Transportation Security SensorNet: a service-oriented architecture for cargo monitoring

Martin Kuehnhausen and Victor S. Frost
Department of Electrical Engineering and Computer Science, University of Kansas, Lawrence, Kansas, USA

Received 24 December 2010
Revised 16 March 2011
Accepted 14 May 2011

Abstract

Purpose – Security and accountability within the transportation industry are vital because cargo theft could amount to as much as $60 billion per year. Since goods are often handled by many different parties, it must be possible to tightly monitor the location of cargo and handovers. Tracking trade is difficult to manage in different formats and legacy applications Web services and open standards overcome these problems with uniform interfaces and common data formats. This allows consistent reporting, monitoring and analysis at each step. The purpose of this paper is to examine Transportation Security SensorNet (TSSN), the goal being to promote the use of open standards and specifications in combination with web services to provide cargo monitoring capabilities.

Design/methodology/approach – This paper describes a system architecture for the TSSN targeted for cargo monitoring. The paper discusses cargo security and reviews related literature and approaches. The paper then describes the proposed solution of developing a service-oriented architecture (SOA) for cargo monitoring and its individual components.

Findings – Web services in a mobile sensor network environment have been seen as slow and producing significant overhead. The authors demonstrate that with proper architecture and design the performance requirements of the targeted scenario can be satisfied with web services; the TSSN then allows sensor networks to be utilized in a standardized and open way through web services.

Originality/value – The integration of SOA, open geospatial consortium (OGC) specifications and sensor networks is complex and difficult. As described in related works, most systems and research focus either on the combination of SOA and OGC specifications or on OGC standards and sensor networks. The TSSN shows that all three can be combined and that this combination provides cargo security and monitoring capabilities to the transportation and other industries that have not existed before.

Keywords Telemetry, Freight forwarding, Security, Tracking, System monitoring, Intermittently connected wireless networks, Communication system software, Data communication, Software engineering

Paper type Research paper

1. Introduction

The theft and tampering of cargo are common problems in the transportation industry. According to Wolfe (2004), the "FBI estimates cargo theft in the USA to be $18 billion" and the Department of Transportation, "estimated that the annual cargo loss in the USA might be $20-$60 billion". Wolfe (2004) also gives good reason to believe that the actual number may be even higher than $100 billion because of two reasons. First, it is assumed that about 60 percent of all thefts go unreported and second, the indirect costs associated with a loss are said to be three to five times the direct costs. With the advances in technology, this problem has evolved into a cat-and-mouse game where thieves constantly try to outsmart the newest cutting edge security systems.

Journal of Systems and Information Technology
Vol. 13 No. 4, 2011
pp. 369-388

1328-7265
DOI 10.1108/13287261111183979

In terms of securing cargo, there are usually two aspects: first ensuring the physical safety of the cargo and second monitoring and tracking it. There has been increasing interest in the latter because many shipments cross national borders and cargo may be handled by a multitude of carriers. All of this leads to a demand for tracking and monitoring systems by the cargo owners, carriers, insurance companies, customs, and many others.

This paper is part of a series that describe the design, various components and conducted experiments of the Transportation Security SensorNet (TSSN). An extended version can be found in a thesis by Kuehnhausen (2009). In this paper, we focus on the software architecture and refer to papers that deal with the other parts of the TSSN; specifically, Fokum *et al.* (2010) give an overview of the hardware utilized and describes in detail truck trials and a short haul train trial. Kuehnhausen and Frost (2010a) present a new and flexible approach to deal with challenges such as intermittent and low-bandwidth communication in mobile monitoring environments and a long haul train trial in Mexico. Furthermore, Kuehnhausen and Frost (2010b) discuss a framework for analyzing and visualizing simple object access protocol (SOAP) messages to overcome the challenges of complexity and disparity that web service monitoring and management approaches face. Security associated with the TSSN and specifically issues that arose when integrating elements from the web services architecture (WSA) led by the world wide web consortium, specifically publish/subscribe communication and service security are described in Komp *et al.* (2010).

Here, an architecture is introduced which builds on open standards and software components to allow "monitoring cargo in motion along trusted corridors". The focus lies on the use of a service-oriented architecture (SOA) and geographical information system (GIS) specifications such as from the open geospatial consortium (OGC) in order to allow an industry wide adoption of this open framework.

In the following we discuss the problems of proprietary systems, the advantages of open standards and the approach of using a SOA in the transportation industry. We introduce the design and architecture and explain the individual components as well as the software components and specifications that are used in the implementation.

The discussion of proprietary systems in contrast to open standards in the following section provides an overview of the challenges that trade and shipping partners face. It explains why it is important to design an open system that is based on standards. Some of the main advantages are a decrease in cost, more efficient shipment management, and enhanced visibility and tracking capabilities. This paper presents the architecture of the TSSN that was implemented to show that such an open system can be built and deployed successfully.

2. Problem area

In order to address the problem of cargo security, research on the TSSN has been conducted. Its goal is to promote the use of open standards and specifications in combination with web services to provide cargo monitoring capabilities. The main question is the following:

> How can a SOA, open standards and specifications be used to overcome the problems of proprietary systems that are currently in place and provide a reusable framework that can be implemented across the entire transportation industry?

The main aspects of this question are discussed next.

2.1 Proprietary solutions

Current commercial systems in the transportation industry are often proprietary. This is because a lot of effort is spent on research and development in order to create intellectual property. The assumption is then that as long as the competitors do not have access to the system and its protocols that intellectual property is safe and provides a competitive advantage. Another common "benefit" of keeping the systems closed is the perceived additional security since in order to successfully attack the system its implementation and protocols have to be reverse engineered.

The problem with this is that these advantages are often one-sided and lead to stove pipe systems provided by a single vendor. Once a proprietary system has been implemented it has to be maintained. What happens if a customer that uses the system invested a lot of money into a its infrastructure and the training of its employees and the company that provides the system releases a new version of it which of course costs money again.

The main point here is that many customers are locked into proprietary solutions that are incompatible with similar solutions offered by competitors. In a 2003 survey by the Delphi Group (2003), it was found that 52 percent of developers and 42 percent of consumers see standards enabling the "approval of projects otherwise threatened by concerns over proprietary system lock-in". Furthermore, an overwhelming 71 percent of developers and 65 percent of consumers feel that the use of open standards "increases the value of existing and future investments in information systems".

The problem of non-interoperability with regard to geospatial processing is the topic of a paper by Reichardt (2004). Because GIS are often immensely complex, companies that invest heavily into this area often only support their company's product.

2.2 Problems with some standard solutions

The idea of open standards and specifications is to define interfaces and protocols that can be used as references for the implementation of a system. There are many standards committees and industry groups, most often focused on a particular area.

