Integrated Operations in the High North

2008–2012

Joint Industry Project

Final Report

For public distibution

Report title:
Final report: All activities

Date of first issue: 24.05.2012	*Project:* Integrated Operations in the High North	
Prepared by: Frederic Verhelst	*Reviewed by:* (reviewer)	*Approved by:* Tom Thomsen
Contributions by: Frederic Verhelst	*Chapter/section:* 1	

See the chapter colophon for details on the contributors in the chapters for the individual activities, chapters 2-6.

Summary:
During a four year period from May 2008 to April 2012, twenty-two companies from different parts in the E & P value chain have joined forces in the *Integrated Operations in the High North* Joint Industry Project to design, implement and demonstrate a reliable and robust software architecture to be used in an Arctic setting.

Operational models for these remote and sometimes hostile environments are often based on a lean local asset organization that is dependent on an extended support network. A prerequisite for these operational models is the continuous need for collaboration across disciplinary, geographical and organizational boundaries. Open standards are essential to ensure interoperability, to facilitate integration, and to transfer data. Shared information and knowledge models based on open standards make collaborative data-to-information-to-decisions work processes more efficient.

This final report starts with giving a context to the project and then continues with giving an overview of the main outcomes of the project. It also lists the individual deliverables and how to get them.

Revision	*SVN*	*Status*	*Date*	*By*	*Reason/Changes*
0.99	3575	Draft	24.05.2012	FVE	Released for review by Steering Committee
1.00	3592	Accepted	05.06.2012	FVE	Accepted by Steering Committee
2.00		Public	20.06.2012	FVE	Issued for public distribution

Contents

Synopsis

During a four year period from May 2008 to April 2012, twenty-two companies[1] from different parts in the E & P value chain have joined forces in the *Integrated Operations in the High North* Joint Industry Project to design, implement and demonstrate a reliable and robust software architecture to be used in an Arctic setting.

Operational models for these remote and sometimes hostile environments are often based on a lean local asset organization that is dependent on an extended support network. A prerequisite for these operational models is the continuous need for collaboration across disciplinary, geographical and organizational boundaries. Open standards are essential to ensure interoperability, to facilitate integration, and to transfer data. Shared information and knowledge models based on open standards make collaborative data-to-information-to-decisions work processes more efficient.

This final report starts with giving a context to the project and then continues with giving an overview of the main outcomes of the project. It also lists the individual deliverables and how to get them.

[1]The participants in the IOHN-project are: ABB, Business Association of Norwegian knowledge- and technology based enterprises (Abelia), Baker Hughes, Cisco, Computas, Det Norske Veritas (DNV), ENI, Epsis, FMC, The Norwegian Defence and Security Industries Association (FSI), IO Centre, International Research Institute of Stavanger (IRIS), National Oilwell Varco, Norwegian Institute of Science and Technology (NTNU), The Norwegian Oil Industry Association (OLF), POSC Caesar Association, Petroleum Safety Authority Norway, Siemens, Statoil, The Norwegian Defence, University of Oslo and University of Stavanger. Two sub-projects are also partly sponsored by the Research Council of Norway; GOICT under the VERDIKT programme (RCN nr. 183235) and AutoConRig under the PETROMAKS programme (RCN nr. 187473).

Chapter 1

Introduction

1.1 Background

1.1.1 The Arctic region

The Arctic region holds vast amounts of extractive energy resources. Most of the Arctic resources lie offshore in environmentally very sensitive areas, beneath thick ice and/or in deep water. Weather conditions, and distance to existing infrastructure and centers of population, add additional operational and logistic challenges. In order to meet all requirements and at the same time maintain profitable operations, the industry needs to create new field development and operational concepts that include heavily instrumented facilities. These operational concepts will often be based on a lean local organization supported remotely by a combination of a remote asset organization, multi-asset support centers and/or external expert centers.

1.1.2 Integrated Operations

Around the year 2000, strategic initiatives were launched by almost all major oil companies, all covering more or less the same basic idea of making better use of modern Information and Communication Technologies (ICT) in oil and gas operations. The main goal was to improve the effectiveness and efficiency of interaction between operations, disciplines and decision-makers, regardless of their respective geographical location. Different terms have been coined for this industry-wide development, and often trade-marketed by individual companies. A few terms are non-proprietary and used in more global contexts. These include the terms *Digital and Intelligent Energy*, used by the Society of Petroleum Engineers (SPE), *Integrated Operations (IO)*, as coined by the Norwegian Oil Industry Association (Norwegian: Oljeindustriens Landsforening, OLF) and *Real-Time Operations*. In essence it is all about integrating operations and people in a seamless collaboration, independently of organization, time and place.

Most of these programs started with getting the digital infrastructure in place, i.e., by installing modern Information and Communication Technologies (ICT) to enable personnel at different locations within the operators' organization to collaborate more effectively. An often used slogan from that time was: "bring the data to the specialist, not the specialist to the data". We saw new work processes being established, typically involving bringing operational people and support functions from the asset team together in a virtual team. This resulted in a more efficient and effective collaboration, leading to more informed, better and faster decision making and in reducing the HSE footprint by minimizing the need for traveling. Essential enablers are the availability of Real-Time Data

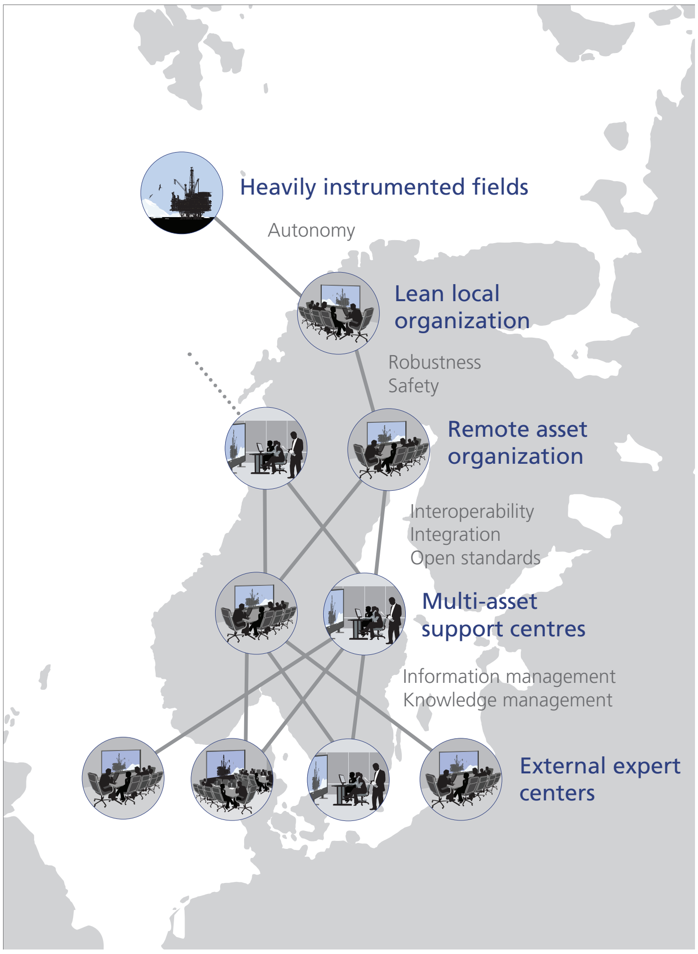


Figure 1.1: Novel operational concepts based on a lean local organization supported from a remote asset organization in combination with multi-asset support centers and external expert centers.

throughout the organization and the possibility to communicate more effectively between people at different locations. The collaboration rooms with video conferencing systems that were installed for this are the more visible components, but the digital infrastructure that makes the real-time data from the production sites available to the whole operators' organization and the dedicated work processes to handle the data are certainly equally important.

1.1.3 Collaboration

Where the first implementations were mostly focused on a one-to-one connection between an operational site and a dedicated asset team, extended support networks have gradually been introduced in the recent years (see figure 1.1 for an example). Different operators are establishing lean local organizations, especially in distant and otherwise challenging locations such as the High North. A lean local organization will be able to offer basic support functions, but will rely on a remote asset organization for more advanced support. The remote asset organization is located in a less challenging, more populated environment.

The support network may be extended further by the introduction of multi-asset support centers, which offer second or third line specialized services to several asset teams simultaneously. Examples include centers for drilling or production support, for condition monitoring of heavy rotating machinery, and instruments for fiscal metering. These centers typically offer services that are not so common and that require special expertise, experience and sometimes tools. Alternatively, these multi-asset support centers may also offer services that may benefit from coordination across individual assets, such as planning and scheduling with relation to logistics and operations. By offering these services from a centralized entity, one may make more efficient use of the available resources. There are also further benefits to be had from these multi-asset support centers, as they are well suited to the transfer of knowledge and experience from senior to more junior experts. In this setting, junior experts are being exposed to a wide variety of cases over a limited period of time, and may ask for guidance from more experienced colleagues in the centre.

Sometimes the extended support network may also involve external expert centers. This may be arranged on an ad-hoc basis, or because some support functions are outsourced to an external service provider. For example, the vendors of heavy rotating equipment may act as experts in a condition based monitoring work process.

1.1.4 Data and information challenges

These extended collaboration networks require an almost continuous interaction between the different locations, and rely upon a robust and secure digital infrastructure. A platform for effective and efficient data and information exchange and decision making becomes mission critical in these extended collaboration networks. To enable collaboration across boundaries in a smooth and efficient manner, a robust and secure digital platform based on open standards is required, supporting a much higher degree of interoperability across multivendor applications, disciplines, geographic locations and organizations than is common today.

There are four particular data and information challenges associated with collaboration across boundaries [1]:

- The first challenge is to find relevant data and information in the huge volumes of real-time and historic data. This is the proverbial "needle in the haystack" problem. In recent years, a huge growth in the volume of data from each single asset has been observed. This results from a combination of an increasing number of measurement points, and higher sample frequencies. Multi-asset support centers have to handle data streams from a multitude of assets and need new tools and work processes to do this more efficiently and effectively.
- A second challenge is to understand the exact meaning of the data and information. Most often the full meaning of a set of data is not given in the data set itself. One needs to be familiar with details of the systems and processes that produced the data and information for a correct and reliable interpretation. This means that currently the use of data and information is not seamless, but to a large extent requires human interpretation and contextual understanding to be useful.
- The third challenge is that with an extended support network, the external experts and the experts from other disciplines give advice to a multitude of assets, i.e., they do not work full time on a single asset. Therefore, they do not always have an intimate and up-to-date knowledge of the detailed setup of a given part of the system, and for instance will not always know in detail the topology of the system. In other words, they may miss (part of) the contextual information regarding that particular asset.
- A fourth challenge is to ensure decisions that are the outcome of a work process are executed in a timely fashion. This is often a bottle neck, since it most often involves human interaction at the operational site for the decisions to be implemented

1.2 Integrated Operations in the High North project

The goal for the *Integrated Operations in the High North (IOHN)* project is to solve the above challenges by designing, piloting and demonstrating a reliable and robust solution architecture and platform that will be able to facilitate collaboration across boundaries [2]. Existing open standards are used and extended when required and new standards are incubated to ensure interoperability, to facilitate integration and to transfer data. To make data-to-information-to-decisions work processes more efficient, information and knowledge models based on open standards are also developed and used.

The IOHN project is set up as five activities organized in a matrix; three related to the *digital platform* and two related to pilots for different *business*

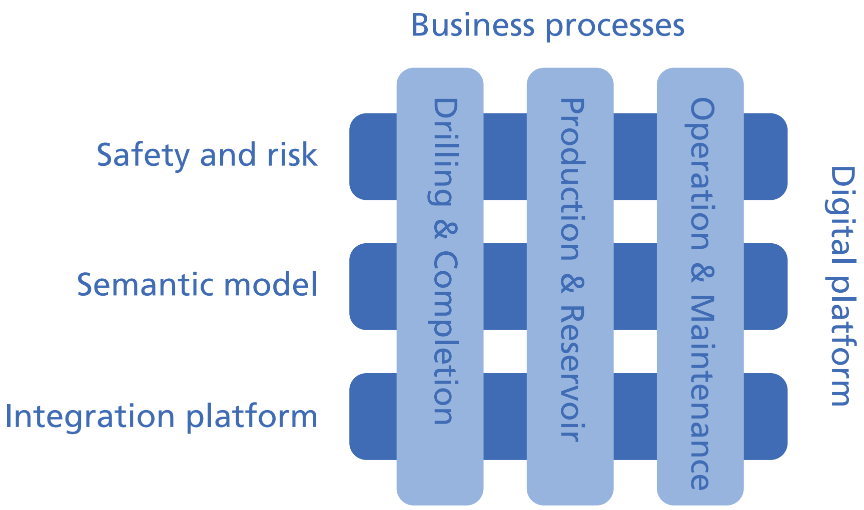


Figure 1.2: The setup of the IOHN project. Operations and Maintenance is not an independent activity, but elements of it are included in the other two Business Process activities.

domains (see figure 1.2). Within the different business domains, architectural relevant use cases have been defined. These use cases form the basis for extracting the requirements for the digital platform and are used later on to test (parts of) the digital platform and to validate the value potential for IO G2.

1.2.1 Digital platform

The digital platform, being a crucial differentiator for IO G2, forms a natural foundation for the IOHN project. Besides an activity on the *Integration Platform* itself, two additional activities are defined working on two very important aspects of the architecture for IO G2. A first additional activity is the *Semantic Model* activity working on the integration of information, i.e. making sure that data and information may be used seamlessly across boundaries using open standards based on ISO 15926, the oil and gas ontology. A second additional activity focuses on specific *safety and security* aspects for highly interconnected and software intensive systems.

1.2.2 Pilots

Pilots within two different business domains are defined in the IOHN project.

The *drilling and completion pilot* focuses on seamless interoperability through open standards at the drilling control level. There is currently a gap in the cost-effective and timely utilization of all the real-time data available during drilling. Improvements will come from systems that are closing the loop, i.e. by automatically analyzing the real-time data stream, autonomous decision making and directly intervening with the drilling control system. Such systems that may integrated with external service providers, rely on a common, computer readable ”understanding” of the drilling domain. This will lead to a better safeguarding of operational limits, less unproductive time and, in the end, better and more efficient well placement. It is also a stepping stone towards concepts as an unmanned subsea autonomous drilling rig under the ice.

The *reservoir and production pilot* focuses on the detection of sand production and associated erosion management, which are common challenges in the industry. Sand production may constitute severe safety, financial and environmental risks as well as lost opportunities. Within the arctic setting, the criticality of these risks only increases and it is therefore crucial to have trustworthy data with respect to both measurements of sand production volumes as well as reliable methodologies for erosion monitoring and predictions. One of the possibilities that arise from the digital platform is increased collaboration and interoperability, which is a key characteristic for IOG2. The pilot is exploring this opportunity by facilitating new work processes for an expert competence center supporting several assets with production-related issues.

1.3 Structure of the report

The different activities have each a separate chapter in this report; first the three activities related to the Digital Platform and then the two related to the Use Cases.

Chapter 2 starts by describing the Integration Platform. In the following Chapter 3 the work on the Semantic Model is described. Chapter 4 discusses the Dependability and Risk aspects of the Digital Platform.

Finally, we get the two chapters on the pilots; Chapter 5 on the Drilling Pilot and Chapter 6 on the Production Pilot.

1.4 References

[1] Verhelst, Klüwer, Skramstad, Myren, Ornæs, and Tvedt. A digital platform for real-time operations in the high north. *World Oil*, 231(10), October 2010.

[2] Verhelst, Myren, Rylandsholm, Svensson, Waaler, Skramstad, Ornæs, Tvedt, and Høydal. Digital platform for the next generation io: A prerequisite for the high north. In *SPE Intelligent Energy Conference and Exhibition*, March 2010.

1.5 List of deliverables

In the following list, preliminary or intermediate versions of deliverables have been left out.

The deliverables are available from the password protected project-internal WIKI-site: http://www.posccaesar.org/wiki/IOHN/Internal.

Title	Date	Contributors	Type
Digital Platform for the Next Generation IO: A Prerequisite for the High North	2010	Verhelst et. al.	*SPE Intelligent Energy 2010* conference paper ☞
A digital platform for real-time operations in the High North	2010	Verhelst et. al.	*World Oil* October 2010 ☞

Report title:
Final report: All activities

Date of first issue: 01.05.2012	*Project:* Integrated Operations in the High North	
Prepared by: Inge Svensson (ISV)	*Reviewed by:* Frederic Verhelst (FVE)	*Approved by:* Tom Thomsen
Contributions by: Inge Svensson	*Chapter/section:*	

Summary:
In this activity we deploy a Semantic Web Integration platform which is used to store sensor data and semantic models for the Oil and Gas upstream industry. Through our research we show that it is possible to develop Semantic models in ISO 15926 formats, store them on the Semantic platform and integrate the models with sensor data through ISO 15926 templates. Client applications connect to the integration platform to fetch data, query the models and even write resulting calculations back to the platform.

Revision	*SVN*	*Status*	*Date*	*By*	*Reason/Changes*
0.9		Draft	11.05.2012	ISV	First draft version
0.99	3558	Draft	24.05.2012	FVE	Released for review by Steering Committee
1.00	3592	Accepted	05.06.2012	FVE	Accepted by Steering Committee

Chapter 2

Integration Platform

2.1 Summary

In this activity we deploy a Semantic Web Integration platform which is used to store sensor data and semantic models for the Oil and Gas upstream industry. Through our research we show that it is possible to develop Semantic models in ISO 15926 formats, store them on the Semantic platform and integrate the models with sensor data through ISO 15926 templates. Client applications connect to the integration platform to fetch data, query the models and even write resulting calculations back to the platform.

2.2 Introduction

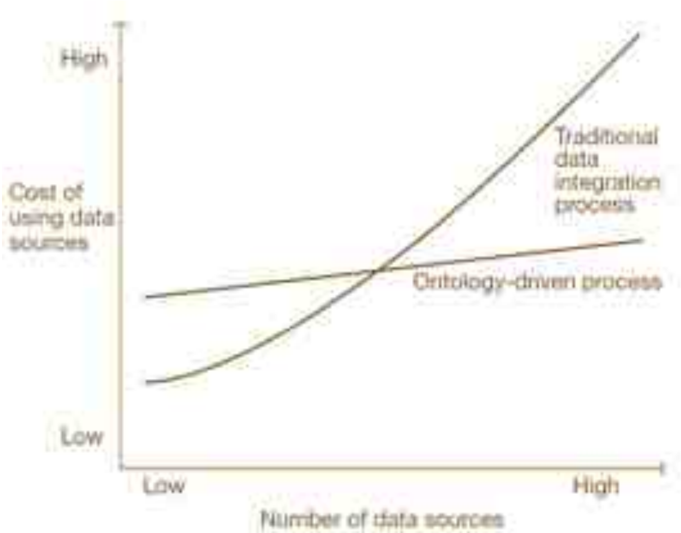


Figure 2.1: Comparison of the cost of integration in function of the number of data sources between an ontology-driven (semantic) and a traditional approach. Figure after PricewaterhouseCoopers [3].

One of the main scopes of the Integrated Operations in the High North (IOHN) project is to develop a seamless and trustworthy software architecture for second generation integrated operations. The architecture will be demonstrated through sub activity pilot projects which uses the developed solution. Essential components in the Service Oriented Architecture (SOA) [12] will make information flawlessly available across boundaries by using open standards based on Semantic Web Technologies proposed by W3C [11], ISO 15926 [1] and the PCA Oil and Gas Ontology [2].

W3C standards providing the foundation for this Web of data include URIs, RDF [8], RDF Schema (RDFS) [9], the Web Ontology Language (OWL) [7] and the Semantic Protocol and RDF Query Language (SPARQL) [10].

Semantic Web technology provides interesting opportunities within knowledge sharing, searching for data, improving data quality, capturing domain knowledge and data integration. Compared to traditional data assimilation processes or systems, semantic data integration can provide a less costly solution when the number of data sources increases according to PricewaterhouseCoopers (see figure 2.1 from [3]). However an Ontology-driven process will have a higher initial cost when the number of data sources is low.

By demonstrating running implementations in the pilots, we provide a way to motivate the use of Semantic Technologies. When partners and potential customers see useful applications that fetch data from multiple sources through a unified framework, it is much easier to motivate them to embrace and implement the Semantic Architecture recommended to adopt data integration on a bigger scale.

2.2.1 Activity scope

The main goal and deliverable of this activity is to implement and support a Semantic Integration platform that will be demonstrated and used by the pilots in the IOHN Joint Industry Project. The work will start by selecting the software platform to use for the development and further going on with implementing the selected solution.

Other work in this activity includes getting familiarized with the software platform, importing and updating models from Semantic Model, updating the server with new versions of software, license control, user access, firewall setup, and operation system and virus protection updates. In addition the scope includes facilitating the clients to start using the platform with various types of consumer software.

2.2.2 Participants in this activity

The main participant in this activity is Baker Hughes. However the implementation of the Semantic Integration Platform has been done in close relationship with the Semantic Model activity and DNV and there is also close interaction with the participants of the Production Pilot. In early stages of the project, Cisco also contributed to Semantic Model.

Baker Hughes has also contributed to Drilling Pilot and the sub project AutoConRig with development of the Drilling Communication standard.

2.3 Preliminaries

2.3.1 Semantic Integration Platform

A prerequisite for implementing the IOHN demonstrators is a working integration layer that provides services to data providers and client applications and connects the data with a semantic model.

An integration layer manages data and data access, and it contains a semantic model to mapped datasets. There are various methods to integrate data. One common technique is to choose standards for transferring, storing and querying data. Developing and supporting the Semantic Integration Layer is the main task of this activity. The Semantic Platform is based on common W3C standards and data is typically stored as RDF data and data and models can be queried using SPARQL.

2.3.2 Semantic Model

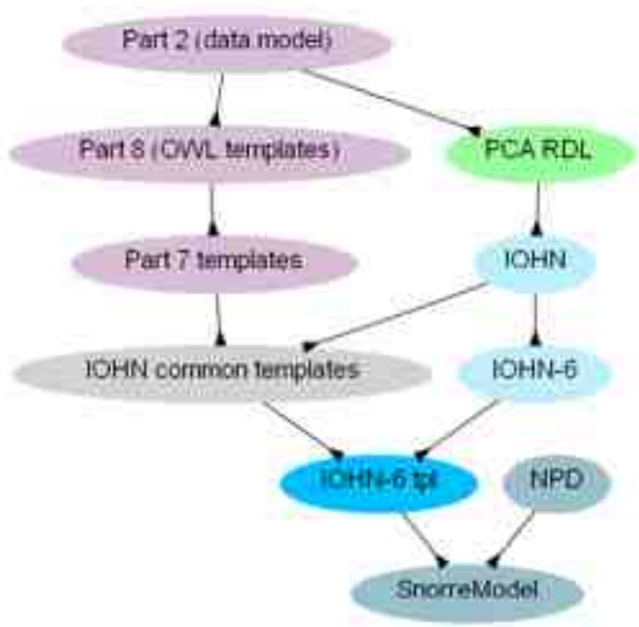


Figure 2.2: Graphical representation of how the models and ontologies relate to each other. ☞

The semantic model describes the structure of oil platforms and wells in a generic language, including information about which parts they consist of and how these parts are related to each other. Particular platforms and facilities are modeled as instances in the semantic model. The model can be queried in many different ways, an example would be to list which sensors are installed on a particular oil well. In the IOHN project, the models templates and ontologies will be based on ISO 15926. The semantic structures are delivered as OWL files

Filename	Description	Graph
data-model.owl	Taxonomy representation of ISO 15926-2 entity types	http://standards.iso.org/iso/ts/15926/-8/ed-1/tech/reference-data/data-model
p7tm.owl	Basic classes for representing ISO 15926-8 template signatures	http://standards.iso.org/iso/ts/15926/-8/ed-1/tech/reference-data/template-model
p7tpl.owl	Signatures for ISO 15926-7 templates	http://standards.iso.org/iso/ts/15926/-8/ed-1/tech/reference-data/templates
pca.owl	PCA RDL	http://posccaesar.org/rdl
iohn.owl	IOHN Nomenclature	http://iohn.org/rdl
iohn6.owl	IOHN-6 (Production Pilot) Nomenclature	http://iohn.org/activity-6/rdl
iohn6-templates-no-AND.owl	IOHN-6 Templates	http://iohn.org/activity-6/tpl
SnorreUML-complete.owl	Snorre Model	http://iohn.org/activity-6/model
iohn6-datamap.owl	Maps each data source in the IOHN-6 Reference Model to a tag name	http://iohn.org/activity-6/map
uio-npd.owl	UiO NPD Fact Page Extract for IOHN Reference Plant Model	http://sws.ifi.uio.no/npd

Table 2.1: List of OWL files from Semantic Model.

Filename	Description	Graph
iohn6-data-sna.owl	Raw data Snorre A	http://iohn.org/activity-6/data-sna
iohn6-data-snb.owl	Production data Snorre B	http://iohn.org/activity-6/data-snb

Table 2.2: List of RDF data files in OWL format.

from Semantic Model. Table 2.1 lists the different OWL-files and figure 2.2 shows how the different ontologies are related to eachother. Please refer to Chapter 3 for more details on the Semantic Model.

2.3.3 Data Mapping

A typical oil platform has several oil wells, and on each well there are numerous sensors that measure properties or status such as pressure, temperature, flow and valve positions. The readings for each sensor are recorded and the produced data are stored in a proprietary system developed by the sensor manufacturer. In order to access sensor data one must have detailed knowledge of each sensor manufacturer's data acquisition system. An alternative is to introduce an integration layer that makes the data from the different sensor systems available in a generic manner, and in addition map the data to a semantic model.

Although the Semantic platform we selected provides the availability to integrate data from various sources like a relation database, we have due to some challenges chosen to provide the data files as RDF files in OWL format and import them to the server (Table 2.2).

2.3.4 Client Application Services

An integration layer provides adapters to which client applications (and possible other integration layers) can connect in order to request data. The communication protocol is interlinked with the semantic model. A client can use the semantic model to locate interesting sensors, and request data by referencing the sensor instances in the model. The client does not need to know the proprietary sensor data systems nor the detailed structure of the platform. SPARQL has been chosen as the query language and the data will be returned as RDF/XML data. The integration Layer also allows for writing data back to the server.

In the project we have multiple clients from various companies connecting to the server, querying for models and data (see figure 2.3. Some of them are also writing results back to the SPARQL endpoint. We also have two different data providers, providing data to the integration platform. However due to some challenges, the data is manually updated. In addition we also have well data originating from NPD.

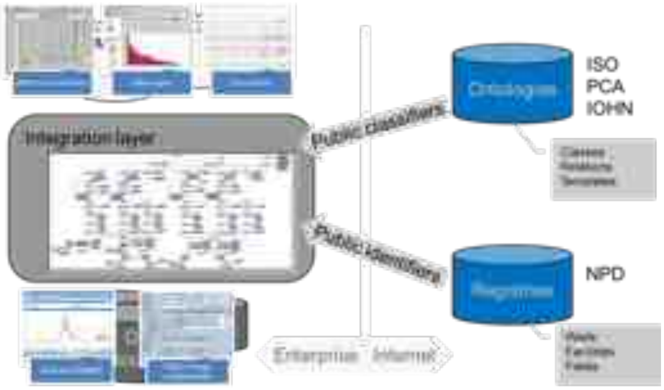


Figure 2.3: Overview of clients and data suppliers.

2.4 Tasks in detail

2.4.1 Implementation

This activity kicked off with a selection process on which Semantic platform to use. Several commercial and open source solutions were evaluated. After several criteria's the Anzo platform from Cambridge Semantics [5] was chosen. One of the main reasons was the ability to integrate data with other systems.

Technology vendor Cambridge Semantics takes a semantic approach to accessing information sources and linking associated data for business use. The founding members of the company realized the business potential of commercializing semantic technology based on work some of them did with IBM's semantic layered research platform. Cambridge Semantics uses virtualized access and navigation across information assets that have a common reference to helps organizations tackle these data dilemmas. Virtualized access navigates through the semantics of the actual information, rather than relying on manual definitions by someone trying to describe the information. What makes this approach unique is its orderly workflow, the ability to tap into a large number of information sources in a unified manner, and the ability to make use of information that to date has been locked up and unavailable.

Cambridge Semantics' Anzo Enterprise product [6] provides a platform and tools to help manage access and linkage of information, along with the workflow, policies and rules required to make information more relevant and meaningful. It has analytics capabilities that can help quickly assemble dashboards for analysis on the virtualized information. Being able to navigate across data was the top-ranked business capability in a business analytics benchmark research, but is not well-established in business systems today. Instead, organizations spend more than two-thirds of their time in analytic processes on data related activities and tasks which Cambridge Semantics helps address.

The Anzo Enterprise platform can be accessed through a Web-based interface that business and data analysts can use without being trained in either database administration or application development. Cambridge Semantics has been a leader in linking RDF and Excel spreadsheets, and provides user-friendly interfaces for querying the RDF database. The product allows for integration with so-called Anzo adapters and provides semantic tools e.g. to be used in Excel or on the Web. The underlying OSGI framework can also allow for custom development of applications and services.

The core of the Anzo Semantic Server is a RDF Quadruple server with an associated SPARQL endpoint. That means it is also using named Graphs persisted for each RDF triple. The platform distinguishes very much between models/ontologies and datasets and typically one dataset are linked to one class in the ontology. Anzo for Excel is essential both as a tool to work with datasets and ontologies but also as a tool for importing, exporting and working with the RDF data.

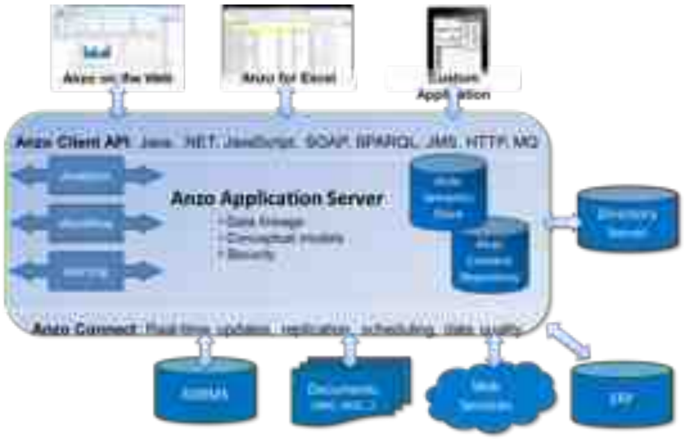


Figure 2.4: The Anzo Enterprise platform.

Figure 2.5: List of models and ontologies in Anzo for Excel ontology management.

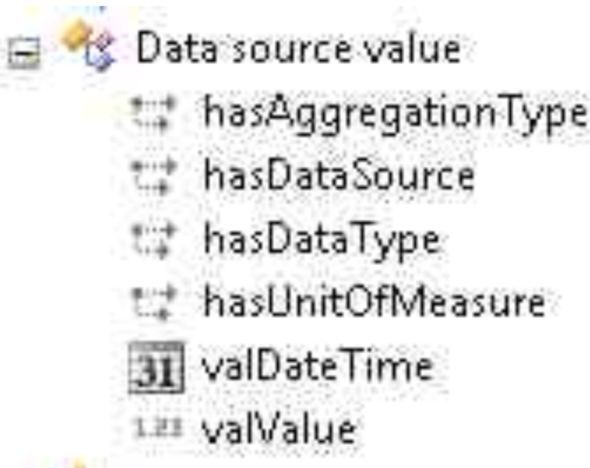


Figure 2.6: Data Source Value Template shown in Anzo for Excel ontology management.

2.4.2 Client support

In addition to the inbuilt Web Server and Anzo for Excel, the Anzo platform also allows for multiple other options to exchange data between applications and the server. The following platforms and communication protocols are supported: Java, .NET, JavaScript, SOAP, SPARQL, JMS, HTTP and MQ. The Anzo suite also comes with its own Java API but open source Java Semantic API's, like Jena, can also be used.

Some client software in the project including test software developed by Baker Hughes has been using dotNetRDF, an Open Source Semantic Web/RDF Library for C#/.Net. The usage of the API has been very encouraging and by using the .net platform it is very easy to integrate RDF data with other software and systems running on the Microsoft platform. An example could be a client which communicates with the drilling communication standard and uses the integration platform to fetch data and query models.

By using Anzo for Excel it is easy to both import en export RDF data to Excel. Anzo for Excel is using the JMS protocol for communication to the server. Even if Anzo for Excel was not actively required for the project, it was successfully tested with simpler models and datasets. Anzo for Excel is also used for Ontology and Data Set management and simple ontologies/models can directly be edited in Anzo for Excel (figures 2.5 and 2.6).

Another option to straightforwardly view RDF data is to use the inbuilt solution Anzo on the web. This solution can be used to create Simple Views of RDF Data using the Anzo Web Server (running on port 80). Again this was successfully tested with simpler models and datasets supplied to the project. However with the advanced ISO 15926 models we experienced some technical limitations in Anzo for Excel.

Some of the data imported to the server where RAW sensor data with high resolution. Even if the period of the time series was quite short, it revealed that it took waste amount of time for the server to respond with a result to the query. It would have helped some to upgrade to the latest state of the art hardware, but even then the results would have shown that RDF is not especially suited for high density sensor data. OPC-UA has been mentioned as an alternative and the semantic model could have provided links to these data sources. However for the platform models and other meta-data, the performance was good and exposed the power of the SPARQL query language and supported semantic technology.

2.4.3 Integration Possibilities

The Anzo enterprise platform promises extended support for integration with other systems. Anzo Connect is a graphical tool for connecting to relational databases, defining the mappings between relational data and ontologies, and creating and managing the schedules for pulling that data into the Anzo Server. The underlying technology is called ETL connections.

Since Anzo allows for SQL Server Integration and SQL Server can replicate data from various sources, we have an entry point for integration with multiple systems including ERP systems. Direct ETL connection to other systems than relational databases, is presently limited.

Currently the extraction processes in Anzo connect, runs only on-demand via the Run button. However, there is the ability to manually update an ETL configuration to run at scheduled times. In the full release of Anzo Connect (future) it will be available via the GUI. However the promised real-time updates are difficult when running on a schedule, and update and delete functions are not yet supported. Each new cycle is expended to take quite some time when you have large amounts of data in the database.

Anzo connect with a direct connection to an SQL Server database was successfully tested in the project. However we had to run with a simpler extracted portion of the data source template. Anzo connect is directly depended on Anzo for Excel for the mapping process and some constructs in the models where not supported (especially in iohn6-datamap.owl). The successful testing also allowed for viewing the data in Anzo on the Web. Actually a lot of time was used to try and get the IOHN models into forms that ANZO could understood.

2.4.4 Security

Security is an important part for ensuring a reliable software architecture for second generation integrated operations. SOIL (Secure Oil Information Link) [4] is a secure collaboration network for the oil & gas industry which originates from Norway (see figure 2.7). So typically data communication between service companies and operators will be done via this secure network.

Figure 2.7: SOIL network.

A company network or rig network will characteristically be split into several DMZ's (Demilitarized Zones) where access control routers prevent access between each Zone and to the outside world. The Zone acts between a company's private network and the outside public network. It prevents outside users from getting direct access to a server that has company data.

The IOHN integration server runs on SOIL and in a DMZ and all outside access is controlled by access control lists. That means only companies that required access were entered on the control list and could access the server/services.

The Anzo semantic platform has inbuilt user control where individual access can be given on each Graph in the system. Anzo enterprise also allows for integration with other user control systems like Windows domain controllers. In IOHN all users where supplied individual users and passwords.

2.5 Conclusions

2.5.1 Lessons learnt

- Semantic Web technology is better suited for meta-data than sensor data due to the high overhead in the RDF data.
- 32 bit OS leads to memory limitations. RDF servers requires waste amount of memory to give adequate performance.
- Developing semantic models is time consuming and requires domain knowledge and strong se-

mantic expertise. Tools for developing ontologies are still in early phases.

- The Semantic Integration server and supported tools was not ready developed.

2.5.2 Successes

- We were able to implement and support the semantic models, datasets and we were able to query both against the model and datasets using SPARQL. The returned RDF/XML data could be parsed to other applications. So the proof of concept worked.
- When an application has been adapted to use the integration layer, it is easily deployed at all company applications supporting the integration layer. This is typically done by developing an API and layered applications.
- The integration layer may offer more data than previous data access, opening up possibilities for extended functionality.
- By writing data back to the integration layer, results are reusable also for other applications from the same vendor or to other vendors.
- Linking of reference classes against the POSC Caesars RDL could lead to better integration with other systems.
- ISO 15926 Templates gives high level access to complicated structures in the underlying ISO 15926 languages.

2.5.3 Dissemination

- Semantic Days, 2010, Presentation
- SPE Bergen, 2011, Presentation
- Intelligent Energy, 2012, Presentation

2.6 References

[1] POSC Caesar Association. Iso 15926 integration of life-cycle data for process plants including oil and gas production facilities, 2012. Available at http://www.posccaesar.org/wiki/ISO15926.

[2] POSC Caesar Association. Oil and gas ontology, 2012. Available at http://www.posccaesar.org/wiki/PCA/IO/OilAndGasOntology.

[3] PricewaterhouseCoopers (PwC). Pwc technology forecast: Semantic web in the enterprise, 2009. Available at http://www.pwc.com/us/en/technology-forecast/spring2009/index.jhtml.

[4] RigNet. The network, 2012. Available at http://www.rig.net/Solutions/SOIL/The_Network/.

[5] Cambridge Semantics. Website, 2012. Available at http://www.cambridgesemantics.com.

[6] Cambridge Semantics. Anzo enterprise product, 2012. Available at http://www.cambridgesemantics.com/products/anzo-enterprise.

[7] World Wide Web Consortium (W3C). Owl web ontology language - overview, 2004. Available at http://www.w3.org/TR/owl-features/.

[8] World Wide Web Consortium (W3C). Resource description framework (rdf), 2004. Available at http://www.w3.org/RDF/.

[9] World Wide Web Consortium (W3C). Rdf vocabulary description language 1.0: Rdf schema, 2004. Available at http://www.w3.org/TR/rdf-schema/.

[10] World Wide Web Consortium (W3C). Sparql query language for rdf, 2008. Available at http://www.w3.org/TR/rdf-sparql-query/.

[11] World Wide Web Consortium (W3C). Semantic web, 2012. Available at http://www.w3.org/standards/semanticweb/.

[12] Wikipedia. Service-oriented architecture entry on wikipedia, 2012. Available at http://en.wikipedia.org/wiki/Service-oriented_architecture.

2.7 List of deliverables

In the following list, preliminary or intermediate versions of deliverables have been left out.

The deliverables are available from the password protected project-internal WIKI-site: http://www.posccaesar.org/wiki/IOHN/Internal.

Title	Date	Contributors	Type
Platform selection document	2010/12	Inge Svensson (BHI) and Johan Klüwer (DNV)	Note
Documents and input related to development of a Drilling ontology		Inge Svensson (BHI)	Report
Various presentations at IOHN workshops	2009-12	Inge Svensson	Presentation
Communications and Security Architecture for the High North	2011	Daniel Keely (Cisco)	Presentation
Network Infrastructure Deliverables Description	2009	Daniel Keely (Cisco)	Report
Network Infrastructure and Platform for services	2009	Daniel Keely (Cisco)	Report
Existing infrastructure in the High North	2008	Kjell Helge Strøm	Report

Appendices: Integration Platform

2.A Cisco: Communications and Security Architecture for the High North

IOHN architecture - executive summary

- **Technical change required to deal with the challenges of the High North will drive core E&P capability transformation**: floating and seabed drilling; subsea production, storage and transportation; environmental monitoring; condition based maintenance; collaborative working environment. This introduces **new requirements for Communications and Security**
- **Advances in ICT have been identified** for meeting the challenges of the new operational environment e.g. autonomous networks, fibre optic sensing, virtualised collaboration
- **Architectures have been defined**, mapping ICT capabilities to the future operational environment across drilling, production and maintenance, with Integrated Operations at the heart of remote monitoring and control
- Applied scenarios in subsea equipment replacement and pipeline incident response illustrate how **ICT can enable new operations and bring new value** to the High North. These scenarios offer the **potential for near term evaluation and piloting**
- Conclusions indicate the same architectural thinking **may also be applied to fields outside of the High North**, creating new value for Integrated Operations in brown-field and green-field sites across a global portfolio

2


Challenges in the High North will drive technical change in the operating environment
Complex environment
Extreme weather
Environmentally sensitive
Remoteness
Lack of infrastructure
New solutions for unfamiliar conditions
Remote 'from the beach' drilling, production and operations
Evolving subsea and maintenance facilities
Automation and autonomous systems
3

Core E&P capabilities will be transformed

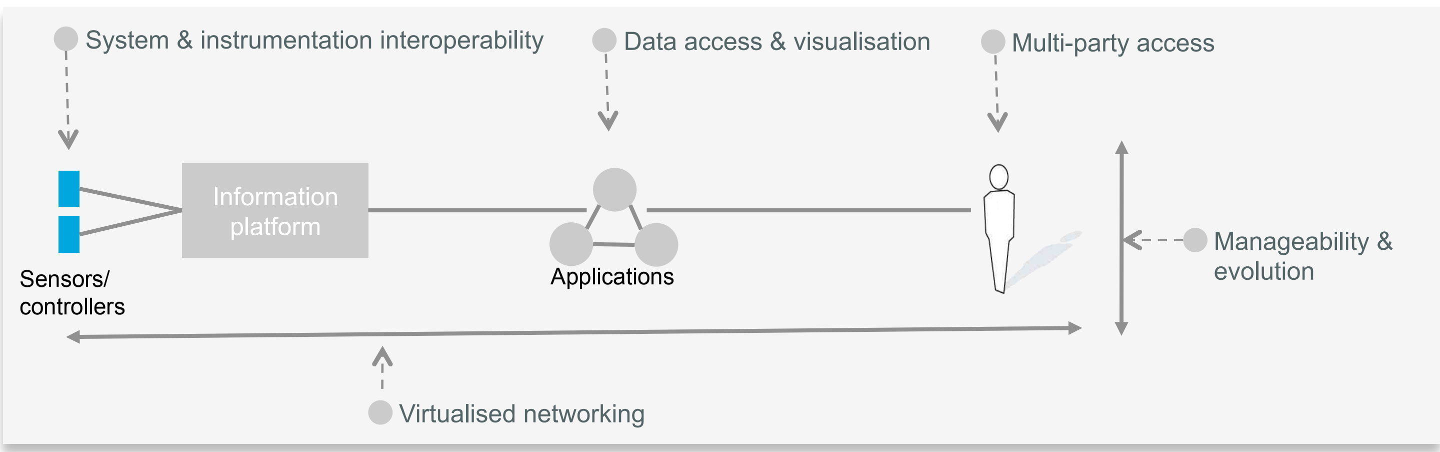
- **Floating and Seabed drilling:** Remote drilling and integrated autonomous drilling systems will develop as well as the rigs: arctic condition floating rigs that can be moved and moored to avoid drifting ice, and the development of encapsulated seabed drilling rigs
- **Subsea production, storage and transportation:** Subsea to shore facilities will be used further from land as subsea boosting and communications technologies evolve, and floating platforms will have to scale to the arctic conditions, high production rates and infrequent platform access. Mooring, artificial buoyant seabed, ice management, power, and remote control systems will all need to evolve
- **Environmental monitoring:** Solutions that demonstrate the sensitive arctic environment is not at risk or being damaged will become increasingly important to meet access and legislative requirements and need to be further developed
- **Condition based maintenance:** the move from time based and reactive maintenance philosophies will need to accelerate towards CBM and place greater dependency upon autonomous technologies for inspection and maintenance
- **Collaborative working environment:** Virtualised collaborative working facilities must be available to enable the greater dependency on partnerships -‘the operational ecosystem’- for managing operations and assets. Techniques and technologies in distributed monitoring and control, remote simulation and training and incident response must all develop

4

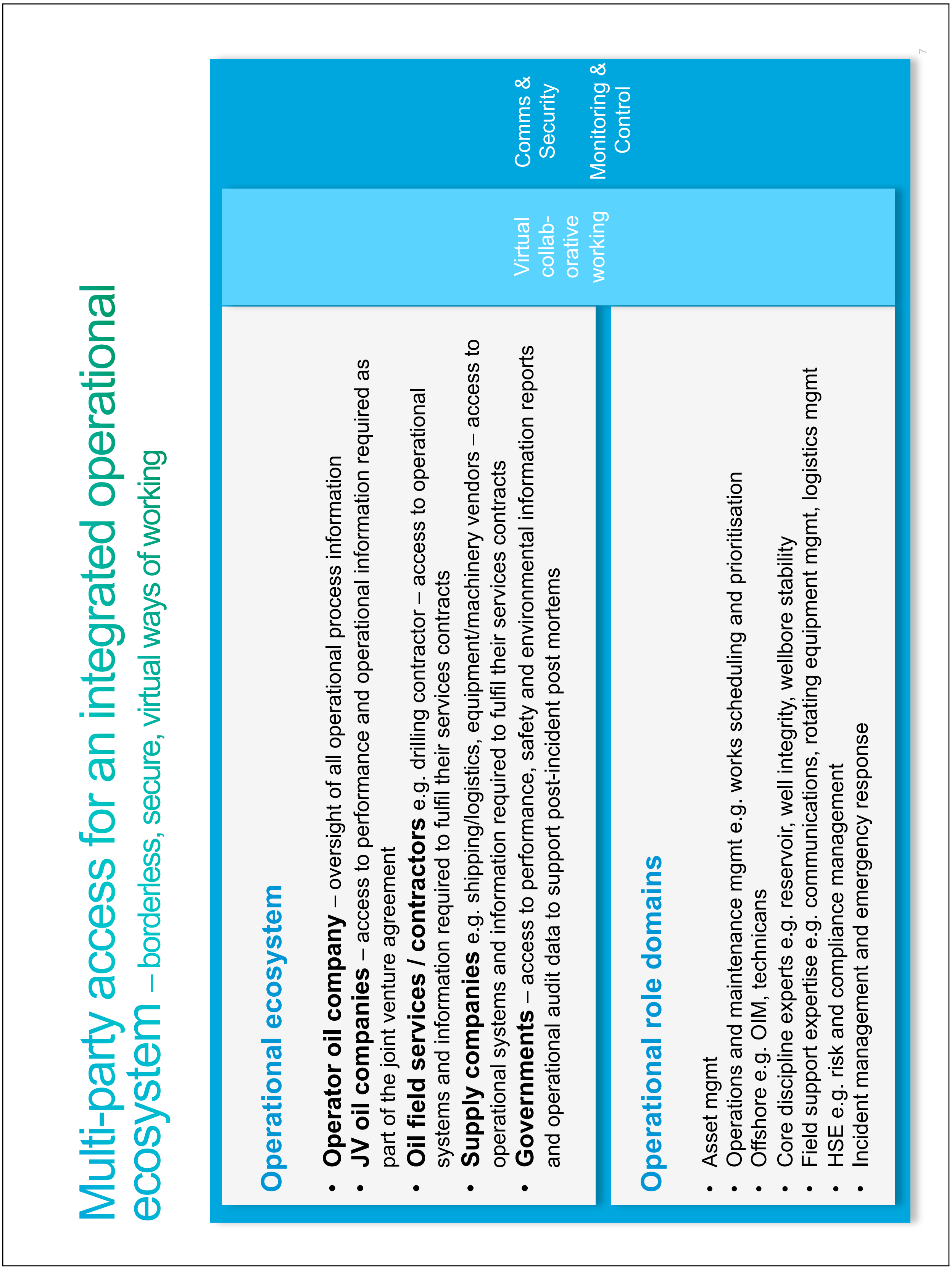

Capability transformation will introduce new communications challenges and requirements
• Greater volume of remote systems and processes to monitor & control
• Greater dependency on inherently safe & reliable systems
New requirements for communications
• System & instrumentation interoperability
• Data access & visualisation
• Resilient virtualised networking
• Multi-party access
• Manageability and evolution
5

New requirements for communications

- **System & instrumentation interoperability:** connection of the growing number of sensors, controllers, actuators using standard IP communications protocol to enable the remote monitoring and control of operational equipment & machinery
- **Data access & visualisation:** transportation of right-time operational data to a common information platform for access by multiple parties and analytical/visualisation applications
- **Virtualised networking:** multiple virtual networks able to offer different service levels and characteristics based on specific operational use cases e.g. process control, video inspection, voice communications
- **Multi-party access:** allowing different organisations to utilise the network as part of the operational ecosystem
- **Manageability and evolution:** ease of deployment, ongoing network management & maintenance, as well as planning for long term installation (i.e. 20+ years) aligned to field development and growth



6


Multi-party access for an integrated operational ecosystem – borderless, secure, virtual ways of working
Operational ecosystem
Operator oil company – oversight of all operational process information
JV oil companies – access to performance and operational information required as part of the joint venture agreement
Oil field services / contractors e.g. drilling contractor – access to operational systems and information required to fulfil their services contracts
Supply companies e.g. shipping/logistics, equipment/machinery vendors – access to operational systems and information required to fulfil their services contracts
Governments – access to performance, safety and environmental information reports and operational audit data to support post-incident post mortems
Operational role domains
Asset mgmt
Operations and maintenance mgmt e.g. works scheduling and prioritisation
Offshore e.g. OIM, technicans
Core discipline experts e.g. reservoir, well integrity, wellbore stability
Field support expertise e.g. communications, rotating equipment mgmt, logistics mgmt
HSE e.g. risk and compliance management
Incident management and emergency response
Virtual collab-orative working
Comms & Security
Monitoring & Control

Changes to the risk management model creates new ICT security requirements

- Shift to remote operation of safety critical activities e.g. drilling, well integrity mgmt
- Greater dependency on technology to automate core operational processes
- Greater number of third party organisations proactively involved with the running of the operational ecosystem
- Harsh environment constrains response and evacuation capability

New requirements for Security

- **Communications resilience and redundancy:** ensuring appropriate availability for safety critical applications
- **Security state monitoring and reporting:** right-time monitoring of the network and information services for compliance with security policy
- **Role based access control:** controlling access to information services based on role and responsibility alignment to given processes, competencies or operations
- **Security management and incident response:** management of security policies, risk profiles and controls maintenance (e.g. security patch management) , as well as response and management of security related incidents

8

Threat model

Assets – what are we protecting?

- Operational processes e.g. remote drilling
- Communications infrastructure i.e. providing availability, integrity and confidentiality to the operational processes

Threats – what do we need to protect against?

Source	Motivation	Threat
Pressure groups/ activists	Political	Hacking/malware
Terrorists	Political	Hacking/malware, physical sabotage
Criminal groups	Financial gain	Hacking/malware, industrial espionage
Hackers	Challenge/ status	Hacking/malware
Employees & partners	Human error	operational policy non-compliance
Geophysical	Force Majeure	seabed movement, adverse weather/ice
Other Arctic activities	Human error	trawling or submarine entanglement

6

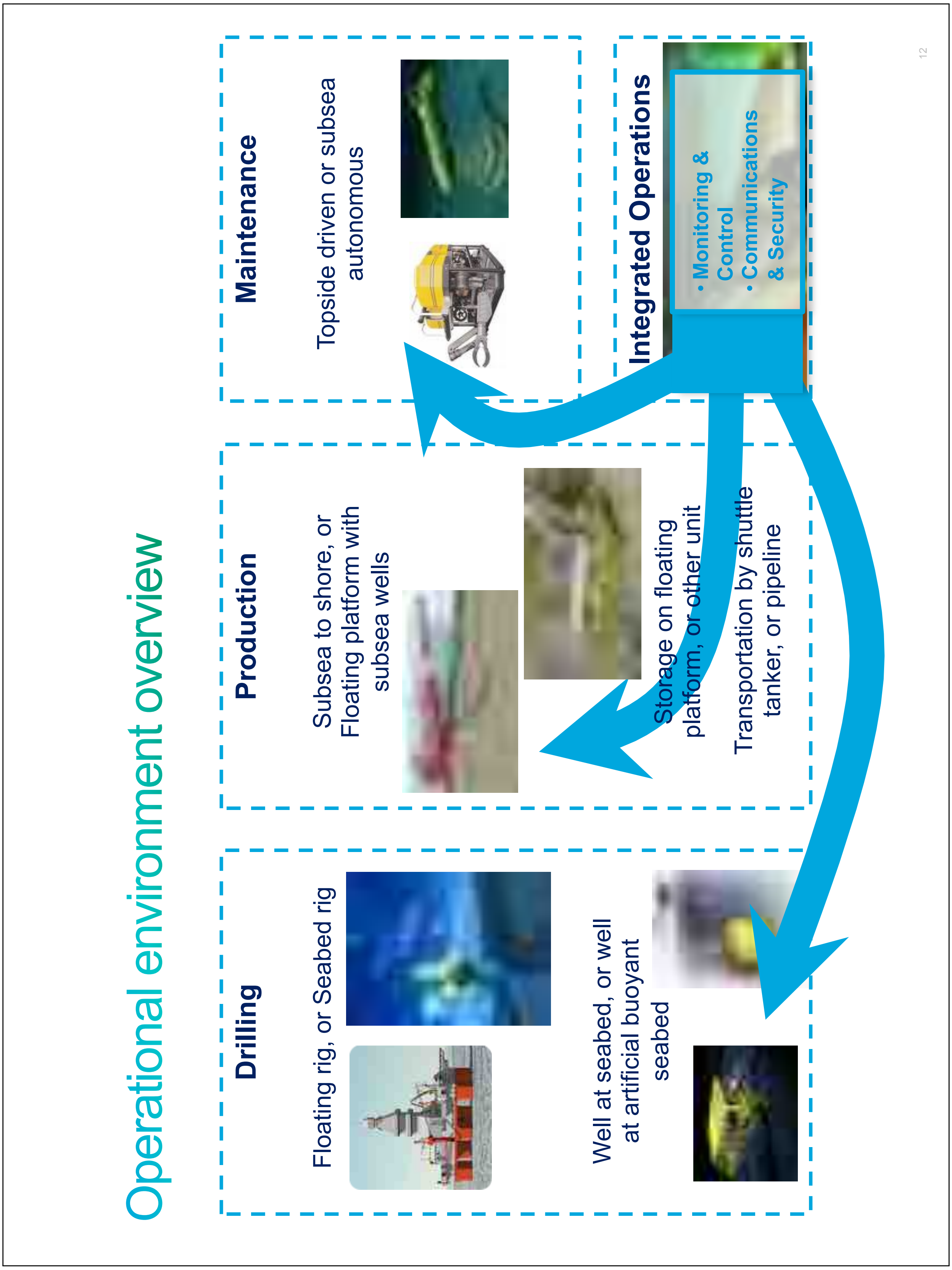
Advances in ICT provide capabilities to address the challenges

Technology	Description	Value for IOHN
IP Convergence and M2M communications	Defacto standard for communications – IP – can be extended to interconnect all equipment used in the field, including subsea and including legacy proprietary protocols. This underpins the ability to enable machine to machine (M2M) communications required for the next level of automation and autonomous systems. For this purpose, small IP stacks have been developed for ultra-low power devices	• Enables the integrated drilling network and associated benefits of full remote visibility and safe control • Enables onshore access to all data generated in the field and adoption of more automated and autonomous systems
Autonomous networks	Enables IP networking equipment (i.e. routers, switches) to adapt to the environment in which it is deployed and self configure and optimise without the requirement for skilled network engineers	• Fast and economical drop deployment of new equipment without the need for pre-configuration or local engineering support • Avoids reconfiguration of equipment during maintenance replacement, movement between fields or assets or after mergers/JVs
Fibre optic sensing	Uses fibre optic cables to monitor attributes (e.g. temperature, vibration, bend) without the need for additional sensors to be attached and managed. Applications include pipeline, equipment and down-hole monitoring	• Simplified, standardised and cost effective solution (use same fibres and connectors as the communications network) for operational and environmental monitoring • Improved instrumentation enables improvement in associated processes e.g. sand detection, well integrity monitoring
Subsea and sea surface wireless nets	Underwater acoustic communications technology is evolving (improving distance/coverage) and the next technology step change is likely to come from laser based communications. These technologies avoid subsea cabling and connections, although distances/bandwidth are limited in the short term. Sea surface wireless nets can be achieved with wireless and/or satellite technology in combination with floating buoys (with a master connection to subsea infrastructure)	• In most scenarios in the near-mid term future it is anticipated subsea fibre will be the preferred solution to acoustics. Surface wireless nets could be of value over certain distances instead of fibre backhaul, and are likely to be useful where transient topside facilities/shipping is deployed • Wireless connectivity can play a role within pipelines or wells to support real-time communications from robots e.g. for monitoring / inspection

Advances in ICT provide capabilities to address the challenges

Technology	Description	Value for IOHN
RFID equipment monitoring	The application of tracking the location of equipment using RFID technology is quite well understood and offers similar use cases and benefits in Arctic and non-Arctic environments. However, in the High North there is a stronger use case requirement to also use RFID as a means of recording and tracking of equipment maintenance history where manual inspection is much harder e.g. drill pipe placement	• Improving utilisation of equipment and helping enable condition based monitoring • Accurate inventory and condition of infrastructure deployed in the field • Enabling equipment to remain within maintenance guidelines
Subsea video	Enables remote visual confirmation of real-time conditions at fixed or mobile locations e.g. down-hole, well head and subsea infrastructure, pipeline (interior & exterior), ROV/ UAV	• Additional high quality information stream to help rapidly understand a problem and formulate an appropriate decision and response
Anomaly detection	Technology to move from pattern based intrusion detection systems to behaviour based systems, enabling activities to be identified outside of normal operational procedures	• Provides earlier identification of potential security incident, creating greater opportunity to prevent an incident before it occurs
Network virtualisation	Technology to enable multiple virtual networks to operate over a single physical network infrastructure, each network being able to operate with different types of services or service levels	• Enables multiple parties to work within an operation in a secure manner • Enabling segregation, prioritisation of safety critical networks and process • Closer alignment between operational requirements of individual processes and communications service levels • Simplifies overall network management
Virtualised collaboration	Enables participants within an entire operational ecosystem to access a consistent view of information and collaborate in real-time regardless of location or organisation	• Reducing time to respond to problems or incidents • Improving quality of informed collaborative decision making

11


Operational environment overview
Drilling
Floating rig, or Seabed rig
Well at seabed, or well at artificial buoyant seabed
Production
Subsea to shore, or Floating platform with subsea wells
Storage on floating platform, or other unit
Transportation by shuttle tanker, or pipeline
Maintenance
Topside driven or subsea autonomous
Integrated Operations
• Monitoring & Control
• Communications & Security
12

