. Andersen	Director

English summary

NATO Coalition Warrior Interoperability Demonstration (CWID) is an annual event focusing on improving the interoperability within the alliance. One of the issues investigated during CWID is the NATO Network Enabled Capability (NEC), and SemSOA, the Norwegian Defence Research Establishment contribution at CWID 2008, aimed to contribute to NATO NEC by investigating semantic service-oriented architecture (SSOA). Loose coupling between services as provided by service-oriented architecture (SOA) is already a NATO NEC recommendation, and SSOA aims at providing an even looser coupling using semantic technologies (see Hansen et al. (2007)).

SemSOA performed technical tests exploring the following areas where SSOA is envisioned to enhance SOA: how to describe services (service annotation), how to find services (service discovery), how to utilise services (service invocation), and how to link services together (service orchestration). In addition to this, two more general areas for semantic technology application were investigated: information integration and information fusion.

During the experiment, we successfully demonstrated the ability to describe, find, and utilise several Web services, some of which were provided by other CWID nations, using SSOA principles. Successful application of semantic technologies for information integration and fusion was also demonstrated.

As this was a technical experiment, ontology modelling was considered outside the experiment scope and simplified auto-generated ontologies were used rather than real-world ontologies. Furthermore, it should be noted that currently available SSOA software tools are still to a large extent experimental and immature and thus limits the available tool support. Extensive software support is a prerequisite for a more widespread use of semantic technologies.

Semantic interoperability is a visionary goal overarching a large number of ongoing initiatives in the NATO research and information technology communities. In the information infrastructure of the future there will be obvious needs for technical solutions that contribute to semantic interoperability, but more research and development is needed before it is recommendable to include the technology of SemSOA in fielded systems. However, the underlying principles of this experiment have been shown to be very much valid, indicating that SSOA has the potential to add value to future information system solutions.

Sammendrag

NATO Coalition Warrior Interoperability Demonstration (CWID) er et årlig arrangement som søker å forbedre interoperabiliteten innen alliansen. Et av temaene under CWID er NATO Network Enabled Capability (NEC), og SemSOA, Forsvarets forskningsinstitutts bidrag under CWID 2008, hadde som mål å bidra til NATO NEC ved å utforske semantiske tjenesteorienterte arkitekturer. Løs kobling mellom tjenester i en tjenesteorientert arkitektur (SOA) er allerede en anbefaling innen NATO NEC, og semantisk SOA fremmer en enda løsere kobling ved hjelp av semantiske teknologier (se Hansen et al. (2007)).

SemSOA utførte tekniske tester for å utforske områder der semantisk SOA er tenkt å forbedre SOA. Disse områdene er hvordan tjenestene beskrives (tjenesteannotering), hvordan tjenestene oppdages (tjenesteoppdagelse), hvordan tjenestene kan benyttes (tjenesteinvokering) og hvordan tjenester kan kobles sammen (tjenesteorkestrering). I tillegg til dette utforsket eksperimentet ytterligere to områder der semantiske teknologier kan benyttes: informasjonsintegrasjon og informasjonsfusjon.

Utfallet av eksperimentet var en demonstrert evne til, ved hjelp av semantisk SOA, å beskrive, oppdage og bruke webtjenester, deriblant flere som var utviklet av andre CWID-nasjoner. I tillegg ble det demonstrert hvordan semantiske teknologier kan benyttes for å gjøre informasjonsintegrasjon og -fusjon.

Siden dette var et teknisk eksperiment, ble modellering av ontologier besluttet å ligge utenfor eksperimentet. Det ble derfor benyttet relativt enkle, autogenererte ontologier heller enn etablerte, store ontologier. Gjennom arbeidet med eksperimentet ble det slått fast at de tilgjengelige støtteverktøyene for semantisk SOA fortsatt i stor grad er eksperimentelle og umodne. Solid verktøystøtte er en forutsetning for en mer utstrakt bruk av semantiske teknologier.

Målet om semantisk interoperabilitet er en fellesnevner i en lang rekke forskningsinitiativer innen NATO. I den framtidige informasjonsinfrastrukturen vil det bli behov for tekniske løsninger som bidrar til semantisk interoperabilitet, men før det kan anbefales å inkludere teknologiene brukt i SemSOA i operative systemer er det behov for mer forskning og utvikling. Dette eksperimentet bekrefter imidlertid at semantisk SOA potensielt kan gi verdi til fremtidige informasjonssystemer.

Contents

1 Introduction

This document describes the research-based experiment titled 'Semantic SOA' (SemSOA) that was performed at the NATO Coalition Warrior Interoperability Demonstration (CWID) 2008. The title indicates an approach to enhance service-oriented architectures (SOAs) by adding the capabilities enabled by semantic technologies.

NATO CWID is an annual event designed to improve interoperability within the alliance. SemSOA was the third in a row of SOA-related CWID trials from the Norwegian Defence Research Establishment (FFI). In 2006, the trial 'Secure SOA supporting NEC' (SecSOA), documented in Rasmussen et al. (2006) was a joint effort between FFI and a number of partnering nations within a NATO RTO research group. In 2007, three FFI projects worked together to create the trial 'SOA - Cross Domain and Disadvantaged Grids', which is documented in Haakseth et al. (2007).

