CoNSIS: Demonstration of SOA Interoperability in Heterogeneous Tactical Networks

Trude H. Bloebaum and Ketil Lund
Norwegian Defence Research Establishment (FFI)
Kjeller, Norway
trude-hafsoe.bloebaum@ffi.no, ketil.lund@ffi.no

Abstract—**The Coalition Network for Secure Information Sharing (CoNSIS) conducted a large scale experiment in Germany in June 2012. During this experiment, multiple aspects of interoperability in the tactical domain were tested in practice. This paper presents the challenges faced by Task 2, which focuses on service orientation, and the use of Web services technology as a means to achieve interoperability between nations. Furthermore, it describes how these challenges were addressed by the different information infrastructures involved. We also present our experiences with several central Web service standards, and describe some lessons learned when it comes to utilizing these standards in tactical networks.**

SOA; Web services;service discovery; publish/subscribe

I. INTRODUCTION

The Service-Oriented Architecture (SOA) concept, most commonly implemented as Web services, is seen as a key enabler for meeting the technical interoperability requirements needed to achieve the NATO Network Enabled Capabilities (NNEC) vision. Within NATO, Web services technology has been the focus of the Core Enterprise Services (CES) working group, which has defined a number of common infrastructure services as core enterprise services. Having a common understanding of how these services are to be implemented and used is critical when attempting to achieve interoperability across national and systems boundaries. An important part of the work towards achieving this common understanding is to utilize these services in experimentation, in which the candidate technologies are tested under conditions similar to those found in an operational network.

The Coalition Network for Secure Information Sharing (CoNSIS) is a multinational group consisting of members from Germany, France, USA, and Norway, with participants from both research institutions and industry. The objectives of this group are to develop, implement, test, and demonstrate technologies and methods that will facilitate the partners' abilities to share information and services securely in ad-hoc coalitions, and between military and civil communication systems, within the communications constraints of mobile tactical forces.

The group has focused on practical application of information infrastructure technologies in a network-of-networks, consisting of a variety of low capability network technologies. The work done within the CoNSIS group has been divided into a number of tasks, each focusing on a different aspect of interoperability issues. This paper focuses on the work done by Task 2, which, together with [1], covers the domain of SOA and its application in limited capacity networks. During June 2012 CoNSIS conducted a large-scale experiment in Greding, Germany, in which all the different aspects of technical interoperability were tested; integrating the work of all the task groups of CoNSIS.

II. BACKGROUND

For Task 2, the goal of the CoNSIS experimentation was to show that by using the Web service standards specified by the NNEC CES as interoperability enablers, independent implementations are able to interoperate with only the service specifications as a common reference. The reasoning behind focusing on standards as a means of achieving interoperability was that it enables us to evaluate not only the implementations used, but also the standards themselves. In other words, we can assess how suitable the standards specified in the NNEC CES are for use in tactical networks.

A. SOA challenges

Assessing Web services (and other SOA implementation approaches) through the use of standards based experimentation is not new; multiple other experiments covering several topics of Web service interoperability have been performed previously. One of the most recent of these was the "Making Services Interoperable"-experiment conducted by the NATO RTO IST-090 working group last year [2]. In this experiment, three different SOA-based information infrastructures were connected together and interoperability was achieved. This experiment does however differ from the CoNSIS experiment in multiple ways:

- The CoNSIS experiment network was considerably more complex than the network used duing the IST-090 experiments. This added complexity meant larger fluctuations in the network conditions experienced by the Web service frameworks being used.
- The IST-090 experiment focused on service invocation, while the CoNSIS experiment covers both service discovery and service invocation.
- IST-090 managed to achieve interoperability between both Web service and Data Distribution Service (DDS) implementations of SOA, but this required the use of specialized gateways which only supported some

service types. The CoNSIS experiments are limited to Web service based implementations only, which allows for a more generic approach to service interoperability.

- The CoNSIS experiment was performed using an IPv6 network, while the IST-090 experiment was done on IPv4. This difference meant that there were additional challenges imposed on the CoNSIS experimentation as the support for IPv6 in pre-existing software and software development tools is limited.

There were primarily two NNEC core services that we wanted to test with respect to interoperability, namely service discovery and publish/subscribe. In addition, we also saw the need for focusing on *topics*, which is a method for classifying information, and as such constitutes the intersection between service discovery and publish/subscribe. These three areas are further described in the following sections.