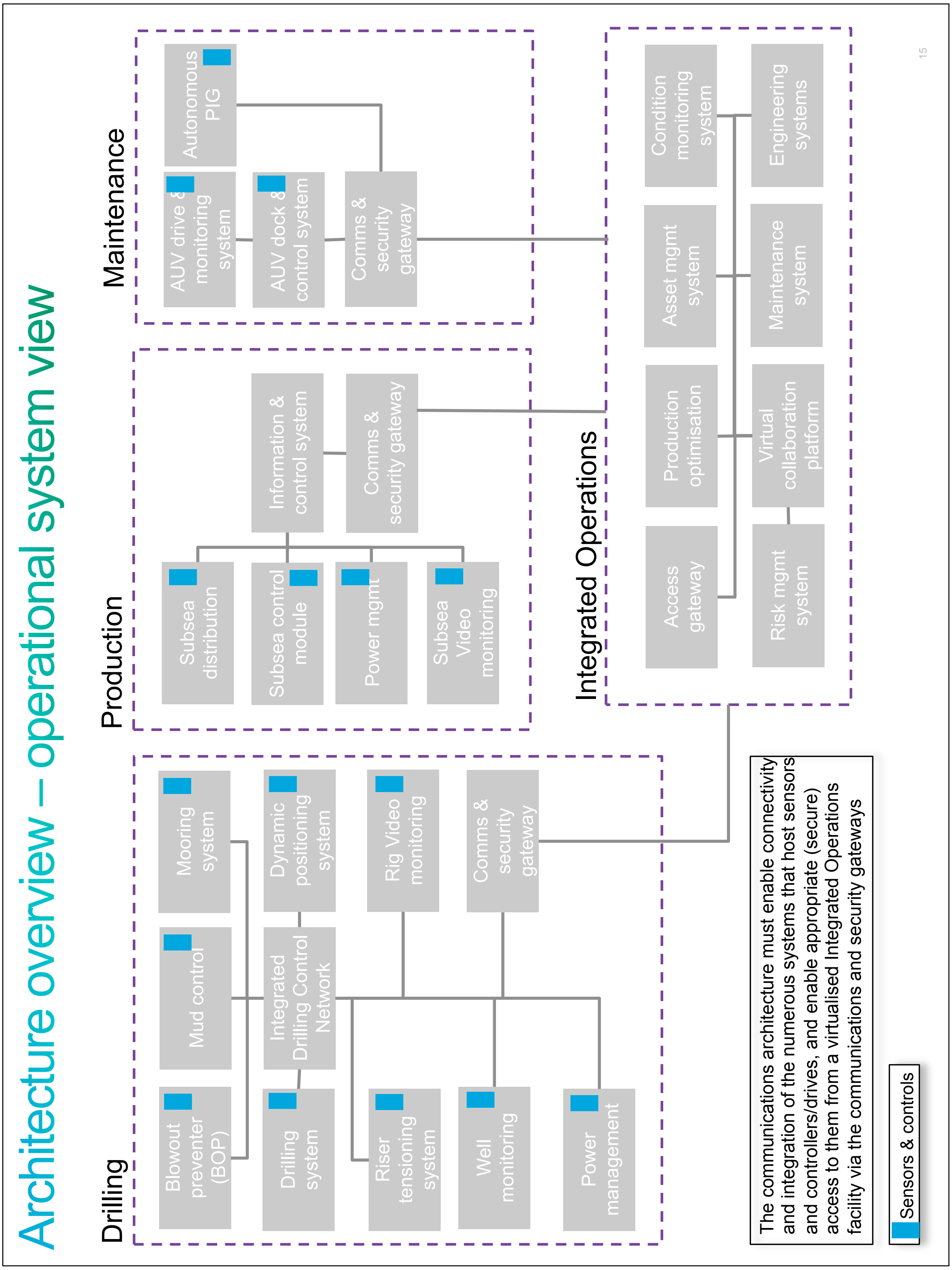
Architectural principles - communications

Requirement	Principles	Enabling Services
System & instrumentation interoperability	• IP and Ethernet as the common communications standard for interoperability • Support for legacy protocol transport within IP during an anticipated migration period to a full end to end IP environment	Switching, routing, IP tunnelling
Data access & visualisation	• Transportation of all field data into a common information platform/repository • Quality of Service (QoS) technology to prioritise important communications • Predictable rich media and application behaviour	Switching, routing, QoS, network protocol optimisation, application aware networking
Virtualised networking	• MPLS Service Provider architecture to create virtual networks • Traffic engineering to enable and manage specific service levels across different virtual networks	Switching, routing, MPLS, MPLS-TE
Multi-party access	• A Business to Business (B2B) distributed closed "Internet", enabling shared access for participating organisations • Standard interfaces and technique for fast onboarding and offboarding of partners onto the network • Facilitation of virtual work environments and dynamic user centric workspaces for collaborative working	Switching, routing, access mgmt, expertise locator, session recording, unified comms, application sharing
Ease of deployment, management & maintenance	• Autonomous networking for simplified field deployment and maintenance of network routing and switching equipment • Common management platform for simplified and common end to end (e.g. offshore instrumentation to onshore support centre, and across multiple virtual networks) network monitoring and management • Equipment that can take advantage of major networking upgrades (e.g. migration to IPv6) without the need to replace physical equipment • Architecture and equipment able to meet the environmental requirements within the arctic e.g. temperature, distance from shore (> 300km), depth (> 500m) and extended life (20+years)	Autonomous configuration, switching, routing
Scalability aligned to field development and growth	• Virtual network creation for rapidly deploying new networks as required by operational or asset/field changes • Standard design patterns to reduce complexity and risk when scaling to new assets, based on pre-defined field characteristic profiles • Ability to add large numbers of nodes and capacity/bandwidth without changing the network architecture	MPLS mgmt

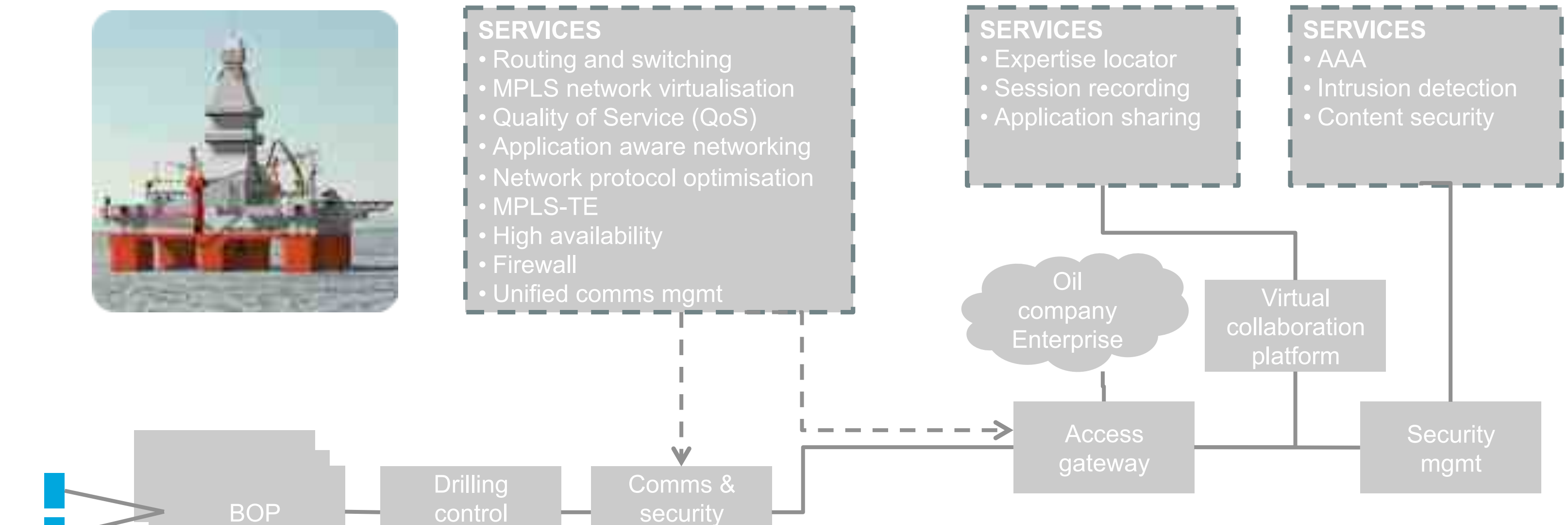
Architectural principles - security

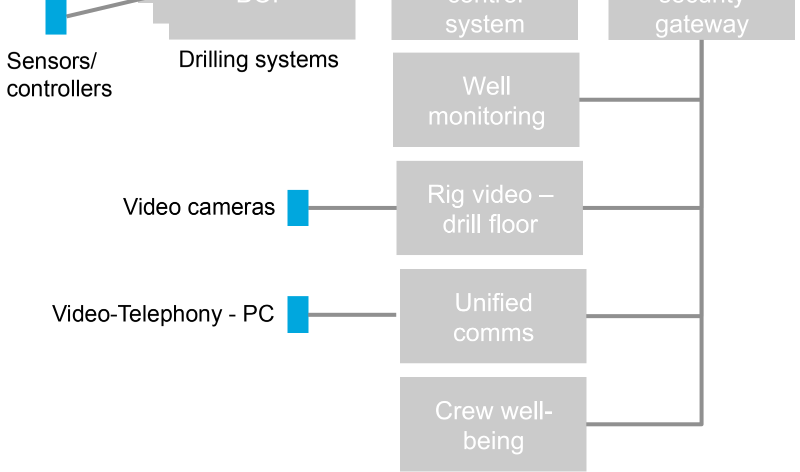
Requirement	Principles	Enabling Services
Infrastructure resilience and redundancy	• Robust, connectionless, multiple path networks for ensuring delivery of end to end communications • Redundancy of critical points in the network	High availability, routing
Real-time security state monitoring and reporting	• Monitoring and reporting of anomalous security related events, outside of security policy • Protection against unauthorised applications and malware entering the network	Intrusion detection, security event mgmt, content security, firewall
Role based access control	• Standardised processes & tools for provisioning of cross-organisation role based user accounts and mgmt of profiles and entitlements	AAA, firewall
Security management and incident response	• Latest security patches/updates consistently applied to network attached systems in a timely manner • Rapid access to relevant real-time and historical security event data, shared with appropriate security investigation / risk management teams	Security update mgmt, security event mgmt, application sharing

14


Architecture overview – operational system view
Drilling
Blowout preventer (BOP)
Mud control
Mooring system
Drilling system
Integrated Drilling Control Network
Dynamic positioning system
Riser tensioning system
Rig Video monitoring
Well monitoring
Comms & security gateway
Power management
Production
Subsea distribution
Subsea control module
Power mgmt
Subsea Video monitoring
Information & control system
Comms & security gateway
Maintenance
AUV drive & monitoring system
AUV dock & control system
Autonomous PIG
Comms & security gateway
Integrated Operations
Access gateway
Production optimisation
Asset mgmt system
Condition monitoring system
Risk mgmt system
Virtual collaboration platform
Maintenance system
Engineering systems
The communications architecture must enable connectivity and integration of the numerous systems that host sensors and controllers/drives, and enable appropriate (secure) access to them from a virtualised Integrated Operations facility via the communications and security gateways
Sensors & controls

Architecture overview – topside drilling: short term



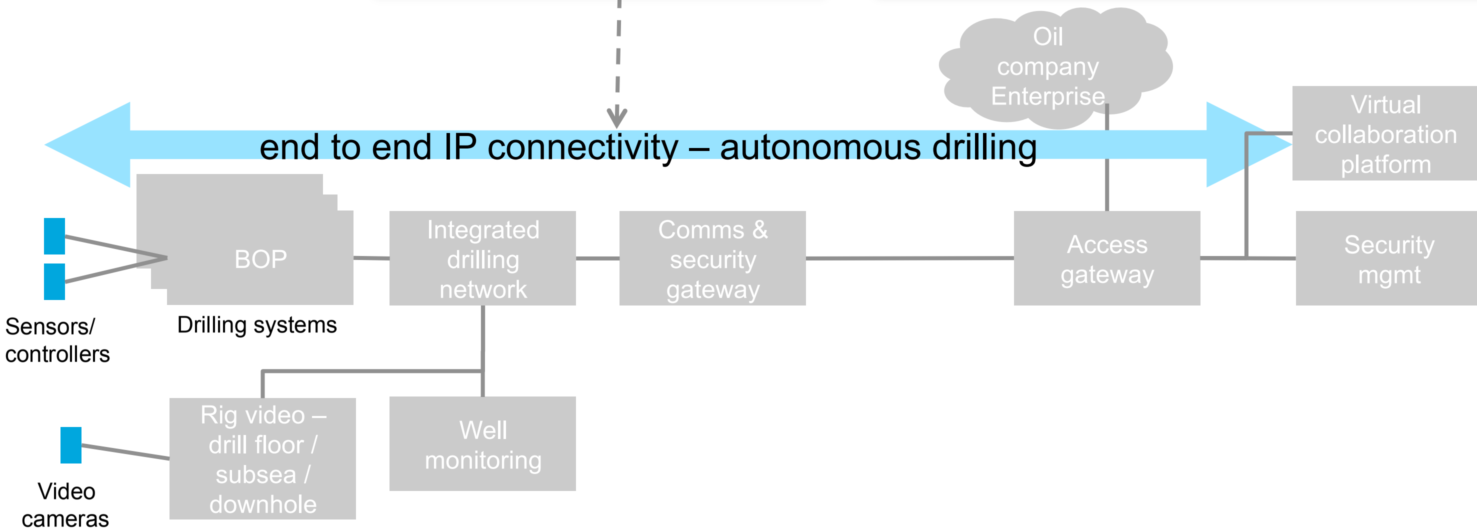


Drilling will be managed by the Drilling Control System (DCS), which will control the drilling through its connections to the drilling systems (e.g. BOP, mud control), which in turn will be connected to sensors and controllers. In the short term, the connections may use legacy protocols, with the IP interface only reaching as far as the DCS. This interface is used for communication with onshore drilling teams (monitoring for local offshore drilling; control also for remote onshore drilling) via the comms and security gateway located on the rig. This gateway also provides direct onshore access to live drill floor video footage and data collected from well monitoring systems. In addition, it enables voice and video communications to the rig using unified communications technology. The gateway provides all the necessary network functions to enable resilient, secure and predictable connectivity between offshore and onshore (connecting through the onshore access gateway, which is the common networking interface point to all field operations. The virtual collaboration platform will enable experts to be located and joined into collaborative sessions and workspaces to share and discuss issues using contextual based information and applications accessed through Oil company and/or IO centre connectivity, and the comms & security gateway. The security mgmt component will ensure access, usage and processes are protected in accordance with security polices.

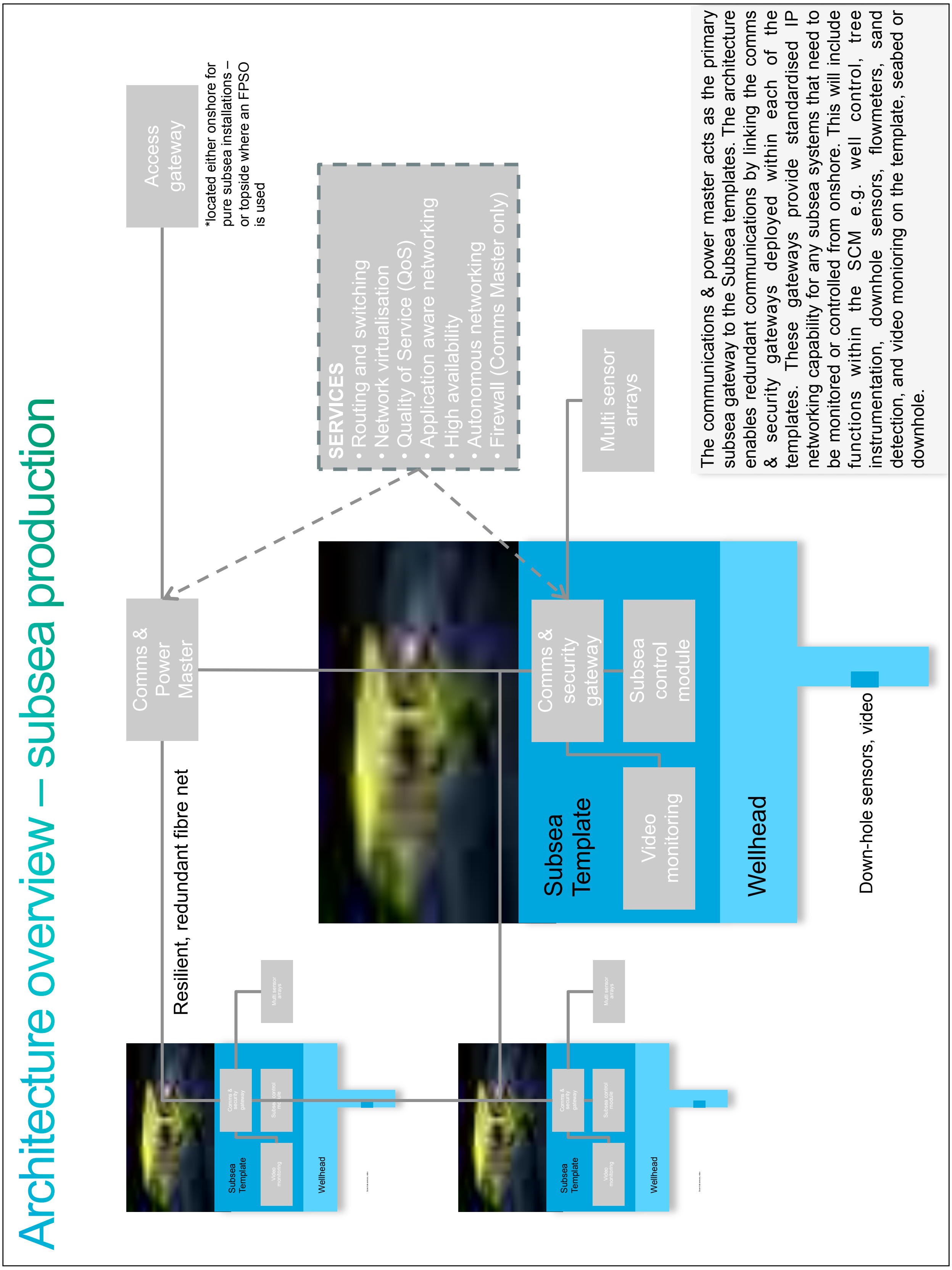
Architecture overview – topside/seabed drilling: longer term

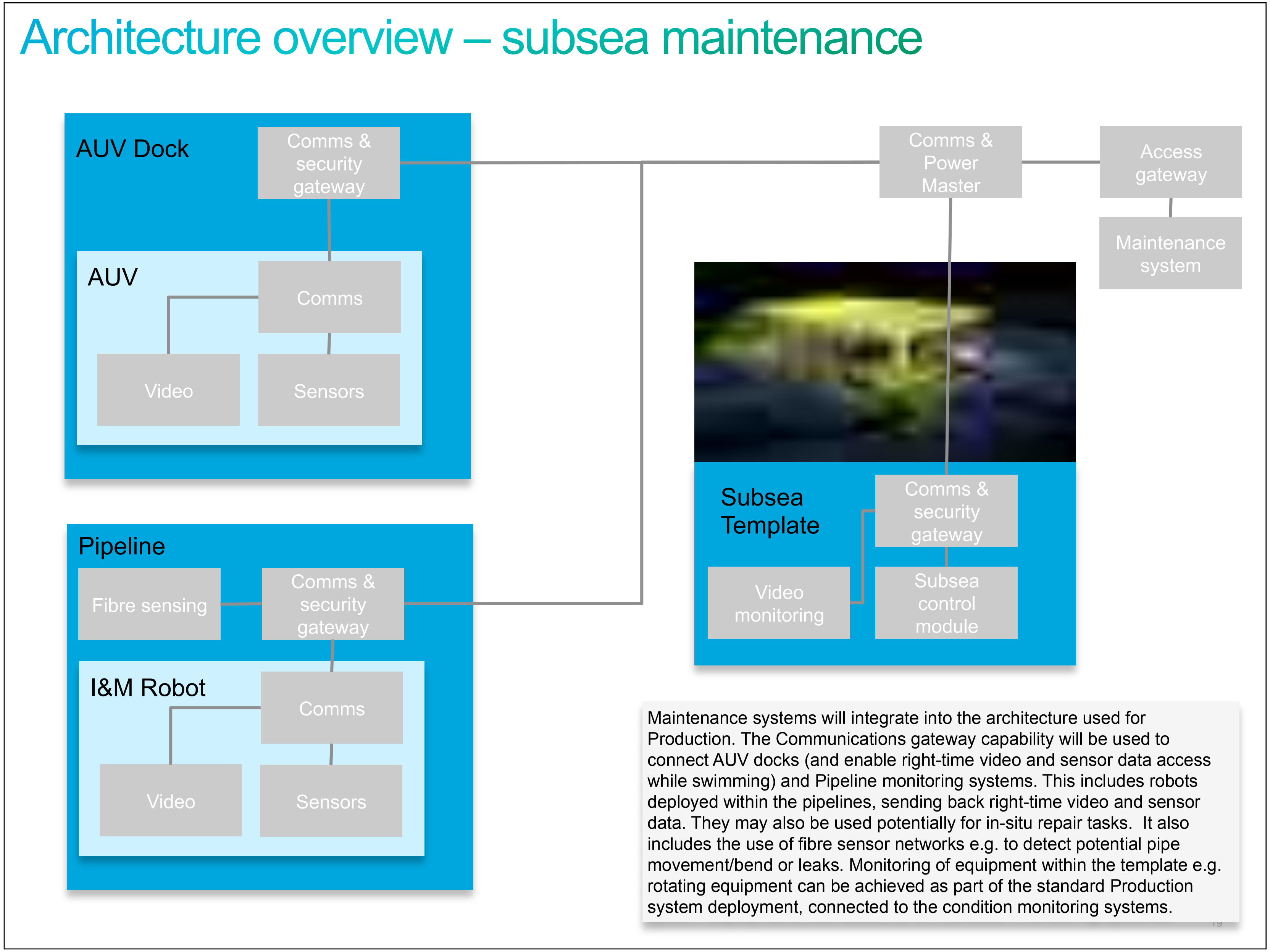
Source: Statoil

The architecture should remain similar longer term as drilling operations potentially move towards seabed drilling. **The main change is likely to be in the Drilling systems – evolving towards control using an integrated drilling network** – this will provide much closer integration of the composite drilling systems that will be required for a higher degree of remote drilling processes in the High North. This should be based on common open standards communications protocols (IP). This enables **true end to end IP connectivity from the IO centre to the subsea sensors and controllers** and will in turn enable greater application of autonomous control. Combined with fibre networks, it will enable greater instrumentation for providing richer data (and video) subsea and downhole. The Comms and Security gateway will support this architecture. For seabed drilling there will be no requirements for unified communications or crew well-being systems, except for mother-ship support where necessary



17


Architecture overview – subsea production
Access gateway
*located either onshore for pure subsea installations – or topside where an FPSO is used
SERVICES
• Routing and switching
• Network virtualisation
• Quality of Service (QoS)
• Application aware networking
• High availability
• Autonomous networking
• Firewall (Comms Master only)
Multi sensor arrays
Comms & Power Master
Resilient, redundant fibre net
Comms & security gateway
Subsea control module
Video monitoring
Subsea Template
Wellhead
Down-hole sensors, video
The communications & power master acts as the primary subsea gateway to the Subsea templates. The architecture enables redundant communications by linking the comms & security gateways deployed within each of the templates. These gateways provide standardised IP networking capability for any subsea systems that need to be monitored or controlled from onshore. This will include functions within the SCM e.g. well control, tree instrumentation, downhole sensors, flowmeters, sand detection, and video monioring on the template, seabed or downhole.

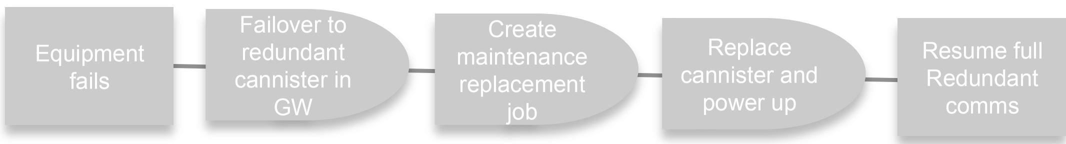

Architecture overview – subsea maintenance
AUV Dock
Comms & security gateway
AUV
Comms
Video
Sensors
Pipeline
Fibre sensing
Comms & security gateway
I&M Robot
Comms
Video
Sensors
Comms & Power Master
Access gateway
Maintenance system
Subsea Template
Comms & security gateway
Video monitoring
Subsea control module
Maintenance systems will integrate into the architecture used for Production. The Communications gateway capability will be used to connect AUV docks (and enable right-time video and sensor data access while swimming) and Pipeline monitoring systems. This includes robots deployed within the pipelines, sending back right-time video and sensor data. They may also be used potentially for in-situ repair tasks. It also includes the use of fibre sensor networks e.g. to detect potential pipe movement/bend or leaks. Monitoring of equipment within the template e.g. rotating equipment can be achieved as part of the standard Production system deployment, connected to the condition monitoring systems.

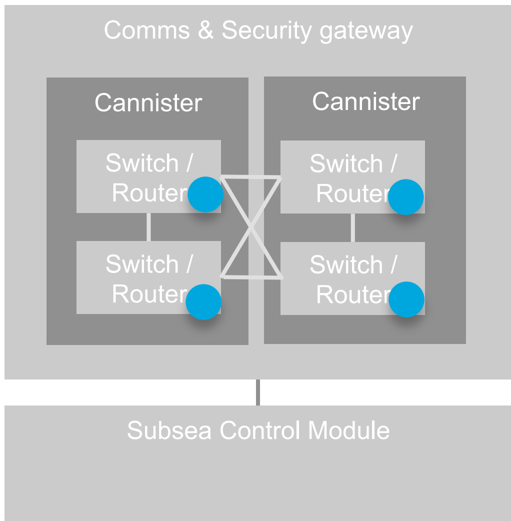
Applied scenario: maintenance – subsea comms equipment replacement

- **Scenario –** it is critical that communications equipment deployed on the seabed remains available to enable continuous subsea production
- **Complication –** equipment may fail during the lifetime of a producing well (20+ years) and need to be replaced. Seabed replacement within a subsea template is complex and expensive
- **Question –** How can we simplify (reduce complexity, time and a cost) the maintenance process for subsea networking equipment?
- **Solution –** Autonomous networking capability within the subsea routing and switching equipment can be used for fast and economical drop deployment of new equipment without the need for pre-configuration or local engineering support. This means a stock of standard replacement equipment can be kept close to the point of activity and deployed (e.g. using an ROV) without having to engage onshore networking experts to configure equipment for use in a specific field and template. The equipment can be replaced as a unit. When powered up it will self-configure using information from nearby networking equipment.

20

Applied scenario: maintenance – comms equipment replacement





If one of the Switch / Routers fails, the network will failover to use a unit in the other Cannister. The first cannister can then be physically removed (potentially without stopping production) and replaced with a new one. When powered up, the autonomous networking service (deployed on each Switch / Router) will self configure using information from equipment in the other cannister or the Comms Power master, and enable fully redundant communications to be resumed without further intervention.
Value – reduces maintenance time and cost.

● Autonomous networking service

21

Applied scenarios: collaborative working environment - incident response

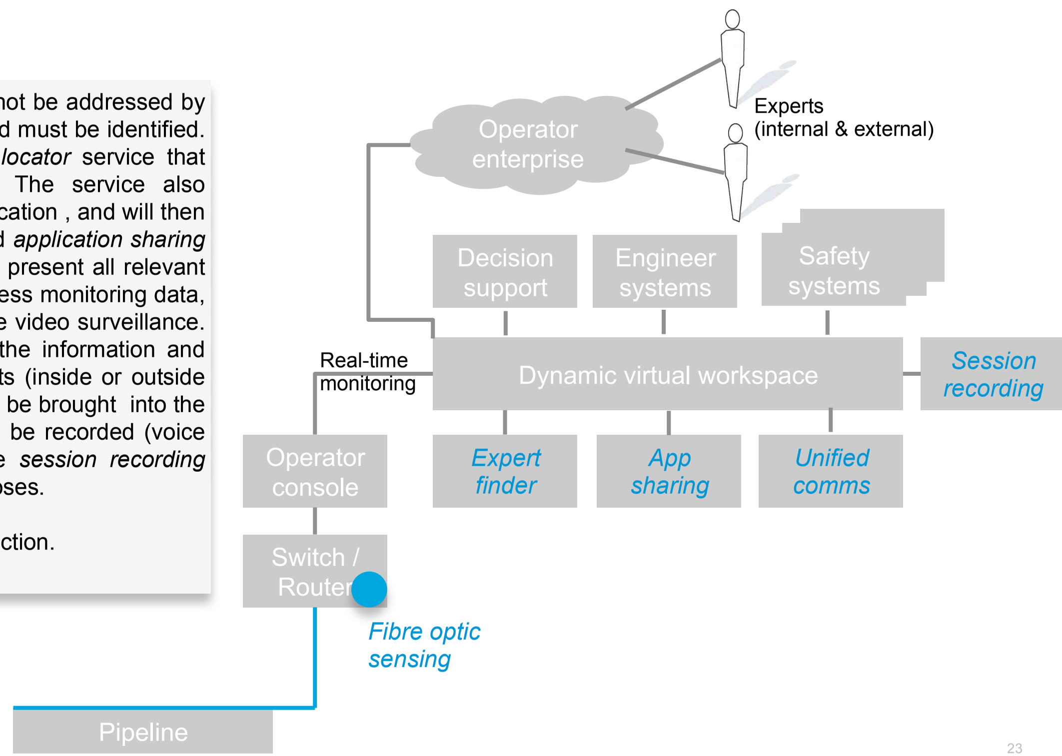
- **Scenario –** it is critical that potential incidents are identified as early as possible, and before they occur or develop (and there are systems within the architecture to help achieve this). However, provision must still be made in case of an incident, with an effective response capability in order to limit the risk and impact
- **Complication –** analysis of the current and evolving situation, as well as deciding upon an appropriate course of action is a complex process given the different expertise that may be required (with limited availability) and the fact that the fields are in remote and harsh locations
- **Question –** How can we rapidly understand the current situation and facilitate appropriate and timely decision making to effectively manage the response to an incident?
- **Solution –** Rapidly identifying the relevant information and experts required, and then providing an environment in which they can engage to analyse, plan and respond

22

Applied scenarios: collaborative working environment - incident response (pipeline leak example)

When an alert is validated as requiring action that cannot be addressed by an operator, the appropriate experts required to respond must be identified. This process can be automated using an *expertise locator* service that matches skills and experience to those required. The service also determines their availability and best mode of communication , and will then create an ad-hoc virtual workspace (*unified comms* and *application sharing* services) and connect them into it. The workspace will present all relevant contextual information e.g. real-time and historical process monitoring data, maintenance history, technical documentation, real time video surveillance. It will enable all experts to see the same view of the information and collaboratively analyse and discuss. Additional experts (inside or outside the organisation) and information can easily and rapidly be brought into the environment as necessary. Collaborative sessions can be recorded (voice video and information /application sharing) using the *session recording* service for future auditing or best practice sharing purposes.

Value – timely and accurate decision making. Risk reduction.



23


Glossary
Application aware networking
network device intelligence to treat application traffic appropriately
Autonomous networking
Self configuring network elements
Firewall
barrier designed to prevent unauthorized or unwanted communications
High availability
system design approach and associated service implementation that ensures a prearranged level of operational performance
MPLS
MultiProtocol Label Switching
Network protocol optimisation
Optimising the performance of TCP-based applications in a Wide Area Network
Network virtualisation
enabling multiple, separate IP networks over a common infrastructure by using MPLS, Virtual Routing and Forwarding (VRF) and Virtual LAN (VLAN)
Quality of Service (QoS) requirements
set of mechanisms to enable predictable transport of traffic with special
Switching
packet switching based on Layer 2 (MAC) information
Routing
packet switching based on Layer 3 (IP) information
24

Report title: Final report: All activities		
Date of first issue: 14.05.2012	*Project:* Integrated Operations in the High North	
Prepared by: Christian M. Hansen, Johan W. Klüwer	*Reviewed by:* Tom Thomsen	*Approved by:* Tom Thomsen
Contributions by:	*Chapter/section:*	
Christian M. Hansen (DNV)	3.1, 3.2.1, 3.3.2, 3.3.3, 3.3.5, 3.3.6, 3.3.7, 3.4.2.4–3.4.2.6, 3.4.3, 3.4.4, 3.4.5, 3.4.6, 3.4.7	
Johan W. Klüwer (DNV)	3.2.3, 3.2.4, 3.3.1, 3.3.4, 3.3.8, 3.4.1, 3.4.2.1–3.4.2.3, 3.4.8, 3.4.9, 3.4.10, 3.4.11, 3.4.12	

Summary:

A prerequisite for the transition to Integrated Operations in O&G is the exchange of information and data across systems and disciplines. However, without the use of common and standardized data formats, data integration and exchange is a costly and hard to maintain.

The IOHN Semantic Model activity addresses the data exchange challenge by demonstrating an integration pilot based on three important technologies: *Linked Data* publication for easy access to shared master data, *ISO 15926-8 Templates* as a standard data representation format, and *Office Semantics*, the use of standard office software both for collecting and maintaining master data, and for distributing master data throughout the organization.

The IOHN integration pilot features a proof-of-concept model of the Snorre B platform with wells, sensors, chokes, valves, pipelines, and separators. The model can be queried for topological information (e.g. “Which components is this pipe connected to?”), and for sensor and production data. The model features inline documentation, and all components are described using a common domain vocabulary.

The IOHN Semantic Model activity has come up with new concepts, as well as novel modifications of existing approaches. The lessons learnt are put to use in several new projects, both research and commercial. Further, the extensive use of open standards in the integration pilot facilitates easy deployment of implementation concepts in a commercial context.

Revision	*SVN*	*Status*	*Date*	*By*	*Reason/Changes*
0.9		Draft	16.05.2012	CMH	For review by PM
0.99	3575	Draft	24.05.2012	FVE	Released for review by Steering Committee
1.00	3592	Accepted	05.06.2012	FVE	Accepted by Steering Committee

Chapter 3

Semantic model

3.1 Summary

A prerequisite for the transition to Integrated Operations in O&G is the exchange of information and data across systems and disciplines. However, without the use of common and standardized data formats, data integration and exchange is a costly and hard to maintain.

The IOHN Semantic Model activity addresses the data exchange challenge by demonstrating an integration pilot based on three important technologies: *Linked Data* publication for easy access to shared master data, *ISO 15926-8 Templates* as a standard data representation format, and *Office Semantics*, the use of standard office software both for collecting and maintaining master data, and for distributing master data throughout the organization.

The IOHN integration pilot features a proof-of-concept model of the Snorre B platform with wells, sensors, chokes, valves, pipelines, and separators. The model can be queried for topological information (e.g. “Which components is this pipe connected to?”), and for sensor and production data. The model features inline documentation, and all components are described using a common domain vocabulary.

The IOHN Semantic Model activity has come up with new concepts, as well as novel modifications of existing approaches. The lessons learnt are put to use in several new projects, both research and commercial. Further, the extensive use of open standards in the integration pilot facilitates easy deployment of implementation concepts in a commercial context.

3.2 Introduction

3.2.1 Thematic Overview

With the transition to Integrated Operations, the O&G industry faces several challenges, some of which are addressed by the IOHN Semantic Model activity. Multi-discipline collaboration in plant operation requires increased focus on *Information Exchange* and *Data Integration* across systems and disciplines. A vast amount of proprietary systems and data formats need to be aligned in order to ensure flow of business-critical information.

There are different approaches to system integration. A point-to-point mapping between each individual system is costly to implement and maintain, since internal changes in one system requires changing all maps for that system. A far better solution is to use common and standardized data formats for data exchange. With this approach, it is sufficient to maintain one mapping to/from the standard format for each system. Internal system changes only requires updating the map to the standard data format.

Data exchange across systems and disciplines increases the demand for *Master Data Management*, i.e. the processes and tools that are used for managing data that are common across systems and disciplines. A consistent management of master data ensures that different systems refer to the same entity in the same way, rather than using individual references. Consistent Master Data Management greatly facilitates data integration and exchange.

The last challenge that is addressed by the Semantic Model activity is *Sustainability*. The lifetime of an O&G field, from exploration, via drilling, to production and decommissioning typically spans 40–50 years. The lifetime of a typical *information system*, on the other hand, spans 10–15 years. As a consequence, business-critical information about wells and platforms will have to be transferred between 3–4 different information systems during its lifetime. In order to ensure a smooth transfer between systems, it is key to avoid application lock-in of information. Using open formats and standards provides vendor independence and facilitates moving data from one system to another.

> Transition to Integrated Operations in O&G raises challenges related to *Information Exchange*, *Data Integration*, *Master Data Management*, and *Data Sustainability*.

In order to address some of the challenges outlined above, the IOHN Semantic Model activity has, with the input from the Production Pilot activity, implemented an integration pilot for Statoil's Snorre B platform. Three important technologies have been put to use in the pilot, as shown in Figure 3.1:

1. Linked Data
2. ISO 15926-8 Templates
3. Office Semantics

The Linked Data publishing paradigm, originally introduced by Tim Berners-Lee, and endorsed by the World Wide Web Consortium (W3C), concerns publication of data in a form that is understandable both by humans and computer systems. In contrast, the web of linked text, as most of us use on a daily basis, is targeted at human readers only, with clickable links between different pages and sites. Linked Data publication ensures that computer systems can browse a web of data, following links between different information, in a similar way as humans do through their web browsers.

The Linked Data paradigm requires that all entities have globally unique identifiers in the form of URIs, much in the same form as today's web page addresses. It is required that the URI identifiers are *resolvable*, i.e. that asking the URI of an entity for information, returns data that describes the entity in a meaningful way, and in a predictable format.

The Linked Data paradigm is highly suitable for Master Data Management, where it is of key importance to be able to easily retrieve information about Master Data elements by both humans and computer systems. By having resolvable URIs as identifiers, a human user can lookup the definition and properties of Master Data elements just by entering the URI identifier in a web browser. A computer system can recieve comprehensible information about Master Data elements by accessing the URI. *The key point is that the access mechanism for Master Data published as Linked Data is readily available, and does not require use of proprietary protocols or formats.*

Note that Linked Data publication does *not* imply that all data must be open to the public. Corporate Master Data can be hidden from the public by using well-known access restriction machanisms, while being easily accessible from the corporate intranet.

> The IOHN Semantic Model integration pilot implements the Linked Data paradigm, and demonstrates how Linked Data publication facilitates easy access to Master Data.

As mentioned above, the O&G challenge of integrating multiple information systems is addressed by using a standardized data format for information exchange. The ISO 15926 standard defines a stack of data representation formats for representing lifecycle information for plants and platforms in the O&G industry, ranging from deep modelling to more high-level formats.

ISO 15926 part 8 defines a *template* format for high-level data representation and exchange. Templates represent data as predicate expressions with a pre-defined set of arguments, easily mappable to proprietary database schemas. As a consequence, ISO 15926-8 templates are highly suit-

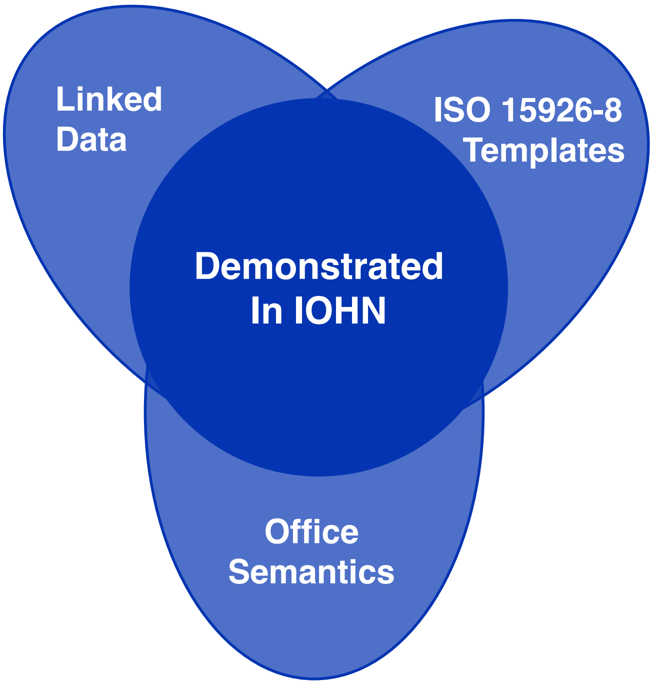


Figure 3.1: The IOHN project demonstrates standards-compliant implementation of ISO 15926-8 Templates, Office Semantics, and Linked Data.

able as a common data exchange format. Further, template data are expressed in open and standard formats recommended by the W3C, which ensures a system-agnostic representation of data, and adresses the Sustainability challenge as well.

The IOHN Semantic Model integration pilot implements data representation as ISO 15926-8 Templates using existing standards and tools.

Proper Master Data Management requires good maintenance tools. Contrary to common belief, such tools does not have to be large and complex. The use of Linked Data publication and data representation based on open standards, opens access to a large box of tools that can be combined and customized in order to obtain a tailor-made toolchain. Further, one can use standard office applications for building and maintaining models of plants and platforms, allowing domain experts to use their modelling application of choice. We use the term *Office Semantics* to refer to the collection and distribution of Master Data using existing off-the-shelf office software tools.

The Snorre B platform model implemented in the Semantic Model pilot is derived from a Microsoft Visio UML drawing of the platform. Further, the vocabulary that defines the components in the Snorre B model is extracted from a Microsoft Excel file.

On the data access side, the simple access mechanism for Linked Data facilitates the use of Master Data in existing software tools, for instance the Microsoft Office suite of applications. In IOHN, there are several examples of dynamically integrating data from the integration pilot in Microsoft Excel and PowerPoint files.

The IOHN Semantic Model integration pilot demonstrates the use of *Office Semantics* for collection and distribution of Master Data.

The IOHN Semantic Model activity belongs to the emerging domain of industrial semantics. On several topics, the activity proceeded from work done in the *Intelligent Data Sets* (IDS) Joint Industry Project, which ran from 2006 to 2008 [37]. Where the IDS project was primarily about the development of new concepts, the Semantic Model activity was explicitly charged with applying and implementing the new concepts in a pilot implementation. The software pilots of the use cases of the other IOHN activities required a model to support integration of data sources and applications.

The implementability requirement gave the Semantic Model activity a strong focus on finding practical, technical solutions.

3.2.2 Participants in this Activity

Main partners in the Semantic Model activity were DNV (activity lead), Epsis, UiO, and OLF. IOHN partners PCA, ABB, Siemens, Baker Hughes, and NOV also made significant contributions. Bechtel (US), representing the iRING project, provided great help in a collaborative effort.

3.2.3 Roadmap, adjustments, and results in brief

A main goal of this activity was to prototype the use of standards for industrial content: to facilitate integration between different domains and work processes by improving consistency. This gave the project a strong focus on using non-proprietary, open vocabularies, and standard formats and access methods. The proprietary formats needed by O&G data sources and applications would be served by means of translation or mapping. The chosen approach was to follow World Wide Web Consortium (W3C) recommendations for schemas, data and access; to apply standard representation patterns according to the ISO 15926 standard; to prefer vendor neutral industry classifications as managed by the POSC Caesar Association (PCA); and to use public registry identifiers where available. This emphasis on open formats was balanced against the need for keeping enterprise data private to the enterprise, with access limited to the Secure Oil Information Link (SOIL) network.

The Semantic Model activity was tasked with building a reference model of selected aspects of the O&G process plant and its information flows. To make this practical, a formalized reference language had to be developed. The solutions developed in the activity would have to satisfy, on the one hand, demands on technical integrity, realism of the data and models, and quality of information and methods – general requirements that belong to the ICT research domain. On the other hand, the Semantic Model activity had to stay close to the O&G subject matter. The solutions needed to provide a method for capturing and modelling the knowledge of domain experts, from the other IOHN aactivities, as they created and developed the industrial use cases.

The aims of the Semantic Model activity required careful consideration of infrastructure and architectural choices, so extensive collaboration with the Integration Platform activity was a given. With emphasis on public vocabularies and access methods, this led to a pilot that in many respects qualifies as a *Linked Data* based architecture, an emerging approach to integration challenges [23].

In March 2009, about half a year into the project, the Semantic Model activity agreed on the roadmap shown in figure 3.2. This plan was followed until late 2010, when a major change in the consortium prompted extensive restructuring of the whole IOHN project. The Semantic Model activity had a leading role in this effort, aiming to preserve the insights and results that had been achieved so far, and to align with recent developments in the semantic technology field. Central to the revision was the replacement of the *test lab* system which had been designated as the pilot execution environment. A recommendation [31] was written, and accepted by IOHN management. On the revised plan, the pilot would implement an integration architecture that conformed to the OLF reference architecture for Integrated Operations (see section 3.4.1), and which followed W3C and ISO 15926 standards to the letter. The revised approach reinforced IOHN's focus on compliance with open standards. In spite of the substantial changes that had to be made, the stated original goals were to a large degree satisfied at project closing, in some aspects with results that went beyond initial ambitions.

We may compare the 2009 list of high-level goals (figure 3.2) to what has been achieved at IOHN's closing in 2012. The following list provides pointers to sections in this report that give further details on each point.

Main deliverable 1: Extended and improved Oil & Gas ontology. Partly successful. At project closing, a viable ontology for sand and erosion is available (section 3.4.5). Drilling resources have been developed, but not integrated with Semantic Model ontologies (cf. section 3.4.7, and the report of the IOHN Drilling Pilot activity). The intention was to extend the PCA Reference Data Library (RDL) with new domains, but this was not implemented; a milestone was reached with RSM classes, but these were later withdrawn (section 3.4.10).

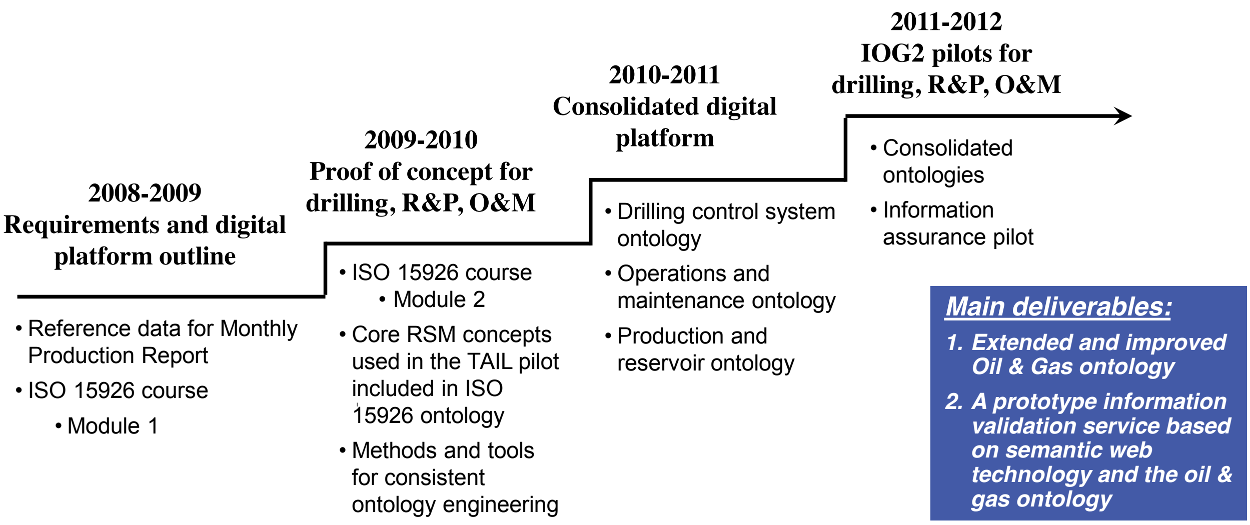


Figure 3.2: Roadmap for the Semantic Model activity, March 2009

Main deliverable 2: Prototype information validation service. Successful. The Semantic Model activity delivered models and data using ISO 15926-8, for which consistency criteria are precisely defined by virtue of the semantics alone (sections 3.3.2, 3.3.3, and 3.4.2). Vital validation criteria for RDF and ISO 15926 have been identified and prototyped (section 3.4.6).

Reference data for Monthly Production Report. Successful, see section 3.4.8.

ISO 15926 course. Partly successful, see section 3.4.9.1. Module 1 delivered, Module 2 not delivered.

Core RSM concepts used in the TAIL pilot included in ISO 15926 ontology. Partly successful; see section 3.4.10.

Methods and tools for consistent ontology engineering. Successful. See section 3.4.4 on the generic methods, and 3.4.2–3.4.5 on industry domain challenges.

Drilling control system ontology. Partly successful. Executed in the Drilling Pilot activity, reference data was developed in interaction with the Semantic Model activity. Integration with other IOHN ontologies was not achieved. See section 3.4.7.

Operations and maintenance ontology. Minor success, as only a small part of the planned O&M activity in IOHN was realized. See sections 3.4.2.5 and 3.4.11.

Production and reservoir ontology. Successful. See section 3.4.2.5.

Consolidated ontologies. Successful, but with reduced scope. A stack of ontologies has been developed and applied, see section 3.4.2. Of the planned ontologies for production, drilling, and maintenance, only the former was developed significantly.

Information assurance pilot. Partly successful. Validation concepts have been developed (see above) on which a service for information could be built, but automated validation has not been developed beyond initial tests (sections 3.4.6.2 and 3.4.6.3).

The consortium change of late 2010 brought a reduction in the resources available to the project. Some of the 2009 goals had to be dropped, while others were replaced by new ones. Noteworthy *new* activities and results include the following.

Office software integration. This has been demonstrated in several ways, both by extracting semantic content from Microsoft Office documents (Excel/Visio), and by enriching Microsoft Office documents with semantic content by means of live queries and linking to data by URIs. (Integration is not restricted to Microsoft products, but can be applied in virtually any office software product.) Cf. sections 3.4.2.1, 3.4.4.4, 3.4.4.3, and 3.4.6.2.

Public registry integration. See section 3.4.3 on how public identifiers for fields and wells were applied directly in the Snorre reference model.

Linked Data. See section 3.4.2 on how IOHN vocabularies and models have been made available as dereferencable resources, and for query at

SPARQL endpoints.

Staged abstractions of semantic content. The Semantic Model content provides a stratified approach that satisfies the ISO 15926 standard, for compliance and coverage of the information scope, while simultaneously providing a simplified layer for convenient access. Cf. section 3.4.4.5.

3.2.4 Impact

The IOHN Semantic Model activity has been a research and development effort, with a "proof of concept" focus on new methods. Impact is therefore to be expected in the first instance in continued research and development, with commercial utility following with some delay.

For the research aspect, it is noteworthy that new projects will benefit directly from the experiences made in IOHN. There is a clear continuity of topics, methods, and teams, ensuring that the methods developed in the Semantic Model activity will be taken further. Especially noteworthy are the following two projects.

- The Norwegian Joint Industry Project *Integrated Environmental Monitoring (IEM)*, 2011, with partners Kongsberg, IBM, and DNV, budget ca. € 20M ☞.
- The European Union Large-scale integrating project (IP) *Optique: Scalable End-user Access to Big Data*, 2012, with 12 partners across Europe, budget ca. €10M.

These projects will benefit from the IOHN approach to domain concepts, modular ontologies, Linked Data, validation, and application of ontology classifiers in an integration platform.

The IOHN Semantic Model activity was headed by DNV's Information Risk Management (IRM) department. At DNV, the IOHN experience has been put to work in a variety of projects, spanning several industrial domains. Engagements include *Equipment Hub* and *Reporting Hub* for the E&P Information Management Association (EPIM), 2009–2011; equipment catalogs for the Russian Research Institute for Nuclear Power Plant Operation (VNIIAES), 2011; ontology development for The European Organisation for the Safety of Air Navigation (EUROCONTROL)'s *Skybrary*, 2011, and Master Data for engineering at Aker Solutions, 2012.

For the PCA–Fiatech collaboration *Joint Operational Reference Data* (JORD, 2010–), the Semantic Model methods have enabled rapid prototyping of an improved ISO 15926 reference data repository, which is scheduled to be used by companies worldwide.

There is consensus among the IOHN Semantic Model participants that the activity has been a great contribution to developing knowledge and capabilities in the new field of semantic technologies. This may be the most important immediate impact of the IOHN project – developing competence, and thereby the ability of the involved companies to deliver products and services that utilize new developments in ICT. To measure this impact in economic terms is difficult, and we will not venture estimates. (There is an abundance of anecdotal evidence and "best practice" recommendations to favor semantics in enterprise information management, but minimal research has so far been done on objective measures of business value.)

The final deliveries of the Semantic Model activity have more of a research character than was envisioned in the original plans of 2008. The change in the consortium, in 2010, shifted the activity from a setting close to the operational reality at an industrial partner (Statoil), to a "laboratory" setup several steps removed from the production and collection of data. This moved the project from operational challenges to a more exploratory mode, with predictable attendant up- and downsides. The benefits are found in how the Semantic Model activity was able to explore and test fresh developments in semantic technology, particularly with regard to the Linked Data approach that is driving new methods in master data management. The disadvantages of a weaker connection to industrial users are however also significant, if hard to measure. The loss of direct contact between IOHN development teams and target users in industry has certainly incurred some *opportunity cost*, as closer collaboration would have helped the transfer of knowledge.

3.3 Preliminaries

In this section, we present the standards and tools that form the basis of the IOHN Semantic Model activity deliveries. During the time span of the IOHN project, standards and tools have matured considerably, and new concepts have been intro-

The IOHN project has piloted O&G data integration using **open standards** and **ISO 15926 representation** of production plants. Data and model information can be queried using a **common domain language**, tightly integrated with standard office software.

duced. A timeline view of relevant standards and projects is given in figure 3.3, p. 54. The year-by-year juxtapositions illustrate how the IOHN project has absorbed and implemented standards at an early stage, in several cases before a standard received its final form. Following a discussion of the timeline, the standards are presented in more detail.

3.3.1 Standards

Semantic technology, methods, and tools have matured considerably during the project period.

The ISO 15926-2 *data model* [34] was published at about the same time as W3C's Resource Description Framework (RDF, see section 3.3.2), and the first version of Web Ontology Language (OWL, see section 3.3.2). An important precursor of IOHN was the Intelligent Data Sets project (IDS, 2006–2009), funded by the Research Council of Norway. IDS pioneered the use of OWL in translating content using full ISO 15926 information models. Several of the concepts that were first developed in IDS have been demonstrated in pilot implementations during IOHN. One case in point is the application of ISO 15926 *templates* (see section 3.3.5) using W3C's semantic technology standards: The W3C workshop position paper "ISO 15926 templates and the Semantic Web" [17], written by the IDS project, outlined a program for the merging of models and formats that was implemented in the course of the IOHN Semantic Model activity.

The POSC Caesar Association's Reference Data Library (PCA-RDL) was made available with an on-line client in 2006, and provided an OWL representation of that library from 2007. This provided a basis and a target for IOHN: The project would deliver data using the W3C standards, following the ISO 15926 patterns for semantics in the industrial domain. The project would place the semantic content it produced in the custody of POSC Caesar, for open access and eventual standardization for the relevant industrial domains. Already in 2008, the Semantic Model activity contributed to the PCA-RDL, for O&G reporting (section 3.4.8).

The IOHN project benefited by standards development at ISO and W3C, which was supported by great advances in semantic software technology. By October 2010, the ISO 15926-7/-8 *statement template* methodology was mature enough that IOHN could arrange a tutorial workshop, to agree on a common work process (section 3.4.9.2). The simplified approach of ISO 15926-8 allowed IOHN to deliver a fully ISO 15926 compliant vocabulary, and to apply it in the asset model of its pilot system.

ISO 15926-8 prescribes RDF for the representation of data. To access data, W3C provides the SPARQL query language (see sections 3.3.2 and 3.3.8), and this was adopted for the IOHN pilot system. Starting in October 2009, SPARQL has seen major advances (under the name "SPARQL 1.1"), several of which have been of great practical utility to the Semantic Model activity.

During the IOHN project period, software for semantics standards has developed rapidly. This includes the widely used *ARQ* reference implementation of SPARQL ☞, efficient reasoners for OWL, including *Fact++* ☞, and RDF server/Linked Data software.

3.3.2 RDF, OWL & SPARQL

The Resource Description Framework (RDF) [39] is a family of World Wide Web Consortium (W3C) specifications originally designed as a metadata data model. RDF has become a general method for conceptual description or modeling of information that is implemented in web resources. There are several syntax formats available for RDF, of which RDF/XML, the XML serialization of RDF, is commonly used for RDF exchange on the web. The Turtle/N3 syntax formats are, however, easier to read and will be used to express RDF in the remainder of the Semantic Model report.

The basic RDF statements are *triples*, consisting of a subject, a predicate, and an object, written in Turtle/N3 syntax as follows:

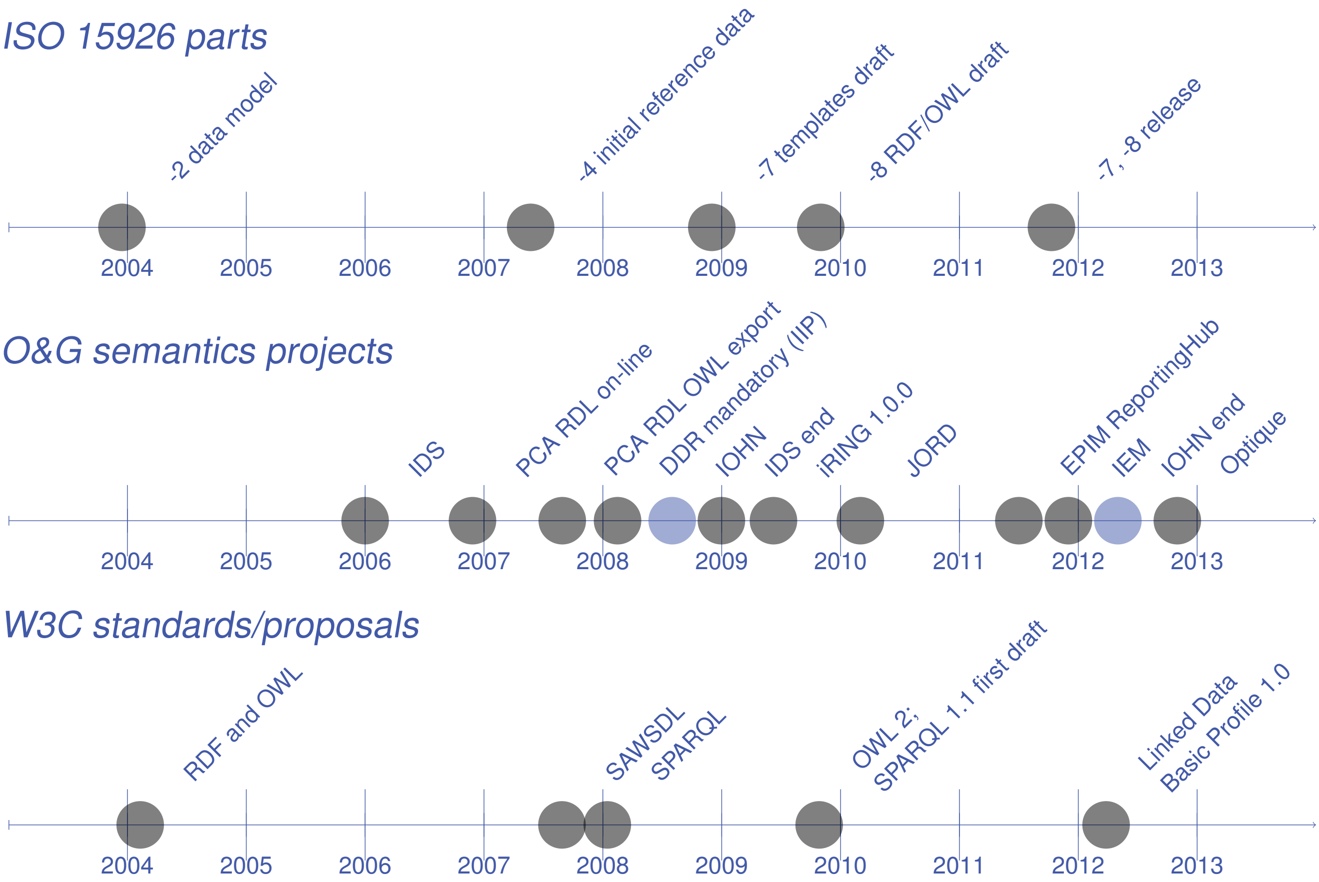


Figure 3.3: Timeline. ISO 15926, O&G semantics projects, and W3C standards

```
subject predicate object .
```

An RDF triple is meant to express that the `subject` is related to the `object` by means of the relation `predicate`. The syntax

```
subject predicate1 object1 ;
        predicate2 object2 ;
        predicate3 object3 .
```

expresses that `subject` is related to `object1` by means of the relation `predicate1`, to `object2` by `predicate2`, and to `object3` by `predicate3`. An RDF *RDF graph* is a set of RDF triples.

The Web Ontology Language (OWL) [13, 38] is a family of knowledge representation languages for authoring ontologies. The languages are characterized by formal semantics and RDF/XML-based serializations for the Semantic Web. OWL is endorsed by W3C and has attracted academic, medical and commercial interest. There are several specifications of OWL, and OWL 2 refers to the 2009 specification. There are several OWL editors and reasoning tools available, both commercial and open-source.