The overall goal of these activities has been to support the NATO Network Enabled Capability (NEC) by facilitating a looser coupling and more efficient communication between consumers and content providers in a military network. For research-based solutions using emerging technologies, as was the case for SemSOA, successful CWID participation confirms that the technical experiments and demonstrations performed are applicable in a military controlled technical environment without Internet access. NATO CWID is also a valuable arena for military IT professionals to meet and exchange ideas. This constitutes an interested audience for presenting research topics with potential future benefits.

The SemSOA experiment at CWID 2008 was designed to show how semantic technologies may add value to SOA. In the tests we added an additional layer of formal semantics to Web service descriptions in a SOA environment in order to improve the following service-related activities:

- Annotation - describing a service
- Discovery - finding a service
- Invocation - using a service
- Orchestration - linking several services together.

Further, the experiment demonstrated how the semantic layer introduced for services could imply important benefits in the areas of information integration and information fusion.

An introduction to semantic technologies in general can be found in Hansen et al. (2007).

This report is organised as follows: Section 2 gives an overall experiment introduction, while Section 3 provides descriptions of the technological goals of the experiment. An experimental information system - hereafter referred to as the demonstrator - was built and used as an arena to explore the behaviour of these activities during the experiment. The demonstrator is described in Section 4. The results of the experiment are then presented in Section 5, while Section 6 sums up and concludes the report.

2 Overall Experiment Description

The main objective of the SemSOA experiment was to test and demonstrate semantic technologies, and in particular their ability to add value to traditional SOA. In order to achieve this, a demonstrator with a browser-based user interface was developed. On the non-technical side, the goal was to establish contact with professionals within the CWID community who share our interest in semantic technologies and their potential benefits.

Before we look more closely at the experiment, it is timely to introduce the concept of semantic service-oriented architecture (SSOA): SSOA is a new-style software architecture that fuses semantic Web and Web service technologies and uses domain ontologies to model semantic information in Web service descriptions. This technology facilitates computer processable service descriptions which can be used to realise high-level operations. SSOA provides an excellent solution to automatic discovery, composition and execution of Web services and is a feasible approach to realise seamless integration of applications and data on the Web (Man et al. 2006). In the SemSOA experiment we focused on SSOA implemented as semantic Web services (SWS)

In order to meet the experiment objectives, the SemSOA trial at CWID 2008 conducted the following activities:

- Development of a demonstrator to support the exploration of semantic technologies in the context of SSOA
- Execution of technical tests with partners from other nations – both planned and ad-hoc
- Demonstration of SSOA to a technical audience
- Presentations of the potential value of SSOA used in future military information systems.

2.1 Demonstration Scenario

To achieve the goal of demonstrating semantic technologies in general, and SSOA in particular, we outlined a simple scenario as a backdrop for a use case. This scenario had no connection to the official CWID scenario other than being set in the same geographical location.

In the use case, a military decision maker is in need of information about military units within a given area. The relevant information resides in multiple heterogeneous back-end systems, and an integrated view to the information is presented to the user by the SemSOA demonstrator. As the decision maker is operating in an environment where service availability varies, s/he will not know which sources are currently available. To overcome this problem, the decision maker searches for available services providing the sought information type.

The search returns a list of information sources that the decision maker invokes in sequence, including:

- a NATO Friendly Force Information (NFFI) (Malewicz 2006) Web service providing information about friendly units in the area
- a logistics Web service providing more detailed information about friendly logistics units in the area.

In addition to these two sources, which are represented as services, the decision maker also has access to a database based on the Joint Consultation, Command and Control Information Exchange Data Model (JC3IEDM) (MIP 2008) with situation information on both friendly and hostile units

The different services provide information in different formats, but helped by the SSOA approach, the information is integrated and presented to the decision maker by the SemSOA demonstrator's browser-based user interface.

A geographical view of the information available to the decision maker is shown in Figure 2.1.

2.2 Technical Tests

The technical aspects of a CWID execution are based on test cases. A test case is a predefined test between partners from different CWID nations where the objective is to test a single well-defined interoperability issue. During the SemSOA experiment we used the test case framework to perform technical tests with trials from Spain, Germany, Canada, and the NATO C3 Agency (NC3A).

SemSOA's activities with international partners were important for several reasons. It was crucial to prove that semantic technologies can be applied to currently deployed real applications, and thereby add value to all parties involved. We also wanted to investigate that this process was not too time consuming for real-world application.

2.3 Demonstrating and Presenting SSOA

At the CWID execution we performed a series of demonstrations and presentations. This was done for two reasons; firstly we wanted to demonstrate how semantic technologies can add value to SOA, and secondly we wanted to establish contact with potential future research partners.