The importance of open standards is emphasized in a paper by McKee (2005). It provides the evolution and success of the internet as the "perfect example" for the use of open standards. In particular, it explains that since the internet is based upon communication and communication means "transmitting or exchanging through a common system of symbols, signs or behavior", the process of standardization can basically be seen as "agreeing on a common system". The other parts of the paper are focused on how openness can help GIS but many of the points mentioned apply to open standards in general.

In particular, the following aspects are associated with open standards:

- compatibility;
- freedom of choice;
- interoperability;
- leverage; and
- open source.

However, there are several problems that can be associated with non-proprietary systems. Implementations are based upon the interpretation of the standards which may differ significantly. Furthermore, some implementations only support a subset of the original specification, may be slower than the reference implementation or use incompatible sub systems.

2.3 Issues in the transportation industry

According to Irmen (2009), automation and efficient communication with partners are the two most important functions in supply chain management which represents the core of the transportation industry. Let us take a look at how the SOA addresses both of them in regard to the individual topics outlined in Irmen (2009).

Automation. A vital part in transportation is the screening process. Companies that transport goods must ensure safety and, therefore, check all parties involved in the trade. An important aspect of this is the use of a denied trade list which lists items and companies that are not allowed to import or export into specific countries. With the reduction in manual labor and transition to a web services-based system that automatically performs these checks, efficiency could be greatly increased.

A closely related topic is accountability. Who is responsible if something goes wrong during the trade process? Since goods are often handled by many different parties, it must be possible to monitor the location of cargo and handovers tightly. This is especially important in cases of tampering or even theft of the cargo.

Furthermore, agencies and customs more and more require electronic trade information instead of paper documents in order to track trade. Because of different formats and legacy applications that are often unable to provide this information in its entirety, additional resources have to be allocated in order to remain compliant with current practices. Web services and open standards can overcome this problem with uniform interfaces and common data formats.

Having the ability to monitor the location not just for perishable goods but also for high value goods is of great importance in the transport chain. Current processes should be able to automatically route cargo based on its needs and cost effectiveness.

Efficient communication. Building a virtual network among the parties involved in the trade process establishes efficient means of communication. It allows the coordination between otherwise disparate entities that is essential to provide cost effective and reliable shipping of cargo. The internet provides the communication layer but it is the standards of web services that enable the integration of different systems.

Information security within the transportation industry plays a big role because trade data is to be kept confidential and only distributed on a need-to-known basis. This puts an additional burden on the parties that are involved, as the parties must exchange data confidentially at each point of interaction. If open standards are used for this, information security is implemented based on interfaces and policies that are easy to manage.

In order to manage the transportation chain in its entirety, a global view is often needed. This is problematic since individual parties often only deal with their respective neighbors. Using open standards and the SOA approach each party could provide an uniform information interface that is accessible to other parties in the chain. This allows consistent reporting, monitoring, and analysis at each step during the shipping process.

3. Related work

In the following sections, related work that is relevant to various aspects of the TSSN such as SOA, web services, communication models, the OGC specifications and sensor networks are analyzed.

3.1 Microsoft – an introduction to web service architecture

Cabrera *et al.* (2004) outline concepts that led to the implementation of SOA and development of the web services specifications that surround them and are used by the TSSN. A lot of the main approaches have been standardized in various committees and organizations by now but were only in the early stages when Cabrera *et al.* first discussed them.

3.2 Service-oriented architecture

The concept of information processing and sharing across various applications using web services is the main focus of this paper. The basic idea is to define components of a system as services and users as clients that can retrieve data from them. Note that interaction between services is done using embedded clients. The services take care of functions such as information processing, data analysis and storage. With all business logic embedded into services and interaction between them clearly defined using open standards an infrastructure is built that is called SOA. Through these web services companies, government agencies, and others have the ability to share and process information in a uniform manner which cuts costs, time, and resources and improves efficiency.

3.3 Enterprise service bus

An adobe technical paper by Nickul *et al.* (2007) outlines general architecture approaches that can be taken when transitioning business processes to the SOA. It mentions a widely used technology called the enterprise service bus (ESB) that provides a standardized means of communication for all services that connect to it. For the TSSN, this is of importance when it comes to asynchronous communication as the JAVA message service (JMS) uses queues that are on the ESB for message exchanges (Kuehnhausen and Frost, 2010a).

In addition to the basic request-response, several other message exchange patterns that go beyond the standardized ones are described. A registry keeps track of service metadata. The service provider is responsible for updating it whenever a change occurs and the service consumer subscribes to the registry for any of these changes. The metadata that is provided is then used to configure a service client. Hence, the client can issue requests and receive responses.

The TSSN essentially uses a very similar approach with universal description discovery and integration (UDDI). Web services automatically register with the UDDI when they are started and clients are able to use specific services by looking them up in the UDDI.

3.4 Open sensor web architecture

An approach to implement the proposed standards of the sensor web enablement that are described in Botts *et al.* (2007) is outlined by Chu *et al.* (2006). The system is called NICTA open sensor web architecture (NOSA) and is focusing on the combination of sensor networks and distributed computing technologies.

The TSSN uses a similar approach but has some significant differences. The goal of both implementations is to integrate a sensor network into a WSA using open standards. NOSA uses a sensor application that is tightly integrated into the sensor operating system and then provides sensor data and control to web services in a non-standard format. TSSN on the other hand implements sensor management and monitoring functionality inside a single service, the sensor node (Section 4.2.1) and allows different sensors to be "plugged in". This allows other services to use standard web service interfaces and SOAP messages in order to access sensors.

Furthermore, the web services used by NOSA are implemented manually according to the OGC specifications which causes them to be limited as not everything that is specified is also implemented. In contrast, the TSSN uses automatic code generation (Section 4.1.1) that enables it to use all OGC specifications. Since their elements and interfaces are generated the only thing that has to be implemented is functionality. This approach significantly reduces development efforts.

3.5 Electronic freight management

The electronic freight management (EFM) initiative by Fitzpatrick *et al.* (2006) is a project that focuses on the improvement of communication between supply chain partners using web technologies. One of the main goals is to provide a common and open technology platform for sharing cargo information among smaller and medium size trade partners. The idea is that the information is only entered once and then shared among members of the supply chain.

EFM because of its SOA approach provides a common electronic communication platform that maintains cargo related information on a web service basis. This information is then shared with authorized users while digital certificates and web service security ensure data integrity and confidentiality. The key benefit here is the improved visibility of shipment information which enables all supply chain members to perform their processes more efficiently and plan ahead better (Troup *et al.*, 2009). The data exchange is standardized and based on the universal business language. Furthermore, each individual transaction is uniquely identifiable by a unique consignment reference.

The TSSN approach is similar but deals in particular with cargo monitoring in mobile environments. The trade data exchange (TDE) as described later is responsible for managing and sharing shipment information.