SPARQL [40] is an RDF query language standardized by W3C. The SPARQL language constructs contain support for querying required and optional RDF graph patterns along with their conjunctions and disjunctions. SPARQL also supports extensible value testing and constraining queries by source RDF graph. The results of SPARQL queries can be tabular result sets or RDF graphs.

The W3C standards RDF, OWL, and SPARQL provide flexible schemas to serve integration, precise semantics to enable reliable validation, and network friendly formats for exchange using the Internet.

3.3.3 Ontology and Model

The term *ontology* has its origin in philosophy, where it refers to the study of the nature of being, existence, or reality as such, as well as the basic categories of being and their relations. In computer

The IOHN semantic model is built with W3C and ISO standards. Standards compliance means the IOHN results are ready for reuse in new projects.

science and information science, however, an ontology formally represents knowledge as a set of concepts within a domain, and the relationships between those concepts. It is the latter definition of ontology that is used in the context of the IOHN project.

Contemporary ontologies share many structural similarities, regardless of the language in which they are expressed. The ontology parts that are relevant to the IOHN project are as follows.

Classes Types of objects.

Relations Ways in which classes and individuals can be related to one another.

The ontologies developed and used in the IOHN project are expressed in the OWL 2 language (see section 3.3.2), with relations expressed in ISO 15926 part 8 Template format (see section 3.3.5).

The term *model*, in the context of this report, means a description of real world objects using classes and relations from one or more ontologies. Observe that a model contains what is referred to in ISO 15926 as *instances*, and a model can therefore be viewed as an instantiation of concepts described in ontologies.

3.3.4 Reference Data and ISO 15926

Information concerning engineering, construction, and operation of production facilities is created, used and modified by different entities throughout a facility's lifetime. The purpose of the standard ISO 15926 is to facilitate integration of data to support the life-cycle activities and processes of production facilities. The IOHN project has adopted this standard for both *instance* and *reference data*. Definitions of these terms, which are central to the Semantic Model activity, are given in the standard's *part 1*.

> *reference data*
> process plant life-cycle data that represents information about classes or individuals which are common to many process plants or of interest to many users. [33, p. 6]

> *instance*
> data that represents, in computer processable form, some real-world thing. [33, p. 5]

Of particular interest is the notion of *life-cycle* information, a term which is used to emphasize the need for tracing the assets and activities of a process plant (in the IOHN context: the O&G platform, with its parts and attendant processes) through time and in successive stages.

> *process plant life-cycle data*
> data that represents, in computer processable form, information about one or more process plants in or throughout any phases of their life.
> NOTE The phases of the life of a process plant may include design, engineering, construction, operation, maintenance, decommissioning and demolition. [33, p. 5]

Reference data needs to be collected and managed to be of interest; a collection is called a Reference Data Library (RDL). The literal definition is as follows.

> *reference data library (RDL)*
> managed collection of reference data. [33, p. 6]

RDLs can come in many forms, ranging from plain vocabulary lists to complex models. ISO 15926 has great expressive power and can be used to build complex libraries. Ontologies with logic-based semantics are naturally suitable for representing RDLs that follow this standard.

For efficient management and use of complex model patterns, ISO 15926 provides a *template* specification, given in parts 7 and 8 of the standard; these parts were available in draft form to the IOHN project, and were standardized by ISO in late 2011. The Semantic Model activity provided early implementations of these parts of ISO 15926. The work in IOHN has contributed to the understanding of how the full scope of the standard can be put to work for practical purposes. While *Part 7* has a formal character aimed at highly explicit modelling, *Part 8* provides a highly simplified format for encapsulating complex patterns and applying them in practical settings, in particular for purposes of information exchange. It is the latter form that has been implemented for IOHN. Selected definitions, from parts 7 and 8, respectively are worded as follows.

> *template* [the ISO 15926-7 abstract notion]
> set comprising of a first-order logic predicate for which a definition is stated as an axiom, a template signature and a template axiom expansion. [35, p. 4]

> *template* [the ISO 15926-8 RDF/OWL format]
> n-ary predicate, represented in OWL reified form as a class with one functional property (role) per variable. [36, p. 6]

For the IOHN project, following ISO 15926 was a given requirement. Early on it was decided to employ the abbreviation mechanism provided by templates, for reasons of efficiency. The original intention was to provide "full" modelling, in accordance with *Part 7*, for all templates/patterns employed in the project. This was carried out for the RSM mapping (section 3.4.10), but resource constraints meant that fully explicit "deep modelling" according to this approach had to be dropped for the final model's plant topology and measurement data templates.

Modelling in semantic languages is not yet mainstream. To enable the IOHN teams to develop ISO 15926 patterns, basic concepts in semantics were discussed and exemplified in several workshops, and a practical tutorial on building Part 8 templates was held in August 2010 (see section 3.4.9.2).

3.3.5 ISO 15926 Templates

An RDF statement consists of three parts: a subject, a property/relation, and an object. However, in many situations one would like to specify properties of the relation between a subject and an object. For instance, assume that a well C2 has a gas/oil ratio of 0.5. This information can be expressed with an RDF triple: (In order to increase readability the syntax is a bit informal; in real-world RDF one would use URIs for the well C2 and the property/relation.)

```
C2 gasOilRatio 0.5
```

What is not captured with this representation is on what date the oil/gas ratio was 0.5 for the well C2. So, given the subject-property-object structure of RDF statements, how can we express the date that the oil/gas ratio was measured? Or, in general, how can we express properties of relations in RDF? Here, ISO 15926 *templates* come to the rescue, with a standardized way to express complex facts. We distinguish between a template *signature* and a template *instance*. A template instance is an expression of the form

```
TemplateName(arg1,arg2,...)
```

where TemplateName is the name of a template, and arg1, arg2, ... are the arguments. For the above example with a gas/oil ratio measurement, assume that we have a template GasOilRatio. A template instance

```
GasOilRatio(C2,01.12.2010,0.5)
```

may express that the well C2 had a gas/oil ratio of 0.5 on December 1st 2010.

A template signature describes a template: the name of the template, the number of arguments, and the type of each argument. ISO 15926 also requires any template to be provided with an (informal) definition that is unambiguous and, as far as possible, phrased in terms of the subject domain. For the GasOilRatio Template, the definition may be worded as follows; note how it is made explicit that this template is for *daily averaged* values only.

> GasOilRatio is a Template that takes three arguments; the first argument is a well, the second argument is a date, and the third argument is a double value. The template expresses that the *daily averaged gas/oil ratio* measured in Sm^3/Sm^3 for this well and date is of the given double value.

An RDF triple store holding template signatures and instances will be structured as follows. There is a top-level class named Template, of which all templates are subclasses. For each argument in a template, the RDF properties argumentXRole and argumentXType provide, respectively, the role that the argument plays, and the type of things that can fill this role. For the example template GasOilRatio, we might end up with something like

```
GasOilRatio rdfs:subClassOf Template ;
   argument1Role hasWell ;
   argument1Type Well ;
   argument2Role valDate ;
   argument2Type xsd:Date ;
   argument3Role valDouble ;
   argument3Type xsd:double .
```

Having declared the property names for each of the arguments of OilGasRatio, we can now represent the above instance in RDF as follows.

```
measurement328 a GasOilRatio ;
   hasWell C2 ;
   valDate "2010-12-01"^^xsd:Date ;
   valDouble "0.5"^^xsd:double .
```

We have declared that *measurement328* is an instance of the template class OilGasRatio, and used the property names from the template signature to specify the values of each argument.

3.3.6 RESTful Web Services

Representational State Transfer (REST), introduced in [5], is an architectural style for distributed systems. In a REST system, each “thing” or resource must have a global identifier. When a client accesses a resource by using its global identifier, the server transfers a *representation* of the resource to the client. A key property is that the server cannot pertain client state information between client requests; all information needed by the server to process the request, must be contained in the request information received from the client. The transfer of a resource representation from the server to the client can be viewed as transferring the *state* of the resource to the client.

A RESTful web service is a web service that adheres to the REST architectural style. RESTful web services communicate over the HTTP protocol [4], and URIs are used as global identifiers of resources. The HTTP Request Methods are used to interact with URI resources as follows. A client requests a representation of a resource by passing a HTTP GET request to the resource URI. In order to replace a resource with a new representation, the client passes a HTTP PUT request, and new resources can be created by passing HTTP POST requests, and, finally, resources are deleted with HTTP DELETE requests.

RESTful web services are not the answer to all distributed system design problems. There are, however, many benefits of designing a RESTful web service. The stateless nature of REST results in a scalable system, since the server does need to keep track of client state between requests. Further, the use of HTTP Request Methods for interacting with system resources provides a simple and general system interface. Also, the large amount of client systems and appliances that implement the HTTP protocol provides a very large base of system users. Finally, the concept of URIs and the HTTP protocol are open standards available for everyone to use. If combined with a resource representation format that is also standardized, like XML or RDF, a system implemented as a (set of) RESTful web service(s) is transparent and avoids locking in business critical data in proprietary systems and formats. Quoting IBM’s article on RESTful web services [28]:

> Exposing a system’s resources through a RESTful API is a flexible way to provide different kinds of applications with data formatted in a standard way. It helps to meet integration requirements that are critical to building systems where data can be easily combined (mashups) and to extend or build on a set of base, RESTful services into something much bigger.

3.3.7 Linked Data

The term *Linked Data* refers to a set of best practices for publishing and interlinking data on the World Wide Web. The best practices were introduced by Tim Berners-Lee in [3] and can be summarized as follows.

1. Use URIs as names for things.

2. Use HTTP URIs, so that people can look up those names.

3. When someone looks up a URI, provide useful information, using the standards (RDF, SPARQL).

4. Include links to other URIs, so that they can discover more things.

Quoting [12]: ”The basic idea of Linked Data is to apply the general architecture of the World Wide Web to the task of sharing structured data on global scale. In order to understand these Linked Data principles, it is important to understand the architecture of the classic document Web.

The document Web is built on a small set of simple standards: Uniform Resource Identifiers (URIs) as globally unique identification mechanism, the Hypertext Transfer Protocol (HTTP) as universal access mechanism, and the Hypertext Markup Language (HTML) as a widely used content format. In addition, the Web is built on the idea of setting hyperlinks between Web documents that may reside on different Web servers.

The development and use of standards enables the Web to transcend different technical architectures. Hyperlinks enable users to navigate between different servers. They also enable search engines to crawl the Web and to provide sophisticated search capabilities on top of crawled content. Hyperlinks are therefore crucial in connecting content from different servers into a single global information space. By combining simplicity with decentralization and openness, the Web seems to have hit an architectural sweet spot, as demonstrated by its rapid growth over the past 20 years.

Linked Data builds directly on Web architecture and applies this architecture to the task of sharing data on global scale."

To summarize, the Linked Data paradigm aims at making data accessible for software systems on the web. In an O&G context, the Linked Data paradigm is particularly useful for publishing shared reference data, aimed for use by both expert users and software systems. The RDF/XML format, the de facto format for transfer of Linked Data, is suitable for machine parsing of Linked Data, but not easily accessible for human readers. In order to present Linked Data in a form that is readable to humans, one can use the Content Negotiation mechanism of the HTTP protocol to serve different version of a resource depending on the client type. RDF-aware software clients can receive an RDF representation, while human users can see a nicely formatted HTML version of the same resource. Figure 3.4 shows an example of a HTML representation of the class `Wellbore` in the PCA Reference Data Library.

Figure 3.4: The PCA RDL's rich set of O&G classes is available as Linked Data ☞

Nally and Speicher, IBM, propose in [22] (and submitted to W3C in [23]) a *Basic Profile*[1] for Linked Data, providing best practices and anti-patterns for Linked Data publishing, classified into four categories: *resources* for basic principles of reading and writing Linked Data, *containers* for subjects related to working with collections of Linked Data items, *paging* for mechanisms that splits large resources into incrementally downloadable pages, and *validation* for properties that describe validation of Linked Data resources.

For the *resources* category, they extend the four Linked Data principles of Berners-Lee with the following rules.[2] After each rule we indicate to what extent the IOHN Integration Layer pilot, ontologies and Snorre B platform model implements the rule.

1. *Basic Profile Resources are HTTP resources that can be created, modified, deleted and read using standard HTTP methods.*
   Basic Profile Resources are created by HTTP POST (or PUT) to an existing resource, deleted by HTTP DELETE, updated by HTTP PUT or

[1] Defined by Nally and Speicher in [22] as a "specification that defines the specification components needed from other specifications, plus provides clarifications and patterns".

[2] The text is a shortened version of the one found in [22].

PATCH, and "fetched" using HTTP GET. Additionally, Basic Profile Resources can be created, updated, and deleted by using SPARQL Update.

IOHN implementation status: Partially. All URIs used as resource identifiers in the IOHN ontologies and platform model are resolvable, and can be fetched with HTTP GET requests. SPARQL Update is provided for the sand detection use case of the Production Pilot activity, but resource modification using HTTP POST/DELETE/PATCH is not supported.

2. *Basic Profile Resources use RDF to define their states.*
   The state of a Basic Profile Resource (in the sense of state used in the REST architecture) is defined by a set of RDF triples.

   IOHN implementation status: Fully. All IOHN ontologies and the Snorre B platform model are expressed as RDF, and when a resource URI is accessed, a suitable set of RDF triples is returned.

3. *You can request an RDF/XML representation of any Basic Profile Resource.*

   IOHN implementation status: Fully.

4. *Basic Profile clients use Optimistic Collision Detection during update.*
   Because the update process involves getting a resource first, and then modifying it and later putting it back on the server, there is the possibility of a conflict (for example, another client might have updated the resource since the GET action). To mitigate this problem, Basic Profile implementations should use the HTTP If-Match header and HTTP ETags to detect collisions.

   IOHN implementation status: Depends on the implementation details of SPARQL Update in the Anzo server of Cambridge Semantics.

5. *Basic Profile Resources use standard media types.*
   Basic Profile does not require and does not encourage the definition of any new media types. A Basic Profile goal is that any standards-based RDF or Linked Data client be able to read and write Basic Profile data, and defining new media types would prevent that in most cases.

   IOHN implementation status: Fully. Only RDF/XML and HTML are supported.

6. *Basic Profile Resources use standard vocabularies.*

   IOHN implementation status: Fully. Both PCA-RDL and ISO 15926 are used, see sections 3.3.1, 3.3.4, 3.4.1, and 3.4.2.

7. *Basic Profile Resources set rdf:type explicitly.*

   IOHN implementation status: Fully. See section 3.4.6.1.

8. *Basic Profile Resources use a restricted number of standard data types.*

   IOHN implementation status: Fully. The only literal data types in use are the XML Schema data types xsd:dateTime, xsd:double, xsd:integer, and xsd:string.

9. *Basic Profile clients expect to encounter unknown properties and content.*
   When doing an update using HTTP PUT, a Basic Profile client must preserve all property values retrieved by using HTTP GET. This includes all property values that it doesn't change or understand. (Use of HTTP PATCH or SPARQL Update rather than HTTP PUT for updates avoids this burden for clients.)

   IOHN implementation status: Not applicable. The Integration Layer pilot with ontologies and platform model concerns the server side.

10. *Basic Profile clients do not assume the type of a resource at the end of a link.*

    IOHN implementation status: Not applicable. The Integration Layer pilot with ontologies and platform model concerns the server side.

11. *Basic Profile servers implement simple validations for Create and Update.*
    Basic Profile servers should try to make it easy for programmatic clients to create and update resources. If Basic Profile implementations associate a lot of very complex validation rules that need to be satisfied for an update or creation to be accepted, it becomes difficult or impossible for a client to use the protocol without extensive additional information specific to the server that needs to be communicated outside of the Basic Profile specifications. Additional checks that are required to implement more complex policies and constraints should result in the resource being flagged as requiring more attention, but

should not cause the basic Create or Update action to fail.

IOHN implementation status: Fully. No validation of Update is performed on the server.

As indicated by the IOHN implementation status of each rule, the IOHN Integration Layer pilot, ontologies, and Snorre B platform model implement most of the Linked Data basic profile rules, and can be viewed as a compliant Linked Data implementation.

3.3.8 Query language: SPARQL, and alternatives

In a traditional setup, getting access to production data across installations is not an easy task. Different installations have different proprietary systems and formats, and documentation is often hard to get hold of. One of the main contributions of the IOHN project is to provide a pilot of an Integration Platform that will provide topological information, i.e. how different sensors and pumps are connected, documentation, and sensor/production data expressed in a common nomenclature.

During the first two years of the IOHN project, it was not obvious which query language, or indeed languages, should be adopted and applied to the pilot system. Table 3.5, from a workshop on query languages (workshop arranged by the Integration Platform activity, 2009.08.25) shows a comparison between the languages IIF QL, SQL, JPQL, and SPARQL, all of which were observed to have strengths and weaknesses [25]. In addition to these generic choices of languages, it was observed that applications might require interfaces with more specialized, domain-specific languages [25, slide 8], and this informed both the architectural and the modelling work of the project.

3.4 Tasks in detail

3.4.1 Architecture

In this section, we review the development of the system architecture adopted for and piloted in IOHN, with a view to showing how the activity Semantic Model is intimately tied to characteristic considerations on architecture that the IO enterprise needs to make.

The activities in Semantic Model of IOHN are directly motivated by the architecture that is widely recognized to be required for Integrated Operations. The enterprise data sources are many, the applications are many, and great resources are wasted because the systems don't speak the same language. The first task of IOHN activity Semantic Model is therefore to provide a generic language for talking about the types of entities in O&G enterprise information (this language is sometimes referred to as a "meta-model"). The second task is to apply this language in a reference model for the enterprise, a rich registry of the specific assets, activities, and streams of data that are managed by the particular enterprise. The Semantic Model activity is of vital importance to an IO system, specifying the canonical form of information that is mediated by the integration layer.

A March 2008 report from OLF described an Integrated Operations *reference architecture* (a generic description of architectural principles to be followed in setting up systems for an IO enterprise) [24]. This report has informed the work in IOHN throughout. The report advises that (p. 14),

> To establish flexible solution architecture views . . . , three important dimensions . . . needs [sic] to be considered:
>
> - The integration pattern
> - The semantic model
> - The services pattern

The *semantic model* aspect is illustrated with some detail in the following figure, where some industry standards that were expected to cover the domains of information needed are indicated.

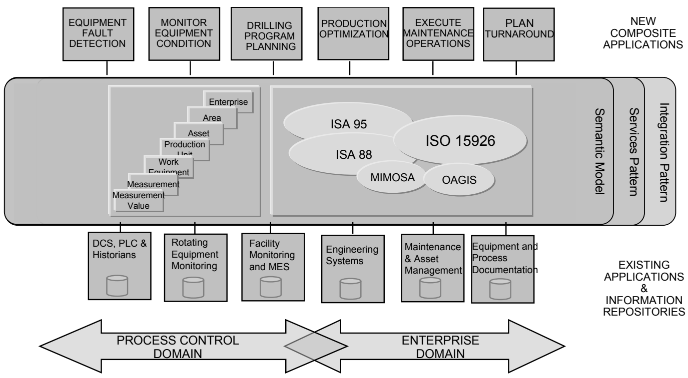


Figure 3.6: OLF reference architecture: "The Semantic Model" [24, p. 14]

Following the OLF reference architecture, the IOHN architecture was designed for the task of

	IIF QL	SQL	JPQL	SPARQL
Acceptance in the industry	IBM Proprietary brand-new	Standard long-existing	Standard established	Standard emerging
Underlying ‚modeling‘ language	RSM	SQL schema definition part	UML/Java Entities	OWL/RDF
Programming model	Proprietary RSM	Generic relational	Generic object-oriented	Generic ontology-oriented
Implication for client applications	• Requires RSM knowledge • Requires understanding of model instance structure • Requires knowledge of RSM specific query API	•Requires RSM knowledge • Requires understanding of model instance structure • Requires deep SQL skills	•Requires RSM knowledge • Requires understanding of model instance structure • Requires deep JPQL skills	•Requires RSM knowledge • Requires understanding of model instance structure • Requires deep SPARQL skills

Figure 3.5: Query languages considered for IOHN [25, slide 57]

uniting major streams of enterprise information, including static design data as well as measurement logs in *historians* and real-time streams form operations.

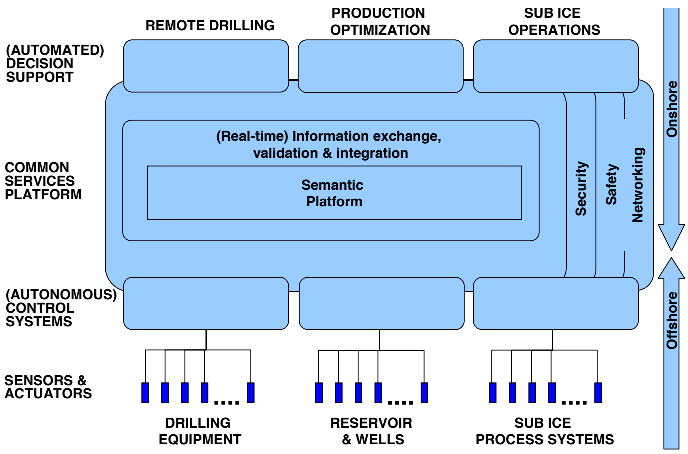


Figure 3.7: "Architecture based on Reference Architecture for OLF IOG2" [18, slide 8]

An *enterprise service bus* would collect and transform data from sources. Client applications would access data by requests to the integration layer. The integration layer would present information in a unified form (i.e., according to the RSM meta-model), structured by the enterprise model. Activity Semantic Model worked to implement the details of how the enterprise model would be set up in detail, with a reliable mapping to the vendor- and domain-independent ISO 15926. This mapping was seen as key to translating the information from diverse systems and standards into a single, common language.

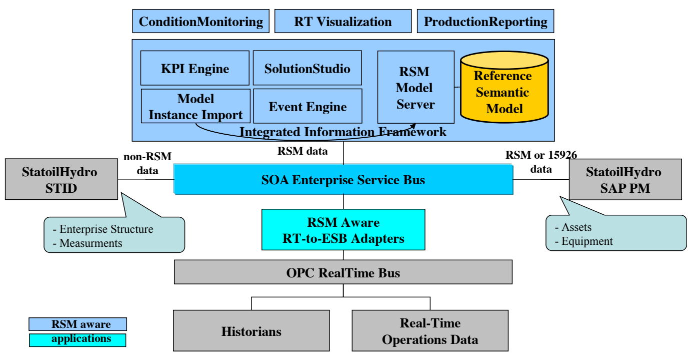


Figure 3.8: Overview of IIF architecture, March 2009 (from IOHN presentation at OLF)

A similar diagram, with greater emphasis on different standards serving different kinds of end user applications, was given in the ISA Expo paper presenting IOHN work in 2009. Here we see RSM represented by the more generic term *Integration Information Model* (IIM), and MIMOSA, ISO 15926, and ISA 95 adapters being envisioned for access from applications to the integration layer services. Still, note that reference classes are assumed to be on ISO 15926 form (in the figure, ISO 15926 reference data is referred to as the "ISO 15926 'bible' "), and to reside with the IIM at the core of the integration layer.

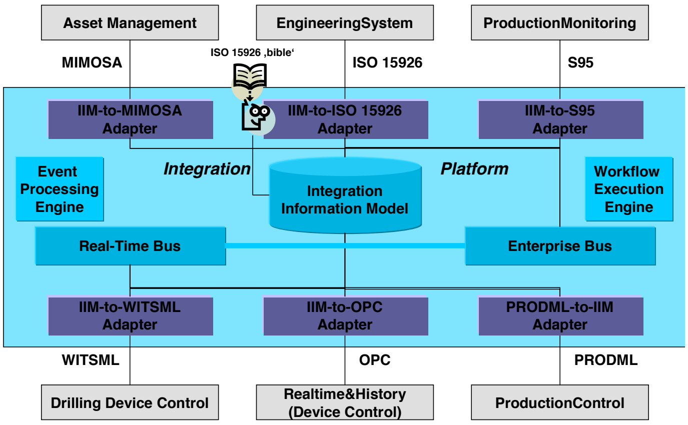


Figure 3.9: "Typical SOA and EDA-based integration architecture for integrated operations applications" [26, p. 4]

The final setup of the IOHN pilot was presented to (and accepted by) the IOHN Steering Committee, May 23, 2011. In this system, we have concrete systems and resources filling in the blanks of the architectural schema.

- Integration layer: RDF server (using Cambridge Semantics' *Anzo Server*)
- Applications: DNV *Erosion Monitor*, ABB *Insight*, Epsis *Decide*
- Data sources: Siemens *PIMAQ*, Tieto *Energy Components*
- Reference data: ISO 15926, PCA-RDL, IOHN ontologies, NPD registry of NCS

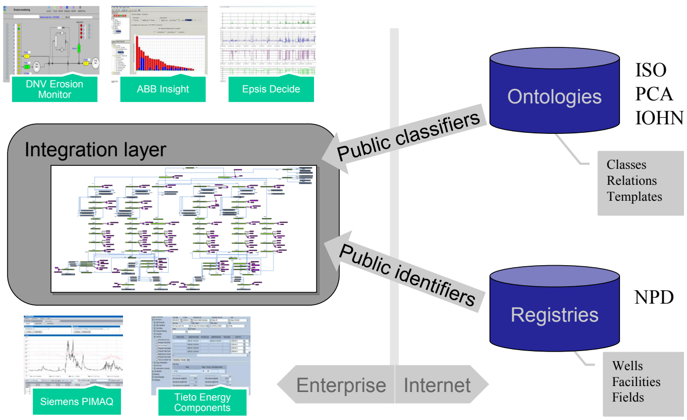


Figure 3.10: The final IOHN pilot setup, with systems and reference data resources [1, slide 4]

In September 2010, OLF presented the following architecture diagram, reflecting some of the implementation work in IOHN, at a Houston meeting. (Note that the role of external *registries* is in this context filled by the MIMOSA registry, which is fully in line with the IOHN approach.)

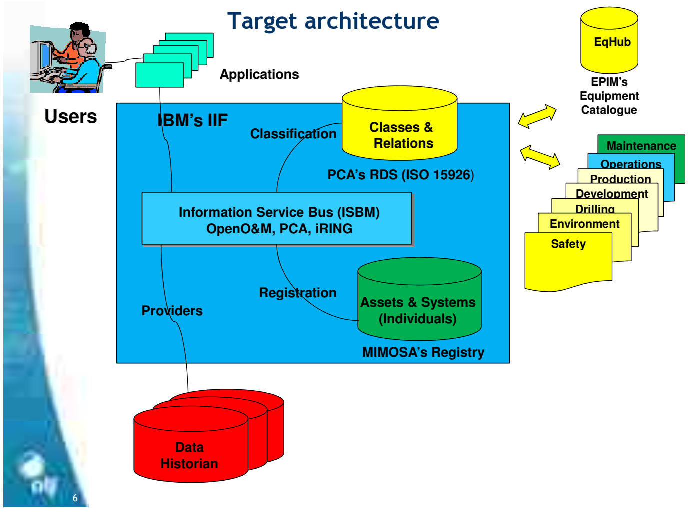


Figure 3.11: The IOHN architecture – OLF 2010 [19, slide 25]

A main feature of the architecture implemented in the final IOHN pilot is the distinction between *public* and *enterprise* information. This crucial distinction is not evident from the original OLF architecture. The integration layer, the enterprise data sources, and the enterprise applications are only available to the closed network Secure Oil Information Link (SOIL ☞). The ISO, PCA, and IOHN ontologies, and the NPD registry are however public resources, and available on the Internet. In this way, IOHN demonstrates how to implement a system that promotes integration and reuse by applying public, shared resources, while retaining full enterprise control over owned data flows.

Summary.

1. Server for platform model: IBM IIF/IIC with RSM model. Shown with iRing at ISA Expo 2009, at Semantic Days 2010, and at Intelligent Energy 2010. Query language and model are proprietary.

2. Server for platform model: Cambridge Semantics with ISO 15926/IOHN model. SPARQL for queries. From 2011.

3. Server for ontology: A Linked Data "hack" with resolvable URIs was demonstrated in June 2010 at Semantic Days, using the PCA server and links to resources on the PCA Wiki. *With the reconfiguration of the whole project during Fall 2010, Linked Data was given top priority, and project data moved to a real server – next point.*

4. Server for ontology: Using the TDB RDF

server running at DNV IRM. From 2011.

5. Server for reference individuals: Using TDB RDF server at DNV IRM, the NPD registry was made available as Linked Data.

6. Server for reference individuals: It was decided to merge with UiO's activities, using their server containing NPD data on fields and wells.

3.4.2 Semantic Model

The main delivery of the Semantic Model activity is a semantic model that provides a language for describing data and artifacts needed by the other IOHN activities. The greater part of the semantic model development process has been in cooperation with IOHN activity Production Pilot domain experts in order to develop a semantic model that supports the erosion, sand detection, and inspection & maintenance use cases.

3.4.2.1 The challenge of collecting information for the model

Getting the information required to develop the semantic model was a challenge that engaged the whole IOHN team. Information had to be collected from engineering and information experts in a variety of roles.

Domain experts. For the production domain, concepts and vocabulary needed to be sourced from erosion and sand management domain experts. A set of Microsoft Excel spreadsheets were set up for recording terms and definitions, and basic taxonomical relationships were specified by listing a superclass (see figure 3.12). O&G experts tend to be very comfortable with working in spreadsheets, and we found it was not difficult to get acceptance for this way of collecting knowledge for the ontology. Domain experts would update and extend the vocabulary, while ontology experts would work on adjusting the content to fill any gaps in the structure, as needed for the ontology to be complete and suitably connected. A script served to convert the spreadsheets was into an OWL taxonomy, as needed for ISO 15926-8 templates (see section 3.4.4.3). This resulted in a taxonomy of classes, suitable for assigning types to all elements of the reference model. Using a spreadsheet meant that contributing updates to the taxonomy was straightforward for the domain experts, and certainly easier than it would have been to use an OWL editor directly.

The reference model of Snorre B is an *individual* described using the taxonomy given by the domain experts. The model had to be developed from scratch, by collecting information from various sources that are not available in a consolidated form (cf. the Production Pilot activity report for details). The model elements needed for the use cases could not, for instance, be generated in any direct way from existing P&ID diagrams. It turned out that the Microsoft Visio drawing tool was well known to the Production Pilot activity experts, so it was decided that the model would be developed as a Visio drawing, rather than with specialized RDF or ontology diagramming tools. With Visio, the experts were able to lay out the needed process plant topology with relative ease. A simple Visio UML profile (using the UML graphical shape set that is bundled with the Visio tool) was chosen for the diagram (see figure 3.13). Using the UML profile allowed for types to be assigned to the process plant entities using the defined vocabulary. Templates with two arguments were represented by UML *links* (shown as blue, labelled lines in figure 3.13) and UML *N-ary Link* shapes were used to represent template instances with three or more arguments (shown as a purple square). Once Visio had been chosen for the task, progress on specifying the platform model, which had been difficult to get started, advanced rapidly. Updates by the experts were also easy to accommodate. The diagrams needed extensive and time-consuming manual reworking by the Semantic Model ontology experts, in addition to scripted conversion into ISO 15926-8 RDF format (see section 3.4.4.4). However, on balance the choice of Visio for developing the model was found to be quite appropriate, like the choice of Excel for describing the taxonomy.

3.4.2.2 ISO 15926, PCA RDL and sources of reference data

ISO 15926 was a basic choice from the start of IOHN. For industry reference classes, the classes defined in ISO 15926-4 and PCA RDL would be used. New classes developed in IOHN, or mapped to ISO 15926 as part of the IOHN work (as would apply to, e.g., the ISA 88/95 standards) would be included in PCA RDL after review in Special Interest Groups (SIG).

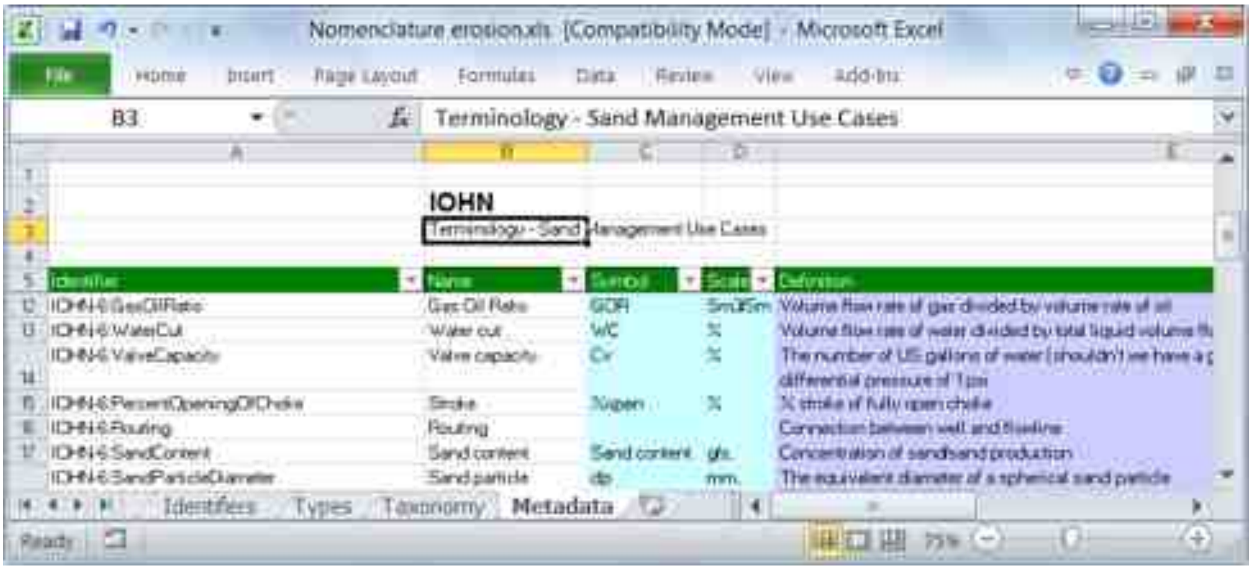


Figure 3.12: Microsoft Excel workbook for specifying vocabulary and taxonomy

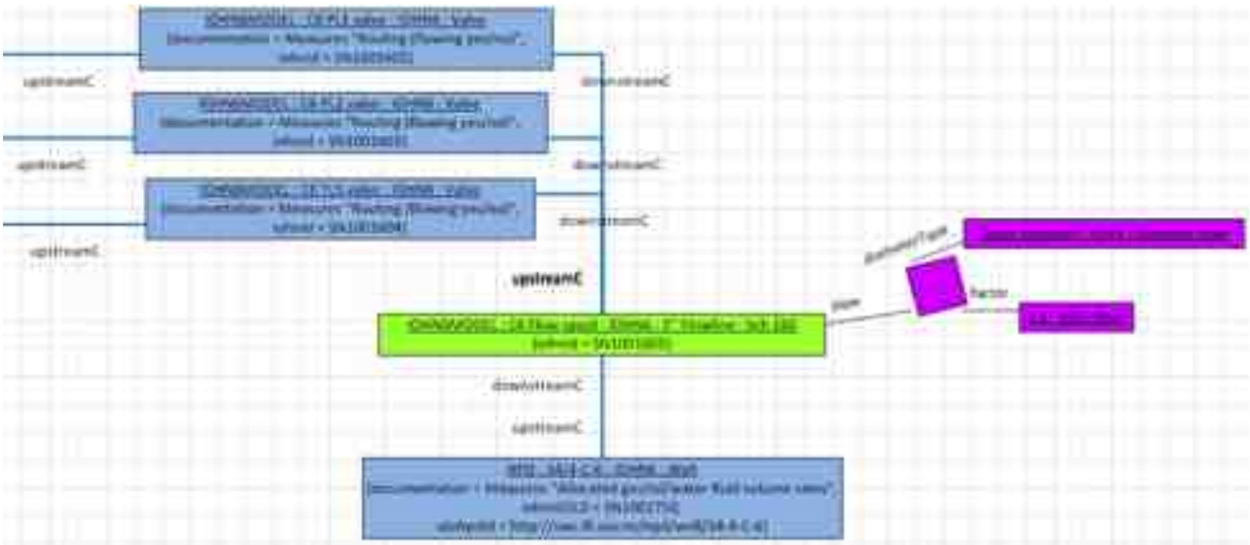


Figure 3.13: Microsoft Visio diagram for describing the Snorre B process plant

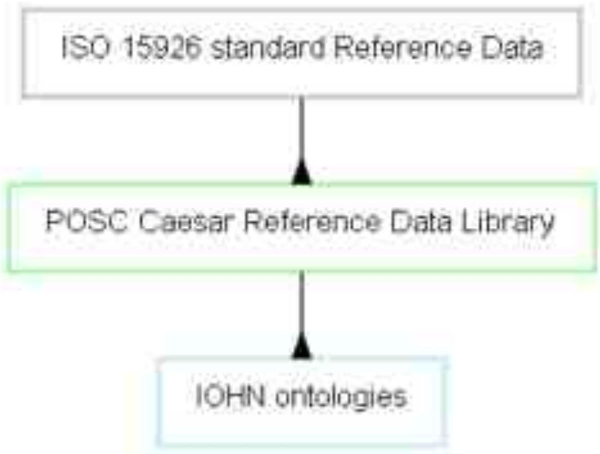


Figure 3.14: Sources of reference classes for the IOHN semantic model activity

The RSM model was chosen as the model for the IOHN pilot system implementation, to be mapped to ISO 15926. The map was delivered early 2009; and RSM 2.0 was sketched, as an extension that would allow for full use of PCA RDL and other sources of external reference data.

At initiation of IOHN, ambitions for semantics were high: the project would integrate content compliant with the ANSI/ISA-88 and ANSI/ISA-95 series of standards for integration "from top-floor to shop-floor" [quote IBM/OLF presentation here], by way of the RSM. Furthermore, PCA would provide industry standard data, and facilitate a standardization of the models and content developed in IOHN. However, the application of reference library classes to the RSM-based system turned out to be problematic in practice.

During the project adjustments had to be made.

- ISO 15926 was retained, but full modelling in the sense of ISO 15926-2 was dropped in favor of the simpler "template signature" approach of ISO 15926-8.

- The PCA RDL was used, but the initial standardization aim of the project could not be realized, much due to unavailability of SIGs that would be able to review new contributions.

- The RSM model was taken out of the project scope after consortium changes. In its place came an IOHN specific ISO 15926 ontology of classes, relations, and templates.

The effort to standardize the IOHN ontologies/templates as part of the PCA RDL was cut short due to changes midway in the IOHN project. However, the ontologies/templates modelled for activity 6 may serve as a starting point for a PCA standardization process.

3.4.2.3 Modular ontology

In 2009 it was determined that IOHN would need to have a modular structure to its ontologies. This eventually led to a *stack* of RDF/OWL ontologies, with ISO 15926 providing a bare schema of generic types, the PCA RDL being sourced for generic O&G industry classes, and IOHN activities extending on these. According to the RDF approach of mixing schema and instance data, the Snorre B example model and the sensor, event, and production data for Snorre B were also expressed as ontologies.

Only the Production Pilot activity actually implemented the ISO 15926 stack setup as laid out by the Semantic Model activity. Activity Safety and Risk developed only local models, and didn't target the ISO 15926 stack. Activity Drilling Pilot delivered an ontology in 2009 that was not built on the ISO 15926 basis, and later plans (2011) to design an ontology of templates for drilling, to fit into the IOHN stack, were not realized due to lack of resources. The resulting IOHN stack of ontologies covers a smaller scope than was planned, but it nevertheless serves to demonstrate the concepts of a modular arrangement that the Semantic Model activity was aiming for. (See section 3.4.7 for how the Drilling Pilot activity benefited from interaction with the Semantic Model activity.)

The ontologies of reference classes, used in activity Production Pilot, are listed in Table 3.1. Figure 3.15 shows the hierarchy of OWL ontologies, from the upper ontology of ISO 15926-8 to the template instances (facts) that describe the Snorre B platform.

Label	Description	Prefix	Namespace
IOHN common RDL	The IOHN common reference data library.	iohn	http://iohn.org/rdl/
IOHN-6 RDL	The IOHN activity Production Pilot reference data library.	iohn6	http://iohn.org/activity-6/rdl/
IOHN-6 templates	The IOHN activity Production Pilot Templates.	iohn6tpl	http://iohn.org/activity-6/tpl/
IOHN-6 platform model	A topological model of the Snorre B platform.	iohn6model	http://iohn.org/activity-6/model/
IOHN-6 Snorre B data	Events, sensor and production data for data sources on the Snorre B platform model.		http://iohn.org/activity-6/data-snb/
IOHN-6 Snorre A data	Raw data for sensors of the well P-9 on Snorre A for a limited period of time.		http://iohn.org/activity-6/data-sna/

Table 3.1: The IOHN ontologies. The Production Pilot activity used to be referred to as "IOHN activity 6", which is reflected in the labels and URIs.

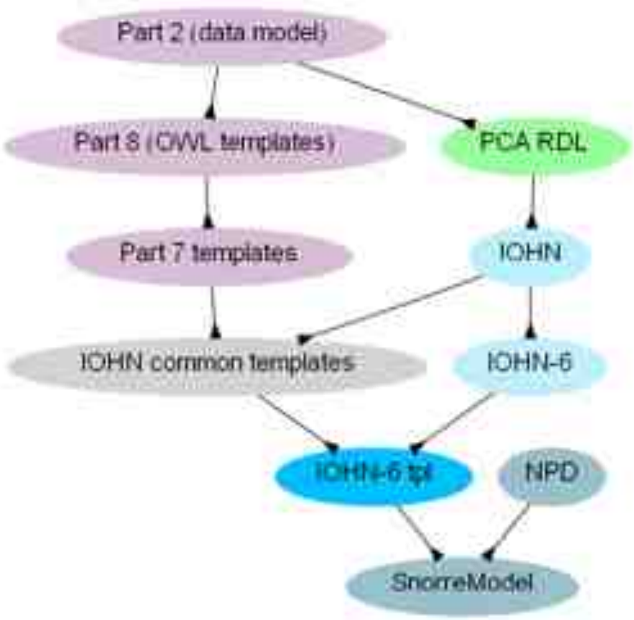


Figure 3.15: The ISO 15926 stack of ontologies for the IOHN semantic model of Snorre B ☞

3.4.2.4 Common Semantics

The IOHN Common semantic model describes common concepts like unit of measure (class iohn:UnitOfMeasure), different measurement data types (class iohn:DataType)[3], and data aggregation types (class iohn:AggregationType). The model also contains individuals representing different units of measure (e.g. iohn:BarGauge and iohn:GramsPerSecond), data types (e.g. iohn:GasOilRatio and iohn:InsidePressure), and aggregation types (e.g. iohn:DailyAveragedValue and iohn:RawValue) needed for data representation.

3.4.2.5 Activity 6

The Activity 6 semantic model is developed in order to support the use cases for erosion, sand detection, and inspection & maintenance. The erosion use cases aims at predicting the amount of erosion on the production lines of the Snorre B platform by using different prediction models and software provided by ABB and DNV. The erosion models require as input production data for the wells of a platform, as well as data for temperature, pressure, and sand sensors. Further, it is necessary to know to which production line each well is routed for a given time period, as well as the radius of curvature of each pipe part of the production lines.

The Activity 6 RDL (prefix iohn6, see Table 3.1) contains a taxonomy of classes that capture *flowlines* of different *schedule*, *separators*, *adjustable choke valves*, *risers*, *wells*, and *wellbores*. Further, the Activity 6 Template ontology (prefix iohn6tpl) contains templates for expressing that two flowline parts are interconnected in a way that al-

[3]Not to be confused with literal data types like *string*, *integer*, and *double*.

lows a flow to pass from one part to another (iohn6tpl:FlowConnection), diameter of flowline parts (ex. iohn6tpl:InnerDiameterOfPipelineType), radius of curvature (iohn6tpl:RadiusOfCurvature), sensors connected to flowline parts, etc.

In order to support sensor and production data, there are templates describing what type of data sensors can provide–including unit of measure, aggregation level, etc. (iohn6tpl:ProvidesDataType), as well as templates for expressing actual data readings for sensors (iohn6tpl:DataSourceValue).

The sand detection use case is about monitoring the raw data stream from sand detectors in order to detect *sand events*–potential sand readings. The sand events are then analyzed in order to determine whether there was an actual sand reading, or just a “false alarm”. The are Production Pilot activity templates for expressing sand events, with a start and end time stamp, as well as a reference to the sand sensor that produced the reading (iohn6tpl:SandEvent and iohn6tpl:SandEventEnd). Further, there are templates to express whether the event has been analyzed, as well as whether the sand event was a true sand reading (iohn6tpl:SandEventProcessing).

The inspection & maintenance use case focuses on using the erosion prediction to target inspection & maintenance to pipe sections that are more likely to be eroded than others. The semantic model supports the use case by providing templates that capture the actual erosion on flowline parts detected during inspection (iohn6tpl:InspectionEvent). For a complete list of templates, see Section 6.A.2 in the Production Pilot activity report.

3.4.2.6 Snorre B Platform Model

The Production Pilot ontology stack contains a topological model of the Snorre B platform (prefix iohn6model), containing wells, flowlines from the wells to topside, topside separators, etc. Pressure and temperature sensors upstream/downstream of the chokes of the flowlines are modelled, in addition to sensors for the wells and separators. Pipe dimensions and radius of curvature are represented. All parts of the platform model are typed using classes from the iohn6 ontology, and all relations are expressed as template instances of the templates defined in the iohn6tpl ontology. For more information on the modelling of the platform model, see the Production Pilot report, and for information on how the Microsoft Visio model file provided by Production Pilot was converted to RDF/OWL, see Section 3.4.4.4.

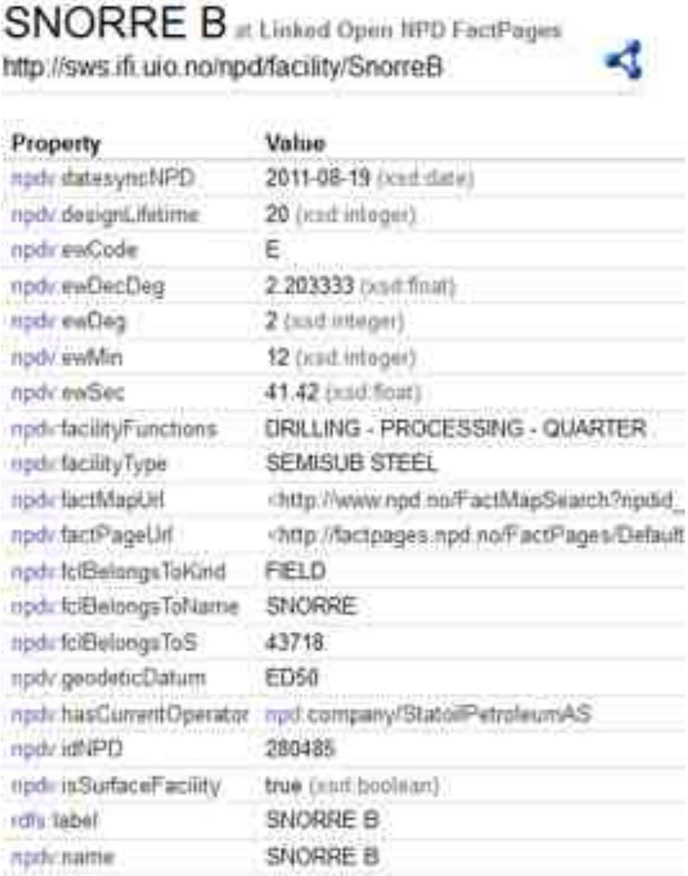
3.4.3 Public Identifiers

There are many stakeholders (operators, contractors, sub-contractors, etc.) that may want to store information about the fields, platforms, and wells on the Norwegian continental shelf. In order to facilitate data exchange, it is of high importance that the different stakeholders use the same identifier for an object. Fields, platforms, and wells are regulated by the Norwegian Petroleum Directorate (NPD), thus, it is natural that NPD organizes common, public identifiers–*reference individuals*–for such objects.

Following the Linked Data approach when developing semantic models, resolvable URIs should be used as identifiers. NPD publishes FactPages[4] containing information regarding the petroleum activities on the Norwegian continental shelf. Ideally, the URIs of the fields, platforms, and wells in the FactPages should be used as identifiers of objects that are under NPD control, and the URIs should have be on a form similar to http://npd.no/rdl/well/.... Alas, the FactPages are *not* published as Linked Data: The URIs are not on a form that is usable as identifiers, and the content provided to RDF-aware clients when resolving FactPage URIs is not in RDF format.

UiO has transformed XML dumps of the NPD FactPages into RDF and published the result as Linked Data on the URI http://sws.ifi.uio.no/npd. The URI for the Snorre field is http://sws.ifi.uio.no/npd/field/Snorre, the URI of the Snorre B platform is http://sws.ifi.uio.no/npd/facility/Snorre B, and the URI for the well C-2 on Snorre B is http://sws.ifi.uio.no/npd/well/34-4-C-2. The URIs are resolvable, and the information extracted from the NPD FactSheets are published as Linked Data. Hence, when accessing the URI for Snorre B, one receives a set of RDF triples describing properties of the Snorre B platform. For HTML clients, the information is nicely formatted as a HTML page, as shown in Figure 3.16, and RDF-aware clients receive RDF.

[4]See http://factpages.npd.no/factpages/



SNORRE B at Linked Open NPD FactPages
http://sws.ifi.uio.no/npd/facility/SnorreB

Property	Value
npdv:datesyncNPD	2011-08-19 (xsd:date)
npdv:designLifetime	20 (xsd:integer)
npdv:ewCode	E
npdv:ewDecDeg	2.203333 (xsd:float)
npdv:ewDeg	2 (xsd:integer)
npdv:ewMin	12 (xsd:integer)
npdv:ewSec	41.42 (xsd:float)
npdv:facilityFunctions	DRILLING - PROCESSING - QUARTER
npdv:facilityType	SEMISUB STEEL
npdv:factMapUrl	<http://www.npd.no/FactMapSearch?npdid_
npdv:factPageUrl	<http://factpages.npd.no/FactPages/Default
npdv:fclBelongsToKind	FIELD
npdv:fclBelongsToName	SNORRE
npdv:fclBelongsToS	43718
npdv:geodeticDatum	ED50
npdv:hasCurrentOperator	npd:company/StatoilPetroleumAS
npdv:idNPD	280485
npdv:isSurfaceFacility	true (xsd:boolean)
rdfs:label	SNORRE B
npdv:name	SNORRE B

Figure 3.16: The Snorre B platform as a Linked Data resource, provided by a University of Oslo server.

The Snorre B platform model use UiO NPD URIs as identifiers for the objects regulated by NPD, like fields, platforms, and wells. Since the URIs resolve to information published as Linked Data, NPD information can be combined with proprietary information only available for the operator when querying the model. Note that using public identifiers for objects does *not* imply that all information stored about the objects must be made publicly available. Corporate information can be hidden from the public, but using public identifiers opens access to whatever information is publicly available about the objects.

3.4.4 Methods and Tools

A main contribution of the Semantic Model activity is the delivery of a semantic model built on input from domain experts, and a set of test data expressed in the language defined by the semantic model. Input to the Semantic Model activity is given in many different formats, including MS Visio and Excel files, and SQL database dumps. The model was developed using a multi-stage process where input from domain experts were transformed into a draft model, which in turn was reviewed by the domain experts, thereby providing improved input to the semantic model.

The Semantic Model activity needed a conversion process that takes file input from multiple sources and automatically converts them into a semantic model and data representations in RDF/OWL format. The conversion process must be easily repeatable and adaptable to new requirements imposed on the input files from the domain experts. In this section we will review the methods and technology used to support the semantic model development process.

3.4.4.1 Open & Standardized Data Formats

An important goal of the Semantic Model activity is to integrate information from different systems and data sources into a unifying and easily accessible integration solution. In order to achieve this goal, it is of key importance to use open & standardized data formats for expressing the Semantic Model itself. Open & standardized formats allow different software on different platforms to easily connect to and use the integrated information. The Semantic Model consists of several ontologies (see section 3.4.2), all of which are expressed in OWL. Further, the OWL ontologies are represented as RDF graphs, serialized using either XML or the Turtle/N3 RDF syntax.

3.4.4.2 Flexible Tool-Chain

The process of converting file input in different formats into OWL/RDF ontologies may include a complex sequence of conversion steps, each of which may depend on preceding steps to complete before it can be executed. Further, it is required to avoid changes to one part of the conversion process or a set of input files to trigger re-run of steps that are not affected by the changes. Also, since the conversion process must be fully automated, it must be built on command-line tools that can be executed when necessary.

The GNU Make [30] tool is used heavily to build binary executables and libraries from source code in complex software projects. Make supports advanced pattern matching, dependency graphs, and flexible definitions of make targets by specifying input parameters to any command-line tool. Hence, GNU Make is ideal for structuring the automatic conversion process from file input to Semantic Model.

The steps in the Make conversion process are mainly based on SPARQL [40] for converting intermediate RDF graphs into new forms, XQuery [42] for traversing XML trees and producing arbitrary output, and Java for writing custom functions that are hard to cover with available tools.

3.4.4.3 Nomenclature: Excel to RDF/OWL

The nomenclatures defined by the IOHN Common RDL and Activity 6 RDL ontologies are based on input from domain experts, mostly engineers, which often prefer Microsoft Excel as their tool for working with structured information. A pre-structured table was setup in an Excel file, in which domain experts filled in classes, descriptions, and sub/superclass relations. A small Java program was created that uses the Apache POI library for reading Excel files, and writing the output to RDF. The conversion from Excel to RDF is captured as a Make target, and can hence be run as often as needed in order to update the RDF/OWL ontology files when changes are made to the master Excel file.

3.4.4.4 Snorre B Model: Visio to RDF/OWL

An important part of the erosion and inspection & maintenance use cases of IOHN Activity 6 is a topological model of the Snorre B platform. The model includes a selection of the Snorre B wells, subsea templates, production and test lines including individual flowline parts, and topside manifolds and separators. For each flowline part, the radius of curvature is represented. Further, a selection of temperature, pressure and sand sensors are modelled, complete with information on what data type, unit of measure, and aggregation level the sensors can provide.

The topological platform model is drawn as an UML object diagram in Microsoft Visio by using the standard UML drawing template. The Snorre B Visio drawing is stored in Microsoft's Visio XML Drawing format. Through a series of conversion steps, essential information is extracted from the Visio XML by means of XQuery scripts. The resulting transformed XML file is then converted to RDF, which is further processed by a sequence of SPARQL CONSTRUCT queries. The final RDF output is an RDF representation of the Snorre B Visio drawing, where each individual is classified using classes from the Activity 6 RDL ontology, and each relation is expressed as an instance of appropriate templates of the Activity 6 Template ontology.

The data type, unit of measure, and aggregation level for each sensor in the model is stored in an Excel file with a pre-defined schema. The Excel file is processed in a similar manner as the nomenclature Excel file described above, producing template instances describing the data provided by each sensor.

3.4.4.5 Derived Properties/Templates

The relationships between objects in the Snorre B model described in section 3.4.4.4 are described in a low-level language, where only atomic/primary relationships are captured. For instance, a Snorre B well is related to the Snorre B platform through a series of relationships connecting the well to the flowline parts that lead to the topside processing equipment. Although the low-level representation indirectly captures the relationship between the wells and the platform, it is in practice quite cumbersome to trace the flowlines from the well to topside in order to list the wells for a given platform.

To remedy, a set of frequently used derived relationships are represented explicitly by means of derived template expressions. The derived templates are instantiated automatically during conversion from Snorre B Visio drawing to RDF representation. Hence, changes to the master Visio drawing automatically generates low-level/primary relationship template instances, as well as derived ones, in the output RDF representation of Snorre B.

When writing SPARQL queries for the RDF representation of the Snorre B model, quite often it is required to follow n steps of binary template instances, for instance when tracing a flowline from a well to topside. If binary templates between objects are represented as OWL object properties, then the object properties can be used instead of the templates when querying the model. Hence, for each binary template with object arguments, a corresponding OWL object property triple is generated automatically during conversion from Visio to RDF. For instance, a flow connection from flowline `a` to flowline `b` is captured as the following low-level template instance.

```
[] a iohn6tpl:FlowConnection ;
   iohn6tpl:hasUpstreamCFTD a ;
   iohn6tpl:hasDownstreamCFTD b .
```

In addition, the following *derived* RDF triple is generated.

```
a iohn6model:flowsInto b .
```

Hence, a SPARQL query that involves querying flow connections can use the derived property iohn6model:flowsInto instead of the low-level template instance, which greatly simplifies writing SPARQL queries for the platform model. Further, the derived property can be queried transitively, using the * SPARQL property modifier. The approach of representing binary templates as OWL/RDF property expressions is, to our knowledge, novel.

3.4.5 Production & Sensor Data

In the beginning of the IOHN project, it was intended to use IBM's IIF/IIC server platform to map proprietary data sources like Tieto Energy Components and Siemens PIMAQ to the Integration Layer, and to dynamically update data presented to end-users by the integration layer. When the project was forced to abandon IBM IIF/IIC, the Anzo server of Cambridge Semantics was chosen as a replacement platform.

The features of Anzo combined with the physical placement of the server at Baker Hughes, in contrast to the original plan to have the IIF/IIC server platform placed in a test lab at IBM/Statoil, forced the project to abandon the dynamic linking of sensor and production data to the integration layer. Instead, a data dump for a chosen period of time would be provided by Statoil/ABB/Siemens for static linking on the Anzo integration server. In this section, we describe the process of collecting the data dumps, transforming them into a suitable format, mapping the data to resources in the Snorre B platform model, and generating ISO 15926 part 8 template instances for the data sets.

3.4.5.1 Mapping Data Needs

The primary users of the production and sensor data available on the Integration Layer are the erosion and condition-based maintenance use cases of the Production Pilot activity. Based on their input, it was decided to get hold of sensor data for the acoustic sand detectors, pressure sensors, and temperature sensors present in the Snorre B platform model. Further, stroke (opening) of chokes and valve positions were needed, as well as production data (allocated rates) for the wells present in the Snorre B platform model. The data was to be provided as daily averaged values for the time period starting on January 1st 2009 to June 1st 2010.

The production rates for the wells are allocated rates, which are checked against actual flow rates at regular intervals. During a well test on Snorre B, the well is routed to the topside test separator, and actual values for gas/oil ratio, water cut ratio, oil rate, gas rate, water rate, well head pressure, opening/stroke of the well head choke, and the separator pressure are recorded. The participants of the Production Pilot activity needed access to well test readings for their use cases.

Additionally, Siemens required for a sand detection use case to be able to fetch raw measurement values for a limited time period for the well P-9 on Snorre A. Hence, it was decided to provide raw values for a set of eight sensors on P-9, starting at 2010-05-28T00:04:41.900 and ending at 2010-06-06T23:59:38.900.

For the condition-based maintenance use case, it was required to be able to access information on choke change events, and inspection events, where actual pipe erosion is measured and recorded.

3.4.5.2 Gathering & Reformatting Data

Once a set of sensors and wells was selected, data dumps for the agreed upon time period were provided by ABB (from Tieto Energy Components) and Siemens (from Siemens PIMAQ). The dumps came in the form of text files and Microsoft Excel workbooks.

The received data needed to be transformed into a format suitable for publishing on the Integration Layer, and annotated with meta-information indicating what unit of measure is used, the aggregation level, and the type of information that is measured. A set of custom software tools where developed, mostly in Java, that parses the text and Excel data sources, adds semantic annotations, and dumps the result in tables according to a pre-defined schema that is close to the definition of the template that will hold the generated template instances.

3.4.5.3 Mapping of Tag Names

A key part of generating template instances for the sensor and production data is to ensure that the correct URI is used to identify the data source from which each data value is recorded. Due to the established regime of *tag names* for components of offshore installations, the text and Excel data sources where equipped with tag name references to source components. Each tag name was mapped to the corresponding IOHN Snorre B model component URI identifier in a mapping ontology (http://iohn.org/activity-6/datamap), which was used when generating template instances for the source data.