B. *Experimentation Networks*

The CoNSIS land mobile part of the network consists of contributions from Germany and Norway. In addition to four German and four Norwegian mobile nodes located on military vehicles, there was also an NGO vehicle. However, this NGO vehicle was not used in the SOA experiments and will therefore not be further described in this paper. Fig. 1 shows the parts of the CoNSIS network that was being using during the SOA experiments.

Two different radio types were used. This is partly because Germany and Norway do not have the same radio systems, but also because different wireless technologies, having different properties with regards to e.g., range, are needed. Together the two radio links form a heterogeneous network. Within the German part of the convoy the IABG HiMoNN radio was used, whereas the Kongsberg WM600 radio was used within the Norwegian part of the convoy. To interconnect the two parts one German radio was placed on a Norwegian vehicle and one Norwegian radio was placed on a German vehicle.

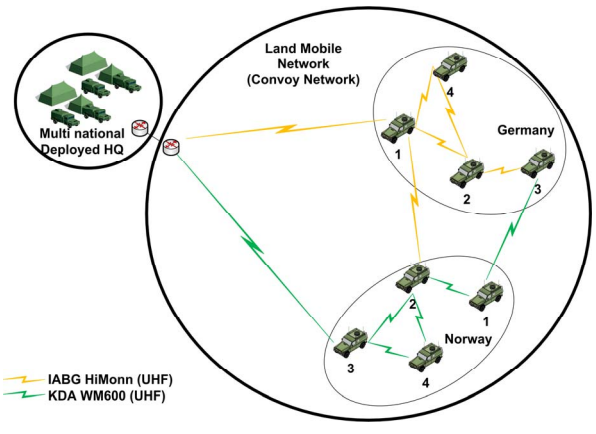

Figure 1: CoNSIS network, initial configuration.

In addition to the vehicle nodes, one mobile network node was physically co-located with the deployable HQ network. This node connected the mobile network to the rest of the CoNSIS network, and was connected to mobile nodes terrestrial radio links as indicated.

The eight vehicles formed two convoys, one German and one Norwegian. In the scenario, these two convoys were separated from the start, but at a later point in time, they merged into one large convoy. In addition, the network topology changed during the experiments, something that affected delay and available bandwidth. One such alternative topology is shown in Fig. 2.

Germany and Norway each provided implementations of selected parts of the NNEC CES, using Web services technology as their foundation. The two implementations were technologically quite different, as the German implementation, Referenzumgebung Dienste (RuDi) [1], is based on an Enterprise Service Bus (ESB), while the Norwegian solution consisted of multiple stand-alone components implementing the set of core services.

In the experiment, the core services were used to provide a infrastructure for functional services providing three types of information:

- *Vehicle positions*: Each vehicle reported its position (through a GPS-based positioning service) to the lead vehicle of its convoy. The lead vehicle then aggregated the positions of all convoy vehicles into a convoy Common Operational Picture (COP), which in turn was delivered to the HQ, where the two convoy COPs were aggregated into a full COP. This full COP was then distributed back to the vehicles.
- *Operational messages*: This is a service for sending messages to other users. The messages can be of the types *Alert*, *Warning*, *Information,* and *Command*. File attachments are also possible.
- *Chat*: This service provides ordinary chat functionality, based on chat rooms.

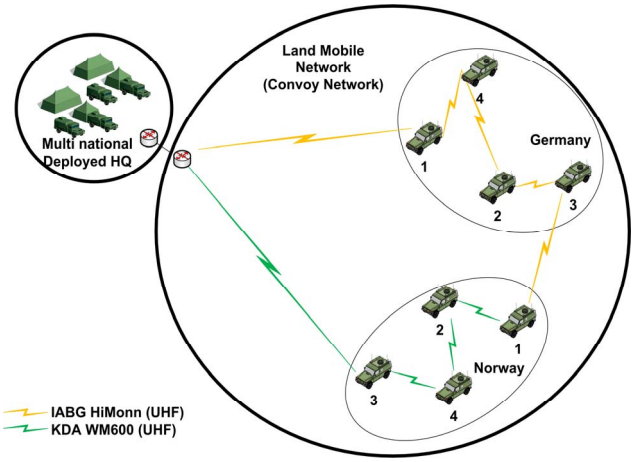

Figure 2: CONSIS network, alternative network topology.