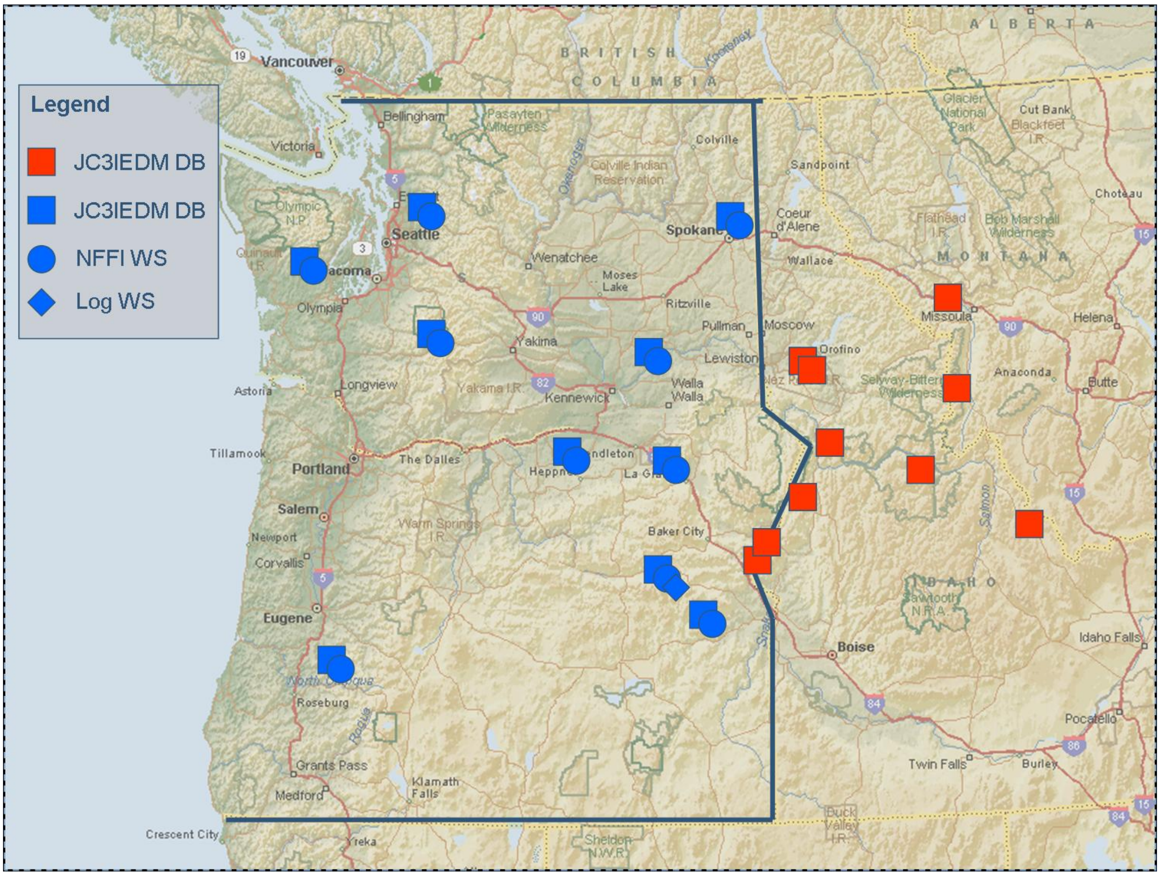

Figure 2.1: A geographical view of the information from the JC3IEDM database (squares), the NFFI Web service (circles), and the logistics Web service (diamonds) used in the demonstration use case

The demonstrations focused on the technical aspects of SSOA and how we chose to solve them in our demonstrator. To a technician, what happens behind the scene is often more important than what happens in the user interface, so all aspects of the demonstrator were available to the audience during these demo sessions.

In addition to the demonstrations, a number of presentations were given about the potential value of using SSOA in future military information systems .

3 Technological Goals

The experiment focused on SSOA implemented as semantic Web services (SWS) , and our activities at the demonstration reflected this. We divided the domain of SWS down into smaller parts, and assigned technological goals to explore each of these. Our aim was to experiment with what we see as the core features and benefits of SWS. We identified four specific goals that warranted study:

- Annotation
- Discovery
- Invocation
- Orchestration.

Furthermore, two generic semantic Web features were studied. These features are not exclusive for SWS, but apply to semantic technologies in general:

- Information integration
- Information fusion.

The individual technologies and the experiments which were undertaken in order to address them, will now be discussed.

3.1 Annotation

Our first goal was to annotate traditional Web services with semantic descriptions. To describe SWS, we chose to use the Web Ontology Language (OWL) for Services (OWL-S) which "enables users and software agents to automatically discover, invoke, compose, and monitor Web resources offering services, under specified constraints" (W3C 2004a). This standardised service description ontology facilitates automated reasoning, and consists of three parts: the service profile, the process model and the grounding (see Figure 3.1). For further reading on the different parts of the OWL-S ontology, consult W3C (2004a).

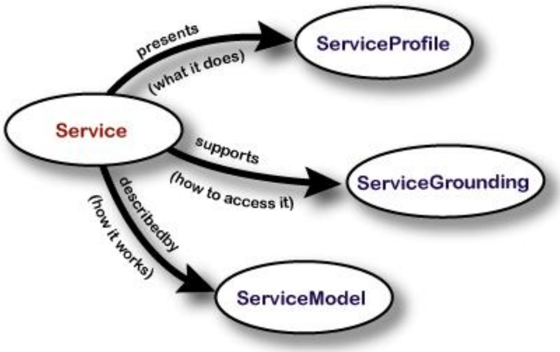

Figure 3.1: Top level of the service ontology (W3C 2004a).

SemSOA experimented with annotations to determine whether Web services that were deployed on the military network could be annotated using the OWL-S ontology. Note that semantic annotations using OWL-S do not imply any alterations to the original Web services but rather augment it.