Tag names are not URIs, and are hence not suitable as resource identifiers in a semantic model. Thus, mappings from tag names to component URIs should not be part of the Integration Layer itself, but be part of the mapping from each proprietary data source to the Integration Layer.

3.4.5.4 Generating Template Instances

Sensor values and production data are represented by instances of the template iohn6tpl: DataSourceValue. The template takes six arguments: the resource for which the value is recorded, the iohn:DataType of the reading (is it an iohn:InsidePressure, an iohn:SandRate, an iohn:GasOilRatio, etc.), the unit of measure of the value, the aggregation type of the value, the time on which the value was recorded, and finally, the value itself. Below is an example of a template instance representing the stroke (opening) of iohn6model:SN1001619, the topside choke of Production Line 3, on February 8th 2009.

```
[] a iohn6tpl:DataSourceValue ;
   iohn6tpl:hasAggregationType
     iohn:DailyAveragedValue ;
   iohn6tpl:hasDataSource
     iohn6model:SN1001619 ;
   iohn6tpl:hasDataType
     iohn:Stroke ;
   iohn6tpl:hasUnitOfMeasure
     iohn:Percentage ;
   iohn6tpl:valDateTime
   "2009-02-08T00:00:00"^^xsd:dateTime ;
   iohn6tpl:valDoubleValue
   "44.371932340047664"^^xsd:double .
```

3.4.5.5 Discussion

Publishing sensor and production data on a common, standardized and semantically annotated format simplifies configuring of data access for software clients. It removes the need for maintaining complicated database connections, where database schemas are quite often not very well documented. The approach taken in the IOHN Integration Layer pilot, however, has some obvious shortcomings. The static representation of sensor and production data requires more maintenance than dynamically linked information. Further, although the semantic annotation of each recorded data value removes any confusion of what the data means, it also results in a considerable overhead that will certainly affect the rate of transfer from server to client.

Considering that the IOHN Integration Layer is a pilot implementation, these shortcomings are not critical, and further, they are easy to overcome in a production implementation. One approach is to use the Integration Layer as a source of meta information on platform topology, what sensors are available, and where to connect to fetch high-speed binary data streams for each sensors.

3.4.6 Validation & Testing

The semantic model used in the IOHN project is expressed in the W3C standardized languages RDF and OWL. Further, all relationships are expressed in ISO 15926 part 8 template format, and classes from the PCA Reference Data Library are incorporated in the class taxonomy where possible. When developing ontologies and modelling plants and platforms in this complex stack of standards and languages, the risk of making mistakes is quite high. Thus, it is of high importance to test and validate the IOHN ontologies and platform model against the different standards involved.

Methodologies and tools for validating RDF/OWL ontologies are readily available, see for instance the list[5] maintained by W3C. There are two main types of errors that occur in OWL models; *syntactic* and *semantic*. A syntactic error occurs when an OWL/RDF serialization contains expressions that are not valid according to the RDF/OWL serialization syntax, and can easily be detected with a parser or validator for the given RDF/OWL serialization syntax (see e.g. the RDF

[5] http://www.w3.org/2001/sw/wiki/Category:Validator

Validator and the OWL Validator on-line syntax checkers).

Semantic errors are related to the formal semantics of OWL ontologies, and occur mainly in the form of *unsatisfiable* classes or *inconsistent* ontologies. A class is unsatisfiable if it cannot have any members, and an ontology is inconsistent if it expresses that an individual is a member of an unsatisfiable class. Checking for such errors can be done in a controlled fashion, due to the precise semantic basis, in formal logic, of the OWL language. (OWL comprises several dialects, and the semantics of each is in full correspondence with a particular Description Logic.) The semantic precision of OWL represents a major advance in comparison to, e.g., the more widely used Universal Modelling Language (UML) family of languages.

The For semantic errors (logical consistency checks), there are several highly capable Description Logic reasoners available that accept OWL ontologies. The IOHN project has mainly used the open-source tools Fact++ and Hermit. Among commercial offerings, the Pellet and Racer reasoners are the most well known. For Pellet, an integrity constraint validator extension (Pellet ICV) also offers “schema” language checks, which should be suitable for checking that a given set of data in ISO 15926 template format is complete (i.e., checking that not only is a set of facts consistent with the type constraints of a template, but also that each fact is a complete and well-formed statement, with every variable instantiated by an explicit resource or data-typed value).

The performance of OWL reasoners has improved dramatically during the IOHN project period. Realistic industrial ontologies, containing thousands of classes, can now be validated with acceptable response times. (Where the problem domain is known in advance, specially optimized reasoners such as TrOWL can provide vastly better performance than generic reasoners. There is considerable potential for tuning reasoners to the O&G models and information that the IOHN project has been investigating, as their structure is largely predictable. It has however not been possible to explore this in detail within the IOHN project.)

In the following, we present some of the tools and methods used to validate and ensure correctness of the ontologies and models developed in the IOHN project.

3.4.6.1 Quality Criteria

The modular construction of the IOHN semantic model provides a flexible setup for combining general ontologies, domain specific ontologies, and ontology instantiations. However, if not implemented carefully, it may increase the risk of introducing modelling errors and reducing the quality of the semantic model. UiO has developed a set of quality criteria for ISO 15926-8 compliant installation descriptions [7]. For each criterion, the document lists an argument explaining why the criterion is proposed, and a description of how the criterion can be checked. Some criteria are easily verifiable with SPARQL queries or use of reasoning tools, while others are on a conceptual level, and cannot be checked algorithmically.

The criteria in [7] are divided into criteria for RDF representations in general, as well as criteria that specifically covers representations in accordance with ISO 15926 part 8. The criteria for RDF representations in general include criteria for separation of ontologies and instance data, as well as separation of generic and domain specific ontologies. This requirement is clearly implemented in the IOHN modular stack of ontologies and models, as described in section 3.4.2.3. Further, it is proposed that every resource is typed, i.e. that it is explicitly declared which class(es) a resource is an instance of. This is in line with rule 7 of the Linked Data Basic Profile described in section 3.3.7, and carefully implemented in the Snorre B platform model. The final RDF requirement is that ontologies are consistent, which for the IOHN ontologies is verified using the RDF/OWL reasoning tools described in section 3.4.6.

The criteria related to ISO 15926 part 8 include the requirement that any domain specific ontology must be a *conservative* extension of any underlying ISO 15926 ontology, i.e. it must not modify the meaning of the underlying ontologies. Further, for an ontology instantiation on ISO 15926 part 8 template format, it is required that the RDF triples are either type declarations or reified template instance expressions. Also, it is required that all arguments of a template are present in each template instance. The ISO 15926-8 criteria are fully implemented in the Snorre B platform model.

3.4.6.2 Validating Visio Model

In order to create the ontology instance for the Snorre B platform model, a Microsoft Visio UML drawing of the platform provided by the Production Pilot activity is converted to template instance expressions by a series of conversion steps (see section 3.4.4.4). For debugging and demonstration purposes, it is convenient to track for each RDF resource in the ontology instantiation for Snorre B which object in the original Visio drawing the resource comes from. Hence, for each resource in the Snorre B ontology instantiation, there is a datatype property iohn6model:VisioID referencing the unique GUID[6] of the corresponding object in the Snorre B Visio drawing.

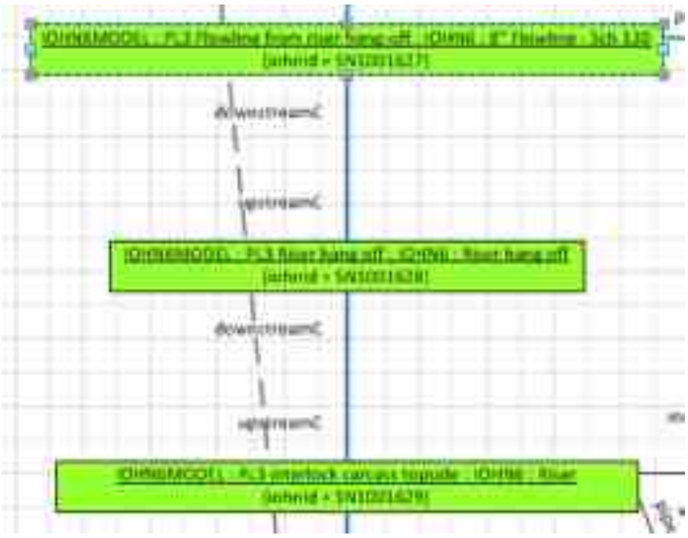


Figure 3.17: Excerpt from Production Line 3 of the Snorre B Visio drawing.

The above image shows an excerpt from the Snorre B Visio drawing, showing some components of Production Line 3. The selected object is designated "PL3 Flowline from riser hang-off". When converted to an RDF resource iohn6model:SN1001627 in the Snorre B ontology instantiation, the resource is annotated with the GUID of the corresponding object in the Visio drawing as follows.

```
iohn6model:SN1001627 iohn6model:VisioID
"{6A4F6CEF-3572-4B10-947B-51292736F285}" .
```

In order to locate an object in the Snorre B Visio drawing based on its GUID, the Visio drawing is equipped with a custom macro "Locate by GUID" that can be accessed from the ribbon menu "RDF Utilities".

Figure 3.18: The "Locate by GUID" macro of the Snorre B Visio drawing.

When the macro is run, it opens a dialog requesting the user to enter the GUID of the object to locate. When the user presses OK, it shows the page on which the object is located, selects the object, and pans the viewport such that the object is centered.

The "Locate by GUID" function has been used frequently when debugging the scripts that convert from Visio UML to ISO 15926-8 templates. When aligning the resulting RDF with the Visio UML drawing, it is easy to see whether mistakes are due to modelling errors in the drawing, or to bugs in the conversion scripts.

3.4.6.3 Jena Eyeball for Validating RDF/OWL

Jena Eyeball [15] is a library and command-line tool for checking RDF and OWL models for various common problems. These problems often result in technically correct, but implausible, RDF. Problems that can be detected include unknown (with respect to a pre-defined set of schemas) properties and classes, bad prefix namespaces, ill-formed URIs, ill-formed language tags on literals, datatyped literals with illegal lexical forms, etc. By using Jena Eyeball on the IOHN stack of ontologies, we have detected (and corrected) inconsistent use of prefixes across ontologies. However, full OWL error detection failed, due to use of the OWL 2 specification in the IOHN ontology stack, which is not supported by the current version of Jena Eyeball.

3.4.7 Interaction with the IOHN Drilling Pilot activity

The Drilling Pilot activity profited from the ongoing IOHN collaboration on semantic methods

[6] GUIDs are Microsoft's implementation of the concept of Universally Unique Identifiers, which is standardized by the Open Software Foundation.

and technology, in workshops and regular meetings. Tutorials on ontology development using ISO 15926, although not used directly, gave valuable insights into good practices of modelling and formats. RDF was chosen as a standard for Drilling Pilot activity data exchanges between systems, and in a wider sense the W3C set of standards for semantics with RDF/OWL and SPARQL, was adopted. That allowed the Drilling Pilot development to execute its engineering efforts in a decoupled manner without losing the integration aspect.

In the early phase of the IOHN project, it was planned to develop an integrated set of IOHN ontologies that would cover the scope of both the Drilling Pilot and Production Pilot activities (see section 3.2.3). During the course of the project, the Drilling Pilot activity had to shift priorities, so its ontologies were not included in the IOHN stack at project end (see section 3.4.2.3).

In a sub-project, the Drilling Pilot activity explored semantic technology for *reporting*. The case study was found in a very real scenario: In 2008, autorities made it compulsory for all drilling activities on the Norwegian Continental Shelf to comply with an XML Schema format for the Daily Drilling Report (DDR) (cf. timeline in figure 3.3, p. 54). Targeting the DDR schema, the Drilling Pilot activity demonstrated how semantic technology can support automated reporting, in work presented at the 2012 Intelligent Energy conference [8]. This was joint work between the University of Oslo, National Oilwell Varco, and Baker Hughes.

Reference data for DDR had been developed in accordance with ISO 15926 guidelines, registered in the PCA RDL, and embedded using SAWSDL links in the XML Schema. The references provide an authoritative and accessible glossary. The Drilling Pilot activity explored how an RDF/OWL ontology for that set of reference terms could do much more, embedded in a tool for automatic generation of DDR reports from existing O&G information systems. This takes semantics from a passive role as a body of reference into an execution environment: Accessing data in existing systems by way of requests framed in the language of an ontology. Thus, in several ways the Drilling Pilot and Production Pilot activities implemented a similar approach to data access.

The Drilling Pilot activity created a rudimentary drilling taxonomy. This provided an ontology with a simpler structure than had been the activity target; in this respect, there are significant similarities to the development for the Production Pilot activity, which followed ISO 15926-8 in only providing a class/subclass taxonomy. While the investigation didn't develop an ISO 15926 ontology for the reporting work, it would clearly be possible to do so in a follow-up development.

Attempts were made in the Drilling Pilot activity to include and extend content from the PCA library, for purposes of integration and to meet the ISO 15926 compliance goal stated for IOHN. However, this effort was hampered by limited accessibility and difficulty of navigating the library's large body of information, and eventually abandoned. (The PCA is adressing the issues that restricted the Drilling Pilot activity's progress, in the JORD project and recently with *native OWL* representations. However, only very preliminary results of this work were available during the IOHN project period.) Drilling Pilot team member Robert Ewald commented on the outcome that "... the biggest potential use of semantic technology is in the exchange and modeling of engineering knowledge. It can radically cut down engineering costs. However, accessibility to data must be improved so that the ontologies may be checked and reasoning can be used to generate new insights".

3.4.8 Semantic annotations for Monthly Production Report (MPR)

Already in 2008, the IOHN Semantic Model activity contributed reference data to the PCA-RDL, in the form of classes and relations that provide semantics for the official Norwegian O&G *Monthly Production Report* (MPR). IBM, DNV, and PCA contributed to the reference data delivery. The definition of semantics in a reference library that is separate from the XML formats for MPR enables a highly desirable de-coupling of domain terminological issues from the formats and software used to receive, validate, and process reports. The IOHN Semantic Model activity adopted this approach, and throughout the project period had a strong focus on developing it further.

The design and management of the MPR is handled by the E&P Information Management Association (EPIM) ☞. The MPR a comprehensive report, and obligatory for O&G operators on the NCS. Reports are delivered by operators to the automated service *License2Share* ☞. Operators must provide their reports using an XML format. PCA reference data provides definitions for this format: Identifiers for the MPR classes are embedded as SAWSDL annotations [41] in the XML schemas that define the MPR format.

The IOHN contribution to the MPR reference data held in PCA-RDL was one of several incremental efforts to improve information management in its domain, going over a period of several years. Development of an ontology for MPR is still ongoing in 2012, with the introduction of EPIM's *Reporting Hub* system [10].

The IOHN delivery is documented in detail in the report "Monthly Production Report (MPR) V. 1.0.0 – Terminology" [29]. Definitions of the core subject vocabulary of the MPR were modelled according to ISO 15926, and incorporated in the PCA-RDL ontology as reference classes. Figure 3.19 shows an excerpt from the list of defined classes, organized as a taxonomy, with definitions and PCA identifiers (in the two rightmost columns).

For the 2008 MPR, PCA identifiers were published in XML format in a way that incorporates much of what has become a norm in Open Linked Data. The URIs are dereferencable, delivering an HTML page if accessed by a web browser and an XML document if so requested. For example, the URI http://rds.posccaesar.org/2009/08/XML/RDL/RDS392257851 represents the set of MPR classes MONTHLY PRODUCTION REPORT NOMENCLATURE (version of August 2009), a second-order ISO 15926 class. Another example is ISOPENTANE COMPOUND, at http://rds.posccaesar.org/2009/08/XML/RDL/RDS482640400.

The classes of the PCA-RDL are in the process of (2012) being made available according to best practices for Linked Data, by way of the Joint Operational Reference Data (JORD) project, a collaboration between POSC Caesar and Fiatech [9]. As of June 2012, the prototype Linked Data server for the PCA-RDL assigns the class ISOPENTANE COMPOUND the identifier http://posccaesar.org/rdl/RDS482640400: PCA key is unchanged from earlier editions, while the namespace has been altered.

3.4.9 ISO 15926 training material

3.4.9.1 Compendium on modelling basics

In 2008, in collaboration with OLF, IOHN delivered a tutorial compendium on ISO 15926. This had the form of lecture notes [6] and an extensive set of slides for classroom use. Written by Henrik Forsell for OLF, the compendium introduces semantic technology for industrial data exchange, describes the ISO 15926 standard, and relates modelling in ISO 15926 to general ontology work. (A planned second part of the tutorial was never completed.) ☞

3.4.9.2 Workshop: Templates in ISO 15926-8

Templates for ISO 15926 patterns: Delivered October 2010. Compendium, script, ontology stack for design of ISO 15926-8 template *signatures* using ISO 15926-8, in RDF/OWL. ☞

The templates tutorial provided the participants with a compendium, and a skeleton ontology stack for creating reference data that the participants could conveniently use as a starting point. It also defined a simple RDF format for describing ISO 15926 template signature, and a shell script that would transform this format into a more complex form compliant with the ISO 15926-8 OWL 2 format. Figure 3.20 shows how the script would generate ISO 15926-8 templates from the specification file, producing an ontology to fit the ontology stack.

C6+	C6+ HYDROCARBON COMPOUND	A hydrocarbon compound that consists of mainly hydrocarbon molecules with six or more carbon atoms.	RDS4826404027
C3+	C3+ HYDROCARBON COMPOUND	A hydrocarbon compound composed of hydrocarbons with three or more carbon atoms.	RDS8015624729
C5+	C5+ HYDROCARBON COMPOUND	A hydrocarbon compound that consists of mainly hydrocarbon molecules with five or more carbon atoms.	RDS4826404020
C10+	C10 + HYDROCARBON COMPOUND	A compound that consists of mainly hydrocarbon molecules with ten or more carbon atoms.	RDS4826404055
isopentane	ISOPENTANE COMPOUND	A compound where the main component is isopentane.	RDS482640400
carbon monoxide gas	CARBON MONOXIDE COMPOUND	A compound in which the main component is carbon monoxide in a gaseous state.	RDS8015624857
mixed butane	MIXED BUTANE COMPOUND	A compound where the main components are butane and isobutane.	RDS418609171
isobutane	ISOBUTANE COMPOUND	A compound where the main component is isobutane.	RDS418623741
hydrogen sulfide	HYDROGEN SULFIDE COMPUND	A compound where the main component is hydrogen sulfide.	RDS978517230
ethane	ETHANE COMPOUND	A compound where the main component is ethane.	RDS418622621
	SALT	A compound in which the main component is salt, i.e. NaCl.	RDS8015624941
salt	SALT AS COMPONENT IN HYDROCARBON COMPOUND	A compound in which the main component is salt, occurring as a component of a hydrocarbon compound.	RDS8015624934
	FLUID COMPOUND	A compound which is able to flow.	RDS13108820
gas lift	GAS LIFT	A fluid compound injected into a producing well to reduce the hydrostatic pressure of the fluid column.	RDS8127142155

Figure 3.19: PCA-RDL classes for EPIM's Monthly Production Report [29, excerpt]

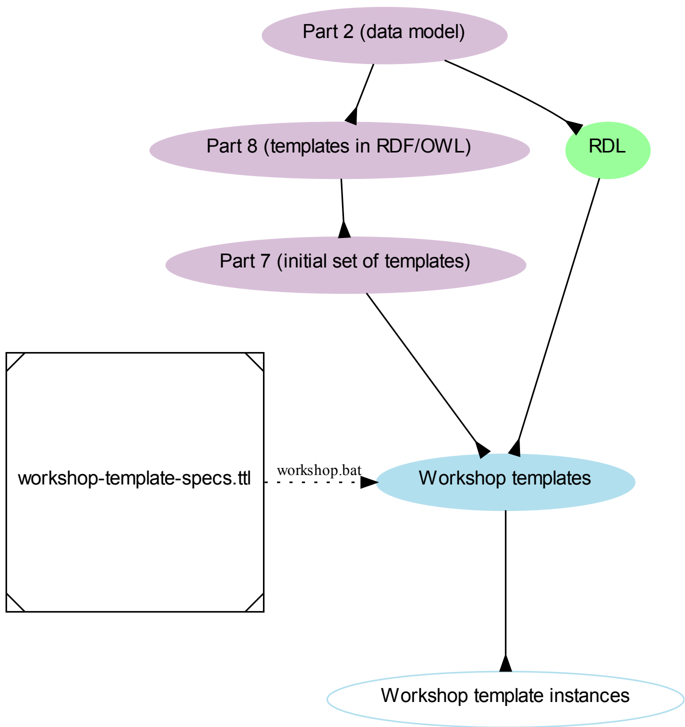


Figure 3.20: Template setup for defining ISO 15926-8 template signatures in OWL [16, p. 7]

In the tutorial, a highly simplified process diagram served as an example for representing in a semantic model. The diagram illustrated two tanks, one open and one closed tank, connected by a pipeline in several sections, equipped with a valve (fig. 3.21).

```
                                              +------+
|     |       c1:H      c2:H    c3:H       |      |
|     |--|><|------||--------||-----()---|      |
+-----+  e:J                          d:I  +------+
  a:F                                        b:G
                  .........
                     c:K
```

Figure 3.21: Simplified process diagram example for ISO 15926-8 templates tutorial [16, p. 10]

As an example of a template defined for this model, a template for assigning control valves to pipelines was given in abbreviated specification form as follows.

```
:ControlValveOfPipeline
  rdfs:subClassOf
      template-model:TemplateStatement ;
  :role1 :hasControlValve ;
      :type1 RDL:Valve ;
  :role2 :hasPipeline ;
      :type2 RDL:Pipeline .
```

An example of a singular fact using this template, and suitable for representing the model sketched above, was given as follows.

```
:myCVPeToc a :ControlValveOfPipeline ;
  :hasControlValve :e ;
  :hasPipeline :c .
```

The need for a simple way to design templates for the Semantic Model activity was emphasized by the complexity of modelling information using the full form specified by ISO 15926-2. A case in point was the information needed for erosion monitoring. A sketch of the elements of the well for which data was needed is shown in figure 3.22 (early 2010). With this sketch, the Semantic Model activity sought advice from well experts on which components of a well were involved, and on what the required modelling would amount to – on how the involved piping, valves, streams, and so forth should be related using ISO 15926 primitives.

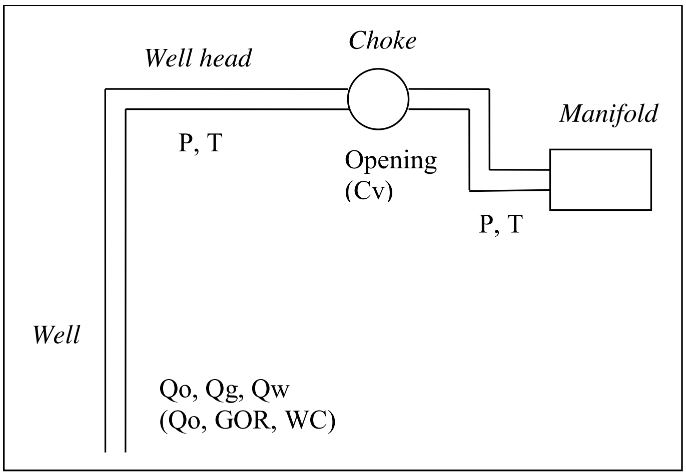


Figure 3.22: Sketch of well elements needed for erosion management [14, p. 1]

Even though it started with a very simple sketch, the analysis produced a rather complex ISO 15926 model, illustrated in figure 3.23. While that figure is not unusually complex for a semantic model, it is too intricate to be useful in communicating between ontology experts and domain experts. (The figure is also a simplification, where ISO 15926 types have been left out.)

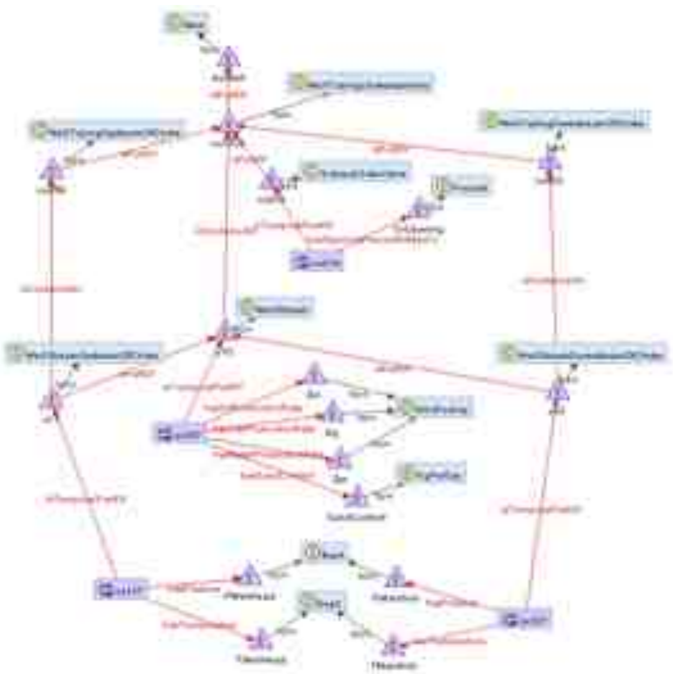


Figure 3.23: The complexity of this *Well Property* ISO 15926 model motivates the template approach [16, p. 1]

The templates tutorial demonstrated a practical approach to creating a semantic model, one that for the first time made it possible to create and instantiate ISO 15926 templates with nothing more than a minimal shell script and a SPARQL query engine. A template signature abstracts from a fact to create a pattern: it only requires a natural-language description of meaning, a list of the variable elements, and a specification of the type of each variable (i.e., given as a class in the domain taxonomy or a data type). A template signature can be set up without attention to the explicit ISO 15926 model that should, eventually, underpin a template with a precise interpretation rule (a template "axiom"). Essentially the same approach – providing the template signatures, but not the full models – was used in constructing the Snorre B reference model from a Visio drawing later in the project (section 3.4.2.6). The template workshop developed a shared understanding between the process and ontology specialists, enabling them to efficiently work together to design the reference model.

3.4.10 Mapping of RSM to/from ISO 15926

3.4.10.1 Overall effort

From the start of the IOHN project in 2008, the Reference Semantic Model (RSM) was targeted as the meta-model for all information. RSM model entities (classes and relations) were expected to provide the format for information residing in of the architecture's integration layer. This would include static structures such as oil platform assets, as well as messages mediated by the

RSM was provided to the the Semantic Model activity as a UML model, developed using IBM's Rational Software Architect (RSA). The IOHN project was provided with RSA development licenses for the purpose of investigating the RSM model in detail.

Presented to Statoil April 2009. RSM meta-model entities mapped to ISO 15926. Requirements on interoperability between RSM models and ISO 15926 were identified, and a proposal for further development of RSM was presented (under the label "RSM 2.0").

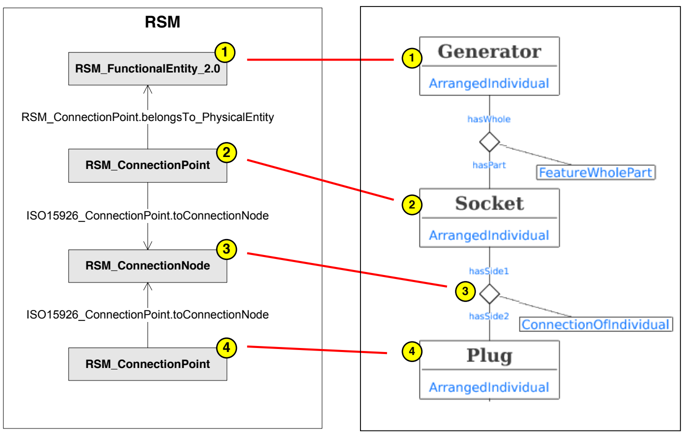


Figure 3.24: Figure from RSM–ISO 15926 mapping

3.4.10.2 RSM and industry standards

The RSM meta-model is set up to integrate several different models of relevance to the O&G process plant. It is mainly based on the ISA S88/S95 standards, but also includes elements from UNCEFACT, Mimosa/OpenO&M, and ISO 15926.

During 2008–09, several Semantic Model workshops focused on the integration of Mimosa's CCOM model and ISO 15926, with a view to developing a combined approach to information that would combine CCOM structures for asset registries and ISO 15926 resources for characterization of assets.

3.4.10.3 UML and RDF models

The IOHN effort to align ISO 15926 to RSM is in many ways characteristic of the work needed to exploit the advantages of ontologies in more traditional, object oriented modelled systems. One of these characteristic challenges is the preponderance of classes needed in the industrial setting. The enterprise needs to handle tens of thousands of types – types of building components, equipment, processes, roles, measurements, etc. Ontologies are naturally suited to representing and managing the corresponding collections of *master data*. A typical object oriented system is however modelled using UML, which is not suitable *out of the box* for managing such big collections of classes (due to the graphical design of the UML language, as well as the capabilities of modelling tools). Typical UML modelled systems are implemented using efficient relational databases (SQL). This implies a fixed schema of tables that doesn't trivially align with the schema-free formats of ontology-based content.

To reap the benefits of ontologies in more traditional systems, such as are built on the RSM, the practical approach needs to be chosen with care. IOHN delivered a sketch of how RSM could be developed, tentatively named "RSM 2.0". The proposed model would support a balanced trade-off between expressive power and efficiency. Some main challenges were pointed out in the IOHN presentation at *Semantic Days*, 20.05.2009. An O&G enterprise needs to manage large numbers of types of equipment. This is reflected in the fact that the PCA RDL contains tens of thousands of classes; and even this set covers only a fraction of what is in use at the assets in scope for IOHN and RSM.

> In the RSM there are many different types of *OrganizationalEntity*, *FunctionalEntity*, *Asset*, . . . In ISO 15926 those types of . . . are represented as classes in the meta-model.
> Adopting this approach for RSM is highly interesting, since it allows for exploiting ISO 15926 equipment classification; has to be done with care, because, e.g., ISO 15926 has 10.000+ equipment types represented as OWL classes; Representing each such OWL class per equipment type as a UML class in RSM would lead to 10.000+ sparsely populated DB tables. [20, slide 32]

In the proposed RSM adaption, main categories of equipment – to be identified statistically by looking at the data in scope for the enterprise use case – would be modelled as UML classes. Specialized equipment types, as are represented by subclasses on the ontology/RDL approach, make up the bulk of reference equipment types. These would be

An O&G Reference Data Library will contain thousands of reference classes – A challenge to relational database design.

modelled as UML instances (see the following figure).

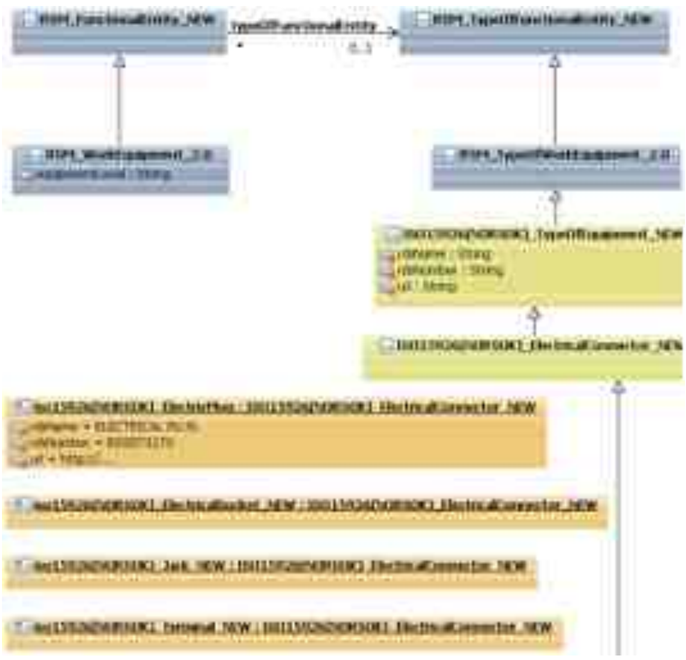


Figure 3.25: Equipment type hierarchy for RSM–ISO 15926 map (excerpt from UML model)

On this RSM modelling approach the number of UML classes in the model, and the corresponding number of SQL database tables, could be tuned for a reasonable tradeoff between expressivity and efficiency, allowing for a large number of equipment types. It was unfortunately not possible to develop this "RSM 2.0" initiative into an implemented system within the IOHN project.

3.4.10.4 PCA RDL inclusion, and SIG review

In March 2009 a set of reference data as required to cover the delivered mapping between RSM and ISO 15926 modelling patterns and PCA RDL reference entities, was included in the PCA RDL. The set contained 56 classes and relations (plus metadata), representing entities from the "core" of RSM chosen for the IOHN project. As the RSM–ISO 15926 activities were eventually taken out of the IOHN scope, this set of reference data was removed from the PCA RDL in May 2011.

It was planned to have a PCA Special Interest Group for Operation & Maintenance review the mapping from RSM into ISO 15926, and the content representing RSM that was added to the PCA RDL. However, it turned out not to be feasible to organize such an interest group, in spite of extended attempts by the activity leaders at DNV. There was insufficient interest in this work in the industry, and little funding available for compensating potential members of a SIG.

3.4.11 Data Exchange using iRING

For ISA Expo 2009, IOHN collaborated with the iRING project and Bechtel, to prototype and demonstrate data exchange for Operations and Maintenance using ISO 15926. The exchange made use of new software developed in the *Camelot* project, a collaboration between POSC Caesar and Fiatech to develop the iRING content integration project [2, 32]. The Bechtel team, with iRING developers Rob DeCarlo and Darius Kanga, contributed great enthusiasm and effort to making this pioneering exchange a success.

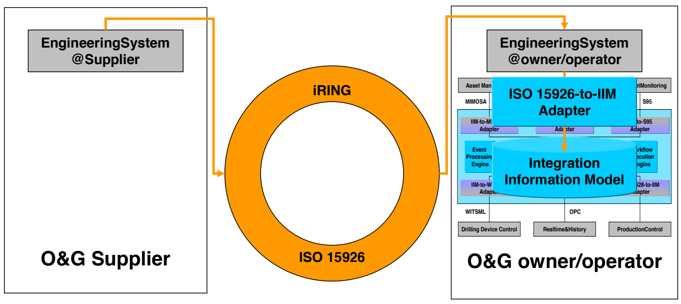


Figure 3.26: "ISO 15926 as a cross-enterprise exchange format for engineering data", demonstration at ISA Expo 2009 [26, p. 7]

The focus on O&M domain topics in this subproject complemented the production focus that otherwise dominated the Semantic Model activity. The exchange methods belonged with iRING's prototyping Part 9 of ISO 15926, although this was not strictly in scope for the IOHN project. Apart from that developed in the Camelot project itself, this demonstration was likely the first use of ISO 15926 templates to exchange process plant information.

Figure 3.27 shows the information flow of the demonstrated exchange, from owner/operator via iRING to an Engineering company. The demonstration was based on the following scenario.

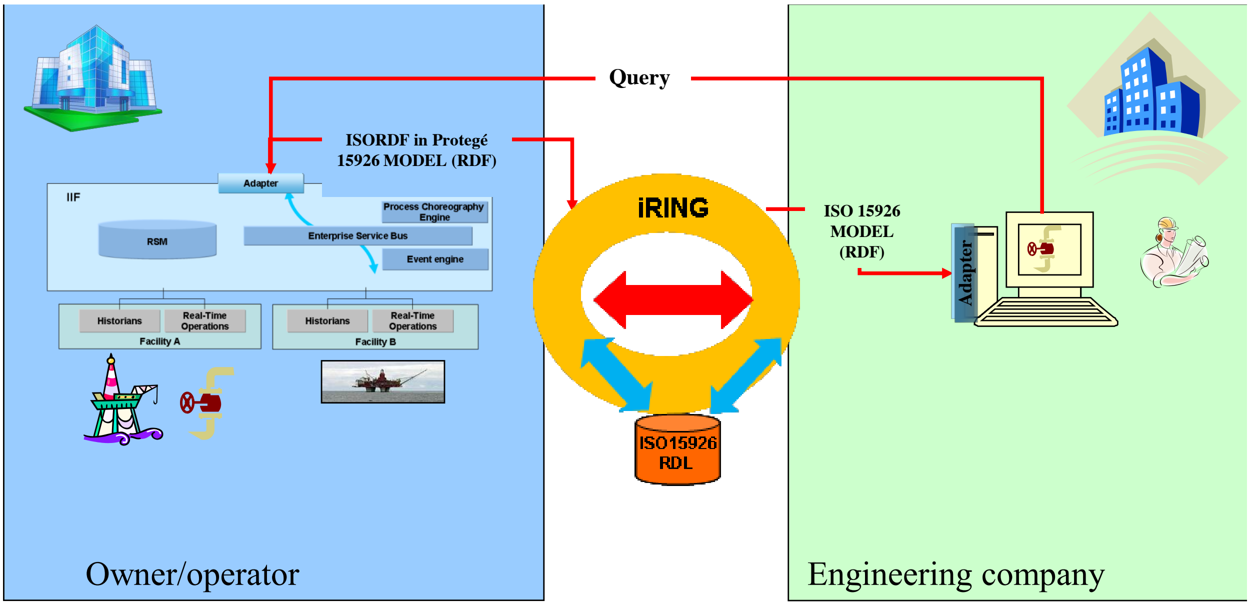


Figure 3.27: Information flow demonstrated at ISA Expo 2009 [21, slide 13]

> The operator has decided to tie in a set of wells on a new field to the Ullrigg platform. Bechtel is the contracted company for the modification job.
>
> The tie in will lead to a higher flow through Ullrigg's separator train, and the well stream from the tied in wells has a higher temperature than the streams already processed on Ullrigg.
>
> Jon is the Bechtel engineer assigned to the job checking if the installed equipment needs modifications or replacements. Related to this he needs as-is information for the valve installed on tag VB3101.
>
> In stead of requesting paper copies of the data sheets for valve VB3101 he will query the operator's integration platform for information from his PC at Bechtel's location, receiving a data set (model) back on "neutral" standardized format. [21, Slide 10]

At the ISA Expo conference, the IOHN and iRING teams gave a live demonstration. An IIF system containing RSM data was queried – a model of an oil platform was accessed, to retrieve data on pressures in a tank. The query results were translated into iRING ISO 15926 templates, and then transferred to a Bechtel service center for investigation in an engineering tool. The IIF server located in Stavanger, and the Bechtel system in Frederick, functioned as iRING nodes. The demonstration was executed with no technical difficulties. Video links were used to show conference attendants how systems were engaged at each end of the exchange.

A minimal plant model (for the fictitious "Polar Champion A" platform) was created for this demonstration, and represented using RSM on an IIF test server at Statoil. "There exists a platform, on which there exists a responsibility area, under which there exists a system, which has a pressure tank, on which there is a pressure sensor. Give me data for that sensor."

For the statements transferred, templates and reference library entities were designed and represents in the formats required by the iRING software (using then-current iRING spreadsheet formats).

3.4.11.1 Adapter: From an RSM model to ISO 15926 statements

The IOHN integration platform, with the RSM model and the IIF platform, was presented as a basis for exchange of ISO 15926 compliant information. Quoting the *summary* slide of the ISA Expo presentation:

> - ISO 15926 → get the terminology right
>   RSM → implement based on the right terminology
> - Bi-directional translations between ISO 15926 and RSM
> - Enable shopfloor-to-topfloor/vertical integration scenarios
> - IIF is an integration layer platform which can be hooked into in an iRING network
>
> [27, slide 24]

Thus, the architecture of a semantics-based integration platform is a natural basis for ISO 15926 compliant exchange.

For the demonstration, content processing at the data source, representing the owner/operator of the Ullrigg oil platform, included the following steps.

1. An IIF query returns an excerpt of the IIF model of the oil platform, as XML data

2. An XQuery query translates that data into RDF, exploiting the RSM–ISO 15926 mapping developed in the project, and PCA-RDL reference classes

3. A SPARQL query converts the model into ISO 15926 templates using the iRING format

The resulting data set was suitable for being transferred to the engineering company (Bechtel), were an iRING node received data in the template format. On that end, the data set was imported in the Bechtel engineering tool for investigation and analysis.

The query language used to retrieve data from the source system was the IIF QL native to IIF (cf. figure 3.5). The following query was used to retrieve an excerpt of the Ullrigg model. On execution in the demonstration, this query was submitted to a live IIF web service, returning results in JavaScript Object Notation (JSON ☞) format.

```
rsmQuery.enterprise('name' eq 'IBM Oil')
  country('name' eq 'Europe').
  field('name' eq 'Stavanger').
  productionUnit('name' eq 'Ullrigg').
  workCenter('name' eq 'Separators').
  workCenter('name' eq
      'CD7101-X').equipment().
  equipment().equipmentCategory()
  equipmentPropertyDefinition()
  equipmentProperty();
```

Figure 3.28 shows the query result, after conversion from XML to RDF (screenshot from the RDF Gravity 1.0 visualization tool ☞). The figure shows an asset hierarchy from the enterprise, via field and plant down to a pressure tank. The pressure tank assembly includes a ball valve, one of the components that the engineering company should look at, for which some performance characteristics are given.

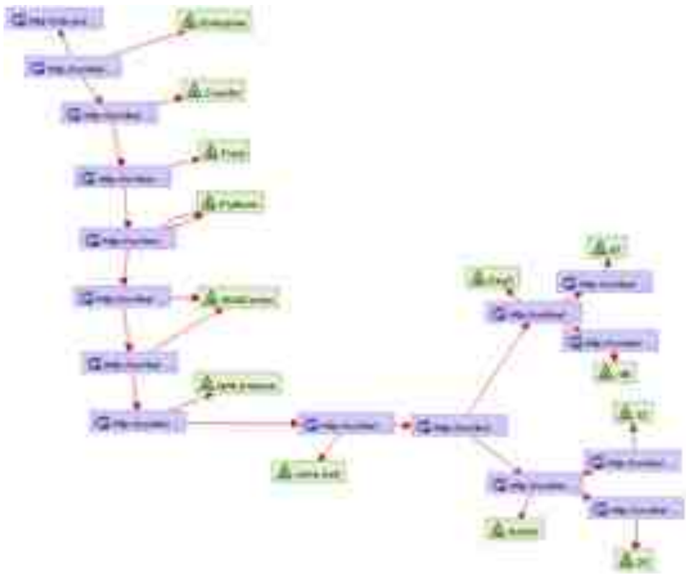


Figure 3.28: *Ullrigg* model fragment exchanged at ISA Expo 2009, in RDF

Figure 3.29 gives a simplified visualization of template instances, with the name of each template in the leftmost cell of each node, and template rolefillers in the subsequent cells. The figure illustrates how facts about the performance range of (made-up) commodity equipment classes *Ball Valve Class 300* and *EV-01-ES-01* are expressed as template instances.

All templates for this demonstration were modelled in accordance with ISO 15926. The template *ArrangementOfIndividualTemplate*, which is used for "*x* is part of *y*" statements, is a member of the ISO 15926-7 *initial set* of templates (a set of generic templates defined in the standard itself). The other templates, such as *WorkCenterContainsEquipment*, were created for the Ullrigg case according to ISO 15926-2. Using the mapping between RSM and ISO 15926 developed earlier in 2009 (as described in section 3.4.10), mapping IIF output into template statements could be done in a straightforward and precise fashion. Figure 3.30 illustrates how facts about the physical composition of the Ullrigg plant *Platform Ullrigg* are expressed as template instances: Ullrigg is part of the *Field Stavanger*, and contains a work center, with a pressure tank that has three valves as parts.

Figure 3.31 shows how the data queried from the Ullrigg model appeared in Bechtel's INSPEC engineering tool.

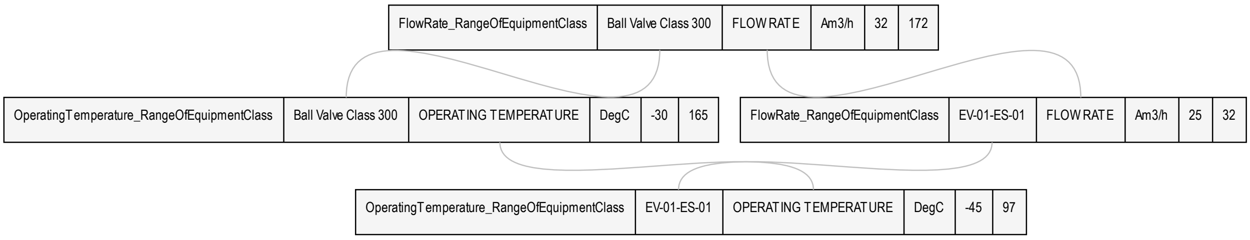


Figure 3.29: *Ullrigg* equipment class characteristics stated in ISO 15926 templates

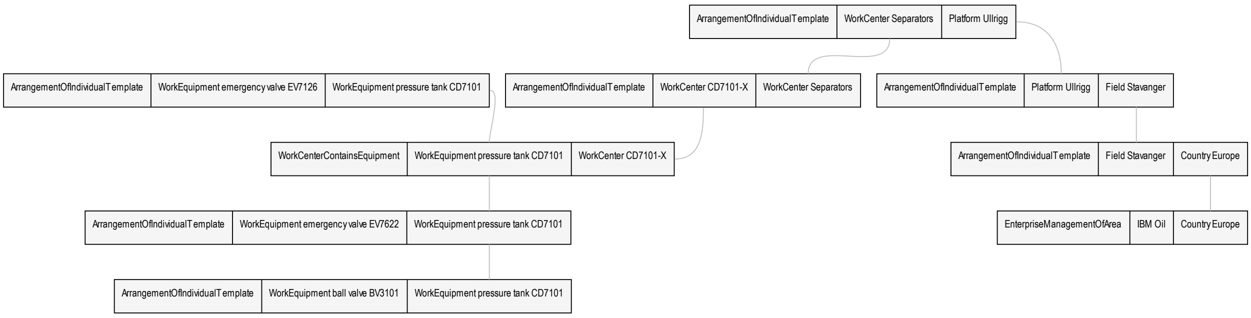


Figure 3.30: *Ullrigg* platform structure stated in ISO 15926 templates

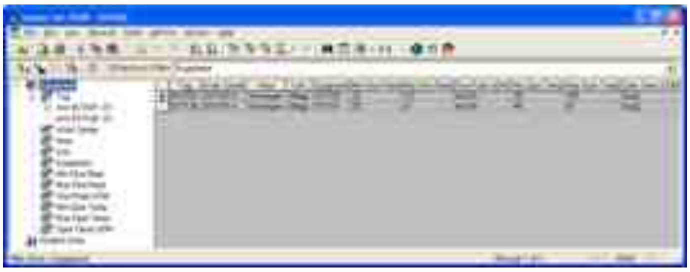


Figure 3.31: "Valve data in Inspec, the Bechtel engineering application" [21, slide 19]

The work done to express the Ullrigg model in the form of ISO 15926 templates contributed valuable knowledge for the IOHN Semantic Model activity. It produced a working example of how a model provided in an arbitrary format, here the RSM representation of Ullrigg, could be translated into ISO 15926 template format using standard tools – in this case, XQuery and SPARQL queries. The templates used in the demonstration were not ISO 15926-8 compliant, but expressed in RDF/XML according to the iRING software format requirements. This non-compliance is only a minor technical detail. The real achievement was to show ISO 15926 templates and reference library data applied in a functioning exchange setting. The reference library classes used in the example were taken from the PCA-RDL, with FLOW RATE and OPERATING TEMPERATURE chosen to characterize properties of valves.

3.4.12 FIIG codification case

The use of broad-ranging, comprehensive classification systems has a long history in the defence sector, at a much larger scale than what we find in the O&G domain. There is clearly a great potential for O&G to learn from this established practice. At the same time, the Integrated Operations work in O&G has developed new concepts and methods that can likely have great value if applied in the defence domain. In a sub-activity led by defence information experts at DNV Industry, it was investigated how the domains can benefit from adopting best practices from each other.

> This limited study has focused on investigating current and parallel practices in NATO and Oil & Gas industry to give recommendations for areas that should be further explored with the aim to uncover common benefits for both Defence and Oil & Gas use. [11, summary]

Within NATO a large number of separate organizational entities (land, sea, and air domains) that rely upon complex technical systems and patterns of operation need to be coordinated. There are obvious analogies to the O&G domain.

The NATO Federal Item Identification Guide (FIIG) codification system was selected for investigation as a characteristic case with wide applicability. Work focused on development of a demon-

While O&G item identification schemes are in their infancy, NATO supply item schemes have existed for decades.

stration system for *supply item* (spare part) management, and on semantic topics in modelling and interpretation across established paradigms, i.e., selected NATO systems compared to ISO 15926.

Figure 3.32: Screenshot from pilot: ISO 15926 for Supply Item codification [11, slide 23] ☞

The FIIG activity explored ISO 15926 for *asset characterization*. This has potential value in the defence domain due to the ontology character of ISO 15926 reference data, including multiple classifications ("multiple inheritance") and explicit relational constraints. These features aren't found in traditional defence classification systems, which have the form of less formal class taxonomies. This is therefore an area where the defence domain can learn from ISO 15926 and the O&G work in integration of information.

W3C formats were used to build the web services and pilot/demo application for the FIIG case, combining NATO and ISO 15926 concepts. Little prior art exists for this kind of combination, and the IOHN effort demonstrated in an innovative way how a software system can exploit two "semantic" systems of characterization, for benefit to the user.

This activity also described for the IOHN community how the FIIG and other NATO reference data systems are organized and operated. The O&G sector has much to learn from the more than 50 years of experience with common classification schemes found in the defence sector. This includes how to run the organization, and how to secure cooperation between many countries. For instance, this will include managing services with well-defined responsibilities for consistency, quality, and response time when introducing new classes to the reference data repository.

The codification study focused on a particular use case: A Norwegian Defence *Recommended Spare Parts List*, a template for recording supply items to be kept in a store of military equipment. We will refer to the spare parts lists as "ARL", abbreviating the Norwegian "*anbefalt reservedelsliste*". This is a typical case, where keeping a properly stocked store for maintenance and repair [check to find right acronym] requires recording and managing parts that are to a large extent commodities, i.e., classes of things that lend themselves to management in a reference data library of the ISO 15926 type.

- Capturing the commodities in the RD language can allow for better maintenance of the information. For instance, automated recognition of interchangeable parts is considerably easier on an ontology approach, where information about characteristic features is machine processable, than using less structured "code list" methods.
- Which parts are required for a given type of (complex) equipment can be naturally captured as part/whole constraints in a reference data library of the ISO 15926 kind.

An ARL records required stock for a given type of equipment. Current practice in the defence administration is to use a spreadsheet with about 25 columns for recording characteristics about each spare part.

The use of a spreadsheet template for ARLs introduces *consistency* challenges. Are the spreadsheets used in the same way across contexts (between sites, among personnel filling out the forms; over time)? Harmonization of practices is clearly desirable. The IOHN Semantic Model work on the ARL targeted two distinct levels of classification.

- At the meta-level, the project described the relationship between the Nato Stock Number (NSN) and . . . [check]. Each codification uses its own specialized vocabulary, which takes study before it can be reliably used. The mappings between codification systems can be quite complex in structure, and require much manual investigation. In the pilot, the terminology of the classification systems themselves (the ISO 15926 and NATO concepts used to set up and structure registries) was made accessible directly in the application.

- At the first-order level, the pilot showed how ISO 15926 and NATO classifications could be combined in an ARL registry.

The user interface of the codification pilot application provided easy access to the definitions of the schemas – FIIG codes and ISO 15926 concepts. This would allow personnel working with ARLs to conveniently look up reference information about the standards.

Integrating the meta-level and first-order level of information, the *codification* activity as an efort to define the ARL spreadsheet patterns by means of ISO 15926 templates. It was considered how an ARL spreadsheet (a filled-in form) could be seen as a set of facts expressed in ISO 15926 template statements. (In this way, the codification activity was analogous to the Production Pilot activity of mapping sand production data.) Due to budget constraints, this template development work could not be completed with a definite delivery of templates. However, it did provide insight into the challenge of bringing the strict methods of ISO 15926, based in formal logic, to the domain of a long-established classification system designed to be used without a formal language (i.e., while the context of use is formalized, the content is not expressed using a formal language expressed as an ontology) and interpreted by humans.

In the NSN, it is for instance not uncommon to find categories that are difficult to model well in the extensional paradigm of ISO 15926 (and, indeed, in first-order languages like OWL). ISO 15926 and similar frameworks doesn't provide primitives for *default* property characterizations, and the use of implicitly negative characterizations such as "other types of X" are non-trivial to accommodate. The use of ISO 15926 raises the bar for precision in content. Pragmatic, useful solutions for applying the semantic approach are of great interest, and the FIIG template development effort is a typical example.

For the ARL spreadsheet template, it was found that the columns provided were less clearly defined than one would have wanted. A mapping to ISO 15926 (and wrapping up the pattern into ISO 15926 templates) would require clarifying what information was required, with stricter definitions. The ARL is designed to be written and read by humans; the ISO 15926 form will, by virtue of the systematic approach alone, incorporate rules for the consistency and completeness of an ARL. It will also imply what a database of ARLs consists in. Clearly this raises the bar on what is an acceptable ARL – but also opens great potential for automated checks on data quality, and more efficient and economical spare parts management.

In the defence sector, there exist much bigger volumes of structured reference data than in the O&G domain. The history of management of institutional reference data management is also far more comprehensive. The FIIG codification activity in IOHN was limited in time and budget, but succeeded in revealing some main challenges and opportunities that deserve further exploration.

3.5 Dissemination

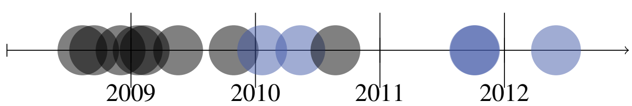
3.5.1 Presentations

The presentations were in general held in collaboration with the Integration Platform, Drilling Pilot, and Production Pilot activities. (IOHN has been presented at more conferences than the ones listed here; this is restricted to the ones where Semantic Model had a major role.)

- Check: Position papers for W3C workshop at Chevron. One paper for RSM, submitted independently for ISO 15926?
- Semantic Days, 2009, 2010, 2011
- ISA Expo, October 2009 (iRING/RSM)
- Intelligent Energy, 2010

3.5.2 Workshops

The following timeline shows Semantic Model workshops in grey, and workshops held together with other activities (Drilling Pilot, Production Pilot, and Integration Platform) in blue.



Initial workshops, until summer 2009, focused on mapping the RSM model to ISO 15926. From the end of 2009, the challenge of developing ontologies for the Drilling Pilot and Production Pilot use cases was the main topic. The restructuring of the overall IOHN project explains why there were no workshops from winter 2010 until fall 2011.

Several of the IOHN Semantic Model workshops introduced and discussed novel advances in semantics. For instance, at the ontology workshop of 2009.10.29, prof. Arild Waaler (UiO) gave an early presentation of Ontology-Based Data Access (OBDA), as applied to producing XML reports for offshore O&G ☞. In the IOHN project, OBDA work was a minor activity. It has however had considerable significance, as preparatory for the upcoming large EU project Optique (see section 3.2.4).

1. DNV, Høvik, 11.08.–15.08, 2008: *RSM/ISO 15926* ☞
   - Included a presentation of ISO 15926 results from the IDS project, providing a basis for ontology work in IOHN.
2. DNV, Høvik, 15.09–19.09, 2008: *MIMOSA and ISO 15926* ☞
3. Statoil and DNV, Stavanger, 01.12–05.12, 2008: *MapIT/GODI, RSM, and ISO 15926* ☞
4. DNV, Høvik, 21.01–23.01, 2009: *RSM in ISO 15926*
   - Slide 6 expresses an IOHN project goal: “By March 1st 2009: Represent in ISO15926 the scope of the Reference Semantic Model (RSM) currently exploited by the F0B/GODI (Tail) project”

   ☞
5. DNV, Høvik, 11.02–13.02, 2009: *RSM in ISO 15926* ☞
6. Conference venue, Stavanger, 18.05, 2009: *Ontology development*
   - This short workshop was held the day before the Semantic Days conference. Slide 29 gives a 4-year roadmap which is interesting to compare to the actual project outcome: many goals have been met.

   ☞
7. DNV, Høvik, 29.10, 2009: *Ontology. Use case requirements; modular IOHN ontology*
8. DNV, Høvik, 23.08, 2010: *ISO 15926 templates* ☞
9. DNV, Høvik, 31.05, 2012: *Final workshop*

Related workshops:

- Integration Platform activity, 25.08–26.08, 2009: *IOHN Activity 2 Workshop.* Included *model query* session discussing the choice of query language for the IOHN architecture. ☞
- Production Pilot activity, 21.01, 2010: *Use case ontology* ☞
- Production Pilot activity, DNV, Høvik, 11.05, 2010: *Interoperability. Query language for IOHN.* ☞
- Drilling Pilot activity, NOV, Stavanger, 04.10–05.10, 2011. *ISO 15926 templates for drilling* ☞
- Integration Platform activity, Baker Hughes, Stavanger, 06.10, 2011. *Anzo RDF server*

3.6 Conclusion, and lessons learnt

In conclusion, the IOHN prototype model of the *Snorre B* platform (section 3.4.2) deserves attention as a proof-of-concept standard compliant reference model. This is the kind of model that a sustainable IO architecture needs (section 3.4.1). The model features of note include the following (some are improvements on existing offerings, others are new developments).

- The model is compliant with the ISO 15926 representation standard, and free of dependencies on proprietary software tools. This promotes vendor independence and a robust framework for content validation.
- The model provides a plant topology that includes descriptions of data sources. This allows for a simple mechanism for finding and retrieving bulk data.
- Using Linked Data for references and SPARQL for search and retrieval, the model supports efficient data access.

The semantic model enables integration across domains: Modelling assets, linking from office documents, harmonizing vocabulary, enabling data access.

- Documentation of the model is integrated into the model itself, by virtue of being expressed in the language of an ontology.
- The model is expressed in a language of classes and relations that is maintained independently, facilitating reuse and interfacing with different contexts.

The ontology used to express the IOHN example model of Snorre B covers only a fraction of the data types and model constructions included in e.g. the POSC/Energistics ☞ , PPDM ☞, or Schlumberger Seabed ☞ data models. The model has interest because it demonstrates how a model can be designed and used without any tie-in to proprietary software or formats. An enterprise can use this approach to model assets, to be employed in a master data role, while keeping full ownership and control over the representation.

The model approach explored in the Semantic Model activity suits a variety of use scenarios. The UML information models of OPC-UA (IEC 62541) is one case in point. Another is the message content of existing enterprise information bus solutions, to which the ISO 15926-8 format promises to combine compliance, semantic precision and ease of use.

The IOHN Semantic Model activity spanned a wide range of subjects over the four-year project period. Some main goals were pursued from start to finish, such as the implementation of standardized reference data across enterprise contexts, from data sources to applications and exchange. Some subjects were taken into the project scope as the result of new technological possibilities; in particular the Linked Data approach belongs in this category. There were also some tasks that were dictated by unforeseen reconfigurations in the consortium, including the change of implementation model, where RSM was replaced by ISO 15926 templates, re-purposed from an exchange setting into the reference model.

With this spread of tasks, the activity team gathered experience on a wide range of subjects. At project closing, some of the *lessons learnt* appear to represent firm and lasting knowledge, while others are preliminary. We offer the following main points to be observed in implementation and research activities that will follow up the Semantic Model activity.