All information types are distributed using WS-Notification, which is the Web service standard for Publish/Subscribe that is specified by the NNEC CES. Furthermore, all information types were distributed among all the vehicles as well as the HQ, as illustrated in Fig. 3.

III. SERVICE DISCOVERY

Service Discovery is the process of finding the services that are available in the network. NNEC CES has recommended the use of the registry based solution Universal Description, Discovery and Integration (UDDI) for this core service. However, when operating in a wireless network environment where node mobility and shifting network conditions can cause network partitions and loss of network connections, it is vital to use a service discovery mechanism that does not rely on the availability of any given node. In other words, we need a fully distributed service discovery mechanism [3]. The only standardized Web service discovery protocol that currently fulfills this requirement by operating in a distributed mode is WS-Discovery [4], which was utilized during the SOA experiments.

WS-Discovery is designed for use in one of two modes: *managed* and *ad hoc*. In managed mode all nodes communicate through a discovery proxy, an entity which performs the service discovery function of behalf of all the other nodes, and which communicates with the other nodes using unicast messages. This mechanism can be used to achieve interoperability between registry based service discovery mechanisms and WS-Discovery.

In ad hoc mode, on the other hand, communication is fully distributed. Requests for service information are sent using multicast to a known address, and each node is responsible for answering requests from others about its own services. The ad hoc mode is intended to be used for local communication only, and the standard recommends limiting the scope of multicast messages by setting the time-to-live (TTL) field of the IPv4 header to 1, or by using a link-local multicast address for IPv6.

The CoNSIS experiment consists of a number of ad hoc networks connected to each other using Multi-Topology Routers (MTRs) [5], forming a IPv6 based network-of-networks. The dynamic character of these networks implies that one cannot rely on a managed mode discovery proxy to remain available, meaning that the distributed ad hoc mode should be used. However, since this mode is limited to link local communication it will not provide the multi-network service discovery capability required in the CoNSIS experiments. In order to work around this issue, we decided to go against the recommendations in the standard, and allow the multicast discovery messages to travel across network boundaries by using a site-local IPv6 address, and increasing the Hop Limit in the IPv6 header. This solution works within a controlled network environment such as the one used during the CoNSIS experiments, but it is less than ideal for use in larger scale networks. That is because increasing the scope of the multicast messages might cause the messages to travel further than intended, and thus cause increased network load in networks where the messages are not needed.

WS-Dicovery is a hybrid discovery protocol, meaning that it has both a *proactive* and a *reactive* element (see [3] for further details on the different types of discovery protocols). The proactive element consists of the HELLO and BYE messages nodes send out when they first make a new service available, and when they remove a service, respectively. Other nodes then store the information gathered from these proactive messages, allowing them to perform service discovery without having to actively query for information. This proactive mode works well under stable network conditions, since the likelihood of these messages reaching all other nodes is high. The CoNSIS network is however not stable, which means that many of these messages will be lost, rendering the proactive element of WS-Discovery unable to provide service information to all nodes. This means that one has to rely on the reactive element of WS-Discovery, the PROBE and PROBE MATCH messages.