3.6 Globus – open grid services architecture

Globus is an architecture that is based on grid computing. It focuses on providing capabilities as services in a grid environment using standard interfaces and protocols. An initial paper by Foster *et al.* (2002) gives an overview of the architecture and design decisions. In particular, Globus supports "local and remote transparency with respect to service location and invocation" and "protocol negotiation for network flows across organizational boundaries". Its service approach is similar to the SOA that is used by the TSSN. Additionally, security concepts that work inside a grid are applicable to SOA and vice versa.

In contrast to the TSSN, Globus makes use of web service specifications in some of its components but also provides custom implementations and interfaces as for service discovery and notifications. The TSSN uses web services specifications and OGC standards almost exclusively which ensures standards compliance and compatibility. For service discovery the UDDI, see Bellwood *et al.* (2002), is used and for notifications WS-Eventing, see Box *et al.* (2006).

3.7 Service architectures for distributed geoprocessing

A research paper by Friis-Christensen *et al.* (2007) outlines the implementation of an application that analyzes the impact of forest fires using web services. The main focus is the transition from a client application to a flexible WSA using OGC specifications. The components include multiple data sources that are made available through data access services like the web map service and the web feature service. A geoprocessing service performs the analysis of the data and provides it to a client. Furthermore, a discovery service serves as the registry for all services and their metadata.

The implementation is interesting in the sense that it exclusively uses OGC specifications which makes it compatible to other GIS. The TSSN aims to be OGC compliant as well but includes specifications that deal with sensor networks such as the sensor observation service and the sensor alert service, something that is not addressed by Friis-Christensen *et al.* (2007).

3.8 Web services orchestration

The problem of reusability of services and "next generation challenges" was addressed by Kiehle *et al.* (2007). The idea here is to increase transparency and reusability by splitting processes into smaller more reusable processes and utilizing a workflow management system called web services orchestration. This is especially important for the integration of the TSSN into systems used in the transportation industry. Its modular design and architecture allow single components to be reused and information flows to be created.

The web processing service specification describes how services can be arranged and combined into service chains that form a process. Two alternatives are commonly used in order to achieve this. A web processing service can be set up to combine and "encapsulate" other individual web services and, therefore, provide the desired abstraction. However, the best way to define workflows is using the business process execution language (BPEL). BPEL enables complex service chains to be defined without the need for custom and potentially not reusable web processing services that just "encapsulate" services.

4. Proposed solution

4.1 Overview

This section describes the architecture of the TSSN. It provides an in-depth discussion of design aspects and the implementation.

4.1.1 Structure of solution

> Service-oriented architecture is a paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains (MacKenzie *et al.*, 2006).

Building a "Service-oriented architecture for monitoring cargo in motion along trusted corridors" makes sense. According to a study by the Delphi Group (2003), companies that collaborate usually request compliance for the following standards: XML 74 percent, J2EE (Java) 44 percent and SOAP 35 percent. The architecture used for the implementation of the TSSN utilizes all three technologies by separating functionality into web services. This allows for high flexibility and is cost effective.

Haas *et al.* (2004) early on proposed various models for web service architectures. The message-oriented model focuses on message relations and how they are processed.

An approach that centers around resources and ownership is the resource-oriented model. The policy-oriented model defines constraints and focuses on security and quality of service. Ideas from all these models have been combined with the service-oriented model into what has become SOA. Of the proposed models SOA has been the most widely implemented.

An excellent overview of Java and web services is provided by Kalin (2009). Note that SOA by definition is programming language and platform independent. It is built on the basis of requests and responses and the independence of web services. The choice to use Java for the implementation was made because the TSSN is built on top of previous research on the "Ambient computing environment for SOA" by Searl (2008) which is written in Java. The main components of the TSSN are sensor management and alarm notifications. An overview of the services and relevant message exchanges is shown in Figure 1.

The TDE (Section 4.4) provides shipment, route, logistics, and relevant cargo information. It is managed externally and used by the system only through its specified interface. The virtual network operation center (VNOC) (Section 4.3) is responsible for the processing of sensor data and alarms. One of the major capabilities that it provides is alarm notification. The mobile rail network (MRN) (Section 4.2) deals with the actual management of sensors on a mobile platform, e.g. a train. Web services at the MRN capture sensor data from the sensors and "preprocess" that data. A detailed description of each individual service is provided later in this section.

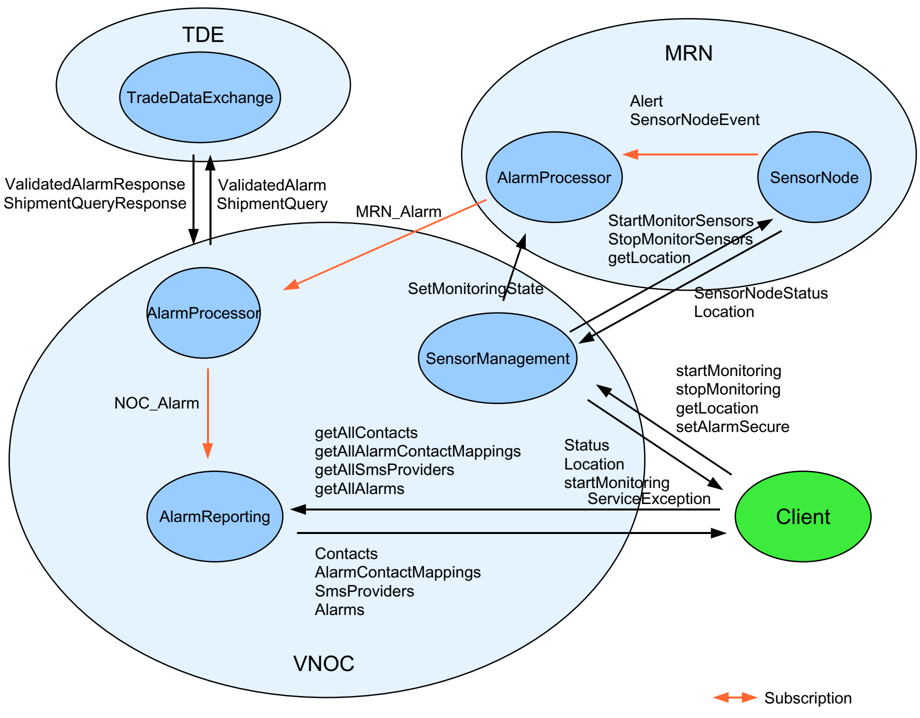

Figure 1.
Service message overview

The architecture consists of web services that are separated into service clouds. These service clouds represent the different geographically distributed locations (e.g. Overland Park, KS for the TDE; Lawrence, KS for the VNOC and on a moving train for the MRN) where services are deployed.

The web services are developed according to the web service specifications and the standards provided by the OGC. This means that they aim to be standards compliant. Since the OGC specifications are at times very complex, the geography markup language for example defines over 1,000 elements, the TSSN was implemented using custom interface definitions first and adding the OGC ones later. This enabled fast prototyping and testing of the system.

The following sections explain in-depth the approaches and technologies used in the architectural prototype and implementation of the TSSN.