- Be radical in avoiding application lock-in (i.e., use *anything*; our primary case is Visio, re-purposed as modelling tool)
- Mix external and enterprise reference data, avoiding local proprietary schemas where possible.
- Use the W3C stack for RDF/OWL to carry data, and SPARQL for access. The W3C standards are usable today: RDF is stable, SPARQL is stable in the 1.0 version and actively developed into version 1.1. Use OWL 2, which provides crucial expressive power and semantic rigor.
- The ISO 15926 standard is ready for use in the RDF form laid down in ISO 15926-8. Using this representation form, difficult modelling decisions, which abound in the O&G subject matter, can be put on hold while the terminology of reference libraries is put to good use. ("Full" or "deep" modelling with the primitives of ISO 15926 part 2 will likely be needed to support enterprise integration contexts with great variety of schemas than were in scope for IOHN.)
- The software tools that are available for semantic modelling need improvement. There are too many (hundreds) competing solutions for mapping between relational/tabular sources, and the graph structures of RDF. The OWL/RDF modelling tools are not mature enough, and not usable for the O&G engineering domain experts whose knowledge we wish to capture for a semantic model. (Note that these points of criticism apply also to UML approaches to data modelling. Modelling for enterprise integration using RDF is not harder than using for traditional formats.)

- Use the PCA RDL, which has a standing as the preferred reference library for the O&G domain, in particular for the NCS. The library has some shortcomings, with hosting on a proprietary system, no modularization, and limited verification of consistency. It still provides great value as a common source of generic industry concepts. Shortcomings are being addressed; a Linked Data form implemented by the JORD project ☞, shows good progress toward a reliable solution.

- Develop clear, verifiable criteria for model integrity. For take-up of RDF/OWL and ISO 15926 content in O&G settings, the work [7] should be developed into a widely accepted rule set.

3.7 References

[1] IOHN Semantic Model team. Sand production and erosion: Integration model. In *IOHN Steering Committee meeting*, May 2011.

[2] Robin Benjamins. The Camelot Project. In *Semantic Days*, June 2009. Presentation of iRING, the IOHN ISA Expo demonstration with RSM, and JORD. Available at https://www.posccaesar.org/browser/pub/SemanticDays/2010/SessionIOHN_Benjamins_IOHN%20CamelotiRING.pdf.

[3] Tim Berners-Lee. Linked Data – design issues, 2006. Available at http://www.w3.org/DesignIssues/LinkedData.html.

[4] R. Fielding et al. *Hypertext Transfer Protocol – HTTP/1.1*. The Internet Engineering Task Force (IETF), 1999. Available at http://www.ietf.org/rfc/rfc2616.txt.

[5] Roy Thomas Fielding. *Architectural Styles and the Design of Network-based Software Architectures*. PhD thesis, University of California, Irvine, 2000.

[6] Henrik Forsell. *Semantic Technology and ISO 15926: An Introduction*. OLF, December 2008. Tutorial, with slides for classroom use. Available at https://www.posccaesar.org/browser/projects/IOHN/Activity%203%20%28Semantics%29/ISO15926_Tutorial.

[7] Martin Giese. Quality criteria for ISO 15926-8 compliant installation descriptions. In *Semantic Days*, May 2011.

[8] Martin Giese, Jens I. Ornæs, Lars Overå, Inge Svensson, and Arild Waaler. Using semantic technology to auto-generate reports: Case study of daily drilling reports. In *SPE Intelligent Energy International*. SPE, March 2012. Available at http://www.onepetro.org/mslib/app/Preview.do?paperNumber=SPE-150225-MS&societyCode=SPE.

[9] Ian Glendinning. The JORD (Joint Operational Reference Data) Project, December 2011. Available at https://www.posccaesar.org/attachment/wiki/FiatechJord/JORDFlyer2011WhiteFinal.pdf.

[10] Angela Guess. Semantic case study: EPIM ReportingHub. In *SemanticWeb.com Industry News*. SemanticWeb.com, February 2012. Available at http://semanticweb.com/semantic-case-study-epim-reportinghub_b26590.

[11] Tore Hartvigsen. Use of product data codification schemes and reference data sources in defence and oil gas applications. Technical report, DNV, May 2010. Available at https://www.posccaesar.org/browser/projects/IOHN/Activity%203%20%28Semantics%29/ReferenceDataCodificationCase/Codification_pilot_report.ppt.

[12] Tom Heath and Christian Bizer. *Linked Data: Evolving the Web into a Global Data Space (1st edition)*. Synthesis Lectures on the Semantic Web: Theory and Technology. Morgan & Claypool, 2011. Available at http://linkeddatabook.com/editions/1.0/.

[13] Pascal Hitzler, Markus Krötzsch, Bijan Parsia, Peter F. Patel-Schneider, and Sebastian Rudolph, editors. *OWL 2 Web Ontology Language: Primer*. W3C Recommendation, 27 October 2009. Available at http://www.w3.org/TR/owl2-primer/.

[14] Knut Hovda. Specification: Data in IIF for IOHN demonstrator – Erosion Management. IOHN project internal report, January 2010.

[15] Jena Eyeball Development Team. *Jena Eyeball 2.1 Documentation*. Apache Software Foundation, 10 February 2004. Available at http://jena.sourceforge.net/Eyeball/full.html.

[16] Johan W. Klüwer. *IOHN Templates Workshop*. DNV, August 2010. Tutorial, with example data and scripts. Available at https://www.posccaesar.org/browser/projects/IOHN/Activity%203%20%28Semantics%29/Workshop_2010-08_Templates/IohnTplWsFall10.pdf.

[17] Johan W. Klüwer, Martin G. Skjæveland, and Magne Valen-Sendstad. ISO 15926 templates and the Semantic Web. In *W3C Workshop on Semantic Web in Energy Industries; Part I: Oil & Gas*, December 2008. Position paper. Available at http://www.w3.org/2008/11/ogws-papers/kluwer.pdf.

[18] Einar Landre, Frode Myren, Udo Pletat, and Johan Klüwer. Model-driven integration architecture for IO G2. In *IO 09 Science and Practice*. Center for Integrated Operations in the petroleum industry, September 2009. Available at http://www.ioconf.no/2009/presentations/parallel6.

[19] Thore Langeland. Integrated Operations reduce risks and improve productivity. In *Offshore Oil Executive Briefing*, Houston, September 2010. IBM. Available at https://www.posccaesar.org/browser/pub/PCA/MemberMeeting/201010/PresentationsDay2/20101021_Alan_T_Johnston.pdf; event website at www.ibm.com/events/offshoreoil2010.

[20] Frode Myren, Udo Pletat, and Johan Wilhelm Klüwer. Model-driven integration architecture for IO G2 information – Reference Semantic Model alignment to ISO 15926. In *Semantic Days*, May 2009. Available at https://www.posccaesar.org/svn/pub/SemanticDays/2009/Session_6_Myren_Pletat_Kluwer.pdf.

[21] Frode Myren, Pål Rylandsholm, Johan W. Klüwer, Knut Hovda, and Jone Aarre. A demonstration of interoperability based on open standards. In *ISA Expo*, 2009. Demonstration at ISA Expo 2009 – "IOHN represented by IBM and DNV in collaboration with Bechtel". Available at https://www.posccaesar.org/browser/projects/IOHN/Activity%202%20%28Infrastructure%26WS%29/Deliverables2009/ISAExpo_demo.ppt.

[22] Martin Nally and Steve Speicher. *Toward a Basic Profile for Linked Data*. IBM developerWorks, 2011. Available at http://www.ibm.com/developerworks/rational/library/basic-profile-linked-data/.

[23] Martin Nally, Steve Speicher, John Arwe, and Arnaud Le Hors. Linked data basic profile 1.0, 3 2012. Available at http://www.w3.org/Submission/2012/SUBM-ldbp-20120326/.

[24] OLF IO project group for IT Architecture. Reference architecture of IT systems for OLF's IO G2: A preliminary report. Technical report, OLF, March 2008. Available at http://www.olf.no/PageFiles/14295/090801_IT_Architecture_IO_G2.pdf.

[25] Udo Pletat. Model Query languages for use within IOHN: Summary of different query language approaches. Workshop presentation, IOHN activity Integration Platform. Available at https://www.posccaesar.org/browser/projects/IOHN/Activity%202%20%28Infrastructure%26WS%29/Deliverables2009/IOHN%20ModelQueryLanguages_V0.2.ppt, August 2009.

[26] Udo Pletat, Johan Wilhelm Klüwer, Einar Landre, Jørn Ølmheim, Frode Myren, and Pål Rylandsholm. ISO 15926- & RSM-based integration architecture for oil & gas industry solutions. In *ISA Expo*, 2009. Available at http://www.isa.org/Template.cfm?Section=Technical_Paper_Collections&Template=/Ecommerce/ProductDisplay.cfm&ProductID=10806.

[27] Udo Pletat, Johan Wilhelm Klüwer, Einar Landre, Jørn Ølmheim, Frode Myren, and Pål Rylandsholm. ISO 15926- & RSM-based integration architecture for oil & gas industry solutions. In *ISA Expo*, 2009. Presentation at ISA Expo 2009. Available at https://www.posccaesar.org/browser/projects/IOHN/Activity%202%20%28Infrastructure%26WS%29/Deliverables2009/ISAExpo2009.ppt.

[28] Alex Rodriguez. *RESTful Web services: The basics*. IBM developerWorks, 2008. Available at http://www.ibm.com/developerworks/webservices/library/ws-restful/.

[29] Pål Rylandsholm. Monthly Production Report (MPR) V. 1.0.0 – Terminology. Technical report, EPIM, June 2009. Available at http://production.posccaesar.org/browser/standards/MonthlyProductionReport/1.0.0/semantic_report_MPR.doc.

[30] Richard Stallman and Roland McGrath. *GNU Make Manual*. Free Software Foundation, 1997–2010. Available at http://www.gnu.org/software/make/manual/.

[31] Inge Svensson and Johan W. Klüwer. Platform selection for IOHN test lab – decision document. Technical report, IOHN activities Integration Platform and Semantic Model, December 2010. Available at https://www.posccaesar.org/browser/projects/IOHN/Activity%202%20%28Infrastructure%26WS%29/Semantic_Platform/IOHN_Note_IOHN-Platform-Selection-v1.1.pdf.

[32] IDS-ADI Project Team. iRING: Camelot project close out report. Technical report, POSC Caesar and Fiatech, June 2009.

[33] ISO 15926-1 team. Part 1: Overview and fundamental principles. In *International Standard ISO 15926: Industrial automation systems and integration – Integration of life-cycle data for process plants including oil and gas production facilities*. The International Organization for Standardization (ISO), 1 edition, July 2004.

[34] ISO 15926-2 team. Part 2: Data model. In *International Standard ISO 15926: Industrial automation systems and integration – Integration of life-cycle data for process plants including oil and gas production facilities*. The International Organization for Standardization (ISO), 1 edition, December 2003.

[35] ISO 15926-7 team. Part 7: Implementation methods for the integration of distributed systems: Template methodology. In *International Standard ISO 15926: Industrial automation systems and integration – Integration of life-cycle data for process plants including oil and gas production facilities*. The International Organization for Standardization (ISO), 1 edition, October 2011.

[36] ISO 15926-8 team. Part 8: Implementation methods for the integration of distributed systems: Web ontology language (owl) implementation. In *International Standard ISO 15926: Industrial automation systems and integration – Integration of life-cycle data for process plants including oil and gas production facilities*. The International Organization for Standardization (ISO), 1 edition, October 2011.

[37] Magne Valen-Sendstad, Ian Glendinning, and Johan W. Klüwer. IDS project final project report. Technical report, Intelligent Data Sets JIP, January 2009.

[38] W3C OWL Working Group. *OWL 2 Web Ontology Language: Document Overview*. W3C Recommendation, 27 October 2009. Available at http://www.w3.org/TR/owl2-overview/.

[39] W3C RDF Core Working Group. *RDF Primer*. W3C Recommendation, 10 February 2004. Available at http://www.w3.org/TR/rdf-primer/.

[40] W3C RDF Data Access Working Group. *SPARQL Query Language for RDF*. W3C Recommendation, 15 January 2008. Available at http://www.w3.org/TR/rdf-sparql-query/.

[41] W3C SAWSDL Working Group. *Semantic Annotations for WSDL and XML Schema*. W3C Recommendation, 28 August 2007. Available at http://www.w3.org/TR/sawsdl/.

[42] W3C XML Query Working Group. *XQuery 1.0: An XML Query Language*. W3C Recommendation, 14 December 2010. Available at http://www.w3.org/TR/xquery/.

3.8 List of deliverables

In the following list, preliminary or intermediate versions of deliverables have been left out.

The deliverables are available from the password protected project-internal WIKI-site: http://www.posccaesar.org/wiki/IOHN/Internal.

Title	Date	Contributors	Type
ISO 15926 training material	2008/12	Henrik Forsell (for OLF)	Tutorial compendium
Reference data for *Monthly Production Report*	2008/12	Henrik Forsell (for OLF), Bård Henning Tvedt (Epsis), Pål Rylandsholm (DNV)	PCA RDL content
Semantic Technology in Data Integration: Primer and Tool Survey	2009	Audun Stolpe (UiO), Martin G. Skjæveland (DNV/UiO), Espen H. Lian (UiO)	Report
Re-engineered version of AKSIO ontology	2009	Lars Overå (grad. student, UiO)	Report and OWL ontology
WITSML superstructure / Drilling Ontology	2009	Lars Overå (grad. student, UiO)	Report and OWL ontologies
Converter mechanism for lifting WITSML XML data to superstructure	2009	Lars Overå (grad. student, UiO)	Report and software prototype
Industry Content Standardization: Applying the IOHN Approach to Harmonization Across Domains	2010/06	Johan W. Klüwer (DNV)	Presentation, *Semantic Days*
Sources for vocabularies relevant for drilling and completion	2010	Inge Svensson (Baker Hughes)	Report
Relevant forms/statements/queries/ templates for drilling and completion	2010	Inge Svensson (Baker Hughes)	Report
Version 2.0 of drilling ontology state of the art report	2010	Inge Svensson (Baker Hughes)	Report
Ontology of classes and templates for sand monitoring and erosion management	2010	Bård-Henning Tvedt (Epsis)	OWL ontology
Pilot + report for FIIG codification case	2010	Tore Hartvigsen (DNV)	Report and software prototype
IOHN common ontology of classes and templates	2010	Johan W. Klüwer (DNV)	OWL ontologies
IOHN ontology development toolbox: ISO 15926-8 templates; Linked Data; Platform.	2010	Johan W. Klüwer (DNV)	Report, tutorial, and OWL ontologies
Quality Criteria for ISO 15926-8 Compliant Installation Descriptions	2011/05	Martin Giese (UiO)	Presentation, *Semantic Days*, Oslo
Semantic model for sand production and erosion – An ISO 15926 use case	2011/05	Johan W. Klüwer (DNV)	Presentation, *Semantic Days*, Oslo
IOHN model and ontologies on-line service	2011	Christian M. Hansen (DNV), Morten Strand (DNV)	Linked Data
ISO 15926 on-line asset model validator	2011	Christian M. Hansen (DNV), Morten Strand (DNV)	Web Service
IOHN-Production Pilot sensor data on-line service	2012	Christian M. Hansen (DNV), Morten Strand (DNV)	Linked Data
Using Semantic Technology to Auto-generate Reports: Case Study of Daily Drilling Reports	2012	Martin Giese (UiO), Jens I. Ornæs (National Oilwell Varco), Lars Overå (PCA), Inge Svensson (Baker Hughes), Arild Waaler (UiO)	*SPE Intelligent Energy* conference paper ☞

3.9 Public available material with contributions from the Semantic Model activity

The Semantic Model activity contributed to the following material that is publicly available. Links to external public sources are provided where available. Note that this material is not considered being a part of the project deliverables.

Title	Date	Contributors	Type
RSM core parts represented in ISO 15926, reviewed and "standardized" through the PCA framework	2009	Johan Wilhelm Klüwer (DNV), Udo Pletat (IBM), Frode Myren (IBM), Pål Rylandsholm (DNV)	Report
Model-driven integration architecture for IO G2 information - Reference Semantic Model alignment to ISO 15926	2009/05	Frode Myren (IBM), Udo Pletat (IBM), Johan Wilhelm Klüwer (DNV)	Presentation, *Semantic Days*, Stavanger ☞
ISO 15926- & RSM- based integration architecture for oil & gas industry solutions	2009/08	Udo Pletat (IBM), Johan Wilhelm Klüwer (DNV), Einar Landre (Statoil), Jørn Ølmheim (Statoil), Frode Myren (IBM), Pål Rylandsholm (DNV)	*ISA Expo 2009* conference paper ☞
Model-driven integration architecture for IO G2	2009/09	Einar Landre (Statoil), Frode Myren (IBM), Udo Pletat (IBM), Johan W. Klüwer (DNV)	*IO 09* presentation ☞
The role of ISO 15926 for applying enterprise service bus technologies in oil & gas industry solutions	2010/06	Udo Pletat (IBM)	Tutorial, *Semantic Days*, Stavanger ☞

Report title:
Final report: All activities

Date of first issue:	*Project:*	
01.05.2012	Integrated Operations in the High North	
Prepared by:	*Reviewed by:*	*Approved by:*
Torbjørn Skramstad	Jing Xie	Tom Thomsen

Contributions by:	*Chapter/section:*
Åsmund Ahlmann Nyre (NTNU)	4.5.1
Jostein Jensen (NTNU)	4.5.2
Thomas Østerlie (NTNU)	4.5.3
Jingyue Li (DNV)	4.1.1
Stig Ole Johnsen (NTNU)	4.3.2

Summary:
Software as an integrated part of critical components used in oil and gas installations, as well as a component of its own right controlling essential systems of physical component is increasingly being used in the oil and gas industry, because it promises more functionality, better operator guidance, lower building costs, lower cost of operation, as well as higher flexibility for future changes and upgrades. In particular, the concept of integrated operations will continue to need more software. The Dependability and Risk activity in IOHN has harvested methods and tools from more mature industries than the oil and gas industry, such as aerospace, railway and automotive and developed guidelines and recommendations for the qualification of cost efficient software intensive technologies in order to have more reliable and safe systems without significantly changing the cost of building and assessing the systems. The guidelines have been tested out in several trials together with FMC, Statoil and others.

Revision	*SVN*	*Status*	*Date*	*By*	*Reason/Changes*
0.1		Draft	11.05.2012	SKR	First full draft version
0.2		Draft	15.05.2012	SKR	Updated after review by Jing Xie
0.99	3558	Draft	24.05.2012	FVE	Released for review by Steering Committee
1.00	3592	Accepted	05.06.2012	FVE	Accepted by Steering Committee

Chapter 4

Dependability and Risk

4.1 Activity Summary

Software as an integrated part of critical components used in oil and gas installations, as well as a component of its own right controlling essential systems of physical component is increasingly being used in the oil and gas industry, because it promises more functionality, better operator guidance, lower building costs, lower cost of operation, as well as higher flexibility for future changes and upgrades. In particular, the concept of integrated operations will continue to need more software.

The overall expected impact of the activity is reduced probability of hydrocarbon production flow stop caused by software malfunction. Software malfunction can also lead to accidents causing harm to humans as well as to the environment. Consequences of software faults can be significant in cost.

The Dependability and Risk activity in IOHN has harvested methods and tools from more mature industries than the oil and gas industry, such as aerospace, railway and automotive, and developed guidelines and recommendations for the qualification of cost efficient software intensive technologies in order to have more reliable and safe systems without significantly changing the cost of building and assessing the systems. The guidelines have been tested out in several trials together with our partners.

Citations:

4.2 Introduction

4.2.1 Purpose

The overall goal of the activity Dependability and Risk (Activity 4) has been to improve the dependability of ICT in the oil and gas industry, especially for integrated operations thus reducing the probability of hydrocarbon flow stoppage or safety or environmental incidents due to software vulnerability or software malfunction. The objective of the activity has been achieved by

- Develop a detailed recommendation for qualification of new, innovative, and cost-efficient software intensive technologies.
- Develop recommendations for how to utilize existing and emerging standards for improved quality at reduced total life cycle expenses for oil and gas installations. Control of interface definitions and follow-up is a major challenge in this respect.

The Dependability and Risk activity has been executed along three main axes or Work Packages, Component Qualification, Architecture and Integration, and Life Cycle Processes. For each of these areas, guidelines are developed in several iterations, including partner trial and evaluation. In addition there have been one Work Package for PhD student research, including one post. doc. researcher and two PhD students, and one Work Package for dissemination of project results. The Component Qualification and the Integration and Architecture tasks are attacking the problem of how to build and integrate large software intensive systems. Large software intensive systems can be considered as a system of systems, since the system is built by integrating software components and subsystems from many different vendors. A System of Systems (SoS) is not merely a large and/or complex system made up of a number of individual systems: there is more to it than that. A key distinction is that a SoS's behaviour is more complex to analyse than by just considering the sum of its constituent elements. A SoS typically demonstrates a combination of characteristics, such as a certain degree of autonomy, dynamic collaboration and emergent behaviour, that pose a number of challenges to dependability analysis and assurance. There is an increasing need for the capabilities delivered by this kind of system, and common engineering challenges. One challenge is how to qualify the single subsystems or components. Another problem is what type of architectural solution to select, and how to integrate the various subsystems and components. The task focusing on the Life Cycle Processes is dealing with which software engineering tasks have to be performed and to what rigidity depending on the criticality of the system to be built. The most common framework for addressing process quality is the so-called capability maturity models. The industry standard for capability maturity assessment is the Capability Maturity Model Integration (CMMI) model developed by the SEI at Carnegie-Mellon University in Pittsburg, USA. However, the CMMI is mostly aiming at assessing organisations' development and integration of business software. The contribution in our activity has been to extend the CMMI to include the dependability aspects such as safety, reliability and security and tailor it to different criticality levels, or confidence levels.

4.2.2 Participants in this Activity

Participants in activity 4 have been DNV, FMC Technology, OLF, PSA (Ptil), NTNU, Kongsberg Maritime, SAS Institute, Statoil, Cicsco, IO Centre (NTNU/Sintef), Tieto Enator, Norwegian Research Council, Invenia and Norwegian Defence. Some of the partners withdrew from the project during the project period.

4.3 Preliminaries

4.3.1 Software Dependability

Software's increasing role in most industries creates both requirements for being able to trust it more than before, and for more people to know how much they can trust the software. A sound engineering approach requires both techniques for producing reliable products and for sound assessment of the achieved results. We use the concept of dependability informally to designate those system properties that allow us to rely on a system functioning as required. Dependability encompasses, among other properties, reliability, safety, security, and availability. These qualities are the shared concern of many sub-disciplines in software engineering (which deal with achieving them), of specialised fields like computer security, and of reliability and safety engineering. Several quality models have been proposed and used, however, a comprehensive and practical model of the properties that are important to critical systems have been lacking. In the dependability model defined in our work we have distinguished between the dependability goals, which are the properties of the software system that the users are concerned about, and the attributes (or ”itilities” that build upon these goals. Figure 4.1 shows the dependability model we have developed and used in this project. The main purpose of the model is to have a common understanding of software dependability. Dependability is regarded as the combined fulfilment of the goals of safety, security and continuity (in the meaning of business continuity, or system availability). These dependability goals are supported by a set of important first-order abilities (i.e. attributes), i.e. availability, reliability, maintainability, integrity and confidentiality. Other complementary lower level, or second-order (i.e. secondary support) abilities can be further incorporated when needed. Figure 4.1 shows the dependability goals at the top and the related quality attributes at the bottom. The trade-off is illustrated by showing the goals as the arm in a balance weight. With reference to this dependability model our research has been directed towards methods to obtain and assess safety, security and business continuity (availability) in systems used in the Oil and Gas industry, and especially in an Integrated Operation context. We have further been dealing with the trade-off concerns related to how to balance between the dependability goals, since these usually are counteracting.

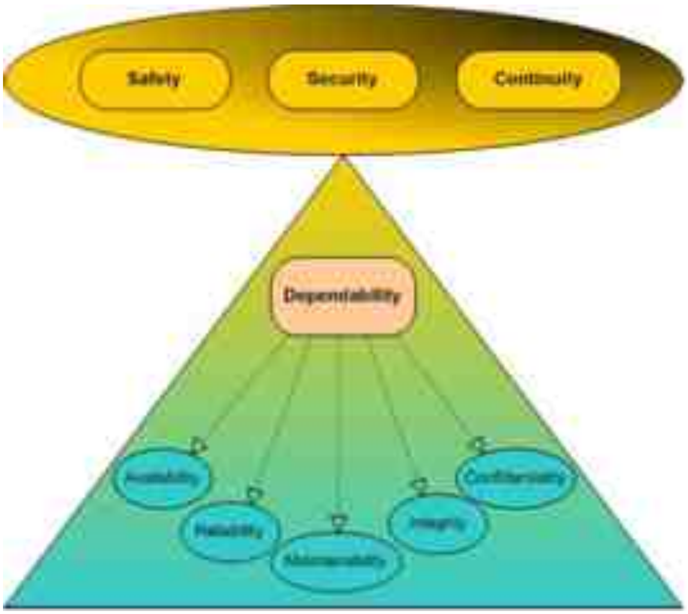


Figure 4.1: The Software dependability model used in IOHN Activity-4

4.3.2 Process versus Product Approaches

The dependability goals safety, security and continuity, represent a subset of the more general software quality properties. Software quality and software quality assurance have for many decades been an important part of systems and software engineering. The dependability goals are in particular important when dealing with critical systems.

Many different factors contribute to obtaining system and software quality. This is the topic of the large amount of systems and software quality standards. Some of the most important factors are: competence and capabilities of the people involved in the development and verification of the software, the tools and methods used in the development process, the system and software architecture and the amount of verification and validation activities, including testing performed. A large portion of the software engineering community believes in the best way to obtain and assess system and software quality is to ensure that the software development process has high quality. Their assertion is that if you master the software process then the software product will have good quality. Even if most professional software engineers believe that there is a relation between the process quality and the quality of the developed product, they claim that assessing process quality alone is no guarantee for a high quality product, and that the real quality of a product can only be assessed by verifying the product itself. At least for the most critical systems most professionals agree that assessing only the process quality is not sufficient. Most software development organisations use both approaches. Most of the research in our project has been focused on product oriented approaches to dependability, however, one of the Work Packages has been devoted to the process approach, Work Package 3; Life Cycle Processes. The CMMI is the de facto ”industry standard” for process oriented software assessments. Our work on the process oriented work has been based on the CMMI. Since the CMMI is not particularly focusing on dependable software systems our research has been to define and develop an extension to the CMMI for dependable software systems.

4.3.3 Goal-based versus Prescriptive Approaches

The safety of systems and software is generally justified either by appeal to a safety certificate of some safety standard, or alternatively, by arguing safety. In the first case, we assume that the system is acceptably safe if it is certified to the standard. A certificate is a formal assurance that the system is safe. In the latter case, safety is justified by providing convincing arguments of the safety properties of the product, documented in a safety case.

4.3.4 Formal Methods and Criticality

Formal methods are mathematical based techniques for the specification, development and verification of software and hardware systems. Formal methods have been believed to be effective when high dependability is required. Formal methods are therefore getting widespread attention as a possible measure for preventing design faults and resulting service failures, whose criticality has been recognised and often emphasised recently. On the other hand formal methods are not generally known well. Especially, it is difficult to find hints to examine their practical applicability, and generally they are believed to be too costly. Formal methods are relevant to the dependability community as they can be used to support approaches to dependable mechanisms, protocols and so on. The figure 4.2 illustrates how the benefit of using formal methods increases with increasing dependability requirements.

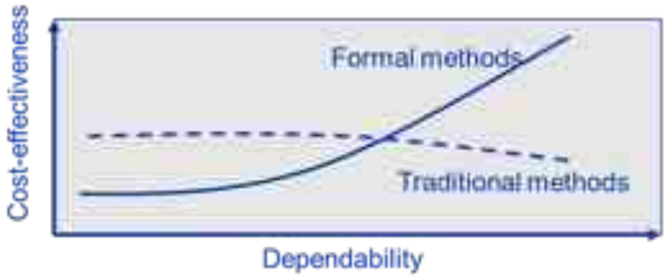


Figure 4.2: Formal specification methods and verification- and validation methods might be cost-effective for highly critical systems

Annex B in part 3 of the standard, [34]. We have further studied goal-based versus prescriptive approaches to assessment of safety certification. This has resulted in documenting a retrospective safety case, [1], based on the GSN-notation, [18]. Based on the experience with the dependability case we realized that by building a library of dependability patterns we could reuse the components from a dependability case, a sort of Pattern Based Approach [36].

4.4 Tasks in detail

4.4.1 Work Package 1: Component Qualification

In industry, production sites are built from new devices composed into new systems. Software delivers an increasing share of the value provided by a device. Traditionally, new production critical parts of a system, or systems, have been subject to a procedure called technology qualification. This procedure establishes whether the device under qualification should be used as is, if risk-mitigating actions should be applied, or if it should not be used at all. However, as the dependence of software increases, identifying qualification steps becomes an increasingly important challenge. How to qualify software intensive systems at the right cost-level, in particular in the long supply chains that exist in the O&G industry? Which inter-supplier information flows are crucial to achieve good integration of devices? In this Work Package we have addressed the challenges from different perspectives. We have developed and tested out a qualification process for software components, and software sub-systems. The idea is that qualified components and subsystems can be integrated to a higher level system. In this Work Package we have also dealt with the problem of assessing dependability critical software systems. We have performed studies on inter-rater reliability of functional safety assessments against the IEC 61508 standard [11], where we analysed the differences in scoring among the auditors on the different elements in the tables in Annex A and

4.4.1.1 A Process Model for Software Component and Sub-System Qualification

The Software-Related Technology Qualification Process (SRTQP) was developed for qualifying systems and components for dependability. The foundation and motivation for SRTQP is based on the fact that systems in the Oil and Gas industry are not built from scratch, but are assembled from a large portion of COTS, specially developed software and software packages, typically protocol stacks. The process has a special focus on integrating goal-based, [36] argumentative case approaches with a qualification approach. While the primary focus of SRTQP is on software-related products and not directly on life cycle processes and business organization, it to some extent touches on these topics as well, promoting a holistic approach to dependability. It is evident that systems and components are interdependent concepts, and many times overlapping concepts, depending on context and tradition. Therefore, we have extended the focus into qualification of architectures, systems and components (ASC). It is possible to also extend this to qualification of life cycle processes and business organizations related to ASC products. The general intention is to cover typical complex software-intensive (CSI) architectures and systems and software-intensive (SI) components used in the Gas and Oil industry. A schetch of the process model is shown in figure 4.3.

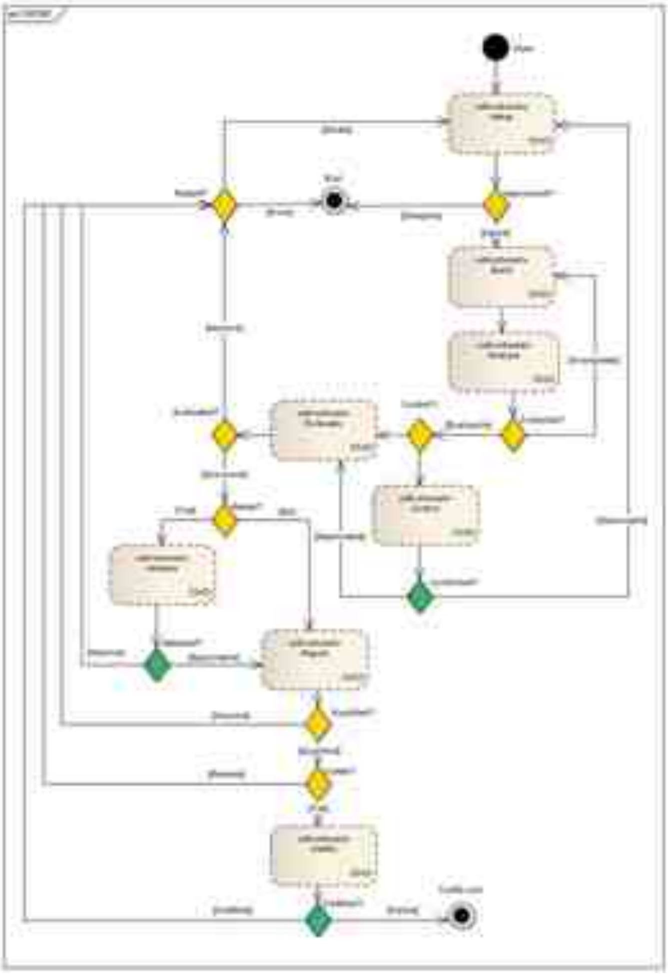


Figure 4.3: Software Related Technology Qualification Process

4.4.2 Goal-based Safety Certification Based on Prescriptive Standards

Existing standards for software safety tend to suffer from a number of limitations and issues. They are prescriptive and process-oriented, i.e. that they prescribe methods and techniques to be used (such as boundary value analysis) rather than being product-oriented, i.e. assessing the product properties. There is little quantitative evidence on how much a specific technique actually contributes to reducing the probability of failures. There is often a lack of consensus between related standards. Assessments may be inconsistent, thereby suggesting that certifications are somewhat arbitrary, because different assessors interpret the content of a prescriptive standard differently, [34]. Recent studies therefore advocate the use of a goal-based, safety case approach for justifying and assessing the safety of software, [6], [7] as an alternative to a standards-based approach. The trend seems to move away from prescriptive standards towards the use of safety cases with evidence tailored to the system developed [26], [37]. One short term, or medium term, problem with the goal-based safety case approach is that it takes time to develop new, more goal-based, product-oriented standards. Meanwhile, customers still need to certify their products to existing standards. We therefore proposed a hybrid approach which combines the benefits of a goal-based safety case approach with the regulatory requirements to comply with existing standards, [38]. We believe that one benefit of the goal-based safety case approach is to make the justification for the outcome of a certification assessment more explicit, transparent, and consistent. Furthermore, we believe that safety assessments would be both more efficient and more consistent across projects if we can enable reuse of elements of a safety case. Our idea and approach are as follows. First, we transform the prescriptive elements in the standard from a table format into a safety case format. Then, the assessment and ratings of the requirements in the standard must be explicitly argued. Second, we extend these skeleton safety case elements with more detail to support more consistent arguments and more consistent assessments. Third, we provide these safety case elements in a pattern format to facilitate their reuse. We have transformed all the requirements in IEC61508 into a collection of safety case patterns using the Goal Structuring Notation (GSN). We exemplify our approach by showing an example safety case pattern derived from IEC61508 and by discussing how the safety case pattern can be used in safety certification in figure 4.4.

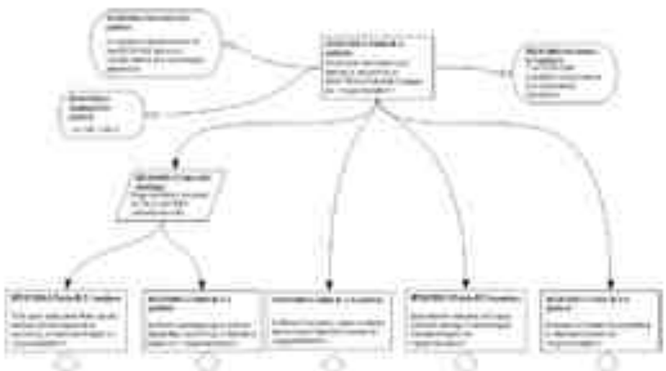


Figure 4.4: A small part of a safety case based on GSN

4.4.2.1 Interrater Reliability of Functional Safety Assessments

We have investigated inter-rater reliability of assessments against the functional safety standard IEC 61508, [11]. Four assessors rated an off-shore oil and gas control system in two separate sessions with approximately two weeks in-between. Each session lasted for about 5-6 hours. In the first session the assessors used the tables in IE61508 and asked questions to the supplier. The questions and answers were given openly within the group and based on the answers and explanations each assessor independently rated the entry. In the second session the assessors had available all relevant system documentation such as requirements, design documents, and test logs. Again the assessors after open discussion in the group rated the same entries once more, independently of each other. The rating scores from the first session were not available during the second session. The rating scores were on a 5-level Likert scale ranging from Weak/Very low to High/Very strong.

Correlation coef.	Interpretation
0.00-0.20	Slight agreement
0.21-0.40	Fair agreement
0.41-0.60	Moderate agreement
0.61-0.80	Substantial agreement
0.81-1.00	Almost perfect agreement

Figure 4.5: Evaluation Framework

Session 1	Session 2	Sessions 1+2
0.29	0.62	0.36

Figure 4.6: Correlation results

We found low to moderate inter-rater reliability. One conclusion for safety certification is that the outcome depends relatively much on the individual assessor. Another conclusion is that the assessors agree more when they have the possibility to investigate all the system documentation.

4.4.2.2 Sequence Based Specification

Formal methods and tools have been considered as promising means to support dependability in complex software intensive systems during development. It is generally understood that failures in many software based systems are caused by wrong or incomplete specifications, leading to that software engineers misunderstand the specification. In IOHN Activity-4 we have investigated how a particular formal method can be applied to produce a more dependant software. The purpose of this work was to evaluate the applicability, scalability, and effectiveness of applying the so-called sequence-based specification (SBS) approach to verify and validate software product. The SBS approach has been invented to verify logical consistency and completeness of requirements, to improve the traceability between functional requirements and test cases, and to facilitate statistical-based testing and to assess software reliability. The SBS approach has been experimented in the German automotive industry and in railway signal systems. However, few case studies have previously been performed to examine this approach in the Maritime and Energy sector.

4.4.2.2.1 Using SBS to Find Inconsistent and Missing Software Requirements

Sequence-based specification (SBS) is to implement rigorous, and practical software specification of the functional behaviour of a system. The essential idea of SBS is to treat the software system as a black box. When implementing SBS, the software behaviour is specified in terms of the appropriate externally observable response to each sequence of external inputs, [25], [5]. Through systematic enumeration of all sequences of external inputs and their corresponding observable responses, SBS produces an arguably complete, consistent, and traceably correct software specification. SBS has been examined in a few industrial applications. For example:

- SBS has been combined with statistical-based testing to estimate the reliability of mirror control unit of car [5].
- SBS has been combined with risk analysis to generate critical test cases to verify railway control systems [39].
- SBS has been used to model the requirement of a hypothetical Thermal Margin Calculator of a nuclear plant [25].
- SBS has been extended to include time, continuous functions, non-determinism, and internal events for embedded real-time systems.

However, SBS has not previously been experimented to verify and validate software used in the

Maritime and Energy sector. This report presents our case study on using SBS approach to Verify and Validate (V&V) micro-controller software according to the software requirements. Results of the study show that SBS is applicable and effective to V&V the examined software. However, unlike systems that have been examined using SBS, the software we investigated is highly configurable. So, there are limitations in tool-set of SBS that need to be addressed in order to use SBS more effectively. We first examined the functional requirements of the software using the SBS approach. In this process, some missing and inconsistent requirements were discovered. Then, we checked whether each requirement had corresponding acceptance test cases. In this process, the traceability between software requirements and acceptance test cases were strengthened, and missing test cases were found. Results of the study reveal that the SBS approach is applicable and effective to (V&V) the completeness and consistency of the functional requirements of the examined control system. The cost of using the SBS approach is durable, in case sufficient domain knowledge is acquired. Although there is a tool to support using the SBS approach, the tool needs to be improved to better support the examined system, which is a highly configurable system. The SBS approach is suitable for modelling event-driven reactive systems. Reactive systems are systems whose role is to maintain an ongoing interaction with their environment rather than produce some final value upon termination. Typical examples of reactive systems are Air traffic control system, Programs controlling mechanical devices such as a train, a plane, or ongoing processes such as a nuclear reactor. The behaviour of such a system can usually be modelled into a state-machine. That is, the system will behave differently in different states. Based on the documents provided by the industrial partner, we selected three subsets of the requirements (we call them sub-systems hereby). The chosen subsets of requirements describe the system's react to certain inputs. The other requirements are not relevant for SBS, because they describe how fast the data should be transferred and the data format. We used free software tool ProtoSeq to construct the SBS model of the three sub-systems respectively. ProtoSeq is a prototype of the sequence enumeration tool developed by the Software Quality Research Laboratory at University of Tennessee. The activities that were included in our study are presented in Figure 4.7 and Figure 4.8. Figure 4.7 shows the procedure of constructing the SBS model which is adapted to the usage of ProtoSeq. This same procedure was conducted for three sub-systems respectively. The input is the requirements of the sub-system and the output is an SBS model. The generated SBS model is the input in Figure 4.8 which illustrates the procedure of generating and comparing test cases.

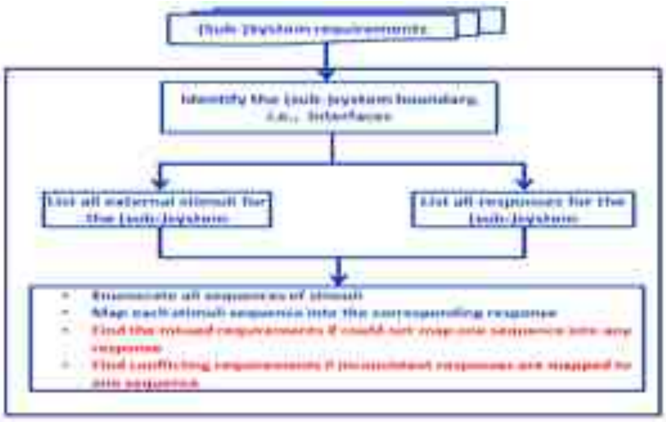


Figure 4.7: SBS activities to verify completeness and consistenmcy of requirements

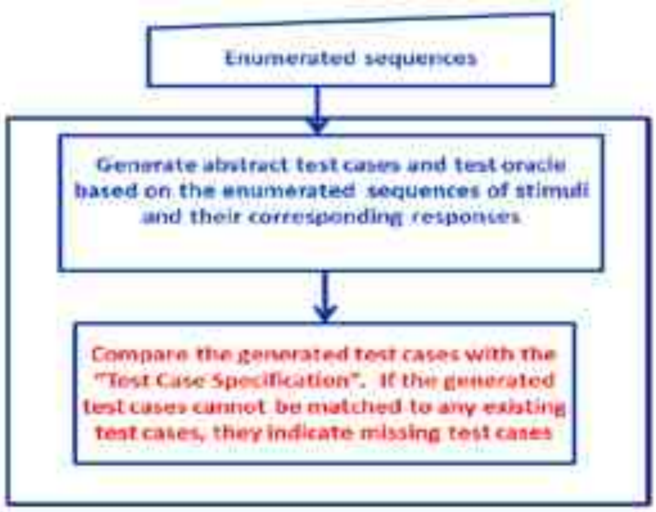


Figure 4.8: SBS activities performed to verify completeness of test cases

4.4.2.2.2 Using SBS to Reverse Engineer a Subsea Control System

The background of this study is that a subsea oil production system integrates devices from many vendors. The communication of all devices is expected to follow a communication protocol, which has been agreed by major oil companies and vendors in 2011 as the standard protocol for level 2 subsea device communication. A company has already developed a subsea oil production system, which has been deployed and is in use. Although the system is already in use, it has a complicated structure and is therefore hard to maintain. Another issue is that the system is probably not compliant with the newest version of the protocol. So the company wants to re-implement the system from scratch to ease the maintenance of the system and to be compliant with the protocol. The challenge is that the company has limited documentation of the functional requirements of the system. Due to the evolution of the code, the requirements are probably outdated. To re-implement the system, the company needs to create solid software requirements and test cases for developers and testers. The idea is to reverse engineer the existing running system, to document the behaviour of the system, and to document the interactions between this system and the systems around it. In addition, the software requirements must consider the protocol to ensure that the newly developed system will be compliant with it. The research question is then "How to explicitly document the functionality and interaction of the integrated existing system, within the context the system and systems around it are expected to follow the same communication protocol?" To address the research question, we have examined to use the sequence based specification (SBS) approach [25] to model the interactions between the system and systems around it. The study includes five activities as follows:

- Identify critical functionalities of the system to be modelled.
- Using SBS approach to model the sequence of input/output message of the examined functionalities based on the communication protocol.
- Exclude the irrelevant parts of the model (e.g. optional parts of the protocol that are not implemented in the existing code) by examining the code.
- Present the model to company engineers and let them confirm the model.
- Convert the model to requirement specification and abstract test cases.

We have tried the study in two functionalities of the system. Results of the study show that:

- SBS can be used to systematically generate software functional specification to present how the system should interact with other systems according to the protocol.
- Comparing the specification generated from SBS with the existing code helped us identify which parts of the existing system are compliant with the protocol and in what manner. More importantly, it helped us identify which parts of the system are not compliant with the protocol and therefore need to be completely re-implemented.

4.4.2.2.3 A Prototype Tool for the SBS Method

Because of the limitations of the ProtoSeq we have also developed an executable prototype to generate state machine and generate abstract test cases. The output from the tool can be exported and used as input to the Jumbl tool used to perform statistical testing [39]. The prototype tool can handle highly configurable systems, which is a limitation in the ProtoSeq tool.

4.4.2.3 Failure Mode and Effects Analysis of Software-Based Systems (SWFMEA)

Failure mode and effects analysis is one of the well-known analysis methods having an established position in the traditional reliability analysis. The purpose of FMEA is to identify possible failure modes of the system components, evaluate their impact on system behaviour and propose countermeasures to supress these effects. FMEA is well understood at the systems and hardware levels, where the potential failure modes are usually known and the task is to analyse the their effects on system behaviour, e.g.see [8]. More and more system functions are realised at software level, which has urged to investigate how to perform FMEA on software based systems, in particular software intensive systems. In our technical report (listed in section 4.7) describes a recommended practice for how to analyse systems incorporating software into the FMEA.

4.4.2.4 Cloned Buggy Code Detector (CBCD)

Developers often copy, or clone code in order to reuse or modify functionality. When they do so, they also clone bugs in the original code. Or, different developers may independently make the same mistake. As one example of a bug, multiple products in a product line may use a component in a similar wrong way. We have made two contributions. First, we did an empirical study of cloned buggy code. In a large industrial product line, about 4 percent of the bugs are duplicated across more than one product or file. In three open source projects (the Linux kernel, the Git version control system, and the PostgreSQL database) we found 282, 33, and 33 duplicated bugs, respectively. Second, we developed a tool, Cloned Buggy Code Detector, CBCD, that searches for code that is semantically identical to given buggy code. CBCD tests graph isomorphism over the Program Dependency Graph (PDG) representation and uses four optimizations. We evaluated CBCD by searching for known clones of buggy code segments in the three projects and compared the results with text-based, token-based, and AST-based code clone detectors, namely Simian, CCFinder, Deckard, and CloneDR. The evaluation shows that CBCD is fast when searching for possible clones of the buggy code in a large system, and it is more precise for this purpose than other code clone detectors, [19].

4.4.3 Work Package 2: Topology, Architecture and Integration

Traditional network architecture/system architecture research has provided good advice on how to architect systems and networks for performance, security, and logical maintainability. However, in the Oil and Gas industry, there is rarely clearly designated ICT system property owners, who can govern decisions all the way into products attached to some network. Thus there are examples of putting critical software executing close to potentially dangerous areas, and examples of how office-style software has contributed to shutting down large scale production equipment due to interconnected networks. The challenge is formulated as ’how do we cost efficiently qualify topologies for the purpose of achieving operational dependability’ in the Oil and Gas domain. The overall system architecture covering the whole chain of components from sensors monitoring the processes through the networks and process control systems to the user interface has to be under control. Definition and control of interfaces between these systems is essential.

4.4.3.1 Resilience of Open Architectures for Integrated Operations

Systems in domains like Integrated Operations (IO) in the oil and gas offshore sector are becoming ever more complex. Typical systems are actually ‘systems of systems’, with many (thousands) of components and subsystems on different platforms and locations, using multiple standards and interacting in many possible and unforeseen ways. As a result, it becomes harder to oversee a priori all possible events and scenarios that may affect the dependability of a system. Unforeseen malfunctions or unexpected interactions between components or subsystems are likely to have a negative impact on the reliability, availability and other aspects of system behaviour. With growing complexity, traditional approaches and methods, which assume more static and smaller scale systems and system environments, are no longer sufficient to provide sufficient business continuity, safety and security. To have the necessary dependability of systems with high complexity, a new approach that is not dependent on a priori or design time knowledge of possible events that may impair dependability is needed. Instead, a new approach should be striving for ‘resilience’: the characteristic of a system to adapt dynamically to changing internal or external circumstances, both foreseen and unforeseen. Research has been carried out to identify existing and new techniques to improve the resilience of systems with a so-called open architecture, [32]. This implies that systems are not considered as static entities, but rather as assemblies of building blocks that may vary over time. In practice, in general but, also increasingly in the Integrated Operations domain, this type of systems is designed and implemented in a Service Oriented Architecture. The resulting overview of resilience techniques is organised in categories for infrastructure, software and information architecture. Finally, a foundation has been laid to come to a recommended approach in using resilience techniques, given the level of dependability requirements (expressed in a confidence level per system function), the maturity and complexity of the resilience technique itself and the complexity aspects of the system at hand. As the name ‘Integrated Operations’ implies, one of the key characteristics of this IO business domain is that processes and their supporting IT systems are increasingly integrated. This is not only true for integration of single processes that may span over different locations and times and that may involve a diversity of existing (software intensive) systems. It is also true for integration across different processes, sharing information and functionality to achieve more optimised coordination at a higher level than the level of single processes or disciplines. Examples of Integrated Operations can be found in a close cooperation between offshore operations and onshore control centres, or even complete control of offshore operations by onshore control centres. Other typical examples of integration can be found in the close interaction between processes for drilling and reservoir management and between operations and maintenance. Following the line of increasing integration, the ultimate 'Integrated Operations' scenario requires a fully Integrated chain of information and functionality covering the whole range of operational data of sensors and actuators, local and remote control systems, local and centralized control rooms, decision support, business applications and global collaboration facilities. Moreover, the envisioned Integrated Operations, involves a myriad of sub-systems and components that may be supplied by a variety of vendors and/or implemented and configured by a variety of system developers and system integrators. Interoperability, portability and maintainability (both corrective and for improvements) are therefore key requirements. In this context it should be noted that levels of integration or 'maturity' of Integrated Operations may vary significantly in practice. A distinction is made between generation 1 and generation 2 Integrated Operations, with "generation 1" having limited system and process integration, more focused on improved collaboration and cooperation. It is only at "generation 2" that the full benefits of Integrated Operations are reached, leveraging full or at least significant integration levels. It is at this level that we focus on in this work: the level at which systems become 'systems of systems', integrating and coupling existing and new systems, both onshore and offshore, automating the whole chain of offshore sensor data and data integration to onshore data analysis and decision support, coupling with (remote) systems of the operator company, suppliers and other parties involved, and back to offshore control systems.

Figure 4.9: Integrated Operations

Figure 4.10: Integrated Operations generation 2

Considering the fact that Integrated Operations are in the core of both tactical (i.e. production planning) and operational (i.e. production) business processes of the offshore industry, it is obvious that the dependability system requirements are very high to support sufficient business continuity, safety, security and environmental protection. Having said that, it is also possible to differentiate dependability requirements for different functions, which are to be mapped to systems, sub-systems or main components, using for instance the confidence level categorisation that is described in DNV's recommended practice for Integrated software dependent systems, [10]. Building on these confidence levels, it is also a requirement that if systems or part of systems can no longer be at the required dependability level, for whatever reason, functions of higher confidence levels should be given priority over functions of lower confidence level, with the effect that if systems start to fail, for instance due to limited resources, the first functions that should become unavailable are those with the lowest confidence level, etc. This requirement is also known as the ability for 'graceful degradation'.

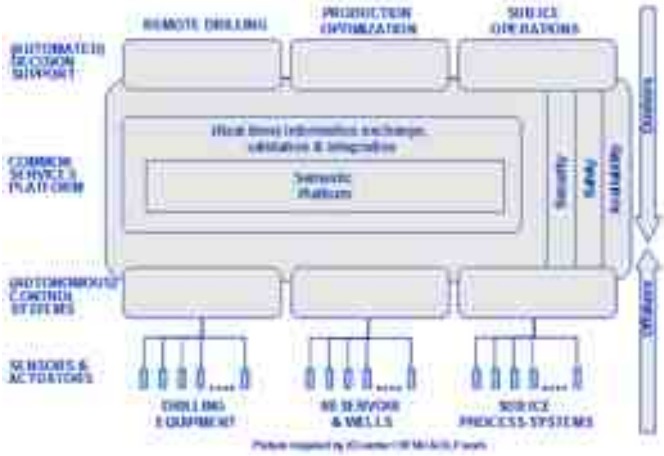


Figure 4.11: High level (service oriented) architecture of the Integrated Operations in The High North' illustrating the use of a common services platform and the need for an enterprise service bus

4.4.3.2 Resilience in Complex Socio-Technical Systems to Improve Safety and Continuity in Integrated Operations

One of the PhD students at NTNU working in collaboration with the IO Centre at NTNU has studied the need for resilience in complex sosiotechnical systems to improve safety and continuity in Integrated Operations, [35]. This research has been conducted in close collaboration with IOHN-Activity 4. Therefore some of the results are reported below. More challenging oil and gas fields must be explored in the future due to increasing need for oil and gas. Integration of technologies and increased utilization of expert knowledge are needed to handle these challenges, but the increased complexity and the operational challenges are pushing the boundaries of safety. Integration of technologies, distribution of information and collaboration in teams creates new vulnerabilities, new complexities and new uncertainties in the systems used to control operations. Catastrophic accidents may happen if operations are not controlled, as seen in, e.g., the Deepwater Horizon accident. To improve safety and continuity we have investigated resilience as "the intrinsic ability of a system to adjust its functioning prior to or following changes and disturbances, so that it can sustain operations even after a major mishap or in the presence of continuous stress", [2]. The work suggests that risk assessments should expand the scope of experience by learning both from accidents and successful recoveries and to explore how resilience can be introduced in order to improve safety and continuity and avoid catastrophic accidents. Accident analyses have seldom explored successful avoidance of catastrophes. Positive actions on safety are usually not recorded, and Risk assessments have seldom systematically explored resilience. Resilience is suggested as an important strategy to improve safety, security and continuity in this environment of complexity and uncertainty. An investigation of the status of safety and security of control-based systems was performed based on surveys on all 40 offshore oil and gas installation at the Norwegian continental shelf, in addition to audits of three installations based on interviews and discussions. The systems can be described as complex, i.e., consisting of many interrelated parts, and coupled, i.e., no buffers or slack between two items and what happens in one item directly affects what happens in the other. Important findings in the survey were poor resilience and lack of awareness of risks. There were missing risk assessments. Only 5 of 46 installations in this study could document a risk analysis of the integration of process control systems with information and communication technology (ICT. No certification of system integration and thus several vulnerabilities may be present. Poor awareness of the potential problems related to integration between process control systems and ICT. A review of accident reports involving these systems has also been done, showing that aspects of the control systems often contributed to the severity of the accidents. The role of control systems in accidents has not been systematically gathered earlier. There is a need to establish common taxonomies and methods to structure these vulnerabilities, [17]. Catastrophic accidents are far between, and to improve knowledge of the steps to an accident, exploration of recoveries can give added insights. But successful recoveries are seldom analysed and documented. It is suggested that past accidents (weaknesses) and successful recoveries (strengths) are discussed in the accident analysis. Improved methods must be used in risk assessment to explore opportunities in resilience and to mitigate risks (threats). A literature survey of intervention studies based on action research and organizational learning involving key stakeholders indicates improved safety, security, awareness and resilience as a result of structured participation and involvement, [16]. Thus action research and involvement of key stakeholders, such as authorities and operators during risk management are suggested. To be able to use resilience in an operational setting, seven resilient principles were proposed based on literature reviews, including redundancy, the ability to perform controlled degradation and the ability to "rebound or recover", flexibility, ability to manage margins close to performance boundaries, establishment and exploration of common mental models, reduction of complexity, and reduction of couplings. We have also seen how the increased automation onboard ships and offshore structures lead to the need to rethink emergency handling, [33].

4.4.3.3 Software Fault Tolerant Techniques and Architectures

Computer-based systems have increased dramatically in scope, complexity, and pervasiveness, particularly in the last decade. Safe and reliable software operation is a significant requirement for many types of systems, also in the oil and gas industry which is the target for the IOHN-Activity 4. Ideally, the processes by which software is conceptualized, created, analysed, and tested would have advanced to the point where software could be developed without errors. The current state-of-practice is such that fewer errors are introduced, but unfortunately not all errors are prevented. Increasing the dependability of software presents some unique challenges compared to traditional hardware systems. Software does not physically wear out, burn out, or otherwise physically deteriorate with time. Software has only logical faults, which are difficult to visualize, classify, detect, and correct. To protect against these faults, one cannot simply add redundancy, as is typically done for hardware faults. So to provide protection against these faults one turns to software fault tolerance. There are two ways to deal with faults, fault prevention and fault tolerance. Fault intolerance aims to prevent the existence of faults. Fault tolerance is intended to handle faults when they occur in an executing system. In engineering, fault-tolerance means that a system has the ability to continue operation, possibly at a reduced level (also known as graceful degradation), rather than failing completely, when some part of the system fails. The concepts of fault, error, and failure are directly linked to a causal relationship: faults lead to errors, which ultimately lead to failures. A fault is a physical defect or flaw within a hardware or software component. This is essentially the definition found in a typical dictionary. An error is the manifestation of a fault: a deviation from accuracy or correctness in state. A failure is an externally observable event representing a deviation from the authoritative service specification. There are two means of fault prevention, avoiding their introduction during production, and removing them before deployment. In both cases faults are dealt with prior to execution. 'Fault avoidance' is a design activity that attempts to prevent faults from being introduced into the deployed system. 'Fault removal' is a design (i.e., implementation) activity focused upon testing. Testing is inherently limited by the inability to test under completely realistic conditions, by the potential for specification errors, and, of course, by the fact that testing can only show the presence of errors, not their absence. In contrast to fault avoidance, fault tolerance schemes consider faults inevitable and deal with them after deployment. As a run-time activity, therefore, fault tolerance may be defined as the ability of a system to continue performing in the presence of faults. If faulty software or hardware components are to be dealt with during execution, some additional resource is necessary. Therefore, fault tolerance is based on one of several forms of redundancy. Figure 4.12 show an overview of different software fault tolerance techniques.

Figure 4.12: Salient features of the approaches to software fault tolerance by design diversity

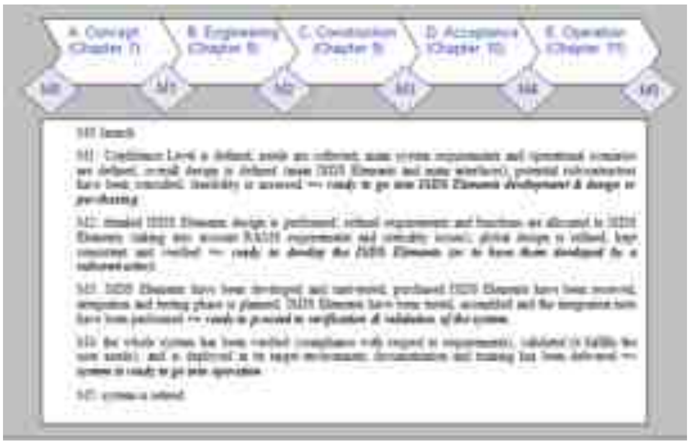


Figure 4.13: DDPRM and RP-D203 Development and Operation phases and milestones

Figure 4.14 shows the structure for assessing the criticality of the system (in DDPRM/RP notation: confidence levels). Confidence Levels define the required level of trust that a control function or system (or controlled function or system) will perform as expected. DDPRM/RP defines Confidence Level 1 through 3 where the higher Confidence Level will provide increasingly more reliable systems or control system functions. That means specifying Confidence Level 3 will provide the highest degree of assurance that the control system will be reliable.

4.4.4 Work Package 3: Life Cycle Processes

4.4.4.1 Dependable Development Process Reference Model

Most of the research and work in this task has been to develop the Dependable Development Process Reference Model (DDPRM) aiming at assessing the maturity level of software organisations' development and integration of critical software intensive systems. The basis has been the CMMI model from Carnegie-Mellon University which is the de facto standard for assessing the maturity of software development organisations. The work was performed by the DNV France team with extensive knowledge of the CMMI and with many years experience from developing and assessing software intensive systems in the automotive and aerospace industry. This work has been the basis for the further development in DNV business units to develop a Recommended Practice (RP) [10] and a DNV Offshore Standard (OS), [9]. The DDPRM (and the RP and OS) divides the project into five distinct phases. Each of these phases identifies core activities that shall be undertaken in any case and activities that are selected based on the Confidence Level required by the function. See figure 4.13.