The annotations also provide computer processable advertisements of the services. These advertisements could be exploited in the following SemSOA experiment activities: service discovery, service invocation, and service orchestration.

3.2 Discovery

One of the envisioned benefits of SWS is that it will improve the ability of discovering candidate Web services. Traditional Web service techniques for dynamic service discovery perform service matching based on the XML Schema Definitions (XSDs) and Web Service Description Language (WSDL) descriptions that the client applications and the Web services support. Furthermore, Universal Description Discovery and Integration (UDDI) matching is mainly based on matching of textual service descriptions and thus relevant Web services might not be found due to minor syntactical differences.

As noted earlier, OWL-S descriptions facilitate the use of automated reasoning to discover potential services based on OWL semantics. Thus in addition to finding services that are identical to the request in terms of type of input and output (according to a formally specified ontology), it will also be able to find services that have input/output types that are equivalent (but named differently), as well as services with inputs/outputs that are subclasses of the requested types (subsumption).

We aimed to test this feature by having a client application search for Web services that fit its specific requirements. The test involved both direct and subsumption matching.

3.3 Invocation

Invocation of semantically annotated and discovered Web services may give several advantages over traditional Web services. By raising traditional Web services to the semantic level, applications using SWS technologies should be able to dynamically invoke Web services that were not known at design-time (not known provider/not known format) and still be able to make use of the information received.

We planned to use our generic client application to demonstrate that it was possible to invoke the services hosted by other nations, and that we were able to utilise the information with minimal work and no need for changes to the client. A run-time conversion from data on the semantic level to the XML-format, expected by traditional Simple Object Access Protocol (SOAP) based Web services, needs to take place during requests, and vice versa for the response. Creating the Extensible Stylesheet Language Transformations (XSLTs) was part of the annotation goal, but the validity of these could not be fully tested before actually invoking the semantically annotated Web services.

This goal had two purposes. Firstly, we needed to perform the invocation goal to verify that the annotation goal had been completed successfully. Secondly, we wanted to check if current tool support was sufficient to perform invocation of real-world semantically annotated services.

3.4 Orchestration

Another predicted benefit of the use of SWS technology is the possibility of performing on-the-fly automated orchestration of Web services. This is achievable if knowledge is represented in a formal way that allows goal-directed machine planning and reasoning. Traditionally, orchestration of services is either performed programmatically or through higher level process languages, i.e. the Business Process Execution Language (BPEL). Further, orchestration is manually modelled by humans and changes in process flow require human intervention. By utilising semantic annotations, orchestration can be automated as annotations can provide sufficient information to construct compound services. For instance, if a sought service is not available, the desired result can be attained by orchestrating (chaining) a set of other Web services.

3.5 Information Integration

A much hailed benefit of the Semantic Web information representation language Resource Description Framework (RDF) (W3C 2004c) versus traditional data representations is that RDF allows for representing data in an abstract way. Porting data serialised in traditional data formats over to RDF makes the data independent of its internal representation. Once the data is in RDF form, it is a simple matter to merge it with data from other sources. Furthermore this allows us to use standardised RDF tools on the data (Herman 2008).

SemSOA planned to perform information integration of multiple heterogeneous sources at CWID 2008. In addition to integrating with various SOAP-based Web services, we integrated information from a JC3IEDM database into our knowledge base. These sources had in common that the information provided by them was not in RDF form.

3.6 Information Fusion

The integration of data from different sources can be used to augment the information objects at hand in order to create a fuller picture. Although this alone can indeed derive new knowledge, the use of rules coupled with ontologies can greatly extend the inference capabilities. SemSOA planned to derive new information based on the resulting information set from the information integration described in Section 3.5.

In the experiment the goal was to infer which logistics units could replenish military units nearby. A logistics unit in our system carries specific equipment and has a limited radius of action. As a result of previously performed information integration, information on all involved units would be in a common data store. With all necessary data available, new information could be inferred by employing rules, and presented to the user in the demonstrator's user interface.

4 The SemSOA Demonstrator

In order to succinctly demonstrate the benefits of the use of semantic technologies and semantic SOA, a software demonstrator was developed. It bound the individual technical experiments together. This provided us with a single testbed in which to conduct our experiments, while at the same time providing us with a graphical demonstration tool that we could use to show the direct user benefits.

This section will now briefly describe the demonstrator, its architecture and its central components.

4.1 Description of System

The demonstrator was implemented as a loosely coupled, distributed, Web-based system. From the user's perspective, the system was available through a Web-based graphical user interface (GUI)[1], accessible through the use of a Web browser. The core system functionalities were provided by individual back-end components implemented as lightweight Web services. These services were responsible for tasks such as discovering candidate Web services, invoking Web services, providing map data, storing knowledge, and performing reasoning. The Web client invoked these services through the use of simple HTTP requests, and received replies asynchronously which were processed client-side, in-browser using JavaScript (Asynchronous JavaScript and XML – AJAX).

The architecture of the overall system is shown in Figure 4.1. Here we see the division between the core services (back-end components providing the functionality), and the external Web services that were used as data sources for the individual tests.
The core components of the system were the Web client and the three core services:

1. The matchmaker
2. The invocation service
3. The knowledge base.

As for the external Web services we specifically highlight the logistics and NFFI Web services as these services were used for both the SWS experiments and the information integration/fusion experiments (the bulk for the external Web services at our disposal were used only for the SWS experiments).