Apache Axis2. is a software stack that allows the development and running of web services and clients. Its architecture as described by Chinthaka (2006) consists of the following main components:

- AXIs object model (AXIOM) for high performance.
- Extensible messaging engine for a variety of different implementations of web services.
- Context model for implementing specifications on a level basis.
- Pluggable modules to be shared among web services.
- Data binding with a variety of frameworks.

SOAP. SOAs make use of SOAP by Lafon and Mitra (2007) as a flexible message format. The TSSN does the same since web service specifications can easily be integrated and applied to SOAP messages.

WSDL. All services in the TSSN are defined using the web services description language (WSDL) version 2.0. An in-depth introduction is provided by Booth and Liu (2007). The combination of WSDL files and XML schemas make up the foundation of a web service. Utilizing the automatic code generator of Axis2 called WSDL2Java, all elements defined in the XML schemas are available as Java classes. The composition of the generated parts, data and external libraries then forms the actual service implementation (Figure 2).

4.1.2 Services. The services that are implemented in the TSSN make use of a variety of components. For long-term information storage, a MySQL database is used. A object-relational mapping tool called Hibernate (Red Hat, 2008), enables objects to be stored and retrieved transparently without the need of complicated database interactions. Esper by Bernhardt and Vasseur (2007) provides complex event and alarm processing and is used at the VNOC. The alarm processor at the MRN currently uses a less complex approach. The sensor node is responsible for the actual communication with the sensors. It must use a device specific protocol by Hi-G-Tek (2009) and a serial connection library for Java called RXTX.

Each component and its particular use is explained in the later sections when each individual service is described. At a high level, one of the main aspects when dealing with web services is the definition of whether they are stateless or stateful.

Stateless. By default web services are meant to be stateless. This is because most message exchanges are completely independent of each other. Web services usually offer calculations, information or capabilities that only require the service

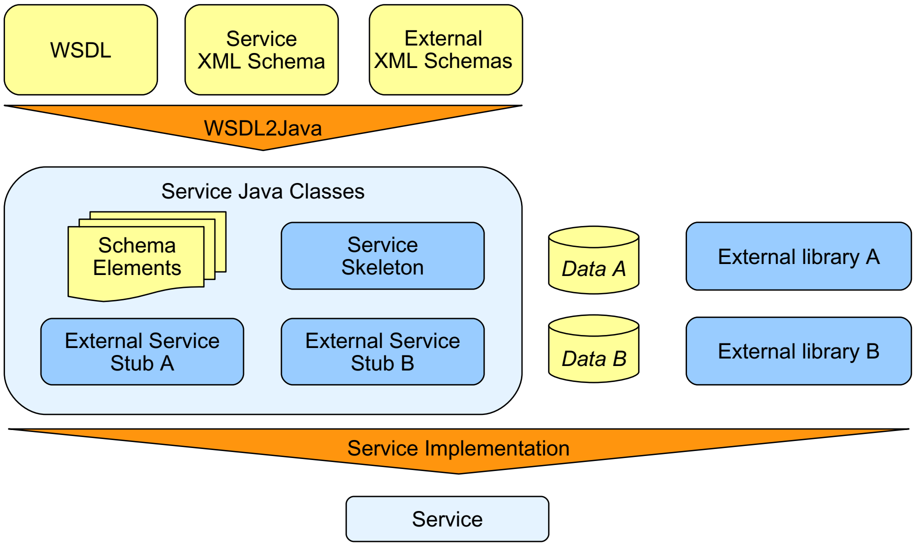

Figure 2.
Service composition

to perform a specific action and give a response. This is part of the autonomy approach of web services.

Stateful. The need for stateful web services has been identified for the TSSN because there are certain limitations in just using stateless web services. Given an online data processor that analyzes sensor data; using a stateless web service, it is impossible to react to trends and complex events because the service is limited to single data objects that it receives.

4.1.3 Clients. Clients are able to make use of the operations provided by the web services. They usually utilize the same modules as the service. This means that in theory all web services could have clients. Since a lot of the services in the TSSN interact independently from users, the number of clients that are available to users is actually smaller.

One of the aspects of clients in the TSSN is the management of the sensors. The sensor management service (Section 4.3.1) provides this among other things like retrieving the location of a particular sensor node. Another aspect is the management of alarm notifications. For this purpose the alarm reporting service defines various management operations for clients.

4.1.4 Modules. Axis2 provides the possibility to “plug in” modules that add functionality or change the way a service behaves. This allows a specific capability to be shared among different services without having to implement it in each of them. In general, the web service specifications that are used in Axis2 are implemented as modules. For more information see Section 4.1.1.

Ping. In order to check the status of a particular service Axis2 provides a module that adds an operation called pingService to a service. This can be used to check the status of either a specific operation or all operations that the service defines. The client part that actually uses this operation was not part of Axis2 and had to be implemented by the author.

Logging. Especially for debugging purposes and performance evaluations, it is of great benefit to be able to see the raw SOAP messages that are sent and received. A logging module was implemented to provide this functionality.

In terms of analyzing the TSSN and its performance the logging module was engaged in all services. Quantitive results obtained using the logging capability can be found in Fokum *et al.* (2010) and Kuehnhausen and Frost (2010a, b); a tool to visualize and animate the timing of the messages is described in Kuehnhausen and Frost (2010b).

Addressing. An implementation of the WS-Addressing specification as described in Gudgin *et al.* (2006) comes as part of the addressing module in the Axis2 core. It fully supports all components of the standard and its ReplyTo and RelatesTo fields are used among other things to allow for asynchronous communication (Section 4.1.6) in the TSSN.

Savan. The Savan module enables web services and clients in Axis2 to make use of various forms of subscription mechanisms as defined by the WS-Eventing specification by Box *et al.* (2006).

Rampart. In order to provide security according to the WS-Security specification by Lawrence *et al.* (2006) for the TSSN the Rampart module was developed by Axis2. It makes extensive use of the WS-SecurityPolicy standard described by Lawrence *et al.* (2007).

4.1.5 Subscriptions. Subscriptions are a fundamental part of the overall architecture of the TSSN. They are used by the alarm processor at the VNOC as well as in the MRN. These web services, that act as information publishers, utilize the Savan module to provide the operations defined in WS-Eventing.

4.1.6 Synchronous and asynchronous communication. By default Axis2 uses request-response in a synchronous manner. This means that the client has to wait and is, therefore, blocking until it receives the response from the service. In certain scenarios, for instance when the service needs a large amount of processing time, the client can experience timeouts. Furthermore, in the TSSN where the MRN is only intermittently connected to the VNOC, synchronous communication shows its limitations. A better option is to make the communication between services asynchronous. This resolves timeout issues and deals with connections that are only temporary.