Figure 4.14: Assessing Confidence levels

The basic principle for the selection of the Confidence Level is that the higher the potential consequence (safety, environmental or business) of a failure of the control system or function, the higher the specified Confidence Level should be. Confidence Levels are assigned to the overall system and then to each function within the control system. This then derives the Confidence Level to the Elements (components or networks within the system). See figure 4.15.

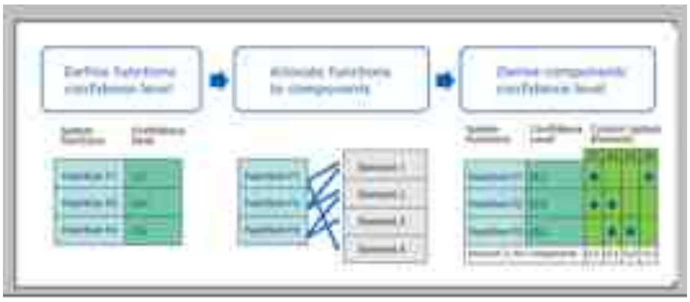


Figure 4.15: How to derive element (component) confidence level

This DDPRM/RP addresses several types of actors of the system development and maintenance. Each of them has specific activities to perform. Five responsibilities have been defined:

- Owner: The Owner is the one who decides to develop the system, and provides funding.

- System Integrator: The System Integrator is responsible for managing the development of the system, in charge of global design, Elements supplier management, and integration and verification of the whole system.

- Operator: The operator is the one who will finally use the system when it is under operations. This covers also the responsibility of maintaining it.

- Supplier: The Supplier represents any subcontractor which is in charge of developing an ISDS Element of the system, under the coordination of the system Integrator.

- Independent Verifier: The Independent Verifier is an organization that is mandated to independently verify that the system is developed according to the expected rules, standards, processes and quality. This responsibility can be undertaken by either the owner or a third party organization. For systems with high Confidence Level (CL3), this responsibility shall be undertaken by a third party company.

The DDPRM/RP for each confidence level and for each role and each life cycle phase defines a set of activities that must be performed. An example is illustrated in figure 4.16.

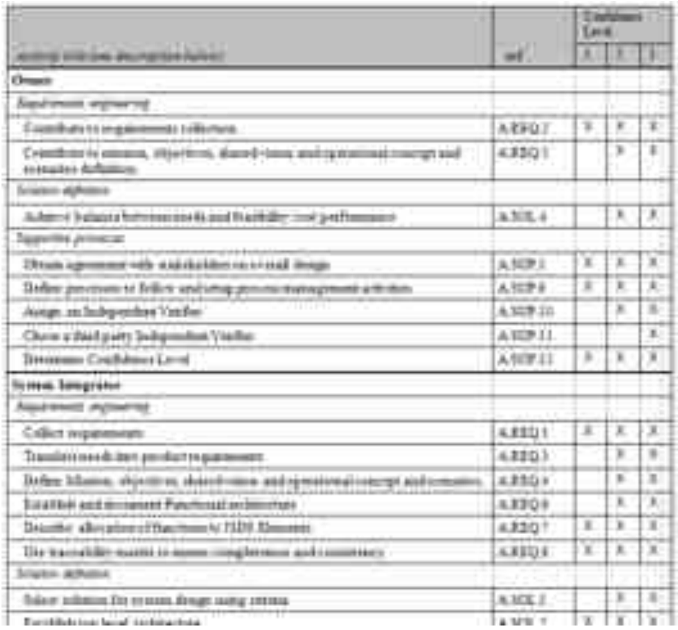


Figure 4.16: The figure illustrates which activities have to be performed in a given phase by a specigic role for a specific confidence level.

The main body of the DDPRM/RP gives a detailed description about each of the activities, e.g. A.REQ.5 is mandatory for the owner (in the Requirements phase) as illustrated in figure 4.16.

4.4.4.2 Security Principles

Security principles, proposed by several literatures within the software security domain, provide a mechanism for expressing security strategies and communicating these strategies to a broad audience. In this work, a security principle represents an established best practice towards security at some level of abstraction and viewpoint. A security principle can target the development of software in terms of the code and technology or the engineering process per se. A security principle can also target the user behaviour or the integration of solutions. Security principles are often expressed without referring to particular technologies, and typically summarized by short phrases that can be explained and understood by everyone involved in security in the organization. Security principles may appear as superfluous or too simple to be useful. However, a security principle is intended to be a driver for achieving security and not to provide an explicit solution. The interpretation of a security principle and the degree of its implementation may vary between organizations. The target and context for which a principle is used may be very different. To illustrate with respect to an organizational interpretation of what needs to be addressed, the defence-in-depth principle (i.e. layered security) may relate to three elements, people, technology, and operations. Achieving defence-in-depth from people's perspective may involve organization, training, physical security, personnel security, awareness, and system security administration. The technical report and prototype database introduce security principles and proposes ideas on how these principles can enhance computer security for security critical infrastructures. How to interpret and implement a particular principle varies with domain and context, thus, categorising them is a difficult task. We have done a structuring based on OLF104 and some other categories. The collected security principles are categorized as follows:

- OLF104 baseline requirements: Which principles support the OLF104 requirements
- System life cycle: Importance with respect to system life cycles
- Potential conflicting principles: Indicate need for trade-off between principles
- SCADA: Principles that may have a different interpretation or implication when used in the general SCADA domain

4.4.4.3 Software Reliability Modeling, Measurement and Prediction

A proliferation of software reliability models has emerged as people try to understand the characteristics of how and why software fails, and try to quantify software reliability. Over 200 models have been developed since the early 1970s, but how to quantify software reliability still remains largely unsolved. As many models as there are and many more are emerging, none of the models can capture a satisfying amount of the complexity of software. Constraints and assumptions have to be made for the quantifying process. Therefore, there is no single model that can be used in all situations. No model is complete or even representative. One model may work well for a set of certain software, but may be completely off track for other kinds of problems. Most software models contain the following parts: assumptions, factors, and a mathematical function that relates the reliability with the factors. The mathematical function is usually higher order exponential or logarithmic. Software reliability models can as mentioned in the above chapter both assess and predict reliability. Assessment deals with measuring past and current reliabilities while predicting provides forecasts of future reliability. This means forecasting in a statistical sense, i.e. an estimate of the probabilities of future failures. In order to give reliable estimates we need good data (accuracy, pertinence) as well as good and relevant software reliability models. The result of a prediction model will be adversely affected by factors such as change in failure data (historical failure data that are input to the reliability model), significant change in the code under the test, and significant changes in the computing environment. We make two assumptions about the software whose reliability we want to measure:

- The software is operating in a real or simulated user environment.
- When software failures occur, attempts are made to find and fix the fault(s) that caused the failures.

In the long term we expect to see the reliability improve. However, there may be short-term decreases caused by ineffective fixes or by introducing new faults. We want to monitor the time between failures in order to get a measure of the reliability of the software. This makes it possible to answer questions such as:

- How reliable is the software now?
- Is it sufficient reliable so we can stop testing and deliver it?
- How reliable will it be if we spend a given amount of further testing effort?
- How long will we have to wait until the reliability target is achieved?

In this activity we have surveyed several software reliability models. Figure 4.17 shows a classification of the different models.

Figure 4.17: Software Reliability Model Classification scheme

The following criteria should be used for evaluation of software reliability models in support for a given project:

- Future predictive accuracy: Accuracy of the model in making predictions beyond the time span of collected data (comparison of future predictions with future observed data).
- Historical predicted validity (comparison of retrospective predictions against past observed data). To compare a set of models against a set of failure data, the fitted models are examined to determine which model is in best agreement with the observed data. This means that the model with the least difference between the retrospective prediction and the actual data is considered the best fit. Best fit can be measured by the criterion of minimum relative error.
- Generality: Ability of a model to make accurate predictions in a variety of operational settings (real time, web applications).
- Insensitivity to noise: The ability of the model to produce accurate results in spite of errors in input data and parameter

Experience shows that we can try all available prediction techniques, but we can demonstrate that none of them produces trustworthy reliability predictions. It is possible for all of them to be very inaccurate. In the technical report on the software reliability modelling and prediction we show how a recalibration technique can help us in many situations by removing noise and biases. NASA presented an interesting approach to integrate software-safety criteria, risk analysis, reliability prediction, and stopping rules for testing the software. This involves using the Schneidewind Software Reliability Model. The approach has been reported successfully for software on board the Space Shuttle. Unlike the traditional approach for obtaining safe software prescribed by standards such as IEC61508, [11], by prescribing specific development methods and architectural solutions this approach is based on predicting time to next failure and through testing predict the system's ability to survive the mission without experiencing a serious failure. Remaining failures (RF), maximum failures, total test time (TTT) required to attain a certain fraction of RF, and time to next failure (TTNF) are shown to be useful reliability measures and predictions for providing assurance that the software has achieved safety goals, and rationalizing how long to test a software component. Two categories of software reliability measurements and predictions are used together to assist in assuring the safety of the software in safety-critical systems:

- Measurements and predictions that are associated with residual software faults and failures;
- Measurements and predictions that are associated with the ability to survive a mission without experiencing a serious failure.

Having predictions of the extent that the software is not fault free (RF) and its ability to survive a mission, TTNF are meaningful for assessing the risk of deploying safety-critical software.

4.4.4.4 Software Quality Standards

The number of software standards is steadily increasing. Figure 4.18 shows some of the most recently available standards and recommendations for dependable software development, assessment and maintenance.

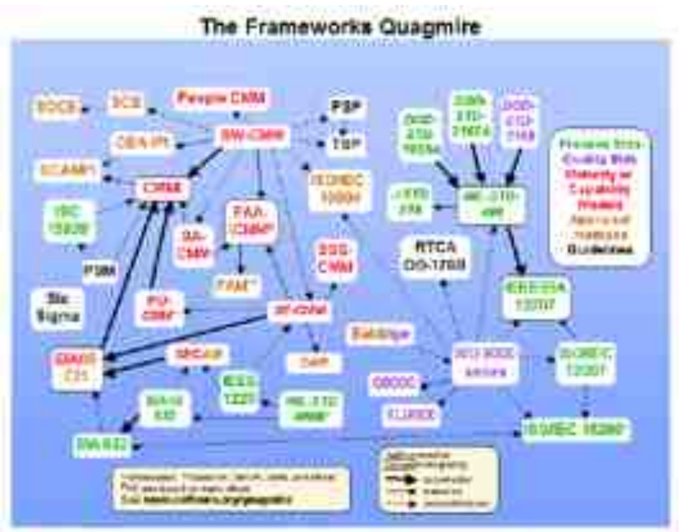


Figure 4.18: Framework Quagmire shows some of the numerous standards aiming at software quality

We have worked with reviewing and collecting information about a large amount of standards related to software dependability. We have also, based on a semantic analysis of the standards' text calculating the relationship strength between the standards. A prototype tool has been developed to search among these standards and also visualize the relationship strength in a graph. The prototype was later extended to search for the GSN patterns for building dependability cases as discussed in section 4.3.3.

4.4.4.5 Ontology for SCADA Security

A master student at NTNU has in her Autumn Project (which is introductory research for the Master thesis work) performed a review of Software Security, Supervisory Control and Data Acquisition (SCADA) security taxonomy and Ontology. The research as part of the master thesis is continuing until end of June 2012 and will result in a working prototype for an ontology search tool for SCADA security. It represents a literature survey of the most relevant subjects contributing in the state of the art in software security for SCADA systems, [21]. The aim is to give a brief survey of the overview of software security, to define the relevant basic concepts of software security and types of taxonomies in this area, the previous works and to compare those taxonomies from different perspective. In general, the taxonomy is a method that the SCADA systems can use it to classify Internet security attacks and incidents. Using taxonomy for security of SCADA networks provides two main facilities. First it supports networks against probable cyber-threats from attackers, and second it facilitates the users and developers to better understand the network domain, leading to define the vulnerability taxonomies of the SCADA network. Previous and existing security ontologies for SCADA systems and security are also presented in general. The specific focus is dedicated to the SCADA networks. The connectivity of SCADA networks with outside network is continuing to grow, leading to an increase in risk of cyber-attacks and crucial need to improve the security of these networks. The study reports the latest attacks performed on SCADA networks. It also has a glimpse overview to protocols, network architecture and security taxonomies in the SCADA systems.

4.4.5 Work Package 4: Dissemination

The dissemination in the IOHN-Activity-4 are devoted to Workshops, internal and external seminars and presentations. In addition we have several papers published in international conference proceedings and in international journals. A more detailed overview is given in section 4.5.3.

4.4.6 Work Package 5: PhD and Post. doc. Research

The project has hosted two PhD students and one post. doc. researcher, all three hosted at NTNU. The PhD students each have a four year scholarship, while the post. doc. researcher had a two-year scholarship. The post. doc. researcher finished his research in 2011, while the two PhD students will finalize their PhD thesis during 2013. The two PhD student have research topic related to information and software security, while the post. doc. researcher has been researching within ICT use for safe and reliable petroleum production. A summary of the research activities for each of them is presented below.

4.4.6.1 PhD 1: Secure Information Sharing in Integrated Operations

4.4.6.1.1 Background

Current access control systems for the most part rely on a central reference monitor to enforce restrictions and are therefore incapable of protecting information outside the system. Hence, whenever information is shared or distributed from one system to another, the originally imposed restrictions are lost. Whether to re-introduce them in the receiving system is determined solely by the receiver. For example when downloading a restricted document from a web server to a laptop, the user (receiver) may forward the document to another user that does not have the required privileges to access it directly. Hence, the originating system is ultimately forced to trust the receiver to take appropriate measures to handle the information. The predominant way of handling this is through contractual means (e.g. Non-disclosure agreements an security policies) that specifies how and for what the shared information can be used. However, these do not cater for the dynamic nature of the content they are supposed to protect and violations are very difficult to detect and punish.Additionally the magnitude and potential sensitivity of the shared information in Integrated Operations substantially increases the risk of misuse compared to the current situation. It is therefore unlikely that the full potential of the IO concept can be realised without technical support for information usage control in addition to whatever procedural or contractual measures taken.

4.4.6.1.2 Research Goals

The main objective of this PhD is: ”To improve the control of shared information within and across organisations in Integrated Operations (IO) through increased usage of enforcement technology.” The project has identified and described existing solutions to distributed usage control and demonstrated the lack of empirical data on their usefulness. Further, the project have conducted a case study and described a new model for understanding users’ attitude towards adoption of usage control technology. Finally, based on previous findings this project will identify the important features of usage control technology in order for it to be successfully adopted by the oil and gas industry, and more specifically the Integrated Operations.

4.4.6.1.3 Expected Contributions

This PhD research will provide new insights and empirical data on enforcement of usage control policies. For researchers:

- A better understanding of how enforcement technologies and their relative advantage are perceived by the business organisations.
- New enforcement models specifically targeted for collaborative environments (Integrated Operations).
- Empirical evaluations of enforcement technology.

For businesses both taking part in Integrated Operations and others:

- A better understanding of how users feel about enforcement technology.
- Knowledge to better evaluate the risk of using enforcement technology.
- Empirical data on the usefulness of the technology.

4.4.6.2 PhD 2: Federated Identity Management (FIdM)

The PhD project titled “Federated Identity Management: An Empirical investigation of benefits, challenges and security risks in an inter-organisational context” aims to collect empirical evidence related to the use of federated identity management within the oil- and gas production environment. During the project companies involved in close inter-organisational collaboration in Integrated Operations are analysed for the purpose of exploring the perceived benefits, security risks and other related challenges to FIdM from the viewpoint of each type of partner organisation in the collaboration. A digital identity is an entity’s digital representation of personal attributes. These attributes can be personally identifiable information, such as the social security number, employee number or hair colour. A digital identity can be used in the digital space to identify, e.g., a user of an online service. Digital identity management (IdM) is the management process that includes creating, updating, using and destroying digital identities. In the business world digital identities are often created during the employment process to give the new employee access to company internal digital resources, such as intranets, e-mail and file sharing repositories. Digital identities are tightly connected to access management systems where users need to prove that they are who they claim to be (authenticate) before they are given service access. Traditionally, each computer/software system has implemented separate Identity and access management systems. In the business scenario this has often meant that users would need to type in their username/password for each service they want to access (and the preferable situation is to keep at least different passwords for each system). Recently, this has been improved, where most companies now offer so called single-sign-on to their internal resources; i.e. type in your username/password once, obtain an identity token, and use this for later and seamless resource access. Figures 4.19 and 4.20 provide examples of current and possible future identity and access management principles. The first figure shows a traditional scenario where an employee of a company has to obtain a digital identity from his or her local company (Company A) to access company internal resources. If the same employee needs access to resources provided by a collaborator (Company B), that company have to define a new user profile before access can be given. Separate login processes, with different access credentials, are required in this scenario. The next figure shows a future scenario where Company A and B have federated their identity and access management solutions, and where inter-organisational single-sign-on is the result.

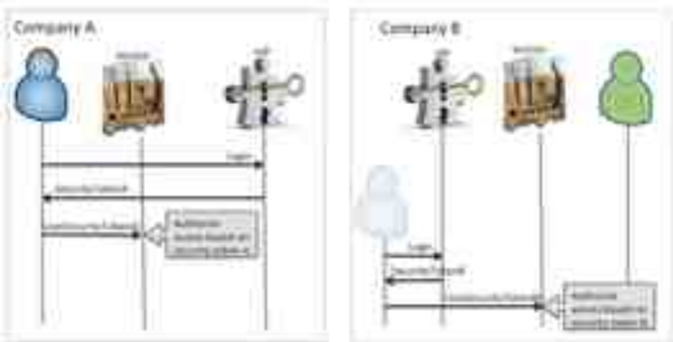


Figure 4.19: Traditional scenario

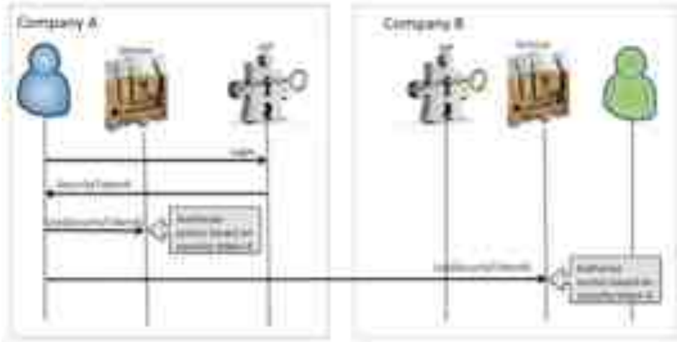


Figure 4.20: Future scenario

FIdM aims to take the identity and access management (IAM) field a step further. In identity federations, companies can link their IdM systems so that users can experience single-sign-on and seamless resource sharing even in inter-organisational collaborations. There are examples of FIdM systems in operation today:

- FEIDE in the educational sector in Norway allows seamless (single-sign-on) access to resources such as the Cristin publication database, intranets and library services.
- MinID is a common Identity and access management solution to government provided services such as Altinn, NAV and Lånekassen.
- OpenID is an identity management solution that can be used to access numerous online services, such as Google, Yahoo, Flickr and myspace.

The industrial uptake of identity federation technologies is relatively low. This is despite the predictions that companies can experience reduced cost related to identity management tasks by avoiding duplication of identity management efforts among the federation partners. Improved data quality can be experienced because user data is stored and maintained at one site, avoiding synchronisation issues. Increased security can be achieved with FIdM since security principles such as avoiding single-point-of-failure and to achieve minimal disclosure of data can be fulfilled, and fine-grained access control can be realised. Service providers can experience reduced complexity by outsourcing identity tasks to specialised identity providers. Last, FIdM promises to facilitate cooperation among federation partners where cross-domain single-sign-on, which may result in seamless service access across company borders. The following research questions are stated, and will be answered within this PhD research:

- What are the benefits and challenges related to FIdM identified and discussed by the research community?

- What are the perceived benefits, security risks and other challenges related to seamless sharing and usage of resources within an inter-organisational (industrial) collaboration context?

- What are the characteristics of current Federated Identity Management standards, and how does these relate (/map/match) to the inter-organisational collaborators' perceived security benefits and challenges.

- What are the main barriers to adoption of Federated Identity Management related to security, and how can these be solved in an (industrial) inter-organisational collaboration context?

The answers to these questions will contribute to bridging the gap between research interests and industrial needs. This can be used by IT strategy planners to understand security opportunities and threats of adopting federation technologies in inter-organisational environments, by other researchers to understand industrial needs related to identity federation technologies, and by developers and product manufacturers to understand company needs and to develop solutions that meet requirements in a professional inter-organisational collaboration context.

4.4.6.3 Postdoctoral Research: ICT Use for Safe and Reliable Petroleum Production

The topic of this post-doctoral project has been ICT use for safe and reliable petroleum production. The main research undertaken in this project is ethnographic fieldwork among petroleum engineers working in a multinational petroleum company's onshore operations center. This facility houses onshore personnel for several oil and gas fields on the Norwegian continental shelf. The fieldwork specifically concentrated on the daily work of the professional community of production engineers responsible for planning, optimizing, and monitoring daily petroleum production from the 50 wells of one of the company's many fields on the Norwegian continental shelf. The intended purpose of conducting the fieldwork was to contribute towards formulating requirements for the pilot software to be developed in IOHN. The production engineers are one of multiple professional communities in the field's larger production organization that encompasses both onshore and offshore personnel. The production engineers are organized under the petroleum technology department, one of the field's three onshore departments. Production planning and optimization is only one of numerous activities in petroleum production. An integral part of the production engineers' daily work is therefore to interact and coordinate activities with the other professional communities on the field, both onshore as well as offshore. This interaction takes place in formalized daily and weekly status, coordination, and planning meetings, as well as in more informal conversations face-to-face, on the phone, or through video conferencing. The Detailed Production Optimization (DPO) room houses the production engineers directly involved in planning, optimizing, and monitoring daily production from the Pan Field's many wells. The DPO room is a collaboration room equipped with multiple large, wall-mounted monitors and video conferencing facilities. The group of production engineers working in this collaboration room is colloquially referred to as the DPO. The DPO draws extensively on real-time sensor data in their day-to-day work. Sensor data is visualized in a host of different engineering applications, corporate reporting databases, as well as in locally developed spreadsheets. The DPO is the formal link between the land-based organization and the two operators manning the central control room (CCR) on the offshore production platform. The formal responsibility between the DPO and CCR is clearly defined: the DPO has the authority with respect to decision-making about how to control production on the field's many wells. The CCR operators have executing responsibility. Yet, the practical relationship between the two groups is characterized by professional deference and respect as they seek to retain good working relations at all times. During office hours, the DPO monitors the field's many wells, instructing the CCR. Outside of office hours, the CCR operators run production according to the priorities specified by the Well Contingency Plan. The Well Contingency Plan is a spreadsheet sent by email to the CCR operators at the end of regular office hours. In case of emergencies, there is always an engineer on an ambulating on-call duty for the CCR to confer with. The DPO is a nexus of activity during regular office hours. Production optimization requires detailed knowledge of the wells and the strategies for draining the reservoir. This is a job left to experienced engineers. The engineers in the DPO all have Master's level engineering degrees or higher. Due to their intimate knowledge of the wells, the DPO becomes a valuable information source for the other engineers working on the Pan Field. Diagnosing wells and assessing their state is part of production optimization. The CCR therefore calls upon the DPO when it experiences incidents with the wells. It is common for engineers from other disciplines to gravitate towards the DPO room during production-related incidents, either to get information or to participate in diagnosis and troubleshooting. Ethnographic fieldwork is a method where the researcher spends a prolonged period of time within the community or social setting that is being studied. Data is typically collected through observations, as well as formal and informal interviews. Three periods of fieldwork over a span of 10 months were performed. During each of these periods, between two and four days a week were spent at the onshore operations center. Each of the three periods of fieldwork lasted six to eight weeks. Throughout this period a permanent seat in the DPO room was available, observing and interacting with the field's production engineers as they went about their daily activities. Throughout this period full access to the onshore operations center was also granted. This made it possible to follow up on events and issues with engineers throughout the onshore organization, either by seeking people in their offices, in conversations over lunch, or during breaks in the communal areas of the operations center. During such field talks, questions were asked about how the engineers used different software tools in performing their many tasks, as well as more generally engage them in conversations about central problems they were dealing with in their daily work controlling and monitoring offshore production. Several other engineers would often join these conversations, making such informal talks an valuable source of information.

4.4.6.3.1 Epistemic Uncertainty and Semantic Interoperability

A key result from this study is to show how epistemic uncertainty is inherent in the way remote sensors generate data. When the computing literature speaks of uncertainty, it is statistic uncertainties related to the value of a measurement. Epistemic uncertainty, on the other hand, is related to the very phenomenon being measured. Rather than being unambiguous, sensor data may resonate with multiple physical phenomena. Take sand in the well flow as an example. To measure sand content, an electro-resistance sand sensor measures changes in resistance across metal probes. This builds upon the physical process where sand erodes the metal probe, which causes resistance to change. Change in resistance is then transformed into a sand content value. However, changes in well flow temperature may also cause resistance to change, even though sand content remains the same. We can therefore say that it is uncertain, just looking at the data generated by the sand sensor, whether there is sand in the well flow or changes in temperature. Although the sand probe is designed to measure sand content, sand data may also resonates with changes in well flow temperature. This has implications for how ICT is used to achieve safe and reliable operations. In making the right decisions, both for optimizing production but equally important during production incidents, the engineers need to sort out resonances in the sensor data. Drawing upon sand in the well flow again, a potentially critical incident, the engineers work to assess the situation before acting upon sensor data. They approach the sensor data with suspicion and caution. The procedures for handling incidents sand in the well flow cause production disruptions. Dedicated equipment has to be diverted from other operations, and the suspected well's production volume is turned down to minimize the effect of sand on the production equipment and the processing plant. The task of the production engineers, however, is to optimize daily production. This is made explicit by tying their annual salary bonuses to annual production volumes. As such, acting blindly upon the alarm is not an option for them. Yet, not acting upon the situation is not an option, either. Even when there is no danger of human injury or environmental harm, sand in the well flow can cause significant production disruptions, potentially ruining the well for good. Furthermore, sand clogs up the processing equipment. This requires clean-up work, reducing the amount of hydrocarbons the topside petrochemical plant can process. Unchecked sand also erodes pipes and valves, and maintenance personnel have to inspect the production equipment for damage, requiring a shut-down of parts of the processing plant, which in turn reduces production. Therefore when the sand alarm sounds, the production engineers choose to assess the situation before making any corrective measures. Sensor resonance has two implications for semantic interoperability, a key concern in IOHN. To achieve semantic interoperability, the content of information exchange requests need to be unambiguously defined – what is sent is the same as what is understood – in order to exchange information meaningfully and accurately between systems. Data generated about the same phenomenon, for instance sand in the well flow, but generated by different sensor types come with different resonances. When exchanging data, the sources of sensor resonance of different sensor types need to be accounted for in the reference model. Current reference models do not. Handling sensor resonances is a key concern that cuts through the activities of DPO engineers, as sorting out resonances is central in making production-related decisions. Much sensor resonance is related to the state of individual sensors. Sensors are mounted in extremely inhospitable environments. They drift, and they are often broken. Knowing the state of individual sensors is therefore critical to sorting out resonances. Yet, once sensor data is moved out of the oil field and then de-contextualized from the local knowledge about the state of sensors, such knowledge about faulty or drifting sensors are lost. When data is exchanged, it is therefore impossible to determine what the data really represent. Is it real measurements, or is the data simply resonating with a faulty or drifting sensor?

4.4.6.3.2 Digital Innovation Dynamics

The result on epistemic uncertainty builds upon the central insight developed through this study: that digital technologies play an integral role in creating, not simply representing, the materiality of the physical phenomena. This insight can also be used to explain the ongoing transformations of the offshore petroleum industry. The core vision of Integrated Operations is fully digital oil fields where mass volumes of sensor data is used for computer-assisted or even completely automated decision-making. This, however, offer limited analytical traction for explaining how digitalization is transforming the offshore petroleum industry. Interpreting the ongoing industrial transformation as a conflation of the material and the digital sheds light on the innovation outcome, increasing digitizing, instead of transformation processes. Building upon the insight above, we show how the on-going industrial transformation of offshore petroleum production is driven by a fundamental shift in the industry's relation to the material basis of offshore petroleum production. Whereas this relation used to be predominantly static, ongoing digitalization of offshore petroleum production shifts it towards an increasingly dynamic relationship. Such a dynamic relationship transforms the offshore petroleum industry as it on the one hand redistributes competency within petroleum companies, as well as between petroleum companies and other actors within the offshore petroleum industry, and, on the other hand, it transforms the nature of work, technology and organizing within the offshore petroleum industry.

4.5 Conclusion

4.5.1 Lessons Learnt

The planning of this project activity started in the autumn in 2006, when we contacted potential partners in order to work out and submit a project application to the Research Council of Norway. The application was submitted in March 2007. After a long evaluation period, the acceptance was received in October 2007. At that time two of our main partners were in the process of merging. The merger became effective in November 2007. We had to renegotiate with our partners, and since at that time the 2008 budgets in most of the companies were almost finalized, we had to find new partners to join the consortium. During this process it was also decided to join forces with the IOHN project, another big initiative were DNV was the project manager. The project activity then was kicked off as one of the activities in the IOHN project in June 2008. During the autumn 2008 one of our partners was acquired by a venture capitalist. The company was totally reorganised and decided to withdraw from the consortium. In spring 2009, a second and third partner went through a reorganization and decided also the withdraw from the consortium. One important learning is that it is difficult to keep a large consortium together in a period of 5-6 years (which is a typical time span from the early discussions to the end of the project). The reason is that companies merge, reorganize or change strategy and the original contact person that had interest in the project are moving on. Another important learning is that partner companies can have conflicting interests, by being competitors, by having a customer/supplier relationship or by other reasons. This makes that they not always want to reveal information to the rest of the consortium. We had two situations where this was a real problem, one of them was the work with Security principles and the other was when working with dependability cases. Since we also had the responsibility for two PhD students and on postdoctoral researcher this put a large risk on DNV, and we had to put more DNV effort into the project than planned in order to release the full financing from the Research Council.

4.5.2 Successes

The research performed in the first part of the project on the DDPRM has been further developed in the DNV business units as a Recommended practice and as an Offshore standard, [10], [9]. These have been very well received in the market and is already gaining considerable business for DNV. We further believe that the results from use of SBS, and on the Software FMEA and Software reliability modeling and prediction to be further developed within DNV's business units and be presented to the Oil and Gas market. In addition the project has been an important driver for moving the research direction in DNV Research and Innovation from having a large amount of research directions spanning wide from process oriented quality, through organisation and web technology towards product oriented verification and validation of software intensive systems.

4.5.3 Dissemination

The research results in the IOHN Activity 4 has been disseminated along three axes: 1) Internal and external presentations, 2) Publications in international conference proceedings and in international journals, and 3) internal and external workshops. In addition one of the PhD students has written a newspaper contribution (Adresseavisen), and work in the activity has also resulted in an article in Teknisk Ukeblad. We have produced a large amount of Technical Report, available from the DNV Library. Some of the Technical Reports are restricted for free distribution in order to disclose partner information. The Technical Reports produced in the activity are listed in section 4.7 List of deliverables. The conference and journal papers produced in the activity are also listed here. In addition we have developed four prototype tools:

- A prototype for search and navigation in dependability standard and among dependability patterns
- A prototype for generating a state machine model from software specifications and to produce abstract test cases
- A tool CBCD that detects software faults due to code reuse
- A Security principles database

4.5.3.1 Presentations

The papers presented at international conferences and in international journals are listed in section 4.7. In additions we have presented project results externally and internally:

- Internal DNV workshop presenting project results and further work in May 2011
- System-of-Systems Interoperability Challenges, April 2012
- Use of Model Checking for Analysing failure propagation in Interconnected Systems, May 2012
- Review of DDPRM, June 2009
- Rewriting Logic Modeling and Reasoning about Dynamiic and Concurrent Systems, January 2011 More details in section 4.7 List of deliverables.

4.5.3.2 International Conferences and Journals

The following papers and articles listed under section 4.6 References, [2] [1] [3] [20] [17] [16] [33]

[32] [36] [34] [31] [30] [27] [24] [22] [15] [23] [42] [4] [40] [41] [12] [14] [13] [19] [28]

4.5.3.3 Workshops

Activity 4 has been represented in several IOHN workshops arranged by the other Activitie. In particular we have participated in most of the architectural workshops arranged by Activity 2 under various Activity leaders. In September 2008, at the very beginning of the project we arranged an IOHN workshop involving most of the partners in order to discuss the scope and the detailed content of the project. In June 2009 we arranged a Workshop on Safety and Security of Critical Infrastructure used in Oil and Gas. The participants included most of the partners and also invited guests from the Norwegian oil and gas industry, a total of 25 participants. Three US experts (prof. Heather Drinan, I3P/Dathmoth College; prof. Jonathan Butts, University of Tulsa and prof. Sujeet Shanoi, also University of Tulsa) and representatives from activity 4 were leading the workshop.

4.5.3.4 Dissemination to public

4.5.3.5 Further Research and Further Work

The research we have done on formal methods, e.g., Rewriting logic, Use of Model Checking, and Sequence-based Specification (SBS) seems to be promising areas for further research. We are in the process of establishing a follow-up research project with European Space Agency on the use of SBS in the space domain. Furthermore the studies we have performed on System-of-Systems interoperability challenges is attacking an important challenge, also supported by [29]. We have started discussions with some of the partners from this project to discuss further research. We are also being engaged by the DNV business units to test out some of the project results on a real world case. *Give a list of press clippings, presentations, etc.*

4.6 References

[1] Andreas Aas, Heidi Stenberg Andersen, and Torbjørn Skramstad. A retrospective safety case for an advanced driller's cabin using goal structuring notation (gsn). In *Proceedings International Petroleum Technology Conference 2009*, Doha, December 2009. Society of Petroleum Engineers.

[2] Andreas Aas, Stig Ole Johnsen, and Torbjørn Skramstad. Experiences with human factors in norwegian petroleum control centre design. In *ESREL Conference*, Prague, Check Republic, September 2009. ESREL.

[3] Andreas Aas, Stefan Elgåfoss, and Torbjørn Skramstad. A design pattern for iso 13407 using goal structuring notation (gsn). In *Proceedings International Conference on Health, Safety and Environment in Oil and Gas Exploration and Production*. Society of Petroleum Engineers, December 2010.

[4] P. Almklow, T. Østerlie, and T. Haavik. Puntuation and extrapolation: Representing a subsurface reservoil. *Theoretical and Experimental Artificial Intelligence*, Forthcoming, Accepted 2012.

[5] T. Bauer, T. Beletski, R. Eschbach, D. Landmann, and J. Poorer. From requirements to statistical testing of embedded systems. In *Proceedings 4th International ICSE Workshop on Software Engineering for Automotive Systems*, Minneapolis, USA, month = , organization = IEEE Computer Society, 2007.

[6] P. Bishop, R Bloomfield, and S Guerra. The future of goal-based assurance cases. In *Proceedings Workshop on Assurance Cases*. IEEE Computer Society, 2004.

[7] P. Chinneck, D Pumfrey, and J. McDermid. The heat/act preliminary safety case: A case study in the use of goal structuring notation. In *Proceedings 9th Australian Workshop on Safety related Programmable Electronic Safety-Related Systems*, Brisbane, Australia, 2004. IEEE Computer Society.

[8] Det Norske Veritas. *Recommended Practice: Failure Mode and Effect Analysis (FMEA) of Redundant Systems*. International Electrotechnical Commision, 2008.

[9] Det Norske Veritas. *Offshore Standard: Integrated Software Dependent Systems (ISDS*. DNV AS, 2011/05.

[10] Det Norske Veritas. *Recommended Practice: Integrated Software Dependent Systems*. DNV AS, 2012/01.

[11] International Electric Committee. *61508 - Functional Safety of Electrical / Electronic / Programmable Electronic Safety-Related Systems*. International Electrotechnical Commision, 2008.

[12] J. Jensen. Benefits of federated Identity Management - A Survey frim an Integrated Operations Viewpointl. *Lecture Notes in Computer Science - Lecture Notes in Artificial Intelligence 2011*, 6908:49–61, 2011.

[13] J. Jensen and M. J Gilje. Security in model driven development: A survey. In *Proceedings of 6th International Conference on Availability, Reliability and Security*, 2011.

[14] J. Jensen and Martin Gilje Jaatun. Not ready for prime time: A survey on security in model driven development. *Journal of Secure Software Engineering*, 2.(4):49–61, 2011.

[15] Jostein Jensen and Åsmund Ahlmann Nyre. SOA security – an experience report. In *Norsk informasjonssikkerhetskonferanse (NISK)*, Nov 2009.

[16] Stig Ole Johnsen, Torbjørn Skramstad, and Janne Merete Hagen. Enhancing the safety, security and resilience of ict and scada systems using action research. In *Proceedings 3rd IFIP WG 11.10 International Conference: Critical infrastructure Protection*, New Hampshire, USA, March 2009. IFIP.

[17] Stig Ole Johnsen, Torbjørn Skramstad, Andres Aas, and Eivind Halvard Okstad. Proactive indicators of risk in remote operations of oil and gas fields. In *ProceedingsSPE International Conference on Health, Safety, and Environment in Oil and Gas Exploration and Production*, Rio de janeiro, Brazil, April 2010. Society of Petroleum Engineers.

[18] T. P. Kelly and R. Weaver. The goal structuring notation - a safety argument notation. In *Proceedings 34th International Conference on Dependable Systems and Networks, DSN-2004*, Florence, Italy, June 2004. IEEE Computer Society.

[19] J. Li and M. D. Ernst. Cbcd: Cloned buggy code detector. In *Proceedings of34th International Conference onSoftware Engineering*, Zürich, Switzerland, June 2012.

[20] Jingyue Li, Jing Xie, Erik Stensrud, and Torbjørn Skramstad. A case study on verifying software requirements and acceptance tast cases using sequence-based-specification. In *Proceedings 22nd International ymposium on Software Reliability Engineering*, Hiroshima, Japan, December 2011. IEEE Computer Society.

[21] Nooshin Aghajnani. *A Systematic Literature Review on Software Security and SCADA Security Taxonomy and Ontology*. NTNU, 2011/12. Master project report.

[22] Åsmund Ahlmann Nyre. Usage control enforcement - a survey. In A Min Tjoa, Gerald Quirchmayr, Ilsun You, and Lida Xu, editors, *MURPBES*, volume 6908 of *Lecture Notes in Computer Science*. Springer, 2011. URL http://dx.doi.org/10.1007/978-3-642-23300-5_4.

[23] Åsmund Ahlmann Nyre and Martin Gilje Jaatun. A probabilistic approach to information control. *Journal of Internet Technology*, 11(3), 2010.

[24] Åsmund Ahlmann Nyre and Martin Gilje Jaatun. On the adoption of usage control technology in collaborative environments. In *Proceedings of the 12th International Conference on Innovative Internet Community Systems*, page Accepted for publication, June 2012.

[25] S. J. Prowell and J. H. Poore. Foundations of sequence-based software specification. *IEEE Transactions on Software Engineering*, 29:417–429, 2005.

[26] W. Ridderhof, H-G Gross, and H. Dorr. Establishing evidence for safety cases in automotive systems - a case study. In *Proceedings 9th Australian Workshop on Safety related Programmable Electronic Safety-Related Systems*, Brisbane, Australia, 2004. IEEE Computer Society.

[27] S. Ruud, T. Skramstad, and I. B. Utne. Comparison of failure analyses of systems in maritime and offshore industries. In *Accepted for*

presentation and publication in proceedings of 11th International Marine Design Conference, Glasgow, Ireland, June 2012.

[28] R. Sassoon, M.. Jaatun, and J. Jensen. The road to hell is paved with good intentions: A story of (in)secure software development. In *Proceedings of 5th International Conference on Availability, Reliability and Security*, 2010.

[29] Don Shafer. Would you like software like that? In *AADE 2005 National Technical Conference and Exhibition*, Houston, Texas, USA, April 2005. American Association of Drilling Engineers.

[30] Torbjørn Skramstad and Heid Brovold. Software fault tolerance in marine systems. In *Proceedings IEEE/IFIP 38th International Conference on Dependable Systems and Networks, DSN-2008*, Philadelphia, USA, June 2008. IEEE Computer Society.

[31] Torbjørn Skramstad and Pieter den Hamer. Reliable it for ontegrated operations. In *Proceedings SPE - Intelligent Energy 2008*, Amsterdam, February 2008. Society of Petroleum Engineers.

[32] Torbjørn Skramstad and Pieter den Hamer. Autonomic service-oriented architecture for resilient complex systems. In *Proceedings 30th IEEE International Symposium on reliable Distributed Systems*, Madrid, Spain, October 2011. IEEE Computer Society.

[33] Torbjørn Skramstad, Stefan Elgåfoss, and Vibeke Dalberg. Maritime automation and emergency handling - software design, system integration and quality. In *Proceedingsg IEEE Electric Ship Technologies Symposium IEEE ESTS 2009*, Baltimore, Maryland, USA, April 2009. IEEE Computer Society.

[34] Torbjørn Skramstad, Stefan Elgåfoss, and Erik Stensrud. Inter-raterreliability of functional safety assessments - a case study of an oil and gas production control system. In *Proceedings IEEE/IFIP 41st International Conference on Dependable Systems and Networks, DSN-2011*, Hong kong, China, June 2011. IEEE Computer Society.

[35] Stig Ole Jonsen. *An investigation of Resilience in Complex Socio- Technical Systems to Improve Safety and Continuity in Integrated Operations*. NTNU, 2012/04. PhD dissertation.

[36] Erik Strensrud, Torbjørn Skramstad, Jingyue Li, and Jing Xie. Towards goal-based software safety certification based on prescriptive standards. In *Proceedings 22nd International Symposium on Software Reliability Engineering*, Hong Kong, China, December 2011. IEEE Computer Society.

[37] R. Weaver and G. Despotou. The heat/act preliminary safety case: A case study in the use of goal structuring notation. In *Proceedings 9th Australian Workshop on Safety related Programmable Electronic Safety-Related Systems*, Brisbane, Australia, 2004. IEEE Computer Society.

[38] R. Weaver, J. A. McDermid, and T. P. Kelly. The heat/act preliminary safety case: A case study in the use of goal structuring notation. In *Proceedings 9th Australian Workshop on Safety related Programmable Electronic Safety-Related Systems*, Brisbane, Australia, 2004. IEEE Computer Society.

[39] F. Zimmermann, R. Eschbach, J Kloos, and T. Bauer. Risk-based statistical testing: A refinement-based approach to the reliabilityu analysis of safety-critical systems. In *Proceedings 12th European Workshop on Dependable Computing*, Toulouse, France, May 2009. LAAS-CNRS.

[40] T. Østerlie. On countering methods erosion: Contributing towards stronger and more empirically grounded information systems theory. In *Proceedings of 45th Hawaii International Conference on System Sciences*, Nov 2012.

[41] T. Østerlie and Vidar Hepsø. Immediacy lost: Managing risks in oil and gas production. In *Proceedings of Collaborative Infrastructures Workshop (co-lcated with ECSCW'2009)45th Hawaii International Conference on System Sciences*, 2009.

[42] T. Østerlie, P. Almklow, and V. Hepsø. Dual materiality and knowing in petroleum production.l. *Information and Organization*, 11(22): 85–105, 2012.

4.7 List of deliverables

In the following list, preliminary or intermediate versions of deliverables have been left out.

The deliverables are available from the password protected project-internal WIKI-site: http://www.posccaesar.org/wiki/IOHN/Internal.

Title	Date	Contributors	Type
Reliable IT for Integrated Operations [31]	2008/2	Torbjørn Skrasmtad, Pieter den Hamer	Conference Article
Software Fault Tolerance in Marine Systems, [30]	2008/04	Torbjørn Skramstad, Heidi Brovold	Conference Article
A retrospective Case for an advanced Driller's cabin using Goal Structuring Notation [1]	2009/12	Andreas Aas, Torbjørn Skramstad	Conference Article
Enhancing the Safety , Security and resilience of ICT and SCADA systems using action research [16]	2009/03	Stig Ole Johnsen, Torbjørn Skramstad, Janne Merete Hagen	Conference article
Maritime Automation and Emergency Handling - Software Design, System Integration and Quality [33]	2009/04	Torbjørn Skramstad, Stefan Elgåfoss, Vibeke Dalberg	Conference article
Proactive indicators of risk in Remote Operations of Oil and Gas Fields [17]	2010/04	Stig Ole Johnsen, Torbjørn Skramstad, Andreas Aas	Conference article
Dependable Development Process Reference Model	2008/12	Thierry Coq, Jean-Pierre Guichenez (DNV France)	Technical Report
Aspects of Software, Systems and Component Technology Qualification	2008/09	Stefan Elgåfoss (DNV)	Technical Report
Software and System Ability Overview	2008/10	Stefan Elgåfoss(DNV)	Technical Report
Software Quality Inventory	2008/11	Olga Cerrato (DNV), Stefan Elgåfoss (DNV)	Technical Report
Security in a Service Oriented Architecture	2008	Magne Rodem (NTNU)	Master thesis NTNU
Resilience of open architectures	2009	Pieter den Hamer (DNV)	Technical Report
Goal Based Qualification of Software Related Technology	2009	Stefan Elgåfoss(DNV)	Technical Report
Review of Dependable Development Process Reference Model	2009/06	Ian Glendinning (DNV), Sigmund Kristiansen (DNV)	Presentation
Prototype of Structured inventory of existing standards and practices for dependability	2009	Stefan Elgåfoss	Technical Report
Federated Identity management in Oil and Gas Industry	2009	Jostein Jensen (NTNU)	PhD Research Plan
Secure Information Sharing in Integrated Operations	2009	Åsmund Ahlman Nyre (NTNU)	PhD Research Plan
Immediacy Lost: Managing Risks in Oil and Gas Production	2009	Thomas Østerlie (NTNU), Vidar Hepsø (Statoil)	Workshop Presentation
Safety and Security of Critical Infrastructure used in Oil and Gas	2009/06	Torbjørn Skramstad (DNV), Stig Ole Johnsen (NTNU)	Report from Workshop
Prototype for search and navigation in dependability patterns and standards	2010	Stefan Elgåfoss (DNV)	Executable prototype, Updated and extended version
A Pattern Based Approach Towards Security Engineering	2010	Ingrid Yu (DNV), Stefan Elgåfoss (DNV)	Technical Report
Concepts of Qualification and Software Related Technology	2010	Stefan Elgåfoss (DNV)	Technical Report

A Design Pattern for ISO 13407 Using Goal Structured notation	2010/12	Andreas Aas (DNV), Stefan Elgåfoss (DNV), Torbjørn Skramstad (DNV)	Conference article
Security Principles	2010	Ingrid Yu (DNV), Stefan Elgåfoss (DNV)	Database
Software and System Ability Overview	2010	Stefan Elgåfoss(DNV)	Technical Report, Updated and Extended version
A Study of Software Fault Tolerance Techniques	2010	Torbjørn Skramstad (DNV)	Technical Report
Software Quality Standards, Processes, Abilities, Methods and Best Practice	2010	Stefan Elgåfoss (DNV)	Technical Report
Security Principles	2010	Ingrid Yu (DNV)	Technical Report
IEC 61508 Part 3 as GSN Patterns	2010	Stefan Elgåfoss (DNV)	Technical Report
Overview of Formal Methods' applicability towards dependable software development and verification	2010	Ingrid Yu (DNV)	Technical Report
A Pattern Based Approach towards Security	2011/01	Ingrid Yu	Technical Report
Software Quality Standards, Processes, abilities, methods and Best Practices Inventory	2011	Stefan Elgåfoss (DNV)	Technical Report and Inventory
Security Principles: A Survey and Potential Application in Security Critical Integrated Systems	2011	Ingrid Yu	Technical Report
Rewriting Logic - Modeling and Reasoning about Dynamic and Concurrent Systems	2011/01	Ingrid Yu (DNV)	Presentation
Overview of Software, System and Component Technology for the Purpose of Qualification	2011/02	Stefan Elgåfoss (DNV)	Technical Report
A Review of the Results from the ModelME project on Model Driven Development of Dependable Software Systems	2011/03	Torbjørn Skramstad (DNV, Stefan Elgåfoss (DNV)	Technical Report
Outlining the Goal-Based Software Related Technology Qualification Process	2011/03	Stefan Elgåfoss (DNV)	Technical Report
Dependability Qualification od Complex Softweare Intensive Architectures - Prescriptive or Goal Based use of Safety Standards?	2011/04	Stafan Elgåfoss (DNV)	Technical Report
Review of Software Security and SCADA Security Taxonomi and Ontology	2011/12	Nooshin Aghajani (NTNU)	Student Project Report
A Case Study of Applying Sequence-based Specification on Verifying and Validating Subsea Control Software	2011/10	Jingyue Li (DNV), Jing Xie (DNV)	Technical Report
Software Reliability Modelling, Measurement and Prediction: A survey of Techniques and a Recommended Practice	2012	Torbjørn Skramstad (DNV)	Technical Report
Constructing Interaction Test Sets for Integrated Software Intensive Systems	2012/04	Jing Xie (DNV)	Technical Report

A Case study on Verifying Requirements and Acceptance Test cases using Sequence-based Specification [20]	2011/12	Jingyue Li, Jing Xie, Erik Stensrud, Torbjørn Skramstad	Conference article
Autonomic Service Oriented Architecture for Resilient Complex Systems [32]	2011/10	Torbjørn Skramstad, Pieter den Hamer	Conference article
Towards Goal-based Software Safety Certification based on Prescriptive standards [36]	2011/12	Erik Stensrud, Torbjørn Skramstad, Jingyue Li, Jing Xie	Conference article
Inter-Rater Reliability of Functional Safety Assessments - A Case study of an Oil and Gas production system [34]	2011/06	Torbjørn Skramstad, Stefan Elgåfoss, Erik Stensrud	Conference article
System-of-Systems Interoperability Challenges	2012/04	Jingyue Li (DNV), Ovidiu Drugan (DNV), Jing Xie (DNV)	Presentation
Use of Model Checking for Analysing Failure Propagation in Interconnected Systems	2012/05	Ovidiu Drugan (DNV)	Presentation
A Prototype Tool for modelling SBS State machines and Generation of Abstract Test Cases	2012/04	Jinyue Li (DNV)	Executable Prototype
Failure Mode and Effects Analysis of Software-Based Systems: A Survey of Best Practice	2012/05	Torbjørn Skramstad (DNV)	Technical Report
CBCD: Cloned Buggy Code Detector	2011/10	Jingyue Li	Executable Prototype tool
A survey of Techniques and Methods for FMECA/FMEA of software intensive systems	2012/04	Torbjørn Skramstad	Technical Report
Comparison of failure analyses of systems in maritime and offshore industries [27]	2012/05	Stian Ruud, Torbjørn Skramstad, Ingrid B. Utne	Conference article
CBCD: Cloned Buggy Code Detector [19]	2012/06	Jingyue Li, M. D. Ernst	Conference article

Report title:
Final report: All activities

Date of first issue:	*Project:*	
01.05.2012	Integrated Operations in the High North	
Prepared by:	*Reviewed by:*	*Approved by:*
Jens I. Ornæs, Roar Fjellheim	Hans Ronny Kjempekjenn	Tom Thomsen
Contributions by:	*Chapter/section:*	
Robert Ewald (NOV)	3.5	

Summary:
The Drilling Pilot in IOHN has been working towards three main results. 1.) A drilling control communication standard. 2.) A description of semantics that needs to be defined for Real-Time drilling data. 3.) A demonstrator of agent technology, for building autonomous behavior into the systems that controls the drilling equipment.
This pilot has also been a separate project, through the establishment of AutoConRig, a RCN-Funded (Research Council of Norway) project. The AutoConRig project has been executed with aligned time-line and goals as the Drilling Pilot in IOHN, and ends simultaneously. Throughout this report, when referring to the Drilling Pilot, it will also be a direct reference to AutoConRig.
A main goal of the project was to transfer the lessons learned into applicable technology to be used in the industry. NOV´s new drilling control system - to be launched in 2012, and demoed at OTC Show in Houston in May 2012, contains elements which is based on the theoretical foundation established in this project.

Revision	*SVN*	*Status*	*Date*	*By*	*Reason/Changes*
0.1		Draft	01.05.2012	JIO	Released for review by HRK
0.2		Draft	16.05.2012	JIO	Updated and released for review by PM
0.99	3558	Draft	24.05.2012	FVE	Released for review by Steering Committee
1.00	3592	Accepted	05.06.2012	FVE	Accepted by Steering Committee

Chapter 5

Drilling Pilot

5.1 Activity Summary

The Drilling Pilot has been conducted in a combination of workshops, and work individually performed by the project participants. The work has been to a large extent theoretical by nature. The pilot was challenged with a task that the drilling industry had not solved yet, namely how different systems - capable of controlling the drilling process could do so, in a generic way, without large integration and commission efforts and through reusing existing network infrastructure. The mechanisms that addresses this are a ”Drilling Control Communication Standard”, a ”common semantic base platform” and ”higher level of autonomy” in the existing drilling control systems and rig equipment. How to define and implement the three parts above was the Drilling Pilot scope of work.

5.2 Introduction

5.2.1 Purpose

The objective of the Drilling Pilot, as stated in the project definitions, was to analyze, develop and test an autonomous and semi-automated drilling control system. Unmanned drilling rigs or drilling rigs placed on the sea bottom can be used to eliminate extreme conditions on drilling facilities in High North areas. An unmanned drilling process requires significant change to the way it is being controlled today, where a high level of systems integration and autonomous decision capabilities is a substantial factor for success. The main objective was broken down into the following sub goals and results:

- Automated tripping sequences (run in and pull out drillstring from well)
- A standard for communicating and integrating with drilling controls systems
- Agent-oriented software architecture for autonomous control
- Demonstration on a full scale drilling rig with both normal operation and unexpected events

5.2.2 Participants in this Activity

- National Oilwell Varco AS (NOV) - Project management and Drilling technology
- International Research Institute of Stavanger (IRIS) - Model development and Test facilities
- Statoil ASA - Requirements Result validation
- Computas AS - Agent platform Software development
- University of Stavanger (UiS) - Decision support
- University of Oslo (UiO) - Ontology development

5.3 Preliminaries

5.3.1 References

There are two type of references in this report. References to work, articles or links related to this report are found as [N], where N is a number. These references are found in part 5.6 in the back of this report. The other type of reference is to deliveries and other results provided by the Drilling Pilot activity. These references are given on the format ”DN” i.e capital letter ”D” followed by a number N. All activity deliveries are listed in part 5.7 of this report.

5.3.2 Project execution

A wiki environment was established to be used for team information exchange and for keeping the results of the activity. Subversion was also used linked to this wiki, for keeping track of versions of shared documentation. This wiki and subversion was shared with other activities in the IOHN project. Information on the wiki was; action lists, call for meeting and minutes, workshop files, background information (both links and uploaded documents), activity deliveries and any other relevant information for the activities, to be shared amongst the team participants.

The Drilling Pilot had also in its intentions to fund the work of a Phd. This was placed at IRIS, where Nejm Saadallah has been participating with his work on a Drilling Control System Modelling approach. His work will be completed at autumn 2012, and a description of his work is provided in delivery D23.

5.3.3 Drilling domain introduction

The initial workshops in 2008 was targeted towards general information about the drilling domain. It was both focussing on the drilling process itself, and on typical infrastructures for information exchange and for control and automation on drilling rigs.

In order to introduce unmanned drilling operations, it soon became clear to the project that a significant job has to be done on the drilling equipment side - towards new designs where the machines themselves are more autonomous, both in terms of executing functions, and in terms of self-diagnostics. The Drilling Pilot therefore had some workshops together with SeabedRig who are building their drilling equipment based on technology from the robotics domain. This work gave some insight in requirements for control of a different type of drilling machines than traditionally adopted in the industry.

Further during the initial workshops a gap was identified in terms of understanding the information and control exchange requirements from a service companies perspective. The entrance of Baker Hughes into the Semantic Model activity of IOHN was therefor very welcomed, and contributed to an improved holistic view of the drilling operational environment.

5.3.4 Standards

Leveraging on the IOHN main objective - ”To design, implement and demonstrate a reliable and robust architecture for Integrated Operations Generation 2” - was a core incentive for the vertical activites in IOHN (the Drilling Pilot and the Production Pilot). This means that the standards and frameworks adopted by the horizontal activities (Integration Platform, Semantic Model and Safety and Risk) should to the extent possible be applied to the Drilling Pilot. Information about these standards and frameworks can be found in the chapters that covers the respective horizontal activities.

5.3.5 Ontology work

The IOHN Drilling Pilot has profited by the expertise of the participants of the Semantic Model Activity. RDF as a standard data format for linked data exchange between subsystems could be established. That allowed for a better decoupling of engineering efforts while still being able to integrate.

The ontology work of ISO15926 and OWL although not used directly gave valuable insights into good practices of data representation. However, modeling the drilling domain ontologically requires both experience in ontology engineering and the drilling domain. The Drilling Pilot only got as far as creating a shallow drilling taxonomy. Attempts were made to reuse definitions of the PCA library, but navigation and accessibility were suboptimal.

Another observation is that the benefits of ontology engineering are visible at a big scale of participants and tend to be obscured by more pragmatic ad-hoc approaches in the small. However any effort starts small and the pragmatic approach often closes the door for wider scaling.

In summary, the biggest potential use of semantic technology is the exchange and modeling engineering knowledge which can radically cut down engineering costs. However, accessibility to data must be improved so that the ontologies may be checked and reasoning can be used to generate new insights.

Even though the Drilling Pilot was not able to define an ontology - the definition work done will be taken further by the industry after the completion of the IOHN project. Status of this work, and plans ahead are discussed in 5.4.3.

5.3.6 Agent frameworks

The main objective of the AutoConRig project, and thus also for the Drilling Pilot, as formulated in the original project proposal to NFR [2], was to demonstrate autonomy in the drilling process:

"The main objective of this project is to analyze, develop and test an autonomous and semi-automated drilling control system."

An autonomous system can be defined as a system able to act without continuous human supervision in complex environments, implying that it needs to be able to reason about and set goals, plan action sequences to meet the goals, and execute and monitor those action plans. In the 2007 AutoConRig proposal, it was foreseen that such capabilities would be required for future drilling operations in challenging subsea and Arctic environments; a view that has been fully confirmed by subsequent evolution of the oil & gas business.

The work in this area has largely been carried out by Computas with domain knowledge provided by NOV, and has been extended and scaled up significantly outside the IOHN Drilling Pilot by NOV's development of the new NOVOS drilling control system [6].

The work carried out by Computas inside the project resulted in the following deliveries:

- State of the Art report: Software Agents
- Agent Pilot 1: Autonomous Tripping Sequence
- Agent Pilot 2: Auto-diagnosis of Drilling Equipment
- State of the Art report: Data Validation Techniques
- Agent Pilot 3: Data Quality Agent

Chapter 5.4.4 provide overviews of these deliveries as well as pointers to further documentation.

5.3.7 Network infrastructures

The IOHN digital platform is from an infrastructure perspective based on TCP/IP. In an automation or control context this infrastructure is considered to be un-deterministic, unreliable and unstable. (This is of course not the case for IT systems that do not share the real-time requirements of automation or control systems). This infrastructure is though nevertheless a design criteria for the Drilling Pilot which means that use of traditional control protocols on dedicated HW was not taken into account when designing the needed capabilities. This drove the project towards investigation of web technologies, and how these technologies could be used in a drilling control context.

5.4 Tasks in detail

The work in the Drilling Pilot in IOHN is in this report split into the following main tasks.