[1]Hereby referred to as the Web client

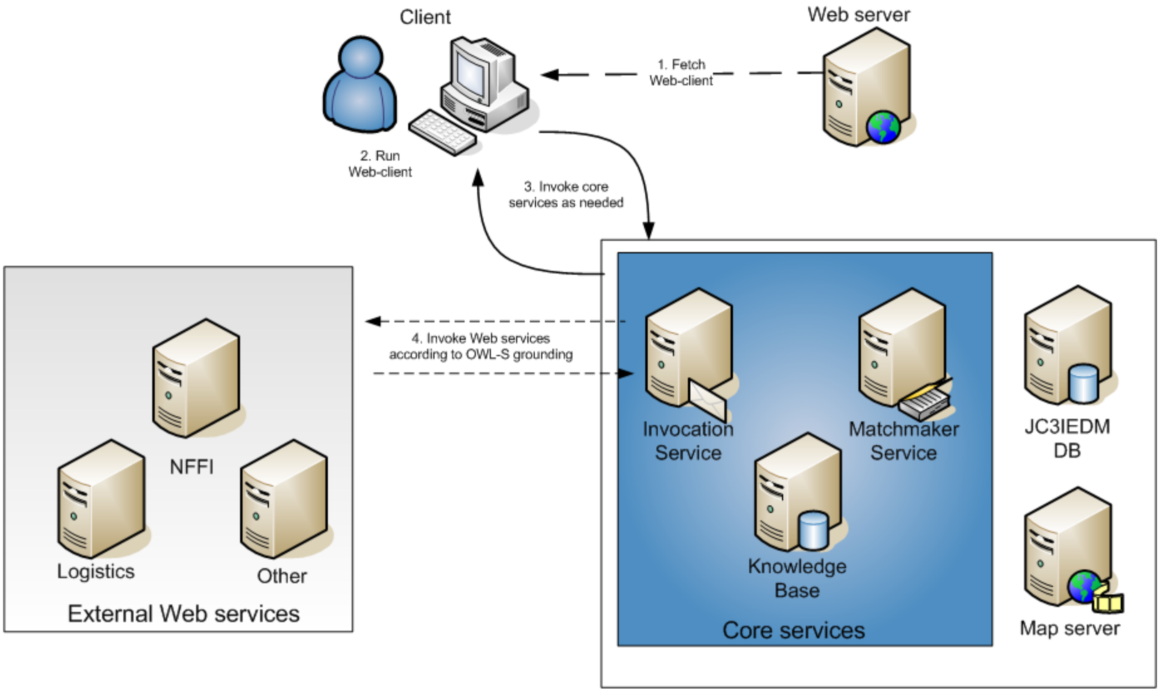

Figure 4.1: Demonstrator system architecture

Note that the map server is a non-critical component of the system, and is thus not shown as part of the core services. Additionally, data from a JC3IEDM database is retrieved in RDF form in order to augment data retrieved from the external Web services.

4.2 Web Client

The Web client consists of an AJAX-enabled web page, hosted on a Web server. In order to utilise the system, the user first has to fetch the web page using a Web browser, which results in the JavaScript code being run. Upon user manipulation of the GUI, the Web client sends requests to the core Web services. This is done in the background without locking the GUI, through the use of asynchronous HTTP.

Upon receiving data, the browser calls the relevant JavaScript code (included in the fetched web page) which processes it and updates the relevant parts of the display with the results (updates the HTML document object model). The reply data is serialised as objects in JavaScript Object Notation (JSON) form rather than traditional SOAP messages. This design decision was taken in order to reduce the necessary in-browser processing to a minimum, as JSON requires less work to process in JavaScript than XML due to the fact that JSON objects are fully valid JavaScript objects.

Furthermore, the Web client utilises a Web Map Service (WMS) enabled map server as a source for map data. The map data is used to plot data in the knowledge base that contain geolocation information on a graphical map. The Web client also uses an open-source SPARQL Query Language for RDF (SPARQL – see W3C (2008)) dataset browser (packaged with D2R, written in JavaScript) that allows users to browse the knowledge base (i.e. browse the instances and relations that exist).

4.3 Core Services

The back-end Web services provide the underlying functionalities of the system. More specifically, they deal with the process of matching and invoking SWS, as well as performing inference and committing knowledge to the knowledge base.

The individual core services will now be described.

4.3.1 Matchmaker

The matchmaker service deals with dynamic service discovery. It does this by matching ideal service descriptions, provided by the client, to actual existing services that it knows about. Candidate services are rated with regards to how well they match the ideal service description in terms of input/output concepts defined in a formal ontology. The set of candidate services include services that are identical to the sought description, services that expect input/output concepts that are more specific than the ideal one thus directly usable according to the OWL subClassOf semantics (W3C 2004b), as well as partial matches that can possibly be utilised in coordination with other services (i.e. matches on output, but not input).

We rely on using the OWL-S Matchmaker software developed by Carnegie Mellon University, to provide the matchmaking functionality (Srinivasan et al. 2004). We chose this software due to it being built on top of jUDDI (a well-known Java UDDI implementation) as well as for its relative maturity. There exists a handful of other OWL-S matchmakers that support more or less the same functionality but differ in underlying architecture, technology and algorithms. Thus, replacing the matchmaking component in our system should be quite straightforward if needed (e.g. if a peer-to-peer solution is necessary or another matching algorithm is preferred).