There exist various forms of transport protocols that are suitable for asynchronous communication. Axis2 by default supports HTTP, SMTP, JMS and TCP as transports but other transports can easily be defined and plugged in. The JMS, for instance, makes use of queues which allow clients and services to store on them and retrieve messages in a flexible manner. This is essential for satellite communication which and discussed in detail in Kuehnhausen and Frost (2010a).

4.2 Mobile rail network

The MRN is a collection of services that is located on a train or in a rail yard. Its services provide the abilities to manage sensors, monitor them and propagate sensor alerts to the VNOC. This section describes them in detail.

4.2.1 Sensor node. The sensor node contains the actual sensor monitoring and management application and its components are shown in Figure 3. It provides several abstraction layers that allow various forms of sensors to be used.

4.2.2 Alarm processor. The alarm processor (Figure 4) on the MRN performs an initial filtering of sensor events generated by the sensor node. It subscribes to of all events of the sensor node, providing interfaces for generic sensor events as well as sensor alerts.

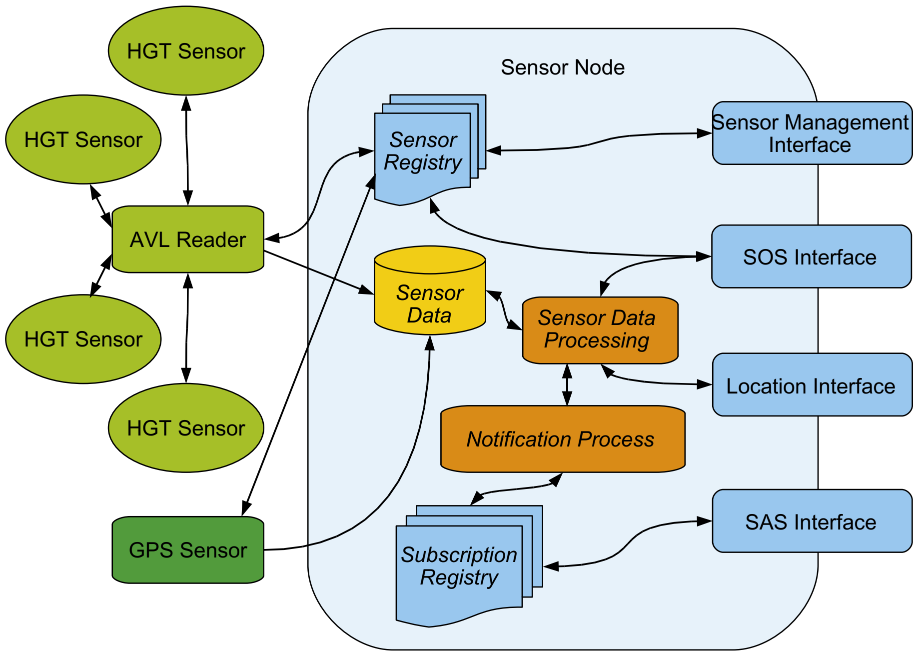

Figure 3.
Mobile rail network sensor node

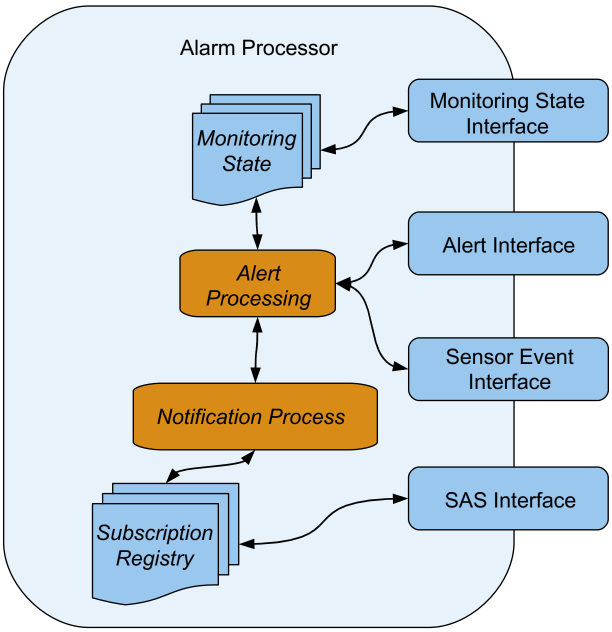

Figure 4.
Mobile rail network alarm processor

The alarm processor handles alerts and events that it receives from the sensor node and classifies them into either information or security alarms depending on its current monitoring state. It is also responsible for deciding whether or not to forward the alarm to the VNOC for further processing and possible transmission to the decision maker.

4.3 Virtual network operation center

The VNOC represents the management facility of the TSSN and consists of services that receive and process alerts received from MRN. It works with the TDE to associate shipment and trade information with a particular alert. Furthermore, the alarm reporting service provides clients with the ability to be notified upon specific events. The processes that are involved in performing these tasks are the topic of this section.

4.3.1 Sensor management. The sensor management service (Figure 5) allows the control of sensor nodes and their monitoring state. Additionally, it is able to retrieve the location of sensor nodes.

4.3.2 Alarm processor. In contrast to the "basic" processing that is performed by the alarm processor at the MRN, the alarm processor as shown in Figure 6 at the VNOC has more resources such as the associated shipment and trade information available which is provided by the TDE and can, therefore, process alarms in a more complex way. This advanced filtering and processing is done using a complex event processing system called Esper developed by Bernhardt and Vasseur (2007).

4.3.3 Alarm reporting. The alarm reporting service (Figure 7) deals with the following two aspects. First, it stores alarms long term to allow for in-depth reporting and analysis. Second, clients that want to be notified of particular alarms can register with the alarm reporting service. Whenever alarms occur notifications are sent out to the registered clients via e-mail and/or SMS accordingly.

4.4 Trade data exchange

The TDE by KC SmartPort (2008) in a sense represents a shipment and other trade data information provider. It aims to be a collection of heterogeneous systems that stores and manages the business aspects of a transport of goods. This is due to the fact