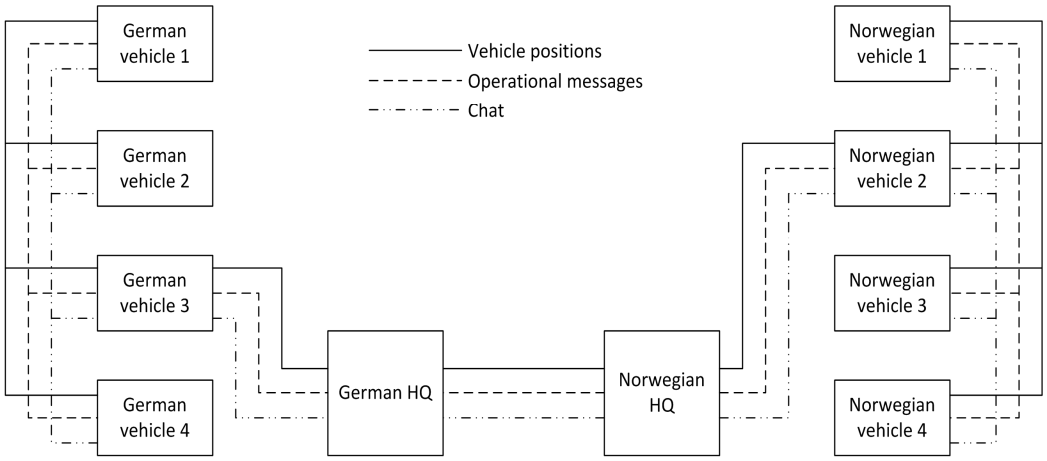

Figure 3: Information types and flows

In this reactive mode a node that requires the use of a service will ask for services matching its needs by sending a PROBE message. This message is sent using multicast, and with the extended scope of multicast messages described above, the probe will reach all other nodes that it currently has a network connection to. Nodes offering a matching service send a unicast PROBE MATCH message back to the probe sender. Note that this reactive mode should be used sparingly in low capacity networks as it generates some network traffic.

The flow of WS-Discovery multicast messages is illustrated in Fig. 4. Since we allowed the packets to flow across routers, a request sent by any one node in the network is received by all other nodes. If the message sent was a probe for available services, then all nodes that did offer a service matching the request would reply with a unicast message to the sender.

On the German side, WS-Discovery was integrated into the RuDi system, and connected to the internal service registry. This meant that any announcement made on WS-Discovery would be added to the service registry, which in turn meant that the announced service could be invoked from within RuDi. This was done by allowing RuDi to periodically probe for information, but at a low enough frequency so as not to overload the network. On the Norwegian side, WS-Discovery was used as the only discovery mechanism. A self-contained WS-Discovery application was therefore used for announcing and searching for services, which made it possible to limit the amount of probes sent into the network by only probing when a new service was needed.

As mentioned above, allowing the multicast packets to traverse routers is not an ideal solution. An alternative is to combine the managed and ad hoc modes in one deployment. When a WS-Discovery proxy announces its presence, all other nodes are asked to enter managed mode, relying on the proxy for service discovery. However, the WS-Discovery specification does not require the nodes to change to managed mode, and by allowing the majority of nodes to remain in ad hoc mode and at the same time keep a link local message scope, one can secure local service discovery without the risk of generating unneeded network traffic in other networks. Combined with discovery proxies that function as relays between the networks, cross-network discovery can be achieved as well.

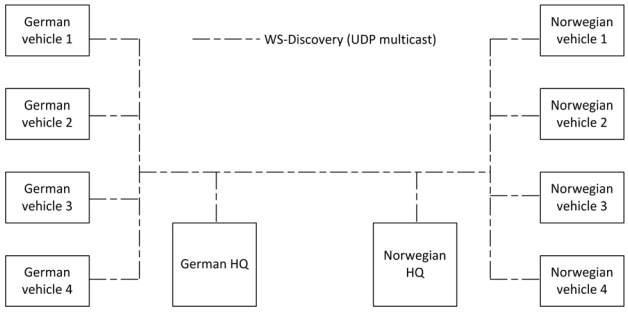

Figure 4. WS-Discovery information flow

Note that, even though the WS-Discovery specification does allow nodes to choose not to enter managed mode when receiving a message telling it to do so, it does not clearly state what the expected behavior of nodes is once the network consists of nodes in both modes simultaneously. This combination of modes is desirable when working with multiple interconnected mobile networks, and therefore a profile of how to use the WS-Discovery standard in this context should be developed by NATO for interoperability between nations.

IV. PUBLISH/SUBSCRIBE

In the CoNSIS experiment the majority of the information exchanged was distributed according to the publish/subscribe paradigm. This means that instead of a node having to repeatedly check if there is new information, the node simply send a *subscription request* to the information provider, asking to be notified whenever new information is available (see Fig. 5). Using Publish/Subscribe instead of the Request/Response paradigm has several advantages: The network traffic is reduced, since the client doesn't have to send periodic requests; the server load is reduced, since there are fewer requests to process; and the client will potentially receive new data sooner, although this is dependent on the request frequency in a Request/Response setting (which in turn will affect network and server load).