4.3.2 Invocation Service

The invocation service is responsible for the actual invocation of the Web services described in OWL-S. It utilises the OWL-S API, an open-source API for invoking OWL-S descriptions originally developed at the University of Maryland (Sirin and Parsia 2004). More specifically, we used the OWL-S API to handle the process of lifting and lowering data between the semantic level (instances of the ontology concepts, as specified in the OWL-S process model) to the specific XML format required by the service WSDLs (as specified in the OWL-S grounding) as well as for performing the actual invocation of the services. This involves performing XSLT transformation and SOAP envelope wrapping and unwrapping.

Ideally, SWS invocation would have been done client-side. However, no OWL-S implementations exist for JavaScript, nor did we find it worthwhile to implement it ourselves.

4.3.3 Knowledge Base

We used the open-source Jena-SDB triple-store (HP 2008b) as our knowledge base. This allowed us to access and modify the RDF triples through the use of the Jena API (which simplified coding) (Carroll et al. 2004) as well as to expose the store through a SPARQL endpoint through Joseki (HP 2008a). Furthermore, it provided us with the ability to employ self-defined rules (Jena rules) for mapping instances from heterogeneous information sources that describe the same thing, as well as for inferring new knowledge not previously explicit.

4.4 Tools

In order to simplify the process of writing and publishing OWL-S descriptions for existing Web services, we developed two tool applications; an OWL-S description editor and an OWL-S publisher.

Our OWL-S editor is a Java GUI application that automates large parts of the process of writing an OWL-S description by extracting information from the target WSDL file, and providing drag-and-drop ontology concept mapping. It is based on the WSDL2OWLS editor included with the OWL-S API, but has been modified extensively to suit our needs. The OWL-S publisher is a Java GUI application that allows us to easily publish OWL-S descriptions to the matchmaker, as well as to perform debugging.

5 Experiment Results

The technical results from the experiment can be divided into two categories; results as output from test cases, and results from the demonstrations.

The test cases, which are listed in Appendix A, focused primarily on three of the four technological goals related to SSOA: annotation, discovery, and invocation.

Additionally, four other test cases not directly related to SSOA were conducted. The topic of these test cases were that of federating the SemSOA jUDDI to other UDDI implementations, and the usage of an external map server as a data source. As neither of these touch upon the main focus of this experiment, they will not be further described here.

The results emerging from the demonstrations addressed the SSOA tests as well as the information integration and fusion tests. Due to time constraints, the orchestration test case was never performed and thus no results were produced.

The results from the experiment are summarised in Table 5.1, and are further elaborated upon in the following subsections.

Technological goal	Result
Annotation	Successfully annotated external Web services using OWL-S, created within a reasonable amount of time (less than half a day).
Discovery	Successfully discovered relevant Web services using semantic technologies matching on direct as well as subsumed concepts.
Invocation	Successfully invoked external, semantically annotated Web services using our generic Web client, thus demonstrating the ability to invoke and utilise Web services that were not known to the client at design time
Orchestration	Due to time constraints no tests were performed on this goal, thus no results were achieved.
Information Integration	Successfully integrated information from three heterogeneous sources by mapping their output to RDF and merging the data.
Information Fusion	Successfully derived new (explicit) knowledge from the integrated data set through the use of rule-based reasoning.

Table 5.1: A summary of the experiment results

5.1 Annotation

In order to determine whether it is feasible to take existing Web services and enhance them with semantic annotations, an annotation scheme was tested on Web services from other CWID nations. The scheme included the use of an OWL-S description editor (Section 4.4) to auto-generate a large part of the OWL-S description from the Web services' WSDLs. However, the most complex part of the semantic annotation, writing the XSLT transformations responsible for lifting and lowering the data between the semantic level and the XML-data level, had to be done by hand. This was a time-consuming task, mainly caused by the fact that RDF/XML can be serialised in a number of ways due to RDF being graph-based, resulting in different internal structures making it more complicated to process procedurally. As a result, the XSLT transformations have to cover all the different possible structures of serialisation. It should be noted that RDF/XML was not created with XSLT transformations in mind.

A possible solution that would reduce the work needed to create XSLT transformations is to first perform a SPARQL query on the RDF data to return only the relevant data. This will be beneficial in two senses; reduce the dataset and having the data in a flat, non-RDF XML structure (SPARQL-result format). The XSLT transformations needed to convert from the SPARQL-results to the sought XML-format should be considerably less complicated to create (W3C 2007).

The experiment demonstrated that the annotation of existing Web services is indeed feasible. The main concern was the time consumption creating the OWL-S description, but we were able to make the necessary annotations in less than half a day. This included the creation the XSLT transformations, and we consider this to be within a reasonable amount of time. With practice, this could be further reduced.

5.2 Discovery

Utilising the SWS annotations from Section 5.1, we were able to automatically find relevant Web services matching on the services' output using the demonstrator's matchmaker (Section 4.3.1). We were able to match on both direct output concepts (e.g. matching output 'Message' against output 'Message'), and on subsumed output concepts (e.g. matching output 'Message' against the more specific output 'TrackMessage').