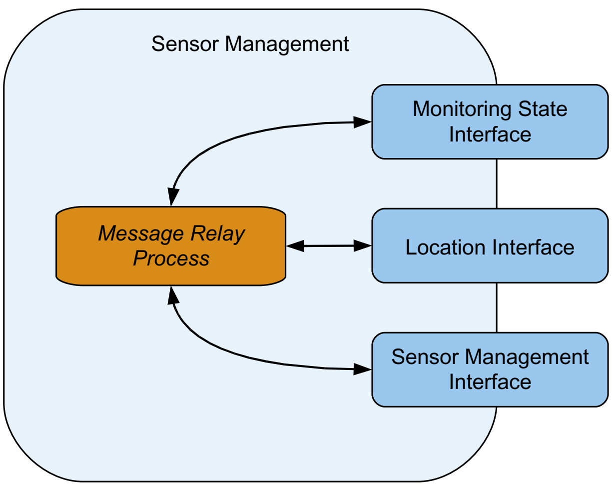

Figure 5. Virtual operation center sensor management

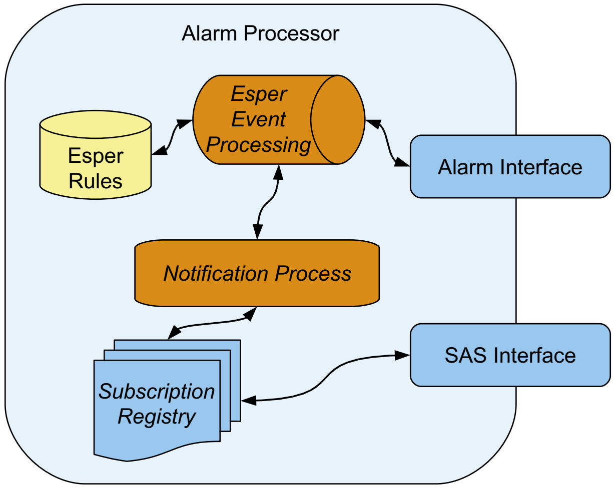

Figure 6.
Virtual network operation center alarm processor

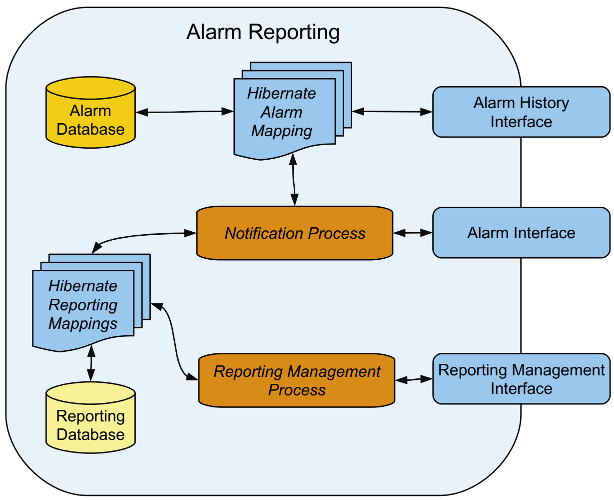

Figure 7.
Virtual network operation center alarm reporting

that there is a variety of different systems implemented by the parties that participate in the transport chain (Sections 2.1 and 2.3). Some provide route information while others manage contracts and shipment data. For the current implementation of the TSSN this "collection" of information and management services is combined into a single service, the TDE service.

The TDE service (Figure 8) interacts with the alarm processor at the VNOC. Upon request it provides shipment and trade information for a specified alarm.

4.5 Open geospatial consortium specifications

As described before, the amount of work that is required to fully implement OGC specifications such as the sensor observation service and the sensor alert service

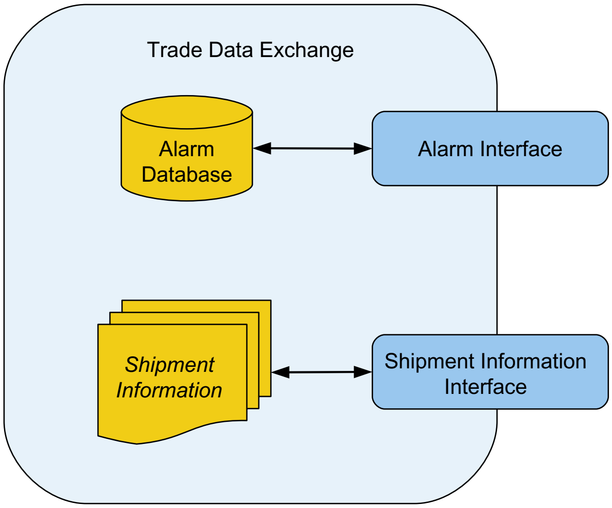

Figure 8.
Trade data exchange service

is immense. The focus of the first stage of the implementation of the TSSN is on the sensor management and alarm notification capabilities. However, at the MRN the sensor node provides an implementation for the sensor observation service as defined by the OGC. Furthermore, services in the TSSN that utilize subscriptions, in particular the alarm processor, are able to receive subscribe requests and publish alerts in a manner that is similar to the sensor alert service. The difference to the proposed sensor alert service specification is that the services that subscribe are already aware of the capabilities, sensor types, and alert types. Therefore, the operations that allow the retrieval of this information need to be implemented in order to be fully compliant.

5. Results

Several experiments were performed at various stages of the TSSN development. First, lab tests were conducted in order to ensure the functionality of the individual web services and their interactions.

Then, as described in Fokum *et al.* (2010), truck trials were completed to test basic interaction of the implemented web services and feasibility of hardware components and sensors in a mobile environment. The message exchanges between web services were correct and the system was able to recover from dropped communication links and lost GPS fixes.

A short haul rail trial was conducted after the successful completion of truck tests. Results of the short haul rail trial are found in Fokum *et al.* (2010). In this trial the MRN traveled 22 miles in five hours. The VNOC was at Lawrence, Kansas and the TDE was in Overland Park, Kansas. One of the goals was to determine the performance of the TSSN when detecting events on intermodal containers in a rail environment. Furthermore, SMS message and e-mail notification of events was investigated and data collected that could be used in the modeling of system trade-offs and communication models resulting in a systems optimization survey by Fokum and Frost (2010). The system performed well: the time it took from detecting an event to generating an alert was about two seconds and the average delivery time of the alert was about 12 seconds.

This is well within the bounds of the requirements of the transportation industry for efficient tracking and monitoring of cargo. Note that for the short haul trial a GSM communication link was used that proved to be stable and reliable. This allowed the web services to interact synchronously with each other.

Enhancements made to the TSSN to work in low bandwidth mobile bandwidth limited and intermittently connected monitoring environments are described in Kuehnhausen and Frost (2010a). In particular, the communication link between the MRN and the VNOC was changed to a dial-up satellite connection and the web services were adapted to utilize a distributed queuing approach and hence communicate asynchronously. The viability of these adjustments was tested in a long haul rail trial in Mexico. Again the TSSN worked well and was able to transmit messages in about 12 seconds whenever connectivity was established. In case the satellite link was down and needed to be established it took about ten minutes on average to deliver messages from the MRN to the VNOC. However, the average case of about seven minutes per message transmission through the system is found to be in range of mobile monitoring environments.

6. Lessons learned

The implementation of the TSSN using the combination of SOA, OGC specifications and sensor networks works. Testing has been completed in a lab environment as well as in the real world and TSSN was evaluated in Fokum *et al.* (2010) and Kuehnhausen and Frost (2010a, b). The complete system provides a web services-based sensor management and alarm notification infrastructure that is built using open standards and specifications. Particular functionality within the system has been implemented in web services that provide interfaces according to their respective web service specifications.

Using standards from the OGC allows the integration of the system into GIS. Although not all the interfaces are fully implemented as of summer 2009, the basic sensor observation service and sensor alert service are. Other OGC specifications can be integrated a lot easier now because enhancements to the Axis2 schema compiler have been made by the author.