WS-Notification [6] is an OASIS standard and consists of a group of specifications that enable Publish/Subscribe-based communication between Web services. It comprises WS-BaseNotification, WS-BrokeredNotification and WS-Topics. While WS-BaseNotification defines which interfaces consumers (clients) and producers (servers) should expose, WS-BrokeredNotification introduces the concept of a message broker, an intermediary node which decouples consumers and publishers, and relieves producers from several tasks associated with Publish/Subscribe. NNEC CES specifies WS-Notification as the standard to be used for Publish/Subscribe in NATO.

The notifications are normally always of the same type, independent of the actual information that is delivered (i.e., the payload of the notification). When a client wants to subscribe to a specific type of data, it therefore expresses the type of information it is interested in by including a *topic* in the subscription request.

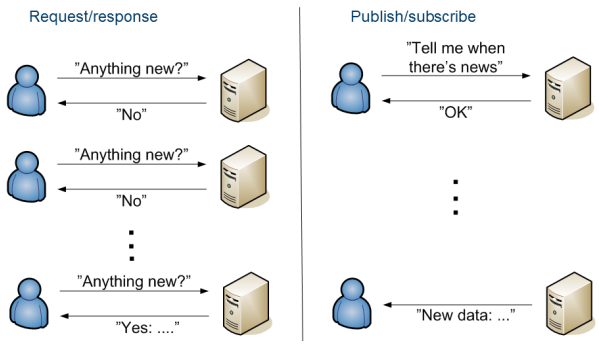

Figure 5. Request/Response versus Publish/Subscribe

For WS-Notification, the WS-Topics standard specifies how such topics should be expressed. It also defines three topic expression dialects, to allow for expression topics of different complexity. This use of topic dialects means that one can express a number of different topic structures within the same standard, including define one's own dialect for handling topics. This topic handling scheme is flexible, but the added complexity using such topics means that one needs to agree not only on which topics to use, but also which dialect they want to use to express topics before communication is possible.

One added complexity when using WS-Notification in a limited capacity network it that the standard is designed to use unicast message transmission only. That means that, even when multiple nodes in the same network want the same information, WS-Notification will send one unicast message to each recipient rather than send one multicast message that reaches all recipients. In radio based networks, where the transmission medium is shared, there is a potential for a significant reduction in network load by switching from unicast to multicast. Note that making such as switch will require further functionality to be implemented into WS-Notification, namely the ability to manage multicast group memberships.

A. Norwegian Infrastructure

The Norwegian infrastructure used during the CoNSIS experiments consists of a number of internally developed components. The core of the infrastructure is an implementation of the core features of WS-BaseNotification and WS-BrokeredNotification. This Java implementation has support for subscribing and unsubscribing, as well as for receiving and sending notifications. While not being a full implementation of the WS-Notification standard, this light-weight solution is well suited for use in test environments like the CoNSIS network. During the experiments all the Norwegian units were running a notification broker, and all clients and services on a node subscribed to, respectively delivered notifications to, its local broker. Between nodes, the brokers subscribe to each other.

Web services, including WS-Notification, normally relies on HTTP over TCP for message delivery, meaning that it is necessary to establish an end-to-end TCP connection between client and service. In the CoNSIS network this means that TCP connections in many cases would have to be established across several radio networks with unstable links and often very high delays. To enable standards-based Web services over such connections, we used our Delay and Disruption Tolerant SOAP Proxy (DSProxy) [7]. These proxies constitute a middleware that hides network delay and disruptions from the applications and also compresses all traffic, allowing XML to be sent over low bandwidth connections.

B. German Infrastructure

The German infrastructure consists of a complete SOA environment, RuDi, covering a range of functionality in addition to the aspects described here. For a more elaborate description of RuDi, including the German national security experiments conducted during CoNSIS, refer to [1].

In order to connect the Norwegian and German infrastructures together, ensuring reliable message delivery in an unstable environment, the DSProxy was used. RuDi supports the use of multiple transport protocols at the same time, and by including the DSProxy as one of these transport options, connectivity between the Norwegian and German infrastructures was achieved.