5.3 Invocation

During the invocation phase of the experiment, we used the demonstrator's generic Web client (Section 4.2) to invoke the external Web services. The Web client used the semantic annotations to determine where to find the service and how to deliver the input and handle the output.

This demonstrated an ability to invoke and utilise Web services that were not known to the client at design-time.

5.4 Information Integration

The information integration was done using output from three heterogeneous information sources:

- An NFFI Web service
- A logistics Web service
- A JC3IEDM database.

The integration concept was quite straightforward: Transform the output from the three sources to an RDF representation and store it in a common knowledge base. In addition to this, the knowledge base was enhanced with rules to tie common items in the three originally separate information sets together. Thus it was possible to retrieve all the available information on a single item in a single query even though the information originated from three separate sources.

5.5 Information Fusion

In order to explore reasoning-based information fusion, the integrated information set described in Section 5.4 was used as a basis.

In this experiment, we chose to demonstrate how information fusion can enhance the information set in a logistics setting. Based on our scenario, we exemplified this by inferring relations between logistics units and friendly units requesting supplies. More specifically, the relations stated whether the logistics units were able to replenish the different friendly units. The necessary rules to infer this relation were specified in the Jena rules format (Carroll et al. 2004) and added to the knowledge base. In essence the rules were in an IF-THEN-form, stating:

1. IF unitA has ordered amountA of product productA
2. IF logUnit has at least amountA of product productA in stock
3. THEN logUnit can replenish unitA

The confirmation of the inferred relation was the presence of the relation in the knowledge base after the inference process.

5.6 Problems Encountered

Several issues appeared during the experiment, and we now present those thought to be of common interest. In addition to the issues described below we also experienced computer performance problems, in particular in connection with reasoning. However, as these problems would have been avoided using more powerful computers, they are not further elaborated upon here.

5.6.1 Variety of WSDL Binding Implementations

The WSDL standard allows the use of a range of different types of bindings. The WSDL descriptions that we discovered at CWID showed that in reality there is no one binding type that is a de facto standard. The OWL-S API did not support all the different types of bindings, thus limited what services we were able to invoke. We added support for some bindings that were not supported off-the-self in the API, but did not find it to be a good use of time to continue this process. However, we were able to determine that our modifications made the OWL-S API support all WSDLs that follow the Web Services Interoperability Organization (WS-I) policy described in WS-I (2007).

5.6.2 Tool Inadequacies

A recurring problem when dealing with SWS is the fact that the tool support is still limited, and the existing tools are often more or less experimental and immature. An example of an experimental and immature tool is the OWL-S API which was used in the invocation service (Section 4.3.2) and in the WSDL2OWLS editor (Section 4.4).

More generally, we would have benefited from better tool support when creating the OWL-S descriptions, and in particular when creating the XSLT transformations. Even though general XSLT tools exist, we were not able to find any tools to support the creation of transformations from XML documents to XML-based RDF documents needed to lower and lift the data between the semantic level and the XML-data level. One potential solution to this problem is outlined in Section 5.3.

6 Conclusion

The experiment SemSOA performed by the FFI project Semantini at NATO CWID 2008 confirmed that semantic technologies can add value in a service-oriented architecture. In an arena of military information systems, it was technically feasible to add capabilities of semantic technologies to existing Web services and exploit these capabilities. Preparation and execution of the experiment gave the participants valuable technical experience, and during the execution period several valuable contacts with peer technicians were established.

The interoperability testing conducted by SemSOA during CWID was technically focused. Adhering to the CWID goal of promoting loose coupling between systems, the experiment focused on providing ways of improving the already loose couplings inherent in SOA. Valuable results were achieved in the primary areas of annotation, discovery and invocation of services. Orchestration of services, which initially was among the technical goals, had to be left open due to time and resource constraints.

Regarding annotation, the experiment showed empirical results of manually adding a semantic layer to existing Web service descriptions. This was done in less than half a day per service, which was considered acceptable.

The semantic annotations were used in the discovery process, where the ability to find services described using subsumed concepts was proved. Further, the same annotations made it possible to invoke the service using a generic client. Finally, the experiment also demonstrated integration and fusion of the retrieved information.

As this was a technical experiment, ontology modelling was outside the experiment scope and simplified auto-generated ontologies were used rather than real-world ontologies. Furthermore, it should be noted that currently available software tools are still to a large extent experimental and immature and thus limits the available tool support. Extensive software support is a prerequisite for a more widespread use of semantic technologies.

Semantic interoperability is a visionary goal overarching a large number of ongoing initiatives in the NATO research and information technology communities. In the information infrastructure of the future there will be obvious needs for technical solutions that contribute to semantic interoperability, but more research and development is needed before it is recommendable to include the technology of SemSOA in fielded systems. However, the underlying principles of this experiment have been shown to be very much valid, indicating that SSOA has the potential to add value to future information system solutions.

References

Carroll, J., Dickinson, I., Dollin, C., Reynolds, D., Seaborne, A. and Wilkinson, K.: 2004, Jena: implementing the semantic web recommendations, *Proceedings of the 13th international World Wide Web conference*, ACM New York, NY, USA, pp. 74–83.

Haakseth, R., Gagnes, T., Hadzic, D., Hafsøe, T., Johnsen, F. T., Lund, K. and Reitan, B. K.: 2007, Experiment report: "SOA – Cross Domain and Disadvantaged Grids" – NATO CWID 2007, *FFI/RAPPORT-2007/02301*, Norwegian Defence Research Establishment.