WS-Eventing plays an important role in the TSSN as it is essential for the alarm notification chain. The specification that is used by all the clients and services is WS-Addressing. Note that HTTP, which represents the underlying transport layer of most the web services, already provides an addressing scheme. This, however, is not as useful as it seems because web services may change their transport layer and messages sometimes require complex routing. The reasoning behind this and other things have been explained in detail.

Overall the TSSN provides a "Service-oriented architecture for monitoring cargo in motion along trusted corridors". This web services-based approach allows for platform and programming language independence and offers compatibility and interoperability. The integration of SOA, OGC specifications and sensor networks is complex and difficult. Most systems and research focuses either on the combination of SOA and OGC specifications or on OGC standards and sensor networks. However, the TSSN shows that all three can be combined and that this combination provides capabilities to the transportation and other industries that have not existed before. In particular, web services in a mobile sensor network environment have always been seen as slow and producing a lot of overhead. The TSSN, as shown by the results in Fokum *et al.* (2010) and Kuehnhausen and Frost (2010a) demonstrates that with proper architecture and design the performance requirements of the targeted scenario can be satisfied.

Furthermore, the TSSN and its SOA allow sensor networks to be utilized in a standardized and open way through web services. Sensor networks and their particular communication models led to the implementation of asynchronous message transports in SOA and are supported by the TSSN.

7. Future work

After evaluating the current implementation, several points of improvement for the specific implementation as well as the architecture were identified.

7.1 Security

The current system only provides entry points for the WS-Security in terms of the Rampart module. There are several issues in the current implementation of the module, especially with regard to attaching policies to web services and clients. This is discussed in Komp *et al.* (2010). Further development is underway to implement WS-Security. In between the VNOC and the MRN communication is secured by establishing a virtual private network. However, this is not practical using a satellite link for performance reasons. In terms of system architecture, the management of sensors is done at the sensor node but as of now there is no support for the secure handover to other sensor nodes. Hence, this approach may not be feasible for large-scale systems and it might be better integrated with the VNOC.

7.2 Service discovery

Owing to several problems in the specific implementation of the UDDI that was used, for the trials most of the services were made aware of the other services through the means of configuration instead of service discovery. Since using a UDDI or another means of service registry provides better scalability in large systems, it must be an essential piece of future versions of the TSSN.

7.3 Multiple service clouds

During the trials, all services were unique which in an operational system is not necessarily the case. There are architectural issues that need to be explored in dealing with multiple versions not only of single web services but multiple VNOCs and MRNs. This is especially important when it comes to managing policies and subscriptions properly in a large-scale SOA.

References

Bellwood, T., Clement, L., Ehnebuske, D., Hately, A., Hondo, M., Husband, Y.L., Januszewski, K., Lee, S., McKee, B., Munter, J. and von Riegen, C. (2002), "UDDI version 3.0", OASIS specification, OASIS, available at: http://uddi.org/pubs/uddi-v3.00-published-20020719.htm

Bernhardt, T. and Vasseur, A. (2007), "Event-driven application servers", available at: http://dist.codehaus.org/esper/JavaOne_TS-1911_May_11_2007.pdf

Booth, D. and Liu, C.K. (2007), "Web services description language (WSDL) version 2.0 part 0: primer", W3C recommendation, W3C, available at: www.w3.org/TR/2007/ REC-wsdl20-primer-20070626

Botts, M., Percivall, G., Reed, C. and Davidson, J. (2007), "OGC SensorWeb enablement: overview and high level architecture", White Paper, OGC, available at: http://portal.opengeospatial.org/files/?artifact_id=25562

Box, D., Cabrera, L.F., Critchley, C., Curbera, F., Ferguson, D., Graham, S., Hull, D., Kakivaya, G., Lewis, A., Lovering, B., Niblett, P., Orchard, D., Samdarshi, S., Schlimmer, J., Sedukhin, I., Shewchuk, J., Weerawarana, S. and Wortendyke, D. (2006), "Web services eventing (WS-Eventing)", W3C member submission, W3C, available at: www.w3.org/Submission/2006/SUBM-WS-Eventing-20060315/

Cabrera, L.F., Kurt, C. and Box, D. (2004), "An introduction to the web services architecture and its specifications", Microsoft Technical Article, Microsoft, available at: http://msdn.microsoft.com/en-us/library/ms996441.aspx

Chinthaka, E. (2006), "Web services and Axis2 architecture", IBM article, IBM, available at: www.ibm.com/developerworks/webservices/library/ws-apacheaxis2/

Chu, X., Kobialka, T. and Buyya, R. (2006), "Open sensor web architecture: Core services", *Proceedings of the 4th International Conference on Intelligent Sensing and Information Processing*, pp. 1-4244, available at: www.gridbus.org/papers/ICISIP2006-SensorWeb.pdf

Delphi Group (2003), "The value of standards", Survey, Delphi Group, Boston, MA, available at: www.ec-gis.org/sdi//ws/costbenefit2006/reference/20030728-standards.pdf

Fitzpatrick, D., Dreyfus, D., Hamilton, B.A., Onder, M. and Sedor, J. (2006), "The electronic freight management initiative", Technical Report FHWA-HOP-06-085, US Department of Transportation, Federal Highway Administration, Washington, DC.

Fokum, D. and Frost, V. (2010), "A survey on methods for broadband internet access on trains", *IEEE Communications Surveys Tutorials*, Vol. 12 No. 2, pp. 171-85.

Fokum, D., Frost, V., Kuehnhausen, M., DePardo, D., Oguna, A., Searl, L., Komp, E., Zeets, M., Deavours, D., Evans, J. and Minden, G. (2010), "An open system transportation security sensor network: field trial experiences", *IEEE Transactions on Vehicular Technology*, Vol. 59 No. 8, pp. 3942-55.

Foster, I., Kesselman, C., Nick, J.M. and Tuecke, S. (2002), "The physiology of the grid: an open grid services architecture for distributed systems integration", paper presented at Global Grid Forum, Open Grid Service Infrastructure WG, available at: www.globus.org/alliance/publications/papers/ogsa.pdf

Friis-Christensen, A., Ostländer, N., Lutz, M. and Bernard, L. (2007), "Designing service architectures for distributed geoprocessing: challenges and future directions", *Transactions in GIS*, Vol. 11 No. 6, pp. 799-818.

Gudgin, M., Hadley, M. and Rogers, T. (2006), "Web services addressing 1.0 – core", W3C Recommendation, W3C, available at: www.w3.org/TR/2006/REC-ws-addr-core-20060509

Haas, H., Booth, D., Newcomer, E., Champion, M., Orchard, D., Ferris, C. and McCabe, F. (2004), "Web services architecture", W3C Note, W3C, available at: www.w3.org/TR/2004/NOTE-ws-arch-20040211/

Hi-G-Tek (2009), available at: www.higtek.com/

Irmen, M. (2009), "10 ways to reduce the cost and risk of global trade management", *Journal of Commerce*, available at: www.joc.com/node/410216

Kalin, M. (2009), *Java Web Services: Up and Running*, O'Reilly Media, Sebastopol, CA.