C. Experiment Information Flow

In addition to these infrastructure components, each vehicle had a GPS component that reads the vehicle's position from a GPS, creates an NFFI message, and delivers this as a notification to the local broker. Furthermore, there was a component for creating Operational Messages, and delivering these as notifications to the local broker. There was also a chat component, which both subscribed to, and delivered notifications to, the local broker. Next, there was an aggregator function that subscribed to the position of each vehicle, and then combined all vehicle positions into one NFFI message which was then delivered to the local broker as a notification. Finally, there was a viewer application, which subscribed to NFFI tracks and Operational Messages from its local broker, and displayed them on a map (see Fig. 6).

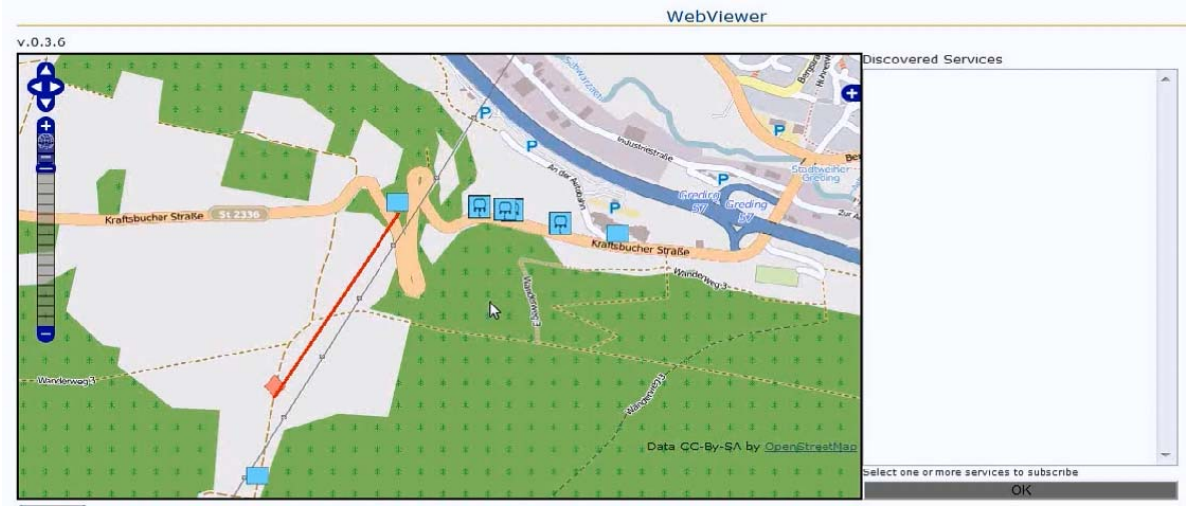

Figure 6. The viewer component

As mentioned earlier, the COP was built in two steps, with the lead vehicle of each convoy building a convoy COP by subscribing to the positioning service of each vehicle, and then the HQ building the full COP by combining the two vehicle COPs. This is illustrated in Fig. 7, and represents the initial flow of position information.

Later in the scenario, the two convoys merged into one. However, since the communication between the two convoy elements was done via the HQ, information flow between them was prone to disruptions and delays even when the two convoy elements were traveling together. In order to improve upon this situation, and take advantage of the higher capacity network connections now available within the convoy, we then changed the information flow: The two lead vehicles stopped subscribing to the full COP from the HQ, and instead started subscribing to each other's vehicle COPs. The full COP could then be built locally at each lead vehicle, and then distributed to the other vehicles in the convoy (see Fig. 8). Due to the flexibility of WS-Notification this change in information flow was easily performed, with the added benefit of both an improved response time for positional updates within the convoy, and less traffic load on the narrow reach-back links.

Thus, because the notification interface is the same, regardless of information type, any subscriber can subscribe to and receive notifications from any broker, as long as the business logic behind is able to parse the payload of the notification.