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

Herman, I.: 2008, Tutorial on Semantic Web, `http://www.w3.org/People/Ivan/CorePresentations/SWTutorial`.

HP: 2008a, Joseki - A SPARQL Server for Jena, `http://www.joseki.org`.

HP: 2008b, SDB - a SPARQL database for Jena, `http://jena.hpl.hp.com/wiki/SD`.

Malewicz, R.: 2006, NATO Friendly Force Information (NFFI) (version 1.2) Interface Protocol Definition IP3, NC3A Working Document.

Man, J.-F., Li, L., Chen, Q. and Deng, X.-H.: 2006, Study on interoperable model for semantic service-oriented architecture, *Technical report*, Hunan University of Technology.

MIP: 2008, MIP Data Modelling Working Group, `http://www.mip-site.org/035_Public_Organ_DMWG.htm`.

Rasmussen, R., Eggen, A., Hadzic, D., Hedenstad, O.-E., Haakseth, R. and Lund, K.: 2006, Experiment report: "Secure SOA supporting NEC" – NATO CWID 2006, *FFI/RAPPORT-2006/02538*, Norwegian Defence Research Establishment.

Sirin, E. and Parsia, B.: 2004, The OWL-S Java API, *Proceedings of the Third International Semantic Web Conference (ISWC2004), Hiroshima, Japan, 2004*.

Srinivasan, N., Paolucci, M. and Sycara, K.: 2004, Adding OWL-S to UDDI, implementation and throughput, *Proc. 1st Intl. Workshop on Semantic Web Services and Web Process Composition (SWSWPC 2004)*.

W3C: 2004a, OWL-S: Semantic Markup for Web Services, `http://www.w3.org/Submission/OWL-S/`. W3C Member Submission.

W3C: 2004b, OWL Web Ontology Language Overview, `http://www.w3.org/TR/owl-features/`. W3C Recommendation.

W3C: 2004c, RDF Primer, `http://www.w3.org/TR/rdf-primer/`. W3C Recommendation.

W3C: 2007, Semantic Annotations for WSDL and XML Schema, http://www.w3.org/TR/sawsdl/. W3C Recommendation.

W3C: 2008, SPARQL Query Language for RDF, http://www.w3.org/TR/rdf-sparql-query/. W3C Recommendation.

WS-I: 2007, Basic Profile Version 1.2, http://www.ws-i.org/Profiles/BasicProfile-1_2(WGAD).html. Working Group Approval Draft.

Appendix A Test Cases

ID	Trials involved	Test description
143	ESP – SOA-ESB NOR – SemSOA	ESP-UDDI registry service will be federated with the NOR-SemSOA UDDI registry service
144	ESP – SOA-ESB NOR – SemSOA	NOR-SemSOA UDDI registry service will be federated with the ESP-UDDI registry service
277	ESP – SOA-ESB NOR – SemSOA	The semSOA system will, based on a semantic annotation of a web service provided by SOA-ESB, discover and invoke the service using a generic client. web service provided by SOA-ESB will be RMP
498	NATO – ACCS-NNEC NOR – SemSOA	An ACCS-NNEC web service will be semantically annotated and then registered, discovered and involed using the semSOA system and generic client
1095	CAN – IOPC NOR – SemSOA	IOPC will expose COP data through a web service. The consumers are responsible for the selection and retrieval of the data
1234	NATO – IMROS NOR – SemSOA	The NMRR will federate its queries to the SemSOA UDDI registry, exposing their registered services/artifacts through the NMRR interface
1235	CAN – IOPC NOR – SemSOA	SemSOA will annotate the IOPC MTBGetTrackService Web Service using OWL-S and register, discover, and invoke it using an experimental OWL-S/jUDDI registry and a generic client
1419	DEU – GeoInf Services NOR – SemSOA	BGIO provides map data via web service (WMS)
1424	ESP – SIGEMOFAS NOR – SemSOA	SemSOA will annotate a SIGEMOFAS Web Service and discover and invoke it using the SemSOA experimental system

Appendix B Abbreviations

AJAX	Asynchronous JavaScript and XML
BPEL	Business Process Execution Language
CWID	Coalition Warrior Interoperability Demonstration
FFI	Norwegian Defence Research Establishment (Forsvarets forskningsinstitutt)
GUI	Graphical User Interface
JC3IEDM	Joint Consultation, Command and Control Information Exchange Data Model
JSON	JavaScript Object Notation
NATO	North Atlantic Treaty Organisation
NC3A	NATO C3 Agency
NEC	Network Enabled Capability
NFFI	NATO Friendly Force Information
OWL	Web Ontology Language
OWL-S	Web Ontology Language for Services
RDF	Resource Description Framework
SOA	Service-Oriented Architecture
SOAP	Simple Object Access Protocol
SPARQL	SPARQL Query Language for RDF
SSOA	Semantic SOA
SWS	Semantic Web Services
UDDI	Universal Description Discovery and Integration
W3C	World Wide Web Consortium
WMS	Web Map Service
WS-I	Web Services Interoperability Organization
WSDL	Web Service Description Language
XSD	XML Schema Definition
XSLT	eXtensible Stylesheet Language Transformation