KC SmartPort (2008), "Trade data exchange: nothing short of a logistics revolution", *Journal of Commerce*, available at: www.joc-digital.com/joc/20081110/?pg=29

Kiehle, C., Greve, K. and Heier, C. (2007), "Requirements for next generation spatial data infrastructures – standardized web based geoprocessing and web service orchestration", *Transactions in GIS*, Vol. 11 No. 6, pp. 819-34.

Komp, E., Frost, V.S. and Kuehnhausen, M. (2010), "Implementing web services: conflicts between security features and publish/subscribe communication protocols", Technical Report ITTC-FY2010-TR-41420-19, Information Telecommunication and Technology Center, University of Kansas, Lawrence, KS.

Kuehnhausen, M. (2009), "Service oriented architecture for monitoring Cargo in motion along trusted corridors", Master's thesis, University of Kansas, Lawrence, KS.

Kuehnhausen, M. and Frost, V.S. (2010a), "Application of the Java message service in mobile monitoring environments", Technical Report ITTC-FY2010-TR-41420-18, Information Telecommunication and Technology Center, University of Kansas, Lawrence, KS.

Kuehnhausen, M. and Frost, V.S. (2010b), "Framework for analyzing soap messages in web service environments", *International Journal of Web Services Practices*, Vol. 5 No. 1, pp. 1-9, available at: http://nwesp.org/ijwsp/2010/vol5i1/ijwsp2010-vol5-no1-01.pdf

Lafon, Y. and Mitra, N. (2007), "SOAP version 1.2 part 0: Primer (second edition)", W3C Recommendation, available at: www.w3.org/TR/2007/REC-soap12-part0-20070427/

Lawrence, K., Kaler, C., Nadalin, A., Monzillo, R. and Hallam-Baker, P. (2006), "Web services security: SOAP message security 1.1 (WS-security 2004)", OASIS Standard, OASIS, available at: http://docs.oasis-open.org/wss/v1.1/wss-v1.1-spec-os-SOAPMessageSecurity.pdf

Lawrence, K., Kaler, C., Nadalin, A., Goodner, M., Gudgin, M., Barbir, A. and Granqvist, H. (2007), "WS-securitypolicy 1.2", OASIS Standard, OASIS, available at: http://docs.oasis-open.org/ws-sx/ws-securitypolicy/v1.2/ws-securitypolicy.pdf

McKee, L. (2005), "The importance of going open", White Paper, OGC, available at: http://portal.opengeospatial.org/files/?artifact_id=6211

MacKenzie, C.M., Laskey, K., McCabe, F., Brown, P.F., Metz, R. and Hamilton, B.A. (2006), "Reference model for service oriented architecture 1.0", OASIS Standard, available at: http://docs.oasis-open.org/soa-rm/v1.0/

Nickul, D., Reitman, L., Ward, J. and Wilber, J. (2007), "Service oriented architecture (SOA) and specialized messaging patterns", Adobe Article, available at: www.adobe.com/enterprise/pdfs/Services_Oriented_Architecture_from_Adobe.pdf

Red Hat (2008), "Hibernate reference documentation 3.3.1", Technical Report, available at: www.hibernate.org/hib_docs/v3/reference/en-US/pdf/hibernate_reference.pdf

Reichardt, M. (2004), "The Havoc of non-interoperability", White Paper, OGC, available at: http://portal.opengeospatial.org/files/?artifact_id=5097

Searl, L.S. (2008), "Service oriented architecture for sensor networks based on the ambient computing environment", Technical Report, ITTC, available at: www.ittc.ku.edu/sensornet/trusted_cooridors/papers/41420-07.pdf

Troup, K., Newton, D., Wolfe, M. and Schaefer, R. (2009), "Columbus electronic freight management evaluation: achieving business benefits with EFM technologies", Technical Report, Science Applications International Corporation, McLean, VA.

Wolfe, M. (2004), "In this case, bad news is good news", *Journal of Commerce*, available at: www.ismasecurity.com/ewcommon/tools/download.aspx?docId=175

About the authors

Martin Kuehnhausen received the Dipl.Inf. (BA) degree from the Berufsakademie Stuttgart, Germany, in 2006 and the MS degree from the University of Kansas (KU), Lawrence, in 2009.

He is currently working toward the PhD degree in Computer Science with the Department of Electrical Engineering and Computer Science, KU. From 2003 to 2007, he worked on various projects with IBM. He was part of the Secure Trade Lane Team, creating an enterprise real-time container tracking solution. He is currently working on the SensorNet Project with the Information and Telecommunication Technology Center, KU. His research interests include intelligent systems, information and data management, web service technologies, and software engineering. He is a member of the IEEE Computer Society, the IEEE Communications Society, and the IEEE Intelligent Transportation Systems Society. Martin Kuehnhausen is the corresponding author and can be contacted at: mkuehnhausen@ku.edu

Victor S. Frost received the BS, MS, and PhD degrees from the University of Kansas (KU), Lawrence, in 1977, 1978, and 1982, respectively. In 1982, he joined the faculty of KU and is currently the Dan F. Servey Distinguished Professor of Electrical Engineering and Computer Science. He was the Director of the KU Telecommunications and Information Technology Center (ITTC) for more than ten years. From 1987 to 1996, he was the Director of the KU Telecommunications and Information Sciences Laboratory. He was a Program Director with the Computer and Network Systems (CNS) Division, Computer and Information Science and Engineering (CISE) Directorate, National Science Foundation (NSF) 2009-2011. He is currently an Area Editor for Communications Simulation of the *ACM Transactions on Simulation and Modeling of Computer Systems*. He has been involved in research on several national-scale high-speed wide area testbeds. He was an Investigator on a gigabit testbed (MAGIC) research effort and the ACTS ATM internet work. His research has been sponsored by government agencies, including the NSF, the Defense Advanced Research Projects Agency, the Rome Laboratory, and the National Aeronautics and Space Administration. He has been involved in research for numerous corporations, including Sprint, NCR, Nortel, Telesat Canada, AT&T, McDonnell Douglas, DEC, and COMDISCO Systems. He has been a Principal Investigator on more than 35 research efforts and has been involved as a Co-investigator on more than 40 projects. As a result of those efforts, he has published more than 100 journal articles and conference proceedings. His current research interests include communications systems and networks, networking testbeds, internet quality of service, traffic management, and integrated broadband communication networks. Dr Frost received the Presidential Young Investigator Award from the NSF in 1984. He was listed in the Kansas City Star Tech 50 in 2000 and 2002, is a Fellow of the IEEE and was a Member-at-Large of the IEEE Communications Society Board of Governors 2008-2011.