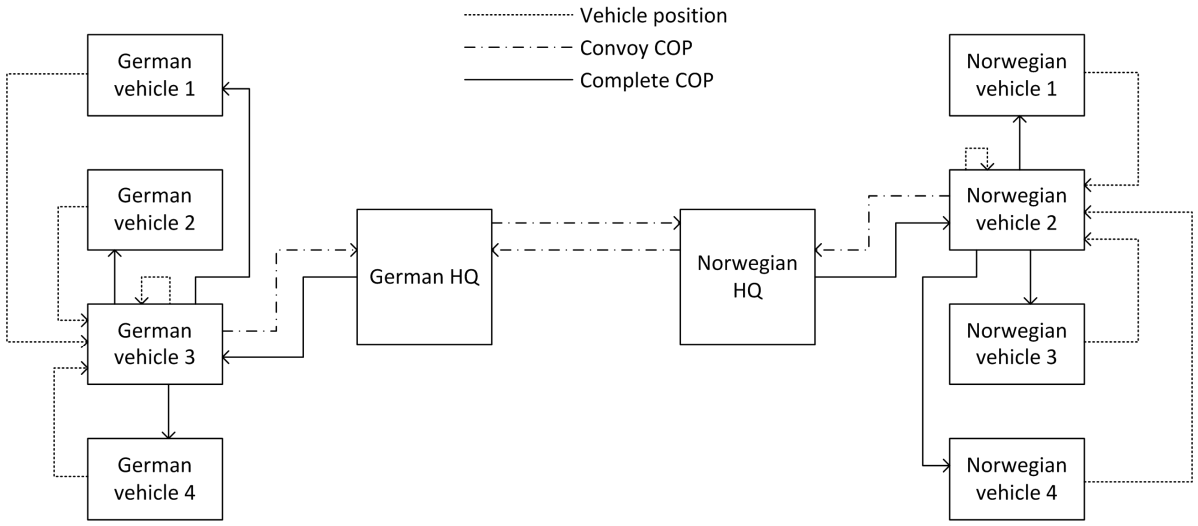

Figure 7: Initial configuration of flow position information

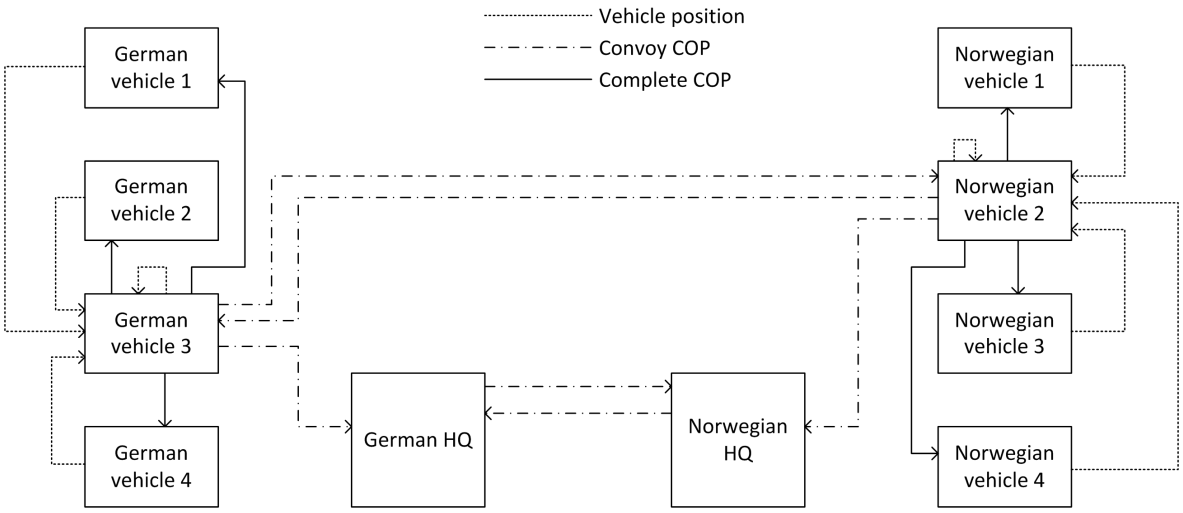

Figure 8: Flow of position information after merging of convoys

V. TOPIC HANDLING

All Publish/Subscribe systems require a mechanism for describing content of interest, and WS-Notification uses topics for this purpose. A topic is a way of classifying content into logical channels, and topics are usually organized into hierarchies. Thus, the highest level topic, the root topic, represents the most general classification, and then an arbitrary number of subtopic levels refine this classification.

This organization of information flows based on topics is fundamentally different from the Request/Response paradigm: When looking for a Request/Response service you are interested in a service with a particular interface. This is because that interface is the only aspect of the service that is known to the consumer, and thus represents the only interface that consumer is about to invoke. The service description for a service, the WSDL file, does not contain information about the actual content provided by the service.