- Automated tripping sequences (run in and pull out drillstring from well)
- A standard for communicating and integrating with drilling controls systems
- Drilling related semantics and Ontologies
- Agent-oriented software architecture for autonomous control
- Demonstration on a full scale drilling rig with both normal operation and unexpected events

5.4.1 Automated tripping sequence

The work on the automated tripping sequence started off trying to define a tripping sequence as a state machine. This was done, both to test if a state machine can be used to represent the tripping process, and also as a tool to allow the project team to gain understanding in how a tripping sequence is executed.

The work on the state machine became the starting point for the PhD study that was funded by the Drilling Pilot, where the essence is to see if it is possible to simplify the design of a drilling control system by looking at it as a Discrete-Event-System.

Also, the defined state machine became the initial use case for a demonstrator application where an agent framework was used to control a trip-in operation.

After the completion of the demonstrator, the plan was to continue developing the system, and add different well-scenarios relevant for tripping. After some initial work on this - it was identified that this work had to be put on hold, due to reasons described in 5.4.1.4.

5.4.1.1 Tripping modeled as a state machine

To describe the tripping sequence as a state machine, it was decided to use the petri net language [7]. Several tools exists, but the team choose to use the free tool JARP Petrinet Analyzer [4]. This model was one of the deliveries from the Drilling Pilot in 2008 and can be found in the list of deliveries, D8.

The state machine covered the physical mechanisms in terms of what one can do with a drill pipe. In this regard, the state machine was built in a drilling process language - instead of drilling equipment language. The work on the state machine gave the team a good alignment in terms of introducing an abstraction between the drilling process and drilling equipment. This became a core design principle in all following work.

5.4.1.2 Modelling a Drilling Control System as a Discrete-Event-System

The general experience during the work on the state machine was that the state machine was well suited to describe the tripping process. After this conclusion this approach became the main area of study for the PhD where the initial tripping model was taken further to also cover other operations than tripping. It also looks at using a discrete-event architecture to model a control system reacting on events from the well.

What has been produced during the corse of the PhD work can be review in deliveries D13, D14, D19 and D30.

5.4.1.3 Demonstrator application for trip in operation

The first agent pilot developed by Computas and UiO in the project represents a major achievement and clearly demonstrated the potential of autonomy in drilling. The objectives for this agent pilot were:

- Demonstrate Autonomous Control of a Drilling Rig, in this case restricted to control of specific tripping sequences
- Act properly in case of communication failure where desirable behavior is described by a set of predefined scenarios

The scenarios describe an initial situation and a sequence of actions resulting in desired behavior with respect to the initial situation. The general setting is that the drilling crew runs a normal trip-in from a remote control center when they experience a communication error, leaving the drilling rig disconnected from the control center. Scenarios were used to verify the system's ability to handle such situations and in addition to this, be used to design the agent architecture. The following scenarios were defined for agent pilot 1 (illustrated in Figure 5.1)

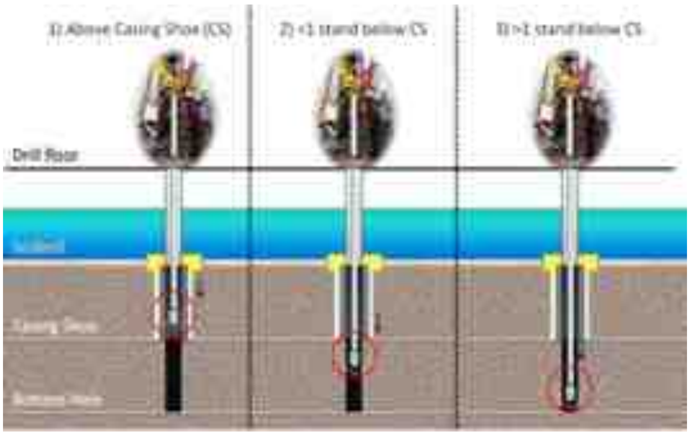


Figure 5.1: Tripping scenario for Agent Pilot 1

The agent pilot was constructed according the BDI multi-agent architecture, which will be described in detail in 5.4.4.1. The set agents and their interaction protocols were closely modeled on the actual organization structure of the drilling crew, as illustrated in Figure 5.2

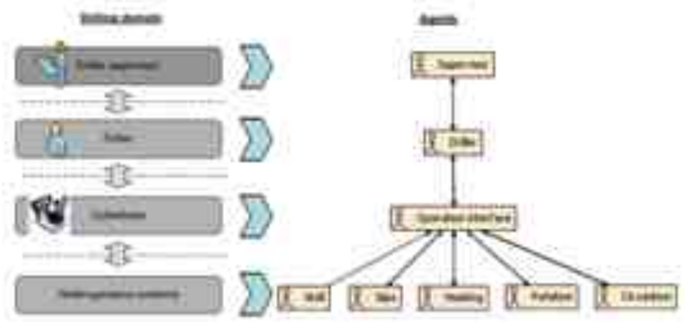


Figure 5.2: Drilling organization and corresponding multi-agent architecture

Each of the agents were developed using the JACK framework (5.4.4.1) for implementing autonomous logic and the LabVIEW [3] system for user interfacing and mathematical models of dynamic behavior (e.g. drill string deceleration). In addition to handling the prescribed scenarios, the autonomous agent was outfit with an embedded planning algorithm to be able to create new plans for unexpected situations "on-the fly".

The agent pilot was subjected to rigorous testing in a simulated environment. An example run of Scenario 2 is shown in the following figures, where Figure 5.3) Draw-work speed, Figure 5.4) Bit position, and Figure 5.5) Hookload are shown as a function of time.

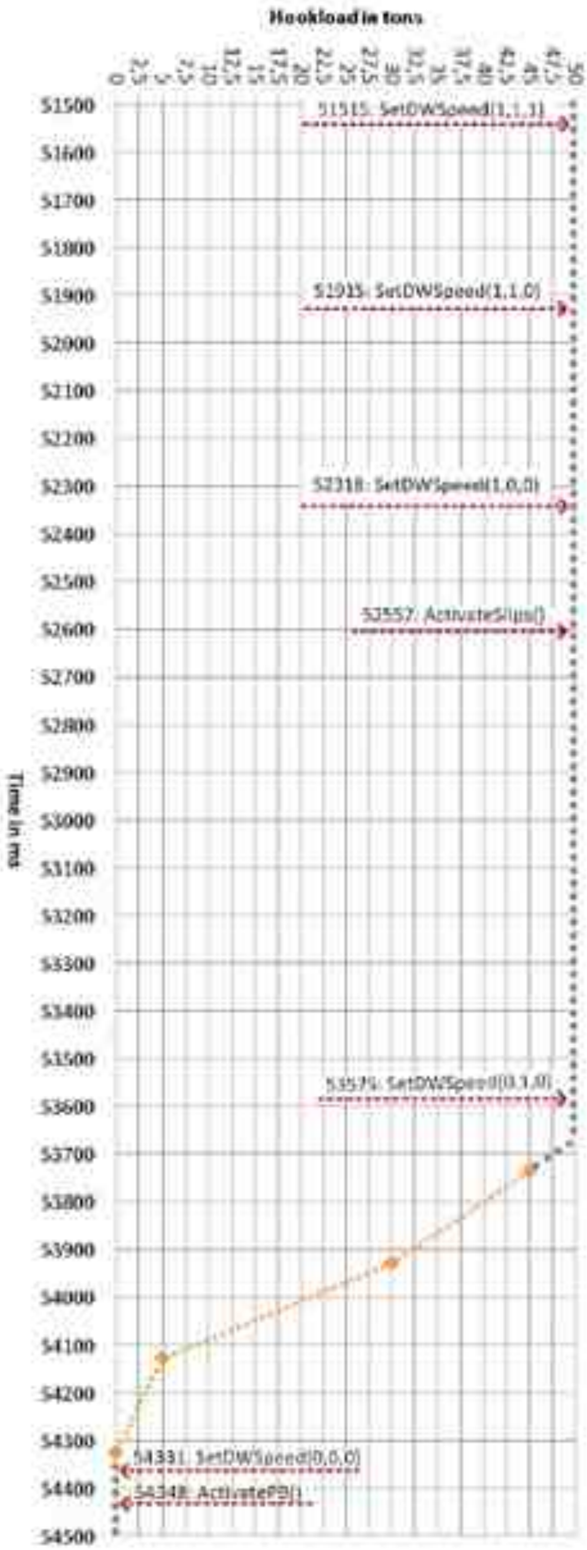


Figure 5.3: Scenario 2 example run of Agent Pilot, Draw-works

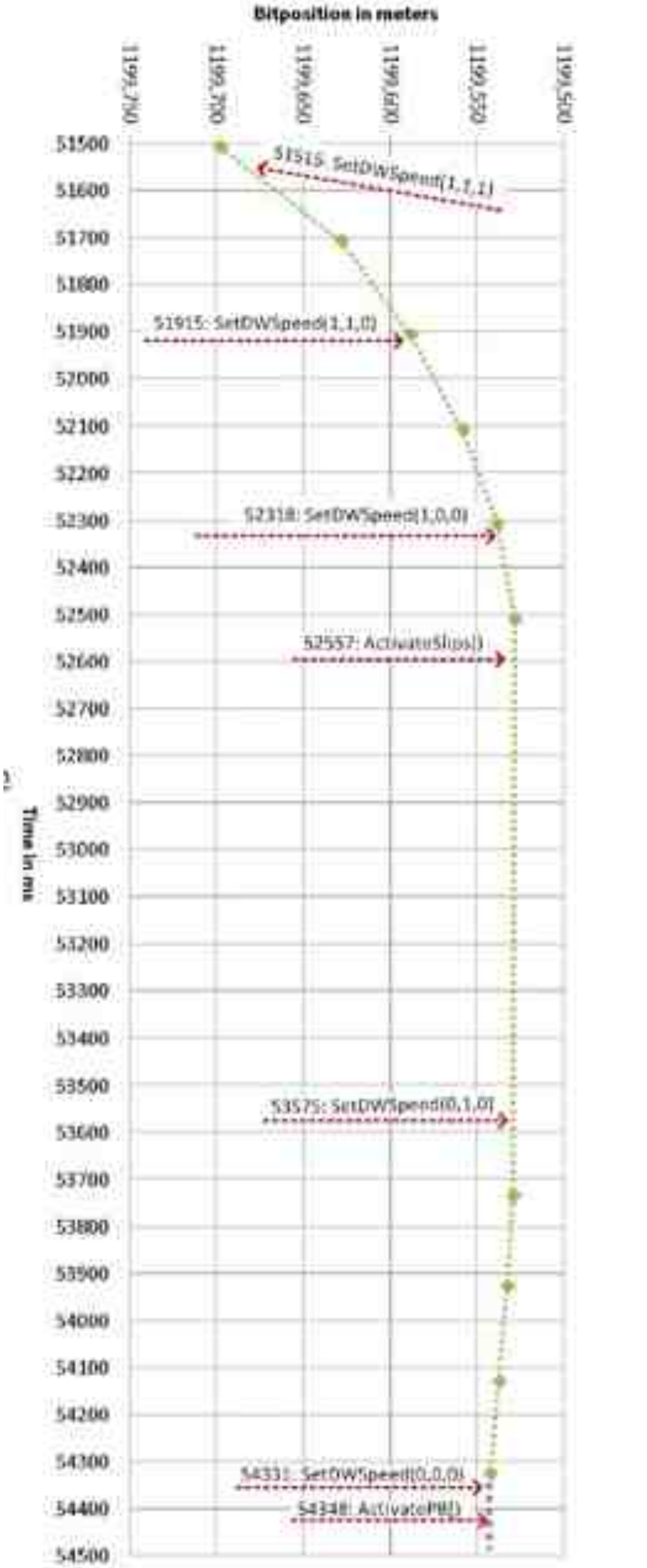


Figure 5.4: Scenario 2 example run of Agent Pilot, Bit Position

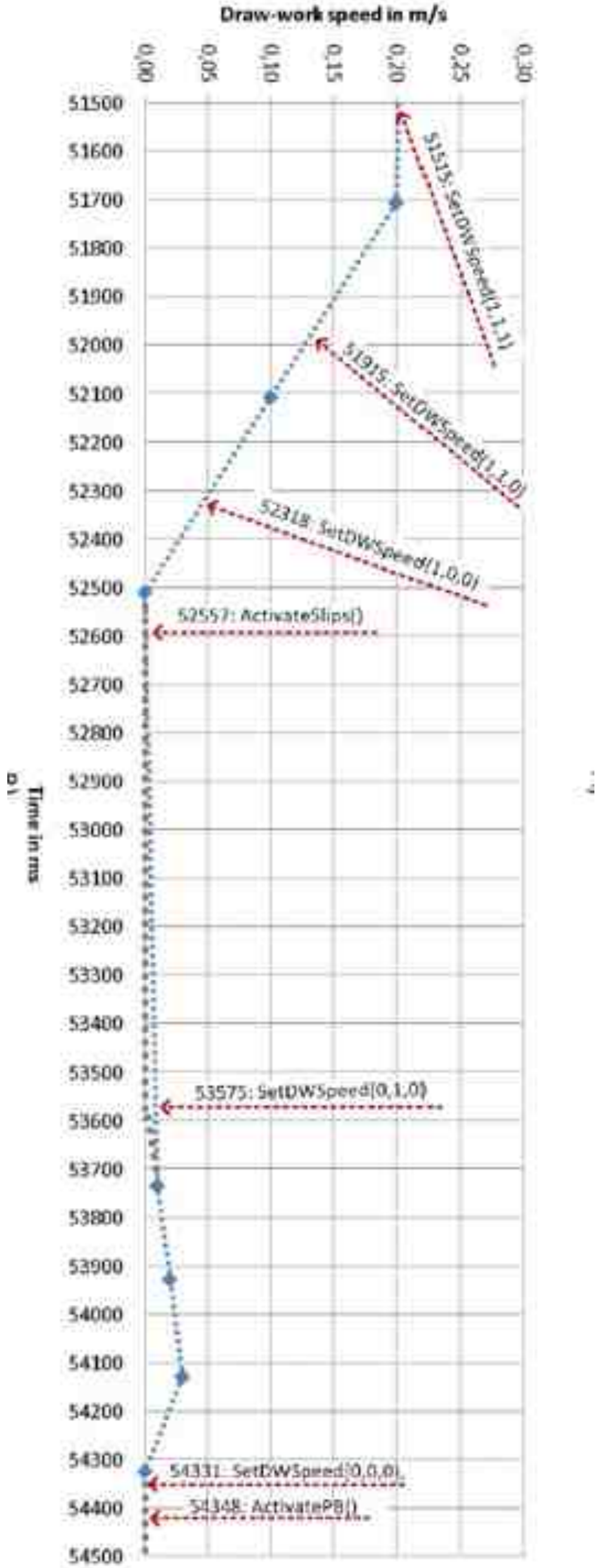


Figure 5.5: Scenario 2 example run of Agent Pilot, Hook Load

In summary the experiments gave the following results:

- The autonomous agent was compliant with the specification of the scenarios in respect to actions selected.
- A high level of autonomy was demonstrated as the system plans and correctly executed sequences of actions for situations unforeseen at design time.
- Both the course of action and the specific action sequences were optimal with respect to the specification and the state of the environment.

The experiment showed that the autonomous agent handled all of the test-cases successfully and fulfilled the experiment's success criteria. Although the experiment was conducted in a simulated setting, it gave an indication on the applicability of multi-agent technology within drilling. We claim that the experiment demonstrated an architecture that realistically can implement autonomous control of drilling processes.

5.4.1.4 Continued work on automated tripping

Even though the demonstrator was a success, this activity was discontinued after the demo. The main reason for this decision was that the next step, which would be to detect well situations, was a topic where some existing IP stopped us from bringing it in to the Drilling Pilot. Also - the demonstrator had showed a complete hierarchical architecture - from supervisory commands, down to detail equipment interaction.

This latter point generated a concern: Should the architecture proposed for an autonomous drilling system be based on autonomy brought down to the level of individual equipment, or should the autonomy be introduced as a layer on top of existing control systems?

The path ahead was pointed out towards introducing autonomy on top of existing control systems, as a supervisory system - almost like mimicking a human operator. The main reasons for choosing this path was:

- Possible to retrofit new technology on the vast fleet of drilling rigs giving them the potential of higher level of automation.
- Possible to generalize and base the autonomous layer on drilling process automation oppose to equipment automation (if the autonomy was introduced on the equipment layer, knowledge of all equipment would be needed in the autonomous controllers)
- Leave the responsibility for safe operation of equipment on equipment suppliers.

The considerations made and the philosophy established was combined into a SPE paper that was presented at the SPE Drilling Conference in New Orleans in February 2010, and provides as delivery D9 in the back of this report.

NOV started work on its next generation Drilling Control Systems in 2010. The technologies explored, and philosophies established in the Drilling Pilot of IOHN became important inputs for this work.

5.4.2 A standard for communicating and integrating with drilling controls systems

The motivations and constructs related to the work on the drilling control communication standard - which is a delivery from the Drilling Pilot in IOHN is also debated in the mentioned SPE Paper from 2010 (delivery D9). The project has developed a web service version of the standard - provided in delivery D11. Implementation of this standard using other technologies is quite possible, but the web service version was chosen to fit the IOHN digital platform. The core concept of the drilling control communication standard is that it is based on non-real time communication. We want the standard to work as an interface where safeguards, operational set-points and parameterized automated functions can be exchanged between control systems - or systems capable of controlling a specific part of the drilling operation. This means that the controller architecture controlling the physical equipment will need to have built-in the controller functions exposed on the drilling control communication interface.

A crucial and tightly belonging part to the drilling control communication platform is the semantic annotation of the parameters and functions. These must be unambiguously understood between a provided and a consumer, a topic further discussed in part 5.4.3.

During the work on the drilling control communication standard, it became evident that a lot of the information we want to capture with this standard is surfacing in the planning stage of the well construction process. The team addressed this in a separate workshop, but we decided that the amount of work involved in trying to map this out from existing well-planning applications would be to large scope for us. It is though a recommendation from this project that work should be done to map out information from the well planning phase that can go directly into the drilling control communication standard. This information would typically be operational speed limits for drill string axial speed (for tripping), drill string rotation, mud flow into the well and weight on bit (for on-bottom drilling). Operational limits for these parameters will in a well plan be specified against the well design - thus as per formation, well section or depth interval.

5.4.3 Drilling related semantics and Ontologies

A core precondition in order to achieve efficient and automatic interoperability between systems within a domain, is to establish a common semantic platform. One of the core drivers in the IOHN project is this exact issue. The Drilling Pilot has addressed this through several tasks throughout the corse of the project. The main goal was to establish a semantic platform for drilling based on ISO15926.

For ontology modelling Protègè was used [8]. In the first year of the project we made a simple Drilling Ontology based on this tool, which is given in delivery D6. Further throughout the corse of the project - the ontology work for the Drilling Pilot was done in close cooperation with the Semantic Model activity. This work is covered in the Semantic Model activity part of the IOHN report.

Now, at the closing of the IOHN project we see that we were not able to establish the common ontology we hoped to establish ref. 5.3.5. There have though been achievements in terms of defining exactly what is needed to efficiently connect different systems together - that will contribute to the well construction process.

From an operational perspective, what this project has found needed as a bare minimum is the following information about a parameter value.

- System: In terms of role of the provider.
- Object: The Object from which the signal is providing data.
- Medium: The medium which is measured e.g. Liquid, Hydraulic.
- Quantity: The physical quantity e.g. Volume, Pressure, Weight.
- Origin: Shall be used to describe sensor placements or principle of measurement in use related to the object.
- SensorInfo: For redundant sensors.
- Output: E.g. Average, Max, Min, Timestamp. When not given, the value is considered being an instant value.

Another important issue, is that all this parameter information should be defined on the basis of the drilling process instead of the equipment. Example: "Standpipe Pressure" is equipment oriented. It does not say anything about what this pressure is for someone not familiar to the drilling domain. The name "Mud Into Well Pressure" is its drilling process equivalent. Now it is humanly understood that this is the pressure of the mud being pumped into the well.

All this information can be part of a semantic signal definition included in a parameters metadata, or it can be defined in the tag nomenclature itself (this latter as the drilling industry has traditionally been dealing parameter semantics).

As a working exercise and quality check this latter approach was tested on a common set of drilling parameters. A report containing this work can be viewed in delivery D22. This report also includes metadata structures containing information regarded useful in an interoperable context.

The work on semantics for Drilling will be continued after the closing of the IOHN project. There will be a meeting in Houston June 28-th for the Standards Leadership Council (http://www.energistics.org/standards-leadership-council) where the continuance of this work will be picked up.

5.4.4 Agent-oriented software architecture for autonomous control

5.4.4.1 Agent-oriented framework

The Artificial Intelligence (AI) field has traditionally been the source of ideas and technology for autonomous systems, (ref. delivery D32) in particular the idea of autonomous software agents. The Drilling Pilot activities therefore started with a comprehensive investigation of the state of the art of agent technology (ref. derlivery D4). The report first discussed agent definitions, and concluded that an agent is a software system with the following characteristics:

- Reactive. Agents are sensitive to changes in their environment and react to these.
- Proactive/Persistent. Agents have goals which set their agenda and drive their actions.
- Autonomous. Agents make qualified decisions based on perception of the environment.
- Social. Agents can collaborate with other agents.
- Flexible. Agents can attempt to achieve their goal in several, alternative ways.
- Robust. Agents can recover from failure.

With respect to software architectures for agents, some main trends have emerged. The most popular agent theory is called the Belief Desire Intention (BDI) model, where an agent's internal representation of the world is represented using these mental states (A plain translation of these terms could be data, goals, and plans):

- Beliefs. Beliefs refer to the information an agent has about the state of its environment.
- Desires. Desires denote the state of mind the agent (ideally) wants to achieve.
- Intentions. Intentions are the subset of desires that the agent is committed to achieving.

The report also surveyed the state of technology frameworks for implementing software agents. The maturity of such frameworks is not at the level of more widely used software approaches, such as standard object oriented development. Many agent frameworks are research prototypes with interesting functionality, but lacking in stability, scalability, documentation, etc. There are a few exceptions, of which an example is the JACK framework, provided by the Australian/UK company AOS - Agent Oriented Software Ltd [5], and is based on the BDI architecture with the following components:

- JACK Agent Language, JAL
- JACK Agent Compiler
- JACK Agent Kernel
- ACK Development Environment, JDE

Another good reasons for choosing JACK was that Statoil had already been using this framework in an earlier study [1].

5.4.4.2 Equipment Auto Diagnosis

When the tripping agent pilot was abandoned, the Drilling Pilot decided to move forward identifying other areas where an agent oriented SW architecture would contribute towards building an autonomous drilling rig.

The subject of the second agent pilot developed by Computas became fault detection and diagnosis in drilling equipment. The complete report from this agent pilot is given in delivery D12. The more obvious extension of agent pilot 1 to more elaborate scenarios (e.g. packoff situations) was proposed by Computas, but rejected by other project participants because of conflict of interest issues. The underlying motivation of agent pilot 2 was that auto-diagnosis capabilities would be required in a future integrated agent system, which in addition to handling external events such as communication failure (agent pilot 1) also should be able to self-diagnose and self-repair in case of equipment failure.

Design criteria for agent pilot 2 included a knowledge-driven approach, where diagnosis of a new piece of equipment could be realized by adding new knowledge, and a modular architecture where independent knowledge bases could be developed and interfaced to the diagnostic reasoning engine via a common protocol Figure 5.6).

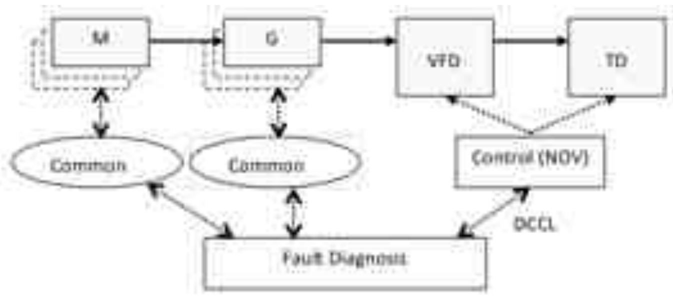


Figure 5.6: Overall architecture of Auto-Diagnosis agent

The diagnostic reasoning strategy for agent pilot 2 is based on the notion of process matrices and problem signatures. The diagnostic engine scans problem signatures stored in process matrices in order to perform automated Root Cause Analysis. A process matrix has one column header for each problem (or root cause) and row headers listing individual effects, or symptoms. Each symptom is defined in terms of a constraint expression describing a comparison test which can be evaluated by a computer algorithm during a diagnostic search. Relations between causes and effects are marked in the table cells, and the table grows until a satisfactory number of causes and effects are covered.

The process matrix approach is based on the assumption that each failure mode has a unique signature comprising the symptoms observed in faulty as well as indirectly affected components. Symptoms of different problems may overlap and so it is the unique combination of symptoms that differentiates one problem cause from another. Implementation of the diagnostic engine was done in LabVIEW. Figure 5.7) shows a snapshot of a process matrix as implemented in LabVIEW.

Figure 5.7: Overall architecture of Auto-Diagnosis agent

The agent pilot implementation includes a user interface to maintain (create, edit, save (binary), load) knowledge bases (process matrices) comprising fault signatures (symptom sets) and problem information (root causes). It also features a user interface to test individual symptoms as well as root cause detection.

A Data Server object associated with each process matrix encapsulates functionality to fetch sensory data for root cause evaluation. This provides a clean separation of diagnostic scan functionalities and connection to outside data sources. The Data Server also features a user interface to define alarms, signals, and data points (arithmetic combinations of signals) in order to allow local/standalone testing of knowledge bases.

The building blocks in the framework are ready to be incorporated in future diagnostic module applications. Such applications will typically include a diagnostic engine controller object administrating any number of process matrices operating concurrently. This allows fault detection and identification knowledge bases (process matrix instances) from multiple process component vendors be incorporated in the same fault diagnosis software module.

Unfortunately, the AutoConRig project failed to produce test results for agnet pilot 2 and its diagnostic engine framework. The main reason is that the domain knowledge required, i.e. process matrices with fault signatures for machine components, turned out to be much more difficult for NOV Subject Matter Experts to acquire and formulate that had been envisaged.

5.4.4.3 Data Validation Techniques

In the spring/summer of 2011, the Drilling Pilot project team had meetings with Statoil's Intelligent and Safe Well Construction project, and it was agreed that Data Quality and Sensor Validation (DQ/SV) in drilling operations are topics of common interest between the two projects. The terms were defined as follows:

- Data Quality is the state of completeness, validity, consistency, timeliness and accuracy that makes data appropriate for a specific use.
- Sensor Validation is the capability to detect, isolate and diagnose a faulty sensor. It may also include replacing the faulty sensor value with a calculated estimate.

It was decided to join forces to survey the State of the Art (SoA) within these areas and produce a

SoA report (delivery D16). Computas assumed the role of editing the report with contributions from Statoil, Baker Hughes, NOV and Computas. The report is split in two parts, where the first part is a selective survey of specific technologies and methodologies that the team found to be of particular relevance:

1. MIMIR. Developed by IFE as a toolbox to support the design, testing, and distribution of advanced condition monitoring applications.

2. PEANO. Developed by IFE for Sensor Condition Monitoring.

3. CBM. Study of Condition Based Maintenance solutions, carried out by Statoil.

4. SmartSensors. Sensors that combine actual measured value recording with complete signal conditioning and signal processing in a single housing.

5. Oil Service Company challenges. Baker Hughes work to improve the quality of sensor readings used in Drilling and Evaluation.

6. PI (Process Information). Enterprise data historian product by OSIsoft Inc. with a suite of software tools for management of real time data and events

7. Microsoft StreamInsight. A .Net-based DSDM (data stream management system) platform for development and deployment of CEP (complex event processing) applications.

8. MSPC. Multivariate Statistical Process Control extracts statistical correlation between process variables in order to employ automated fault detection and identification (FDI).

9. LabVIEW. Data-flow based, graphical software programming environment specifically targeting test, measurement, and control application development and deployment

10. IMP-SV. Inferential Modeling Platform for Sensor Validation provided by ABB is a software package for development and deployment of sensor validation strategies.

11. IO Center Surveys. Studies on challenges regarding data quality performed by Sintef Technology and Society and Sintef Petroleum Research.

12. ValiUpstream. Software suite for model based process data validation and reconciliation from the Belgian company Belsim S.A.

13. IIC. Integrated Information Core is an IBM product for real-time integration of information across multiple disparate systems based on industry standards.

The second part of the SoA report surveyed relevant international standards, including:

1. OPC-UA. (OLE for Process Control - Unified Architecture) aims to enable robust, cohesive, and secure integration between software applications and embedded devices.

2. SPE DSATS.

3. Semantic metadata standards. Includes standards published by the Open Geospatial Consortium (OGC) and W3C (World Wide Web Consortium).

4. GoICT. Subproject of IOHN deals with IT risk management, incl. a classification of architectural patterns as a meta-level in the design of reliable systems.

5. ISO 8000. Framework for improving data quality for specific kinds of data that can be used independently or in conjunction with quality management systems.

It was planned that Statoil and NOV would prepare a separate report (for confidentiality reasons) that matched their own DQ/SV requirements with the capabilities surveyed in the SoA report.

5.4.4.4 Data Quality Agent

The final delivery of the Autonomy and Software Agent stream of work conducted by Computas in the Drilling Pilot is agent pilot 3: Data Quality Agent (DQA) (delivery D18). In this delivery, results from previous deliveries (specifically agent pilot 2 and the DQ/SV report) are synthesized in a software module to be embedded in an open architecture for autonomous drilling control, such as NOV's NOVOS system.

The purpose of DAQ is to monitor and flag data quality problems in real time data streaming from the drilling operation. A small set of requirements for DQA to demonstrate some principles of sensor validation techniques was proposed by NOV. The specification listed nine different process variables to be monitored by the agent pilot software, and specified the principles/techniques to be employed as:

- Abnormal Parameter Variation
- Parameter Values Out of Limit
- Rule Based Dependency Violation

The requirements could be implemented in the LabVIEW framework established in agent pilot 2 Auto-Diagnosis, as the relevant knowledge bases could be represented in equivalent formats. The main addition to the already existing framework was a module for reading WITSML data from a WITSML server (initially a NOV test server, then the Ullrig on-line server). In addition specific DQA application and user interface modules needed to be developed. Figure 5.8) and Figure 5.9) show how DAQ reports data trends and status indicators, respectively.

Figure 5.8: DQA data trend visualization

Figure 5.9: DQA data quality indicators

The Data Quality Agent was interfaced to NOV's NOVOS system and passed a basic feasibility test at the NOVOS test session at Ullrig, Feb. 2012. It was also displayed at NOV's stand at the OTC exhibition in Houston, May 2012. We expect DQA to be further developed and scaled to industrial standards after the completion of the AutoConRig/IOHN project.

5.4.5 Demonstration on a full scale drilling rig with both normal operation and unexpected events

5.4.5.1 Description of the system that has been tested

NOV has over the last 2 years designed its next generation drilling control system, NOVOS [6], on several main areas based on the technology awareness and design philosophies established in the Drilling Pilot of IOHN. NOVOS is being designed to sit on top of existing drilling control systems, and will take over tasks that are currently done by the drillers, or the drill crew.

NOVOS is being configured with either an upload, or a manual typing of the well plan. A goal-driven decision engine is responsible for executing the well plan, which means that it will be able to both drill and trip autonomously, given that the existing rig control system and underlying equipment is capable of full automatic operation.

The Drilling Pilot in IOHN have in 2012 been part of 2 demoes. The first one in February at Ullrigg in Stavanger, and the seccond one in Galena Park land rig construction field, during OTC in Houston.

The system that was tested was the NOVOS system built by NOV. One strength of doing two tests was that we were able to demonstrate the flexibility of the architecture by retrofitting an autonomous control system to two highly different control systems. For the first demo at Ullirigg - NOVOS was retrofitted to an NOV Cyberbase system. For the second demo at Galena Park, NOVOS was retrofitted to an NOV Amphion system. The difference between these systems are quite significant. Cyberbase is built on off-the shelf industrical components such as industrial PC´s and PLC´s. The Amphion system - designed for the low cost land rig market is based on home grown controllers and embedded systems.

5.4.5.2 Ullrigg Demo - February 2012

The essence of this demo was that the architecture, based on the Drilling Pilot philosophy was tested, and that we showed that the drilling control communication standards abstraction approach allows a third party, in this case Baker Hughes - to give input to the execution of the drilling process though simple web service interface. Computas was using an agent framework - fed by real-time data from the drilling control system, via WITSML, to assess the quality of the monitored data (DQA in the figure below). The test report from Ullrigg demo is provided in the Drilling Pilot delivery D20.

The figure below shows a simplified topology of the system that was used. In this figure DQA stands for Data Quality Agent, and DA stands for Drilling Advisor.

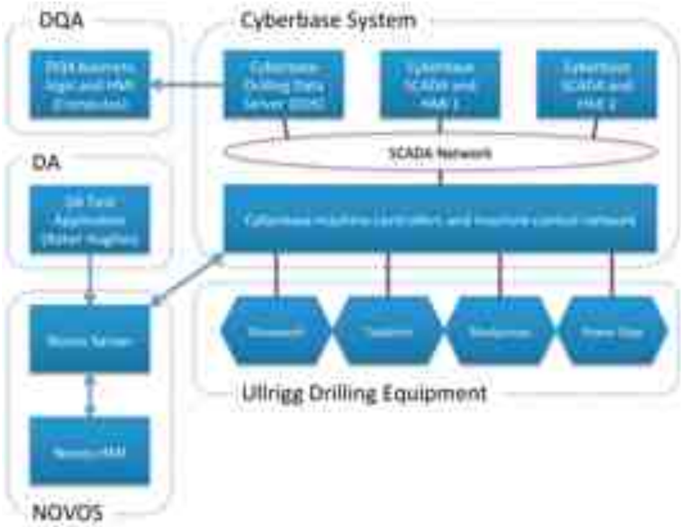


Figure 5.10: Ullrigg Simple Topology

5.4.5.3 Galena Park Demo - May 2012

At Galena Park an NOV landrig was set up to drill a shallow well during the OTC show in Houston, early may 2012. This demo was a repetition of the Ullrigg test, but on a different underlying control system, and different rig equipment. At the OTC show, information was provided about the IOHN project, and in particular the Drilling Pilot. Twise a day, busses were taking special interested parts of the audience to the rig, to watch how the rigs of the future will be operated. The test report from Galena Park demo is provided in the the Drilling Pilot delivery D21. The figure below shows a simplified topology of the system that was used.

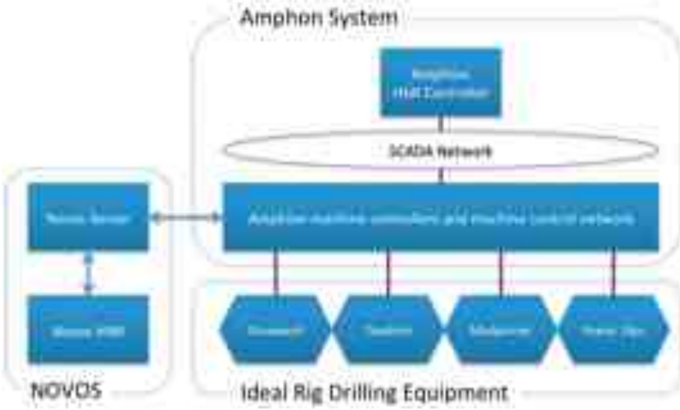


Figure 5.11: Galena Park Simple Topology

5.5 Conclusion

5.5.1 Lessons Learnt

During the 4 years of work on the Drilling Pilot in IOHN, a lot of lessons has been learned. Some of the delivery we thought was easy turned out quite hard to complete, such as building the ontology needed for a drilling operation. Others - which we considered to be hard, actually became manageable, such as building a drilling control system based on agent frameworks.

Agent Framework exists, and in terms of using them to build a supervisory structure on top of existing drilling control systems - this project has showed that this is quite feasible. An important observation though has been that introducing autonomous behavior on top of existing control systems - designed to be operated by a human operator, will require improved process feedback. Adding new sensors, and strengthen the existing sensors in terms of making them more robust or redundant is therefor a likely scenario. NOV´s new drilling control system NOVOS is based on this principle and will enable a new way of integrating drilling functions from several individual actors during the drilling process.

Using agent frameworks to build a distributed control system - with autonomous controllers for each drilling machine has showed to be more difficult, and hard to build an economical incentive for. However - addressing this holistically, together with a new approach to the design of drilling machines will make sense. Autonomy in a given function will never be possible without a repetitive and safe execution of that function by the end equipment. A lot of the equipment in use today does not have the sufficient level of repeatability when executing functions. A human operator can compensates for this, through his observation and correction. To enable a computer program to do the same correction the feedback from the machine needs to be improved (to replace the human operators senses) or the design of the machines need to change to improve the repeatability. It is in this context not difficult to look towards the robotics industry. Seabed Rig is a company that seems to attack the drilling process from this latter angle.

For integration of drilling functions, and general interoperability between systems, the core conclusion is to make the integration between systems happen on a generic level, abstracted from the equipment. The project had an initial focus of trying to define an ontology that modeled the drilling equipment. A simple model was completed, but the challenge we found was that such a model would only be valid for a specific set of equipment. The work in extending and maintaining such a model over the vast variation of rigs and rig equipment suppliers will just not be practically. The abstraction approach on the other hand is much easier to extend and maintain. The physics of a well construction process is not subject to that much change, and new functions can easier be added without over-riding old ones.

A drilling process language has been established by the project, and the work to get this into a standardized platform including the necessary semantics will continue after the ending of the project. An important note here is that for semantic work, it is important to define the boundaries for the work up front - to avoid a too broad modeling exercise.

A last lesson learned by this activity (even though it seems obvious) is that any autonomous system need to be pre-setup with some kind of a plan. This plan in the drilling domain would be the well plan. A well plan will contain top level activities, which can be broken down in to lower level activity. Some of these activities will be tasks that humans need to do, and other can be taken over by computers. The decomposition of the well plan in a structured way will enable organization to plan their way towards higher levels of automation on drilling rigs. The organizational aspect of the drilling process (RACI: Responsible, Accountable, Consult, Inform) would also be possible to model based on such a decomposition.

5.5.2 Successes

The Drilling Pilot team feel that the project has been a success for the partners involved. Knowledge about the drilling domain has been gained by the technology partners and universities. Web technology knowledge has been gained by the drilling domain participants. The following four bullets sums up our successes:

- Defined a standard interface approach for automated drilling functions.
- Proof of concept of autonomous tripping sequence.

- Leveraging of proven concepts in the construction of a new Drilling Control System by NOV
- Establish drilling domain common understanding on preferred architecture for distributed automation for drilling.

5.5.3 Subsequent work

As mentioned in 5.4.3 the work on drilling semantics will be taken further into meetings with the Standards Leadership Council. We believe that the establishment of the Standards Leadership Council will make it easier to get an agreed integration platform in place for the Drilling domain.

Well plan integration is an other topic that will be taken further by NOV based on the conclusions in this project. A vast number of different well plan formats exists. There have been both open and commercial efforts to build an information exchange platform between well plans. (One of the largest ones by OpenSpirit, www.openspirit.com). There has though not to this projects knowledge been any efforts in identifying the information present in well plans, that can be uploaded and used by a drilling control system to autonomously drill a well. NOV have started work on this, and simple first version interfaces are in place in the NOVOS drilling control system newly introduced by NOV. NOV wants to work with the drilling industry on this topic to generate an integration standard for this as well.

5.6 References

[1] Quale Olmheim, Landre. Spe 112078 – improving production by use of autonomous systems, 2008. .

[2] Fjellheim Sandsmark, Jansen. Autoconrig project proposal, submitted to the nfr petromaks program, Oct 2007. .

[3] Website. Labview. Available at http://www.ni.com/labview.

[4] Website. Jarp petrinet analyzer - free software, 2006. Available at http://sourceforge.net/projects/jarp.

[5] Website. Jack intelligent agents on wikipedia, 2012. Available at www.agent-software.com.au.

[6] Website. Novos, drilling control system (by nov), 2012. Available at http://www.nov.com/NOVOS.

[7] Website. Petri net information on wikipedia, 2012. Available at http://en.wikipedia.org/wiki/Petri_net.

[8] Website. Protégé ontology editor for semantic web, 2012. Available at http://protege.stanford.edu/plugins/owl/download.html.

5.7 List of deliveries

In the following list, preliminary or intermediate versions of deliverables have been left out.

The deliverables are available from the password protected project-internal WIKI-site: http://www.posccaesar.org/wiki/IOHN/Internal.

5.7.1 Drilling Pilot official deliveries

Title	Date	Contributors	Type
D1: Scenarios for Drilling Control System Communication Standard	2008/12	NOV, IRIS	Document
D2: Requirement for Drilling Control System Communication Standard	2008/12	NOV	Document
D3: Sequence Definitions - Tripping	2008/12	NOV, UiS, IRIS	Document
D4: Software Agents State Of The Art	2008/12	Computas	Report
D5: Position Paper, NOV for W3C Workshop 10.12.08	2008/12	NOV	Paper
D6: Drilling Ontology	2008/12	NOV, UiO	OWL Model
D7: Master Thesis: Towards Autonomous Control of Drilling Rigs	2008/12	Computas, UiO	Thesis
D8: State Diagram - Abstracted Drilling Machines	2009/12	UiS, IRIS, NOV	Petri net model
D9: SPE Paper: Closed Loop Control for Decision-Making Applications in Remote Operations	2009/12	NOV	Paper
D10: Drilling Control Communication Standard Specification	2009/12	NOV	Document
D11: Drilling Control Communication Standard Web service definition	2009/12	NOV, UiS	Web Service (xsd)
D12: Diagnostic Engine Proof of Concept	2010/12	Computas	Report
D13: PHD: A Goal Based Approach on top of Petri Nets	2010/03	IRIS	Abstract
D14: PHD: A Goal Based Approach on top of Petri Nets	2010/03	IRIS	Poster
D15: Data Quality and Sensor Validation - Scope	2011/02	Computas, NOV	Document
D16: Data Quality and Sensor Validation - Survey Report	2011/06	Computas	Survey Report
D17: New Technology Adoption in Drilling Controls, for Distinguished Lecturers Series	2011/08	NOV	Abstract
D18: Data Quality and Sensor Validation - Report	2011/12	Computas	Report
D19: PHD: A Simple Machine in A Complex Environment: A Petri Net Approach	2011/12	IRIS	Paper
D20: AutoConRig and NOVOS Demo at Ullrigg Feb. 2012	2012/02	NOV, Computas	Demo
D21: AutoConRig and NOVOS at OTC in Houston May. 2012	2012/05	NOV	Demo
D22: Tag Nomenclature using a process-oriented approach.	2012/05	NOV	Document
D23: PHD: Thesis Abstract	2012/05	IRIS	Abstract

5.7.2 Presentations and Aricles

All items in the list below can also be downloaded from
https://www.posccaesar.org/wiki/IOHN/Internal/InformaionAndDissimilation

Title	Date	Contributors	Type
D24: Semantic Technology and Information standards for drilling	2008/12	UiO, NOV	Presentation
D25: Towards autonomous control of a subsea drilling rig - an agent oriented approach	2009/12	Computas	Presentation
D26: Oil and Gas Journal: News article after SPE conference	2010/02	International Press	News article
D27: SPE Paper: Closed Loop Control for Decision-Making Applications in Remote Operations	2010/02	NOV	Presentation
D28: The America Oil and Gas Reporter: Feature Article in 2010 August edition	2010/08	International Press	Feature Article
D29: Think Outside The Planet - Presentation at OTC 2010	2010/08	NOV	Presentation
D30: PHD: Work presentation - Oct. 2010	2010/12	IRIS	Presentation
D31: New Technology Adoption in Drilling Controls, for Distinguished Lecturers Series	2011/08	NOV	Presentation
D32: AI and Autonomy in Oil and Gas, Conference	2012/03	Computas	Presentation

Report title:
Final report: All activities

Date of first issue:	*Project:*	
30.04.2012	Integrated Operations in the High North	
Prepared by:	*Reviewed by:*	*Approved by:*
Bård Henning Tvedt	Frederic Verhelst	Tom Thomsen

Contributions by:	*Chapter/section:*
Espen Breivik (Siemens)	6.1, 6.2, 6.3, 6.4 (-6.4.5, -6.4.6, -6.4.7), 6.5 (-6.5.4), 6.8
Christian M. Hansen (DNV)	6.4 (-6.4.4, -6.4.5, - 6.4.6)
Knut Hovda (ABB)	6.4 (-6.4.4, -6.4.5) , 6.6 (-6.6.7), 6.7 (-6.7.7)
Oddmund Kvernvold (DNV)	6.1, 6.2, 6.3, 6.6 (-6.6.6), 6.7 (-6.7.6), 6.8
Einar Stokke (DNV)	6.6 (-6.6.6), 6.7 (-6.7.6)
Morten Strand (DNV)	6.4.5
Bård Henning Tvedt (Epsis)	6.1, 6.2, 6.3, 6.4 (-6.4.5, -6.4.6, -6.4.7), 6.5 (-6.5.1, -6.5.5), 6.8
Åse Unander (ABB)	6.1, 6.2, 6.3, 6.6 (-6.6.7), 6.7 (-6.7.7), 6.8

Summary:
The main objective of the Production Pilot was to carry out case studies, relevant to the sand management domain, on a system architecture provided by other parts of the IOHN project. Work has been performed on three different cases, namely, Sand Detection, Erosion Management and Condition Based Inspection and Maintenance. All cases have successfully demonstrated new opportunities made possible by improved data integration and easier access to additional data.

Revision	*SVN*	*Status*	*Date*	*By*	*Reason/Changes*
0.1		Draft	15.05.2012	BHT	Issued for review of PM
0.99	3575	Draft	24.05.2012	FVE	Released for review by Steering Committee
1.00	3592	Accepted	05.06.2012	FVE	Accepted by Steering Committee

Chapter 6

Production Pilot

6.1 Activity Summary

The main objective of the Production Pilot was to carry out case studies, relevant to the production domain, on a system architecture provided by other parts of the IOHN project. The case studies aimed to explore:

- the system architecture's integration functionality, which is why it is commonly referred to as Integration Layer
- new opportunities for the production domain enabled by the improvements in data integration.

A scope containing the entire production domain was considered to large, and sand management was identified as an ideal sub-domain for the project. Three case studies were developed, namely:

- Sand Detection Case
- Erosion Management Case
- Condition Based Inspection and Maintenance Case

The cases require the Integration Layer facilitate integration of operational data, inspection and maintenance data and topology and configuration data. The different data types typically reside in different data bases. Operational data, including sand production data, are stored in data bases like IMS Historians like Aspentech Infoplus 21, Energy Components (EC) and Prosty. Inspection and maintenance data are typically to be found in SAP or Excel work sheets. Topology and configuration data on the other hand can reside in P&IDs, Excel work sheets and data sheets. To be able to access all these data through one interface is considered a large benefit by application developers as well as production and maintenance engineers. The data used in the project are courtesy Statoil and the Snorre license.

In order to integrate data the sand management domain had to be described and the following tasks have been performed:

- A nomenclature and information templates have been developed for data and facility equipment
- A topological model of Snorre B has been developed to provide the link between data and equipment, and their semantic descriptions

The description of the domain has been a joint effort of the Production Pilot group, and all participants have developed applications successfully retrieving data from the Integration Layer. The general opinion is that there is a large potential to be gained by seamless data integration, but the process of bringing it about is costly, and whether it is cost efficient or not is still an unanswered question.

6.1.1 Sand Detection Case

The participants in the Sand Detection Case have implemented new functionality in their applications, Epsis Decide and Siemens' IMS system PIMAQ, to enable SPARQL queries against the Integration Layer. The queries retrieve operational data and post new information about sand events. The latter enables in fact 2-way communication through the Integration Layer.

PIMAQ is also extended to detect significant variations in sand measurements, which is an indication of a potential sand event. Epsis Decide use a priori information to categorize sand events, and classify new events accordingly. The implementation is a prototype, but the initial results look promising. If events are correctly classified the result is a verification of sand measurements and can be used to maintain a set of corrected values as done by PIMAQ.

6.1.2 Erosion Management Case

In the Erosion Management Case both participants have extended their erosion applications, ABB's ErosionInsight and DNV Erosion Monitor Application, with functionality to perform SPARQL queries against the Integration Layer. The data retrieved in this case are operational data, welltest data, and configuration data like piping dimensions.

ErosionInsight has been set up to demonstrate erosion calculations at Snorre A, B and UPA installations, while DNV software has been similarly set up for the Snorre B installation. The work illustrates how results from erosion monitoring tools, like flow velocity, sand rate and erosion rate can be used within Production Optimization, as an acceptance criteria for safe production.

6.1.3 Condition Based Inspection and Maintenance Case

ABB and DNV have in this case extended the interaction towards the Integration Layer. SPARQL queries have been developed to now also include inspection data and event data like information about choke changes.

The inspection data can be used to verify and tune the erosion models and input data to improve the results from the erosion monitoring tools and make them more reliable. Erosion indicators like accumulated erosion and Cv difference can be used as a condition monitor for when components like bends and chokes have to be inspected. Such an approach will replace todays procedure of performing inspections at a specified time interval, and can thus change the way the Production and Maintenance domains collaborate as indicated in figure 6.1.

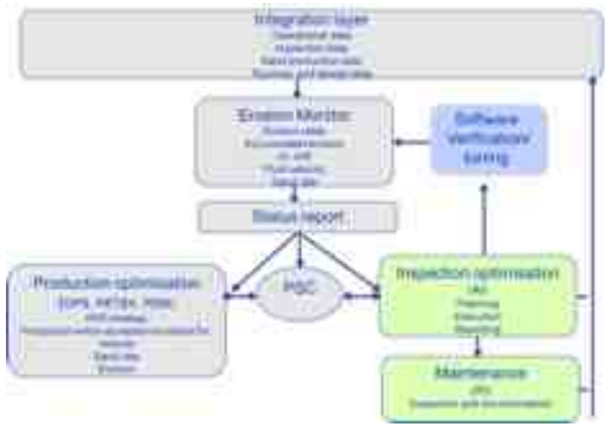


Figure 6.1: Schematic layout demonstrating data interoperability, data flow and use of erosion monitor systems as input to enhance collaboration between various disciplines within an oil company (Example from Statoil)

6.2 Introduction

6.2.1 Project scope

The IOHN project set out to pilot the second generation of Integrated Operations (IOG2) as described by The Norwegian Oil Industry Association (OLF). One of the main pillars of IOG2 is heavily automated facilities supported by extensive instrumentation. Newer installations and facilities are already well instrumented and large amounts of data are collected every day. However, the majority of data is unfortunately never utilized, and one of the bottlenecks is access to and integration of data.

The overal goal for this activity is to demonstrate the potential of increased decision support and data interoperability in a multi-asset production support center setting. Multi-asset support centers are typically second or third line support functions. The ambition is that these centers will become more proactive, and be able to monitor and manage specialized production-critical components and systems across assets. The main tasks can be summarized as:

- provide the project with domain requirements for data integration
- evaluate the potential of the integration functionality of IOHN's system architecture relative to current solutions
- evaluate new opportunities for the domain arising from the increased information availability

When identifying a production related case for the project, three key elements were assessed. First, it had to be an interesting case for the industry. Secondly, it had to be a well-defined problem that would demonstrate value in its own right, as a general production and optimization scenario was considered a too large scope. Lastly, it should involve the utilization of real-time data sources. Sand management was chosen because it met these requirements.

The project scope has not been altered throughout the project, but in order to obtain better results, changes have been made to the applied methodologies. Agent technology was initially given a prominent role, but was gradually replaced and finally not used at all. Semantic technologies are instead used in larger scale within integration, and machine learning is used for domain specific analyses.

6.2.2 Sand management

Sand production is a common challenge across the Norwegian Continental Shelf (NCS) and may constitute both severe safety risks, financial risks and environmental risk as well as lost opportunities as shown in figure 6.2. It is therefore crucial to have trustworthy data both with respect to operational data and measurement of sand production volumes, as well as reliable methodologies for erosion monitoring and erosion modeling.

If a producing well is satisfactory monitored lost opportunities can be explored. The most common production strategy involving sand is for instance the Maximum Sand Free Rate (MSFR) strategy, which can be a considerable limitation on production. By adopting an Acceptable Sand Rate (ASR) strategy production and profits can be increased.

The Production Pilot is divided into the following three cases:

- Sand Detection
- Erosion Management
- Condition Based Insepection and Maintenance

The purpose of the Sand Detection Case is to validate sand measurements. The validation results are made available through the Integration Layer and in doing so removing the separation between data providers and data consumers. Integration Layer is the Production Pilot's name for IOHN's system architecture.

In the Erosion Management Case, two state-of-the art erosion applications are modified to retrieve the necessary production data through the Integration Layer. The applications are implemented with very different strategies, and the aim is to show that the Integration Layer can be easily accessed independently of application implementation.

The Condition based Inspection and Maintenance Case proposes erosion predictions as an equipment condition monitor. The case integrates data from both the production and maintenance domain and aims to optimize well inspection schedules.

The first two cases were part of the original project, and the third was added after the Maintenance & Operations activity was canceled. All

Figure 6.2: Important elements when considering a production strategy involving sand

the cases are connected as the results from the the former case can be used as input to the next. The Production Pilot incorporates therefore the entire sand management loop.

6.2.3 Production Pilot Participants

The companies that have participated in the Production Pilot are ABB, DNV, Epsis, Siemens and Statoil. The participants have made contributions within the following areas:

- ABB- Data integration, Erosion Management, Condition based Inspection and Maintenance
- DNV- Data integration, Erosion Management, Condtion based Inspection and Maintenance
- Epsis- Activity lead, Data integration, Sand Detection
- Siemens- Data integration, Sand Detection
- Statoil - All

The work in the Production Pilot had not have been possible without the contributions made by DNV and Baker Hughes in Semantic Model and Integration Platform activities.

6.3 Preliminaries

6.3.1 System architecture

The focus of the system architecture proposed by the IOHN project is data integration. The Anzo platform from Cambridge Semantics was chosen to host the solution mainly because of its integration capabilities. At its core the Anzo Semantic Server is a Resource Description Framework (RDF) server with an associated SPARQL endpoint. It is referred to the work performed by the Integration Platform activity for more information about the system architecture and the Anzo server.

6.3.2 ISO 15926 Oil and Gas Ontology

The domain descriptions defined in the Production Pilot are standardized by linking them to the relevant counterparts in ISO 15926 Oil and Gas Ontology. The domain descriptions are described in chapter 6.4 and appendix 6.A, and it is referred to the work performed by the Semantic Model activity for more information about the use of ISO 15926.

6.3.3 Data and test facilities

In order to properly test and evaluate the solutions proposed by the project, relevant domain data were required. The project has been given access to data from Statoil's Snorre field, and the data that have been gathered are:

- operational data
- well test data,
- well inspection data
- topology and configuration data
- facility information

Production and well test data have been collected from both Statoil's hydrocarbon accountant Energy Components and Information Management System (PIMAQ). Inspection data for Snorre B were received as an Excel document, and the facilities were described in reports from previous project conducted by DNV's Flow Department.

The projects intentions were to establish a test facility. This facility would connect the Integration Layer to different distributed data sources through the SOIL network. By using the Integration Layer, it would have been possible for participants to fetch increased amount of data as work progressed. For example, the PIMAQ system was setup at the lab locally at Siemens and contained process data that spanned several years. However, due to challenges like security, cost and the change of integration server software; it was decided to copy needed data into the Cambridge Semantics integration server, and to request raw data from the IMS system on demand. Even though it was not an ideal situation to share data on these premises and the amount of data was limited, the participants have been able to test all concepts of communication successfully.

6.4 Interoperability

An interoperable solution indicates a future where data are exchanged seamlessly between applications, without need for manual interaction. The IOHN project's efforts to enter this new paradigm are gathered in the Integration Layer.

In chapter 6.4.1 the motivations and expectations in adapting the Integration Layer are described. Chapter 6.4.2 contains a record of the domain's efforts to describe information as required by the Integration Layer, while chapter 6.4.3 provides an overview of how the Integration Layer is applied. Finally, in chapters 6.4.4 to 6.4.8 the participants share their experiences.

6.4.1 Motivation

Unneland and Hauser [13] claim that many professionals in the oil industry spend 60-80 % of their time finding and preparing data instead of focusing on improving the quality of decisions. Increasingly sophisticated input functionality in expert applications greatly reduce this time, but still interesting data can be inaccessible or in need of validation. In addition new challenges arise as the need for configuration of applications increases and the number of data transformations to be maintained grows larger.

The motivation for adopting an Integration Layer can be summarized as follows:

- the maintenance of multiple data transformations is replaced by the single transformation towards the Integration Layer
- the vendors' data access costs are reduced since it is handled by the Integration Layer
- grants easier access to more data that has not been available previously
- reduce configuration costs as these data can be accessed through the Integration Layer
- an application's results can be made available to other applications
- easier to deploy applications at sites and companies supporting the Integration Layer

6.4.2 Domain descriptions

In order to enable integration of information the necessary parts of the domain had to be described unambiguously. It was quickly realized that the necessary parts constituted data to be transfered, including measurements and the equipment producing the measurements. Other parts, like for instance terms used within erosion theory, are omitted even though they are relevant in describing the domain as a whole.

The terms given in the nomenclature are described in chapter 6.4.2.1, while combinations of terms to describe measurements and equipment are given in chapter 6.4.2.2 and 6.4.2.3, respectively.

6.4.2.1 Nomenclature specification

The sand management domain has an extensive terminology, but it was challenging to set clear boundaries as to what information to include in the nomenclature. It also became apparent that only a small subsection was needed to describe the information intended shared through the Information Layer. The initial aim to create a complete nomenclature was therefore modified to incorporate only information needed to transfer data between applications.

An important consideration in creating the nomenclature was to decide on a level for terms' complexity. A term can be expressed in a unary form like for instance "Pressure" and "Bar" describing a physical property and a unit of measurement. However, terms can also be described in a n-ary form. The term "InsidePressureAtWellInBar" gives the pressure inside the well in bar for example. All terms can be combined to create more precise and detailed meanings. Simpler terms can be combined in a more versatile way than concatenated terms, but the resources needed for combination rules increases. The concatenated terms requires, on the other hand, less combination rules since this is already done, but in creating all combinations the number of concatenated terms grows rapidly. Generality and flexibility have been two important principals for the Production Pilot's participants, and the terms have mainly been kept simple.

A total of 86 terms have been identified and described, and they can be separated into the following 6 categories:

- facility component type (36) - describes the equipment in the topological model. Examples are "Well", "10 inch Flowline - Sch 120" and "Pressure Element"
- measured data types (16) - describes the available measured and derived operational data. Examples are "Stroke", "Inside Pressure" and "Oil Volume Flow Rate"
- configuration data types (3) - describes the available configuration data in the topological model. The elements are "Diameter Type", "Inside Diameter" and "Outside Diameter"
- value aggregation types (11) - describes whether the data are raw measurements or grouped in some way. Examples are "Raw Value", "Daily Average" and "Accumulated Value"
- units of measurement (19) - examples are "Sm3/d", "g/s" and "bara"
- logical types (1) - boolean = true/false

Each category contains a parent term, which all other concepts in the category are children of. "DataType" is for instance the parent of measurements. Its children are "InsidePressure", "InsideTemperature" and so on and must therefore not be confused with common datatypes like "string" and "integer" in computer science. A complete overview of the nomenclature is given in Appendix 6.A

6.4.2.2 Information templates

A measurement itself is only a value, for instance a decimal number, and some sort of description is required to give it meaning. Minimum requirements are to know what it represents and its unit of measurement. Additional information is often needed to describe it fully. In the nomenclature this information is given as several simple terms, which can be combined to give a satisfactory description.