For Publish/Subscribe, on the other hand, all services are equal with respect to the actual interface, and you need information about the content the service offers in order to distinguish between content providers. A consequence of this transition from Request/Response to Publish/Subscribe is that traditional service discovery becomes less useful. This is because all Publish/Subscribe endpoints will appear as the same service type, generating a need for additional meta-information about services, namely the topics. This shift in interest from service types to information types makes *topic discovery* an important issue when dealing with WS-Notification.

In [8] we have described how WS-Discovery can be used to distribute information about which topics a service covers, while at the same time remaining backwards compatible with the WS-Discovery standard. As a preparation for the CoNSIS experiment, initial testing with topic discovery was performed at the Coalition Warrior Interoperability Experiment (CWIX), where WS-Discovery with topic support was tested by multiple partners. During this initial testing, as well as during the CoNSIS experiment execution, we discovered that while this approach provides enough information for nodes to be able to distinguish between the content offered by the different providers, certain extra functionality is desirable.

In particular for notification brokers (as described in the previous section), which can serve many nodes and offer information on many different topics, it would be very useful to be able to query the broker itself about topics: For instance, which topics a broker currently provides notifications on, which topics it knows about (i.e., has seen at some point), and if and when it has last seen notifications on a given topic.

One challenge when working with topic based information exchange is that it requires all the involved parties to have prior knowledge about how topics are organized. In order for an information consumer to get the information it desires, it needs to know in advance which topic to request from the broker. In the CoNSIS experiment we were working with two partners, making a priori distribution of topic information possible. We decided that a client normally needs information following a given schema, and we therefore chose to have a 1:1 relationship between root topic and the XML Schema of the information in question. Thus, we had the root topics “nffi”, “OpMsg” and “Chat”. However, in other contexts, other classifications may be better suited.

In general, for larger scale implementations of topics, it is necessary to utilize a common information model that describes how topics are organized. This means that NATO should be the driving force behind such a model, which would then be used by all member nations.

VI. CONCLUSION

Performing practical experiments with the technologies that will form the foundation of future operational networks is vital to ensure that these technologies will be capable of meeting the interoperability requirements of complex operations. During the CoNSIS experiment we had the opportunity to test Web services in a complex network, allowing us to verify that Web services can be used as an interoperability enabler also in limited capacity tactical network. Using the Web service standards as the common reference between nations made interoperability possible, but there is a need for further development and profiling of standards in order for them to fully support the interoperability challenges faced by the nations.

Due to the potential performance benefits of using the Publish/Subscribe paradigm in tactical networks, use of the WS-Notification standard is recommended. To be able take full advantage of the benefits of Publish/Subscribe however, multicast support for notification should be implemented. In addition, topic handling must be addressed, preferably by introducing a NATO recommendation addressing both the issue of incompatibilities between different topic expression dialects, and containing a common topic vocabulary and structure.

REFERENCES

[1] H. Seifert and M. Franke, "SOA in the CoNSIS coalition environment", IEEE MCC, Gdansk, Poland, 2012, in press.

[2] F. T. Johnsen, T. H. Bloebaum, L. Schenkles, J. Śliwa, and P. Caban, “SOA over disadvantaged grids experiment and demonstrator”, IEEE MCC, Gdansk, Poland, 2012, in press.

[3] F. T. Johnsen, T. Hafsøe, and M.Skjegstad, ”Web services and service discovery in military networks”, 14th ICCRTS, Washington D.C, US, June 2009.

[4] V. Modi and D. Kemp (eds.), “Web services dynamic discovery (wsdiscovery) version 1.1,” http://docs.oasis-open.org/ws-dd/discovery/1.1/wsdd-discovery-1.1-spec.pdf, July 2009

[5] M. Hauge, J. Andersson, M. Brose, J. Sander, “Multi-topology routing for QoS support in the CoNSIS convoy MANET”, IEEE MCC, Gdansk, Poland, 2012, in press.

[6] OASIS, Web services Notification TC, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsn

[7] K. Lund, E. Skjervold, F. T Johnsen, T. Hafsøe, and A. Eggen, ”Robust web services in heterogeneous military networks”, IEEE Communications Magazine, October 2010

[8] F. T. Johnsen and T. H. Bloebaum, “Topic discovery for publish/subscribe web services”, IEEE IWCMC, Limassol, Cyprus, August 2012, in press.