A schema for combining terms is called a template, and 31 templates have been defined in the project in close collaboration with the Semantic Model group. Templates can for instance describe measurements, properties of equipment, relations between equipment and relative positions, and examples of such templates are given below.

iohn6tpl:DataSourceValue
Description: *States that a given iohn6:DataSource has a given data value (in xsd:double) at a given time (as xsd:dateTime) for the given iohn:DataType, iohn:UnitOfMeasure, and iohn: AggregationType.*

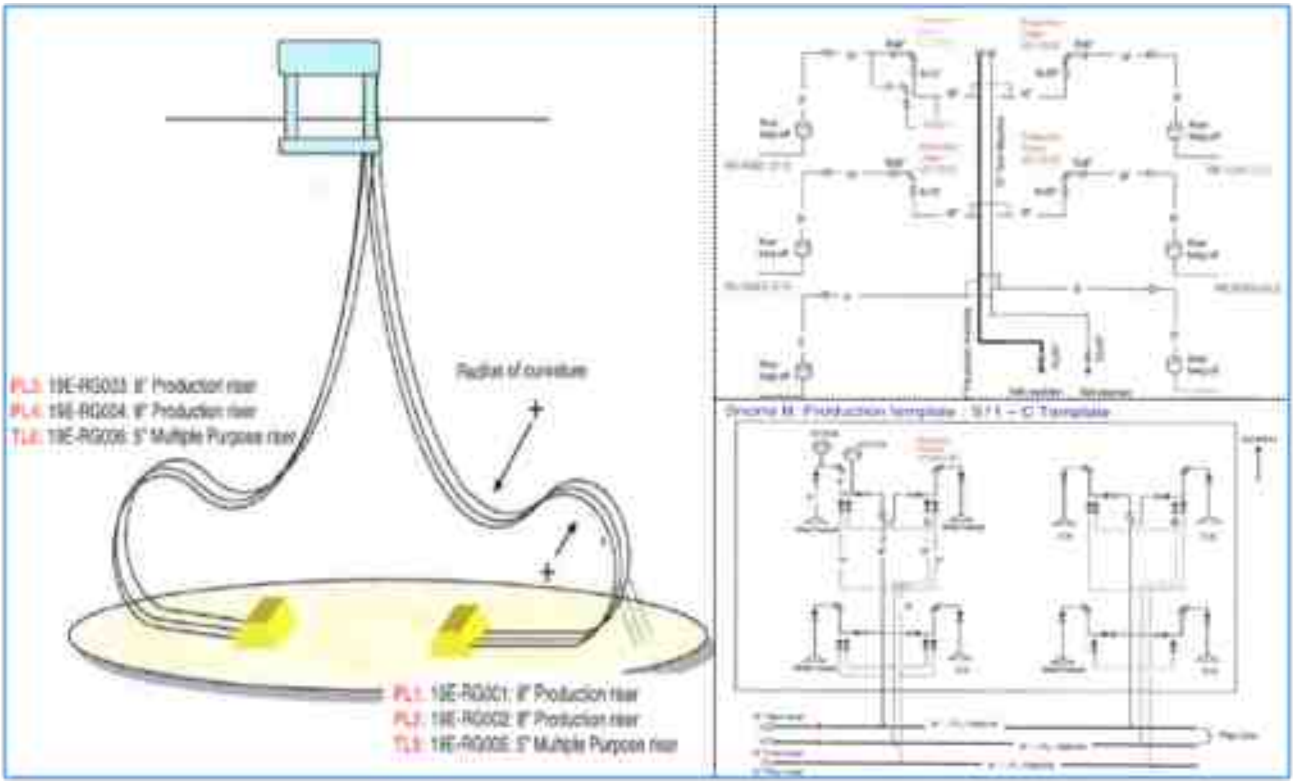


Figure 6.3: *Schematic layout of Snorre B (left) subsea template (right lower) and topside piping (right upper)*

#	Role	Type
1	iohn6tpl:hasDataSource	iohn6:DataSource
2	iohn6tpl:hasDataType	iohn:DataType
3	iohn6tpl:hasUnitOfMeasure	iohn:UnitOfMeasure
4	iohn6tpl:hasAggregationType	iohn:AggregationType
5	iohn6tpl:valDateTime	xsd:dateTime
6	iohn6tpl:valDoubleValue	xsd:double

This example illustrates templates' generic features. The *Type* iohn6:DataType that fills the *Role* iohn6tpl:hasDataType is as mentioned in the previous chapter a parent term. iohn:DataType can therefore be replaced by iohn6:InsidePressure, iohn6:InsideTemperature or any other term in this category. By making similar and appropriate replacements for the other roles all time-series used in the project can be expressed.

iohn6tpl:MeasurementDeviceAttachment

Description: *A connection of a measuring device to some fluid conducting artefact, which allows some measurement to take place.*

Derived property: iohn6model:attachedTo

#	Role	Type
1	iohn6tpl:hasMeasuringDevice	iohn6:DataSource
2	iohn6tpl: hasFluidTransportDevice	iohn6:ContinuousFluid-TransportDevice

The iohn6tpl:MeasurementDeviceAttatchment provides the important connection as to where a sensor or measuring device is attached. This template makes it possible to query a well for its available measuring devices.

iohn6tpl:RadiusOfCurvature

Description: *The radius of curvature of a given pipe is the product of the length of the given diameter type and the factor.*

#	Role	Type
1	iohn6tpl:hasArtefact	iohn6:Artefact
2	iohn6tpl:hasDiameterType	iohn:DiameterType
3	iohn6tpl:valFactor	xsd:float

In figure 6.3 the radius of curvature

of a pipe is shown. Piping configurations, such as radius of curvature, are important in erosion calculations and are today part of the manual configuration work of erosion applications. This template makes it possible to retrieve this information directly from the platform model.

iohn6tpl:Upstream
Description: *Location of the measuring device relative to choke with respect to flow.*

Derived property: iohn6model:upstreamMeasuringDevice

#	Role	Type
1	iohn6tpl:hasChoke	iohn6: AdjustableChokeValve
2	iohn6tpl:hasMeasuringDevice	iohn6:DataSource

Measurements are made both upstream and downstream of a choke. This particular template enables queries that ask for a measurement device upstream of a choke. The alternative is to know the measurement device's identifier. It is in other words one of the templates that simplifies the navigation of the Snorre B model.

For a complete list of production templates developed in the project it is referred to Appendix 6.A.

6.4.2.3 Topological model of Snorre B

Today, accessing and retrieving data can be a cumbersome process. First hand knowledge of which data that are available, where they are stored and how to retrieve them is necessary. Data exist in many forms and of the different kinds that often are required are

- continuous - *operational timeseries*
- (semi) static - *configuration and topology data*
- periodic - *choke changes, well inspections*

The project aims to improve on the current situation by enabling search for and access to these data through the Integration Layer.

The terms in the nomenclature describe non-specific items, for instance a pressure sensor, but in order to retrieve data they must be linked to particular sensors. A model representing actual components on a platform has been created to provide the connection between actual data, equipment and the semantic description. The platform used in the project is Snorre B and the schematic layout of Snorre B is shown in figure 6.3. This figure and additional platform information were obtained from Kvernvold and Torbergsen. [11]

The model of Snorre B is created in Microsoft Visio, and contains relations between different facility components from the topside platform, trough its pipelines, and down into the wells. A total of 6 wells are connected to 2 manifolds, which again is connected to 2 production lines and 1 test line. The production lines are connected to a inlet separator train, while the test-lines are connected to a test separator.

The purpose of the model is to enable navigation of data sources and retrieving related data. It is therefore primarily focused on sensors or measurement devices, and the components that describes their locations. All in all the model can be said to be made up of four different kinds of facility components:

- navigation components - linking other components enabling navigation of the model
- container component - containing a set of smaller components making navigation easier
- data source components - linked to related data making timeseries retrievable
- property components - providing retrievable property information of other components

There is a total of 80 navigation, 6 container and 111 data source components. Most of the navigation components are also linked to property components providing for instance pipe dimensions. Properties can be expressed in three different ways as described later in this chapter.

Parts of one of the production pipelines is shown in figure 6.4 and figure 6.5. It is a container component made up of 10 flow components that are attached to 7 data sources providing pressures, temperatures, stroke, raw sand values and estimated sand rates. The pressure, temperature and sand rates measured upstream choke are shown in figure 6.6.

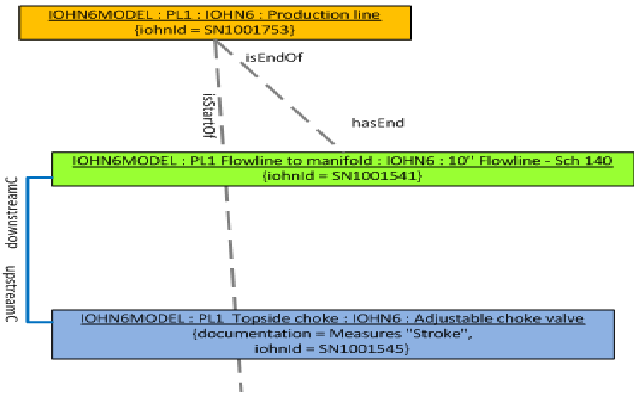


Figure 6.4: End of production pipeline P1 from the Snorre B topology model

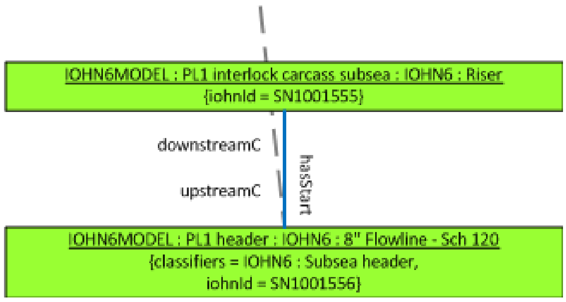


Figure 6.5: Start of production pipeline P1 from the Snorre B topology model

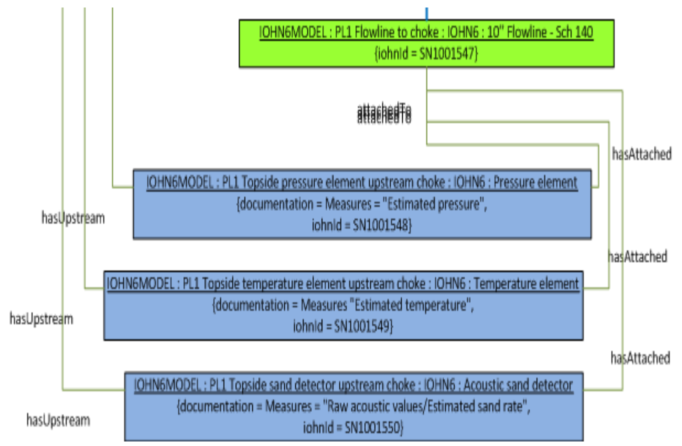


Figure 6.6: Sensors connected upstream choke at the P1 production line

The data represented by the data sources in figure 6.6 are all actual measurements, and different measurements are available at different parts of the model. Operational data are for instance available from the sensors connected to wells and production pipelines, while well test data are available from the test separator. The routing of wells to the different flowlines are modeled as valves connecting a well to the flow spools of the different flowlines as shown in figure 6.7. 3 valves exist for every well directing the flow onto one of the production lines or the test line. A well can only be routed onto one flowline at the time and thus one valve is open while the other two are shut.

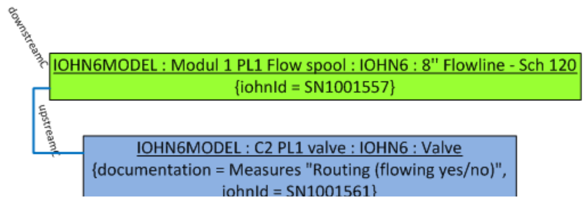


Figure 6.7: Example of valve used for routing a Snorre B well onto flowline.

In some cases data are not measured directly. An example is fluid volume rates, which are allocated, and therefore not connected to a real flow meter. However, being a data source is considered an abstract property of the component. Wells, as depicted in figure 6.8 are defined as a data source providing fluid rates even though they are not performing any measurements.

Periodic data or events are also linked to the model through templates as described in the previous chapters. A sand detector can for instance have sand events, while chokes can have choke changed events.

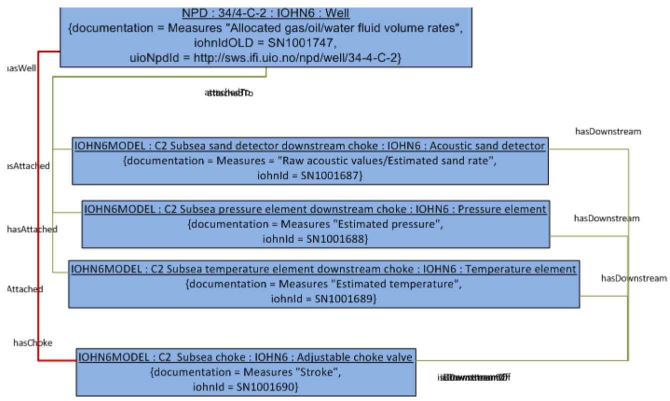


Figure 6.8: Examples of typical data sources connected to a Snorre B well

The pipe objects in the Snorre B platform model are annotated with wall thickness, inner diameter, and outer diameter. For pipes with dimensions given according to the Nominal Pipe Size (NPS)[1] system, the pipe class is annotated with the dimensions. Figure 6.9 shows a 8”

[1]Nominal Pipe Size is a North American set of standard sizes for pipes, where pipe size is specified with two non-dimensional numbers: a nominal pipe size for diameter based on inches, and a schedule for wall thickness.

pipe of schedule 120, and figure 6.10 shows how the corresponding class is annotated with dimensions.

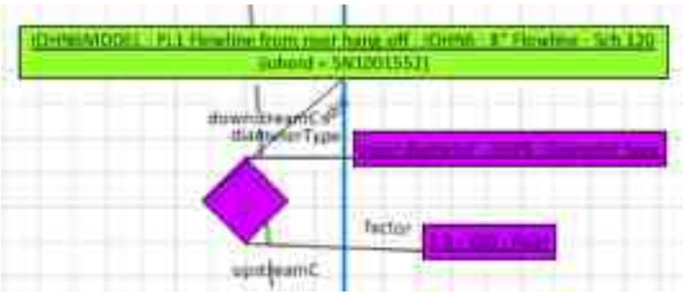


Figure 6.9: Example of pipe of a schedule class.

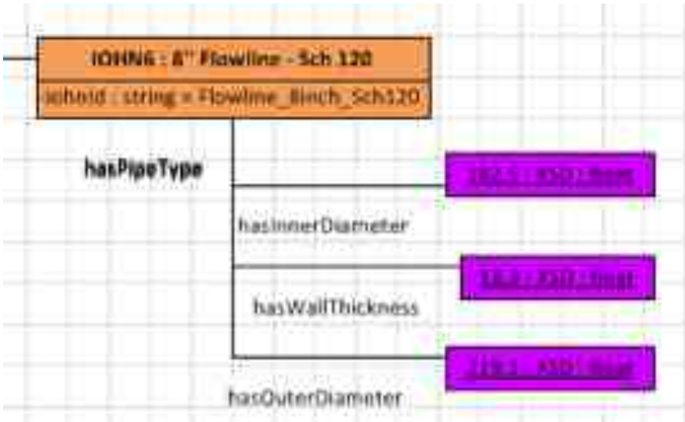


Figure 6.10: Example of pipe schedule class for which dimensions are annotated on class level.

For non-standard pipes, the dimensions are given as annotations directly on the pipe object, as shown in figure 6.11.

For estimating pipe erosion, it is important to know the radius of curvature for the pipe segments in which production fluid flows. Each pipe object in the Snorre B model is annotated with a radius of curvature, shown as purple diamonds in figures 6.9 and 6.11. The diamonds are interpreted as instances of the three-place template iohn6tpl:RadiusOfCurvature (see appendix 6.A.2.18). The template expresses the radius of curvature as a factor that is to be multiplied with either the inner or the outer diameter of the pipe. Thus, instances of the template have the pipe object as the first argument, the diameter type (inner or outer) as the second argument, and the factor as the third argument.

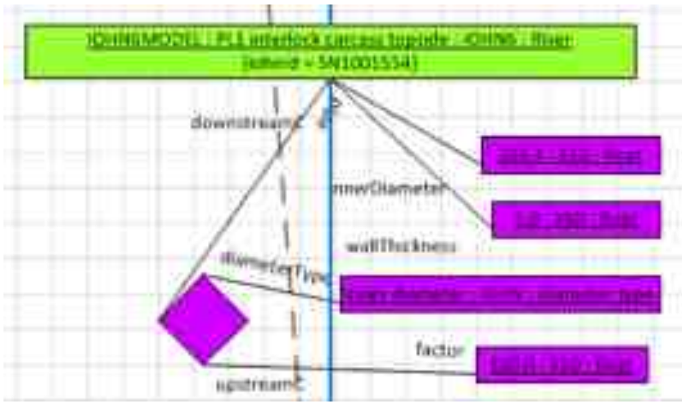


Figure 6.11: Example of pipe for which dimensions are annotated directly.

The model was created to demonstrate data exchange through the Integration Layer, and not necessarily to provide an exact representation of Snorre B. The model is based on 8 year old information and errors exist. However, its quality is more than sufficient to fulfill its purpose.

The topological model of Snorre B is one of the project's deliverables and it is referred to the Microsoft Visio file [1] for a complete overview.

6.4.3 Integration Layer

Figure 6.12 shows the client applications and data sources that interact through the Integration Layer in the Production Pilot. The Integration Layer components, i.e system architecture and reference data are shown as black boxes and it is referred to the IOHN activities Integration Platform and Semantic Model for a theoretical treatise of these components.

The three client applications used in the project are Epsis' Decide, ABB's ErosionInsight and DNV Erosion Monitor Application (DNV-EMA). Data from Siemens' Information Management System PIMAQ, EC's hydrocarbon accountant and Excel sheets have been used. In addition, topology and configuration information are available from the Snorre B model. Interactions between applications and the Integration Layer are done by SPARQL, which is a database query language for retrieving and posting data stored in Resource Description Framework format.

As mentioned in chapter 6.3.3, the original topology planned in the project involved a more distributed system where data sources would be hosted in environments local to dif-

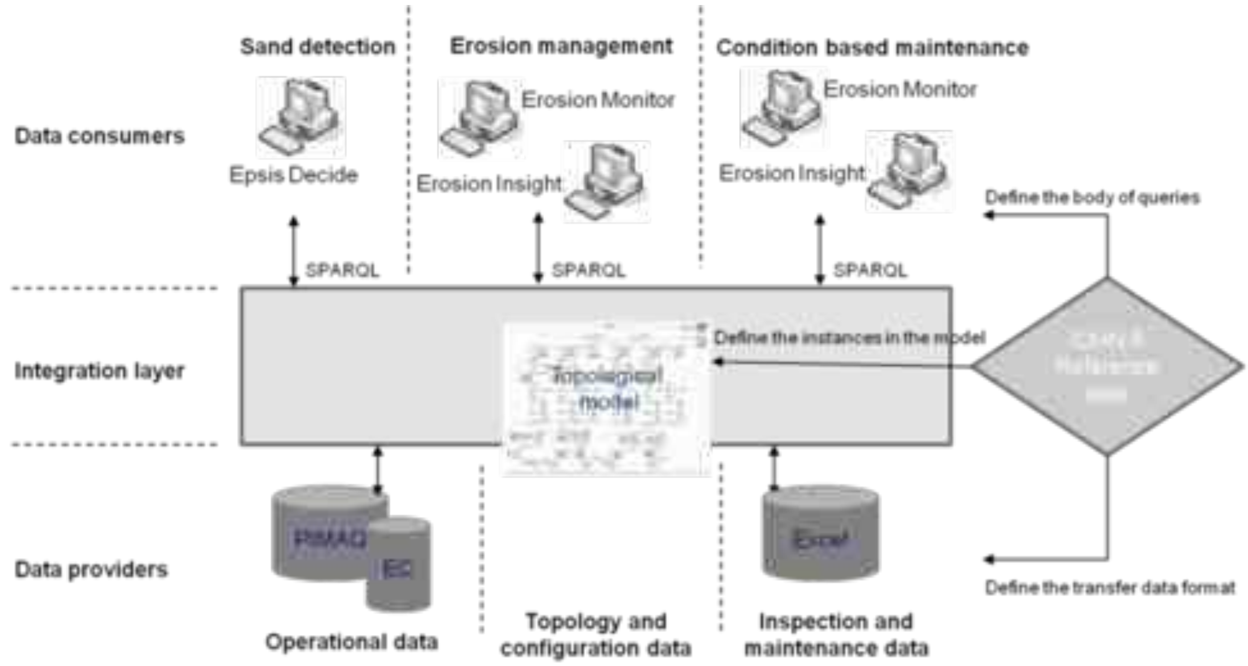


Figure 6.12: Interoperability between client applications and data providers

ferent providers, like for instance the IMS data available directly from the Siemens lab through SOIL. Due to changes in scope and course for the project, the data were copied to the Integration Layer instead of being routed. In future and real-world setups, there would be a need for having mechanics to re-direct requests to various vendors if needed. Duplicating large amounts of ” raw” data can lead to increased complexity of the architecture.

The representations of domain knowledge are collected in the nomenclature specifications, information templates and topological model of Snorre B described in chapter 6.4.2. These representations have been turned into semantic counterparts by IOHN-3, and its the latter representations that in reality enable information through the Integration Layer.

The cases are created to examine different aspects and functionality of the Integration Layer. In the Sand Detection Case the traditional separation between data providers and consumers are challenged. A potential sand event is detected in the PIMAQ system and made available through the Integration Layer. Epsis Decide, a prototype decision support tool, query the Integration Layer for sand events and pushes a sand event validation back. The validation can be queried by the PIMAQ system who uses this to update a corrected sand rate time series. In order to calculate sand rates, the PIMAQ system also needs to fetch data from EC to capture Well Test data. This means that PIMAQ can also be considered a client in this architecture. This specific functionality was not currently implemented in the IOHN project however.

The Erosion Management Case explores how applications requiring the same data can equally apply the Integration Layer, even though their implementation strategies differs. The applications are ABB’s ErosionInsight and DNV Erosion Monitor Application, which today access operational data from a database and Excel spreadsheets, respectively. In addition to operational data, the applications can access a subset of configuration data through the Integration Layer. These configuration data have previously been added manually.

An extension to the production related focus is provided in the Condition based Inspection and Maintenance Case, where inspection data from the Maintenance & Operation domain are made available to the erosion applications.

6.4.4 Interoperability within the Sand Detection Case

The Sand Detection Case is based on 2-way communication between Epsis Decide and Siemens' PIMAQ system, which is facilitated by the Integration Layer. The communication between the applications are tightly connected and are in the following therefore treated according to data flow rather then by application.

Both companies use dotNetRDF [3] as the API for posting SPARQL queries. This works well for SPARQL SELECT queries, which are queries that retrieve data. It is, however, more difficult with SPARQL UPDATE queries, which are queries that write information to the server. Both the Cambridge Semantics server and dotNetRDF API claim they support the SPARQL UPDATE protocol, but this involves using a more low level approach and raw query functionality in the API. SPARQL UPDATE queries can also be realized with standard HTTP support available within the Microsoft .NET framework.

6.4.4.1 Creating and querying a sand event

After a sand event is detected by the application layer in Siemens PIMAQ, as described in chapter 6.5.2, a SPARQL UPDATE query, similar to the one shown in figure 6.13 is posted to the Integration Layer.

```
PREFIX xsd:          <http://www.w3.org/2001/XMLSchema%23
PREFIX iohn6tpl:     <http://iohn.org/activity-6/tpl/>
PREFIX iohn6model: <http://iohn.org/activity-6/model/>
PREFIX detection:    <http://iohn.org/activity-6/detection/>

INSERT DATA {
    GRAPH <http://iohn.org/activity-6/detection> {
        detection:sandEvent_1001823 a iohn6tpl:SandEvent ;
        iohn6tpl:valSandEventId "1001823"^^xsd:integer ;
        iohn6tpl:hasDataSource iohn6model:SN1001773 ;
        iohn6tpl:valD ateTime
                  "2007-06-21T22:03:07.400"^^xsd:dateTime .
    }
}
```

Figure 6.13: SPARQL UPDATE query for creating a sand event

This request adds a new sand event at the integration server with the following properties:

- sandEventId - an unique identifier for the sand event, which in this case is *1001823*
- dataSource - a particular sand detector given by its identifier in the Snorre B model, namely *iohn6model:SN1001773*
- dataTime - the start time for the event

The current Integration Layer does not offer subscriptions to applications. This feature was considered outside of the project scope, and Epsis Decide queries instead the Integration Layer at regular intervals for new sand events. For the prototype implementation in the IOHN project this approach works well and removes unnecessary complexity from the software architecture.

When the query returns a new sand event the identifier for the sand detector is used to first query for the well the it belongs to, and further on to query the well for its attached data sensors. The last query is shown in figure 6.14.

```
prefix rdfs:          <http://www.w3.org/2000/01/rdf-schema#>
prefix iohn6model: <http://iohn.org/activity-6/model/>
prefix iohn6:         <http://iohn.org/activity-6/rdl/>

SELECT ?equipment ?type ?label ?comment
FROM <http://iohn.org/activity-6/model>
WHERE {
  <http://sws.ifi.uio.no/npd/well/34-7-P-9> a iohn6:Well ;
      iohn6model:hasEquipment ?equipment .

  ?equipment a ?type ;
        rdfs:label ?label ;
        rdfs:comment ?comment .

  ?type iohn6model:subClassOfTransitive iohn6:DataSource .
}
ORDER BY ?label
```

Figure 6.14: Query returning data sources for a particular well

The dotNetRDF returns the equipment in question as a XML, and by parsing this file the equipment's identifiers in the Snorre B model is retrieved. An extract of the XML is shown in figure 6.15.

```
<IOHN>
 <variables>
  <variable>equipment</variable>
  <variable>type</variable>
  <variable>label</variable>
  <variable>comment</variable>
 </variables>
 <results>
  <result>
   <binding name="comment">
    <literal lang="en">
     Measures = "Estimated pressure (Annulus between W & M)"
    </literal>
   </binding>
   <binding name="equipment">
    <uri>http://iohn.org/activity-6/model/SN1001777</uri>
   </binding>
   <binding name="label">
    <literal lang="en">
     P9 Annulus W/M pressure element
    </literal>
   </binding>
   <binding name="type">
    <uri>http://iohn.org/activity-6/rdl/PressureElement</uri>
   </binding>
  </result>
 <result>
```

Figure 6.15: An extract of the XML returned from the data source query.

Variables used in the query are given at the beginning of the XML file followed by a series of results. The first result is in this case a particular pressure element with its equipment identifier as given in the Snorre B model. In the next chapter this identifier is used to retrieve the timeseries related to the sensor.

6.4.4.2 Querying timeseries

The wells in the Snorre B model are connected to sand detectors providing both raw values and sand rates, pressure elements, temperature elements and a subsea choke. The measurements from these sensors are analog i.e. it is recorded as the data changes. The resolution is dynamic, and can be as low as 100 milliseconds. Siemens' Information Management System can therefore contain thousands of records for a single sensor for a single day. To keep queries simple and minimize the strain on the integration server it was queried for one measurement type at the time as shown in figure 6.16.

```
prefix iohn6:     <http://iohn.org/activity-6/rdl/>
prefix iohn6model: <http://iohn.org/activity-6/model/>
prefix iohn6tpl:   <http://iohn.org/activity-6/tpl/>
prefix iohn:      <http://iohn.org/rdl/>

SELECT ?dateTime ?value ?dataType ?unitOfMeasure ?aggregation
FROM <http://posccaesar.org/rdl>
FROM <http://iohn.org/activity-6/model>
FROM <http://iohn.org/rdl>
FROM <http://iohn.org/activity-6/tpl>
FROM <http://iohn.org/activity-6/rdl>
FROM <http://iohn.org/activity-6/data-sna>
WHERE {
  ?dataSourceValue a iohn6tpl:DataSourceValue ;
        iohn6tpl:hasDataSource iohn6model:SN1001774 ;
        iohn6tpl:hasDataType ?dataType ;
        iohn6tpl:hasUnitOfMeasure ?unitOfMeasure ;
        iohn6tpl:hasAggregationType ?aggregation ;
        iohn6tpl:valDateTime ?dateTime ;
        iohn6tpl:valDoubleValue ?value .

  FILTER (str(?dateTime) >= "2010-06-05T08"
       && str(?dateTime) < "2010-06-05T10")
}

ORDER BY ?dateTime
```

Figure 6.16: Query a particular sensor for its related timeserie

In table 6.1 and 6.2, the transfer times and sizes of the received results are given for 5 sensors. It is asked for data for periods of 2, 8 and 24 hours. Due to the dynamic nature of the measurements there are large differences in size. Raw sand values are highly variable while changes in stroke usually are kept at a minimum.

Dur (h)	Sand #1 (s)	Sand #2 (s)	Pres (s)	Temp (s)	Choke (s)
2	5.8	5.5	0.3	0.5	0.8
10	40.9	42.5	2.7	2.8	0.8
24	62.4	65.8	2.5	4.4	0.8

Table 6.1: Transfer times of retrieved query results

The sizes of the two series of raw sand values are nearly 7 Mb for 24 hours. A Comma Separated Value file containing time stamp, equipment tag and value will amount to approximately 1 Mb. This is not surprising since each measurement is encapsulated in its full semantic description represented as XML. The semantic description contains more information about every single measurement than is normally stored in a database. XML is also text-based and therefore larger than a binary representation of the same data.

Posting queries with large result sets appear to demand a lot of resources at the server. Tests with one to two weeks of data caused the server to time out several times and it was unable to deliver the requested output. The project's work around has been to post a series of queries with duration 24 hours.Scalability of XML is a limiting factor on the data transfer, and the returned results ought to be semantically annotated in another way.

Dur (h)	Sand #1 (kB)	Sand #2 (kB)	Pres (kB)	Temp (kB)	Choke (kB)
2	584	605	9	12	1
10	3 375	3 394	127	154	1
24	6 695	6 744	244	303	1

Table 6.2: Transfer sizes of retrieved query results

6.4.4.3 Updating sand event

After all operational data relevant to a sand event is retrieved an analysis of the sand event is performed. The analyses itself is described in chapter 6.5. The conclusion, represented by a boolean variable real sand = true/false, is uploaded to the integration server by a SPARQLUpdate request as shown in figure 6.17. Similarly the end time for the event can be posted by using the iohn6tpl:sandEventEnd template.

```
PREFIX xsd:            <http://www.w3.org/2001/XMLSchema>
PREFIX iohn6tpl:       <http://iohn.org/activity-6/tpl/>
PREFIX iohn6model:     <http://iohn.org/activity-6/model/>
PREFIX detection:      <http://iohn.org/activity-6/detection/>

INSERT DATA {
 GRAPH <http://iohn.org/activity-6/detection> {
  detection:sandEventProcessing_1001807 a
     iohn6tpl:SandEventProcessing ;
  iohn6tpl:valSandEventId "1001807"^^xsd:integer ;
  iohn6tpl:valIsTrueSandEvent "false"^^xsd:boolean .
 }
}
```

Figure 6.17: Posting a sand event conclusion

In the example query the conclusion is set to false. This information can now be accessed through the Integration Layer by others, and the PIMAQ system uses it to correct the calculated sand rate by setting this to zero. It can also perform a simple correction of the calibration tables as described further in 6.B.4.

6.4.5 Adapting DNV Erosion Monitor Application to the Integration Layer

The DNV Erosion Monitor Application (DNV-EMA) is an application written in Microsoft Excel and Visual Basic for Applications (VBA). For more info on DNV-EMA, see Appendix 6.E.

In order to retrieve and apply sensor data from the IOHN endpoint to the DNV-EMA calculations, a set of VBA classes was added to DNV-EMA's code base. These classes, hereafter ”the integration software”, consist of three classes.

- IohnCom - The integration interface. Used by DNV-EMA to retrieve IOHN data.
- ResultSet - A class used to format and append results.
- SPARQLEndpoint - The class that handles communication over HTTP with the IOHN endpoint.

While the ResultSet and SPARQLEndpoint classes are generic and may be used in future integration efforts, the IohnCom is written specifically for Snorre B, the DNV-EMA application and IOHN. It can not be used as-is in other projects, but may serve as a blueprint for other integration software for Microsoft Office products using Visual Basic for Applications. In particular it will be easy to employ the SPARQL class for querying SPARQL endpoints.

The goal of retrieving data from the IOHN endpoint and integrate it with the DNV-EMA application can be broken down into these tasks:

- Identify which data is needed by the application
- Locate what resources in the model holds these data
- Prepare queries to retrieve data from the identified resources
- Hand over the retrieved data to the application in a suitable format

6.4.5.1 DNV-EMA Data

The DNV-EMA application holds data for three types, or groups, of equipment. These are

- Well,

- Pipe (i.e. the connection from the well to the topside separator lines) and
- Topside separator line

For brevity, these three types or groups are referred to as *equipment types*. An instance of an equipment type, for instance a particular well, is called an *equipment*. Each equipment is associated with various sensors, and data from these sensors provide the input data to DNV-EMA.

As the DNV-EMA application is written in Excel, the data from the sensors can be found in Excel sheets of the application. Each data sheet in the application corresponds to a piece of equipment. Each row in a sheet holds measurements for each of the equipment sensors at a given time. The first column holds the date and time for the measurement, and each subsequent column holds the measurement for a sensor. The column layout is the same for each equipment type, but may vary between equipment types.

The DNV-EMA integration software passes on data to the DNV-EMA application by filling out these sheets with data from the Integration Layer. In that way, the DNV-EMA application can be completely agnostic to the fact that an integration operation has happened, and is guaranteed to function precisely as before.

There was no one-to-one correspondence between the DNV-EMA application and the data model with respect to names for the data sources. It was therefore necessary to use the integration model to identify the correct data sources. There are two possible methods of doing this. Using the integration layer, it is possible to create generic queries that can identify the various data sources given a point of origin. For instance, every data source associated with a particular well could be retrieved from the integration model given the model identifier for that well. The other option is to go through the integration model and identify each data source manually. As the automatic way of doing this is very taxing on the server, we chose to do it the manual way. It was also thought that as long as data is transfered through the integration layer, the mode of finding that information is less important.

The integration information, that is, the information pertaining particularly to Snorre B, was stored in the IohnCom class. That class is the only class a user of the integration software is in direct contact with.

6.4.5.2 Templates and queries

In order to limit the strain on the endpoint, it was decided that the queries should be kept as simple as possible. As a result, each query asks for values form one sensor only, instead of e.g. the whole set of sensors. Each query was of the form:

```
SELECT distinct ?dateTime ?value
FROM
<http://iohn.org/activity-6/data-snb>
{
?record
  iohn6tpl:hasAggregationType
?aggregationTypes ;
  iohn6tpl:hasDataSource
   ?dataSource ;
  iohn6tpl:hasDataType       ?dataType ;
  iohn6tpl:hasUnitOfMeasure
?unitOfMeasure ;
  iohn6tpl:valDateTime       ?dateTime ;
  iohn6tpl:valDoubleValue    ?value .
FILTER (?dateTime > ?startDate )
FILTER (?dateTime < ?stopDate )
}
ORDER BY ?dateTime
```

The values for aggregation type, data type, and unit of measure was retrieved from the integration model and entered in the IohnCom class along with the data source. Note that this query only gets a (date, value) pair for one data source. As each piece of equipment holds several data sources, this exercise had to be repeated for each data source.

The stop date and start date values are entered by the DNV-EMA application, and given as input values to the integration software.

6.4.5.3 Endpoint communication

The SPARQLEndpoint class handles querying to a SPARQL endpoint. To our knowledge, there is no SPARQL software written specifically for VBA. Instead we used the ServerXMLHTTP object that comes with MSXML 3.0 and later. This object provides methods and properties that enable the establishment of an HTTP connection between files or objects on different Web servers. In our case, we use this HTTP connection to communicate with the SPARQL endpoint.

6.4.5.3.1 Usage

Upon initialization of the class, the parameters needed for query are set:

- Prefixes used in queries
- The endpoint URL
- The endpoint username
- The endpoint password

To perform a SPARQL query using SPARQLEndpoint class, the SPARQLQuery function of that class is used. This function takes a query string as argument and returns a Variant. The Variant is a multidimensional array, where each row is a query match and each column denotes a variable. The order of the variables in the query dictate the order of the variables in the array. Very simple string manipulation is done to preserve double and date values. Any other data type is not handled specifically, and is just interpreted as string. If the query has no matches, the returned variant will have the VBA type Empty.

6.4.5.4 Formatting the answer

Each SPARQL query returns values for only one sensor pertaining to one piece of equipment. To deliver these data to the EM application, they have to be placed in the correct spreadsheet and in the correct column in that spreadsheet. In theory, this is achieved by querying the sensors in the order they appear in the spreadsheet, and append them sequentially to the same spreadsheet. In practice, some sensors may output results more often than others, and a naive approach would then display the sensor values with incorrect dates. This means that we have to sort the values according to their corresponding dates before we append them.

The class ResultSet was written to handle this problem. A ResultSet object will digest a two dimensional array of dates and sensor data, and append it to an internal array of dates and query answers. Subsequent arrays will upon request return a list containing all received query answers sorted by date in a VBA array. The order of the answers will be preserved.

If the input to a ResultSet object is an empty list, it is not skipped, but appended to the data sources list. This was done to preserve the order of the sensors for an item, whether that sensor have any data for the interval or not.

6.4.6 Integration with ABB ErosionInsight

The software product ErosionInsight from ABB is a commercial tool for erosion management. See appendix 6.F for details on ErosionInsight functionality.

On the software-technical side, ErosionInsight consists of three main components:

- A relational database (Oracle), storing configuration data, input and results from online erosion calculations.
- A calculation module, performing automated runs in batch mode to retrieve input data, perform calculations and store results.
- A desktop client GUI, which is a tool for configuration, visualization and decision support.

Both the calculation module and the GUI are implemented in Java. The data access from the Integration Layer is typically performed from the calculation module. There has been a strong focus on object-oriented code design to enable reuse of code and ease extensions; this has been valuable for the project work.

6.4.6.1 Architectural benefits with an Integration Layer

Appendix 6.F discusses the architectural benefits for ErosionInsight with the Integration Layer in detail. As seen by figure 6.69 in appendix 6.F , the project Prosty/EC access can be exchanged with the Integration Layer access. A SPARQL data- wrapper module fetches data from the Integration Layer via an endpoint by using SPARQL to query the RDF graphs. The data is stored in the ErosionInsight application database in sql tables similar to the ones in place today.

The benefits of this approach is that access to multiple data sources (there are currently four separate Prosty/EC databases that ErosionInsight relates to in Statoil, one for each of the fields Gullfaks, Statfjord, Snorre and Sleipner) could have been replaced with access to a single integration layer if this was available for all four fields. This would simplify the data access, even if the data access today is standardized by having the four different data sources implement a database view with the same format.

6.4.6.2 ErosionInsight data

As shown in figure 6.18, the data flow in ErosionInsight consists of five different groups of data:

1. Input: Production data, welltest data, event data
2. Input: Sand data
3. Input: Configuration data
4. Output: Model results
5. Output: Decision support

The input data (1 and 2) are retrieved from a data source (in Statoil this is an Oracle database delivered by Energy Components), while configuration data (3) are entered manually in ErosionInsight when the system in configured (a one-time operation.) One exception regarding (3) is choke type, these are configuration data which are retrieved automatically from the data source (EC). Output data (4 and 5) are written to the ErosionInsight database.

The focus in activity 6 has been to contribute to the specifications of a nomenclature and a model that enables parts of these data to be retrieved from an integration layer. The project result was that most of the data in (1) could be retrieved, but also examples of data in (2) and (3). Regarding (1), even if the examples below focus on production data, the model can also offer welltest data along with selected types of event data such as choke changes. Regarding (2), the model supports acoustic sand detectors, and currently not erosion probes which Snorre B has, but still it proves that the concept is in place. Regarding (3), configuration data like pipe diameters and curvature radius are available in the model.

As seen from figure 6.18, most of these input/output data are in ErosionInsight related to a well. However, from a physical point of view the measurement sensors are related to some other physical components, which again are (or can said to be) related to a well. A typical example is choke opening, which physically is related to a choke, which again is related to a pipeline which is related to a well. Data can also be related directly to pipelines.

6.4.6.3 ErosionInsight query examples

Today, SQL views on onsite sql databases are used as interfaces to the data. In this project these are exchanged with the Integration Layer endpoint and SPARQL queries. In this section we will have a look at three simple SQL queries and their equivalent SPARQL implementation.

All queries have the same input parameters: An object (well) to retrieve data for (identified by 'wellname'), and a timestamp (identified by 'proddate') for which we should retrieve a value. Here, the data retrieved are daily averages, so we just need a date, no time.

The sql queries below use this input to retrieve the three parameters oilrate, well head pressure (whp) and choke opening (chokeopen). Normally, these parameters would all be retrieved in one single sql query since they are all available from the same database view (v_erosion_prod).This is because the database view hides much on the complexity of the underlying database schema.

With SPARQL however we need to relate to the details in the model, and this of course introduces some pros and cons. The pros are definitely increased flexibility, as we can ask a platform for its wells and by that detecting any missing (e.g. added) wells in our manual configuration. We can also ask a well which data sources it has, which enables use of different logic depending on which data are available. The cons are some increased complexity at this point, as there are currently not defined anything similar to a database view for the SPARQL queries. This may change in the future, so complexity is reduced.

Note that in the examples below, the SPARQL queries do currently not limit to a single well, but will list all wells.

```
Example 1 - SQL
Select oilrate
From v erosion prod
Where proddate
= '01-01-2009' AND wellname = 'A-01'

Example 1 - SPARQL
prefix iohn6tpl:
<http://iohn.org/activity-6/tpl/>
SELECT ?well ?oilrate
FROM
<http://iohn.org/activity-6/data-snb>
```

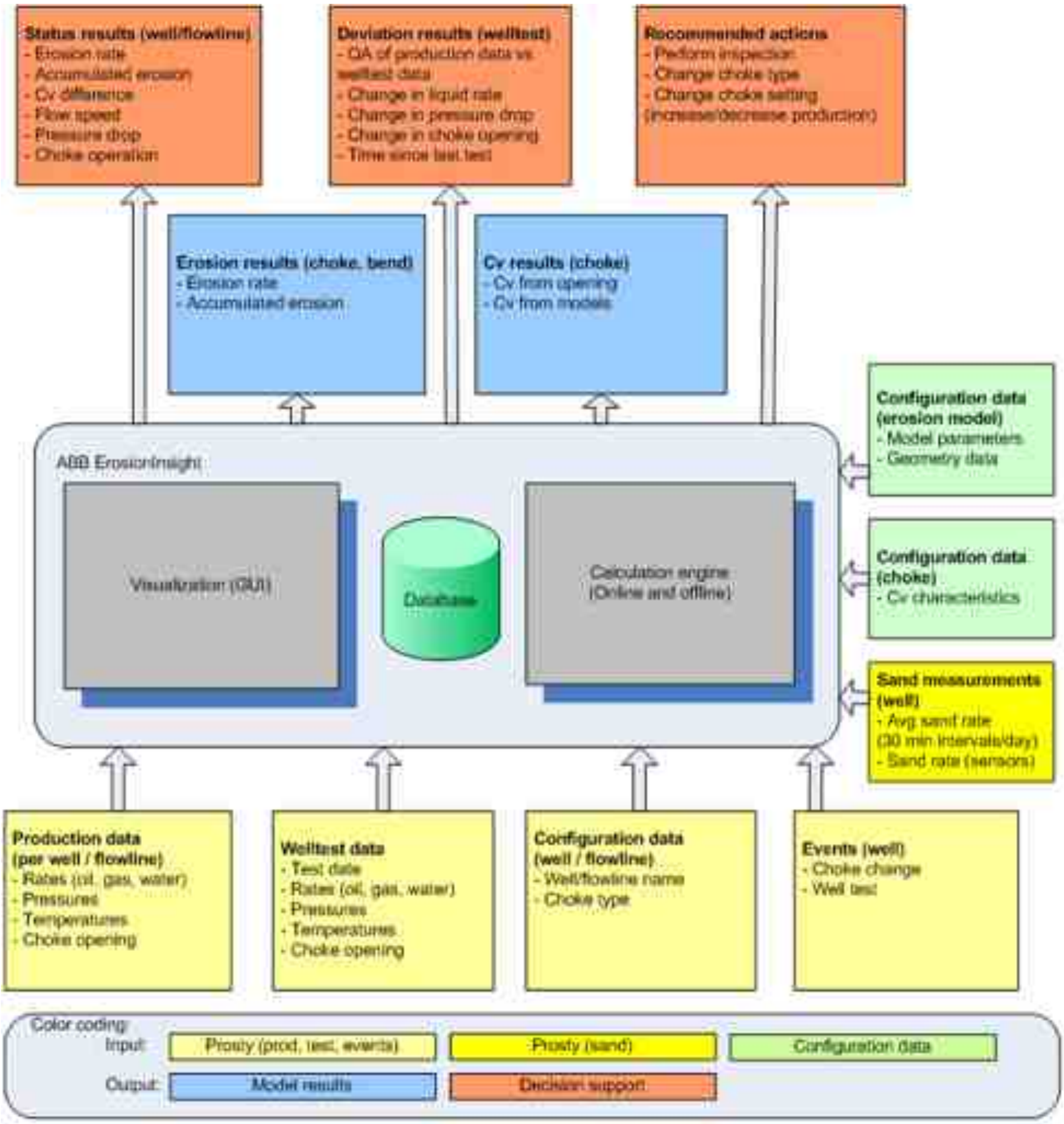


Figure 6.18: Data flow in ErosionInsight. Note that some minor detail may differ in the final IOHN implementation, such as the addition of inspection data (accumulated erosion) and configuration data (PVT parameters), but the figure still illustrates the main data flow.

```
FROM <http://iohn.org/activity-6/model>
WHERE {

   [] a iohn6tpl:ProductionData ;
     iohn6tpl:hasWell ?well ;
     iohn6tpl:valDateTime ?date ;
     iohn6tpl:valOilRate ?oilrate ;

  FILTER
(str(?date) = "2009-12-19T00:00:00")
}
ORDER BY ?oilrate ?well
Example 2 - SQL
Select whp
From v erosion prod
Where proddate
= '01-01-2009' AND wellname = 'A-01'

Example 2 - SPARQL
prefix iohn6tpl:
<http://iohn.org/activity-6/tpl/>
prefix iohn6model:
<http://iohn.org/activity-6/model/>
prefix iohn6:
   <http://iohn.org/activity-6/rdl/>

SELECT ?well ?whp
FROM
<http://iohn.org/activity-6/data-snb>
FROM <http://iohn.org/activity-6/model>
WHERE {
        ?well
iohn6model:hasChoke ?choke .

       ?choke
iohn6model:upstreamMeasuringDevice
?whpSensor .
              ?whpSensor
a iohn6:PressureElement .

  [] a iohn6tpl:DataSourceValue ;
     iohn6tpl:hasDataSource ?whpSensor ;
     iohn6tpl:valDateTime ?date ;
     iohn6tpl:valDoubleValue ?whp .

        FILTER
(str(?date) = "2009-12-19T00:00:00")

}
ORDER BY ?well ?whp



Example 3 - SQL
Select chokeopen
From v erosion prod
Where proddate
= '01-01-2009' AND wellname = 'A-01'

Example 3 - SPARQL
prefix iohn6tpl:
<http://iohn.org/activity-6/tpl/>
prefix iohn6model:
<http://iohn.org/activity-6/model/>
SELECT ?well ?chokeopen
FROM
<http://iohn.org/activity-6/data-snb>
FROM <http://iohn.org/activity-6/model>
WHERE {
        ?well
iohn6model:hasChoke ?choke .

        [] a iohn6tpl:DataSourceValue ;
     iohn6tpl:hasDataSource ?choke ;
     iohn6tpl:valDateTime ?date ;
     iohn6tpl:valDoubleValue
?chokeopen .

        FILTER
(str(?date) = "2009-01-01T00:00:00")

}
ORDER BY ?well ?chokeopen

Example 4 - SQL
Select
wellname, proddate, gor, oilrate,
gasrate, waterrate, wc, onstreamhours
From v erosion prod
Where proddate = '01-01-2009'

Example 4 - SPARQL
prefix iohn6tpl:
<http://iohn.org/activity-6/tpl/>
prefix iohn6model:
<http://iohn.org/activity-6/model/>
prefix iohn6:
   <http://iohn.org/activity-6/rdl/>
prefix iohn:     <http://iohn.org/rdl/>
prefix xsd:
     <http://www.w3.org/2001/XMLSchema#>
prefix rdfs:
<http://www.w3.org/2000/01/rdf-schema#>

SELECT
?well ?date ?oilrate ?wc ?gor ?gasrate
?waterrate ?onstreamhrs ?facility
FROM
<http://iohn.org/activity-6/data-snb>
FROM <http://iohn.org/activity-6/model>
WHERE {
  ?well
iohn6model:hasPlatform ?facility ;
       iohn6model:hasChoke ?choke .
```

```
  ?choke
iohn6model:upstreamMeasuringDevice
?whpSensor .
  ?whpSensor a iohn6:PressureElement .

  [] a iohn6tpl:ProductionData ;
     iohn6tpl:hasWell ?well ;
     iohn6tpl:valDateTime ?date ;
     iohn6tpl:valGasOilRatio ?gor ;
     iohn6tpl:valOilRate ?oilrate ;
     iohn6tpl:valGasRate ?gasrate ;
     iohn6tpl:valWaterRate ?waterrate ;
     iohn6tpl:valWaterCut ?wc ;
     iohn6tpl:valOnStreamHours
?onstreamhrs .

  FILTER
(str(?date) = "2009-12-19T00:00:00")
}
ORDER BY ?well ?date
```

6.4.6.4 User Experience and usability

One major difference comparing SQL to SPARQL can be seen in the where clause. In SQL you specify your property on a relation in the 'where' clause. In RDF triplets you need an implemented template for this. Thus, before specifying queries, you need to know which templates are in place on the server. It is however possible to query the server for this information, and also to define templates that could simplify the queries.

The initial impression is that the syntax for the three queries first in SPARQL should be quite similar since the SQL queries are similar. This was, however not the case. Performance wise there were also a big difference. We have not performed a full benchmarking of the performance so we can report the numbers, but the difference was quite notable.

The fourth example demonstrates that when retrieving multiple parameters, this does not extend the SPARQL query much if the data are found on the same object. But as the three previous queries show, retrieving the three parameters oilrate, wellheadpressure and chokeopening require that we access different parts of the model. It is however possible to access these different parts in the same query, they are just noted as separate queries here to illustrate where in the model the parameters are located.

6.4.6.5 Utilizing the Integration Layer in ErosionInsight

Since the internal data model in ErosionInsight has an object-oriented design, it is well suited for handling the data in the format provided by the SPARQL queries. Many of the modeled objects in the RDF model can also be found in the ErosionInsight object model. However, as for most object-oriented systems, storage is still based on a relational database, so an OR (Object-to-Relational) mapping is performed when the data are stored in the ErosionInsight Oracle database.

In order to test SPARQL and verify the concept, a SPARQL client has been developed in Java using the Jena open source software. This solution integrates well with the Java-based ErosionInsight calculation module. Instead of a direct data access using SPARQL from the calculation module, the architecture selected was to store the results from SPARQL to the ErosionInsight database, where they later can be utilized by the calculation module. This approach using an intermediate storage layer has proven very useful in a research project like IOHN since it allows for a detailed review of the SPARQL results before they are used in actual calculations. The client does not retrieve all data needed for a full erosion calculation (e.g. a well test), but enough to verify the retrieval concept.

Another benefit of this approach is that it allows combination with data retrieval also from other sources than the Integration Layer. This has been very useful as it allows quality assurance by comparing data from multiple sources, and also allows utilizing data from other sources if there are data in the Integration Layer which are missing or have insufficient quality. This kind of approach to verify an ensure data input quality is particularly important to an automated online application like ErosionInsight, since it enables the system to run in a stable manner also in an early phase of an Integration Layer implementation, where we can not assume data contents are fully complete.

6.4.6.6 Retrieving data from IBM's integration layer with ErosionInsight using web services

At the start of the IOHN project (2008 – 2010), a considerable amount of work was done by ABB to interface the ErosionInsight application with the IBM commercial integration layer named IIP (later renamed to IIC). This work resulted in two IOHN-presentations and demo sessions at conferences. The first at Intelligent Energy 2010 in Utrecht, The Netherlands in March 2010, and the second at Semantic Days in Stavanger, Norway in May 2010.

The main work items conducted by ABB within this scope were:

- Describe in detail to IBM which input data were needed by ErosionInsight
- As IBM preferred providing these data through a web service layer, ABB specified the needed design of these web services.
- The web services were tested manually through SOIL by ABB, and any quality problems with the retrieved data were reported and resolved.
- A demo application was developed by ABB in Java to consume these web services.
- Benchmarking was done with regards to performance of data retrieval.
- A presentation was given at the two conferences referred to above.
- A demo of data retrieval from the IIC integration layer, as well as a demo of ErosionInsight functionality that these data were used for, were given at the two conferences referred to above.

The value of this work was primarily to demonstrate the value of an integration layer approach, and to demonstrate the steps needed to achieve this. This provided useful experience for the further work on using SPARQL with a different integration layer which was started when IBM left IOHN.

Some of this experience was that the IBM approach using web services did not fully offer the required flexibility in accessing the model. While web services did give a performance benefit as compared to the later SPARQL approach, it proved difficult to develop a fully generic set of web services that would be relevant for all kinds of applications we could expect would access the integration layer. However, to meet specific application needs, the web service approach worked well.

The work was also valuable to prepare ErosionInsight data retrieval from an integration layer, which simplified the second step of integration using SPARQL.

6.4.6.7 Summary – experiences from a product point of view

Having an available Integration Layer which can provide all the input data an application would need, with sufficient quality and performance, is of course extremely positive to any software vendor. It enables the development resources to be spent on functionality and smart algorithms, and not on data retrieval. (However, familiarity with the model and competence on SPARQL would still be required in order to perform the queries, just as accessing a relational database requires familiarity with the database scheme and SQL.) This clearly demonstrates the value of the approach and of the intended goal.

With the current IOHN results we are not quite there yet, but well on the way. The developed RDF model currently contains much of the needed data (but not all). Regarding model usability this has been a learning process for both the modelers and the application vendors.

The main challenge is currently found on the performance side. However, this is to be expected, and the focus in IOHN has not been to develop a commercial integration layer package, but to utilize available technology to demonstrate concepts and model the data related to the sand and erosion domain as accurately as possible. In this respect the work has brought all participants a long step forwards.

6.4.7 Querying additional data

The chapters about adaption of the erosion applications focus mainly on the retrieval of operational data. Other kinds of data, like well test data and configuration data, are also retrieved by the applications and SPARQL queries are described in the following.

6.4.7.1 Well Test Data

Well tests are captured with the Templates iohn6tpl:WellTestEvent and iohn6tpl:WellTestValue. For each well test, there is an instance of iohn6tpl:WellTestEvent with the well on which the test was conducted, and the date of the test as parameters. Further, for each well test, there are instances of iohn6tpl:WellTestValue for each of the recorded parameters; gas/oil ratio, water cut ratio, oil rate, gas rate, water rate, well head pressure, opening/stroke of the well head choke, and the separator pressure. The following query will produce a table of all recorded well test data in the system.

Prefix	Namespace
xsd	http://www.w3.org/2001/XMLSchema#
rdfs	http://www.w3.org/2000/01/rdf-schema#
iohn6	http://iohn.org/activity-6/rdl/
iohn6model	http://iohn.org/activity-6/model/
iohn6tpl	http://iohn.org/activity-6/tpl/
iohn	http://iohn.org/rdl/

```
SELECT DISTINCT ?well ?date ?gor ?wc
  ?oilrate ?gasrate ?waterrate ?whp
  ?chokeopen ?seppressure
FROM
<http://iohn.org/activity-6/data-snb>
FROM <http://iohn.org/activity-6/model>
WHERE {
  ?well a iohn6:Well ;
        rdfs:label ?wellLabel .

  [] a iohn6tpl:WellTestEvent ;
     iohn6tpl:hasWell ?well ;
     iohn6tpl:valDateTime ?date .

  [] a iohn6tpl:WellTestValue ;
     iohn6tpl:hasWell ?well ;
     iohn6tpl:valDateTime ?date ;
     iohn6tpl:valDoubleValue ?gor ;
     iohn6tpl:hasDataType
       iohn:GasOilRatio .

  [] a iohn6tpl:WellTestValue ;
     iohn6tpl:hasWell ?well ;
     iohn6tpl:valDateTime ?date ;
     iohn6tpl:valDoubleValue ?wc ;
     iohn6tpl:hasDataType
       iohn:WaterCutRatio .

  [] a iohn6tpl:WellTestValue ;
     iohn6tpl:hasWell ?well ;
     iohn6tpl:valDateTime ?date ;
     iohn6tpl:valDoubleValue ?oilrate ;
     iohn6tpl:hasDataType
       iohn:OilVolumeFlowRate .

  [] a iohn6tpl:WellTestValue ;
     iohn6tpl:hasWell ?well ;
     iohn6tpl:valDateTime ?date ;
     iohn6tpl:valDoubleValue ?gasrate ;
     iohn6tpl:hasDataType
       iohn:GasVolumeFlowRate .

  [] a iohn6tpl:WellTestValue ;
     iohn6tpl:hasWell ?well ;
     iohn6tpl:valDateTime ?date ;
     iohn6tpl:valDoubleValue
?waterrate ;
     iohn6tpl:hasDataType
       iohn:WaterVolumeFlowRate .

  [] a iohn6tpl:WellTestValue ;
     iohn6tpl:hasWell ?well ;
     iohn6tpl:valDateTime ?date ;
     iohn6tpl:valDoubleValue ?whp ;
     iohn6tpl:hasDataType
       iohn:WellTestPressureWellHead .

  [] a iohn6tpl:WellTestValue ;
     iohn6tpl:hasWell ?well ;
     iohn6tpl:valDateTime ?date ;
     iohn6tpl:valDoubleValue
?chokeopen ;
     iohn6tpl:hasDataType
       iohn:WellTestStroke .

  [] a iohn6tpl:WellTestValue ;
     iohn6tpl:hasWell ?well ;
     iohn6tpl:valDateTime ?date ;
     iohn6tpl:valDoubleValue
?seppressure ;
     iohn6tpl:hasDataType
       iohn:WellTestPressureSeparator .
}
GROUP BY ?well ?date ?gor ?wc ?oilrate
  ?gasrate ?waterrate ?whp ?chokeopen
  ?seppressure
ORDER BY ?well ?date
```

6.4.7.2 Pipe Dimensions & Radius of Curvature

It is possible to query the Integration Layer for radius of curvature for each pipe object in the system. Below is a query that lists each pipe object, along with its dimensions and radius of curvature. In order to correctly retrieve pipe dimensions, one must distinguish between NPS standard pipes, for which dimensions are given as class annotations, and pipes for which dimensions are given as direct annotations on the pipe object. This is reflected in the query by a UNION of the NPS pipes (the first result set), and the non-NPS pipes (the second result set).

Prefix	Namespace
rdfs	http://www.w3.org/2000/01/rdf-schema#
owl	http://www.w3.org/2002/07/owl#
iohn6	http://iohn.org/activity-6/rdl/
iohn6model	http://iohn.org/activity-6/model/
iohn6tpl	http://iohn.org/activity-6/tpl/

```
SELECT DISTINCT ?fluidTransportDevice
  ?fluidTransportDeviceLabel
  ?fluidTransportDeviceType ?factor
  ?diameterType ?innerDiameter
  ?outerDiameter ?wallThickness
FROM <http://iohn.org/activity-6/model>
FROM <http://iohn.org/rdl>
FROM <http://iohn.org/activity-6/tpl>
FROM <http://iohn.org/activity-6/rdl>
WHERE {
 {
   [] a iohn6tpl:RadiusOfCurvature ;
      iohn6tpl:hasArtefact
        ?fluidTransportDevice ;
      iohn6tpl:hasDiameterType
        ?diameterType ;
      iohn6tpl:valFactor ?factor .

    ?fluidTransportDevice
      a ?fluidTransportDeviceType ;
      rdfs:label
        ?fluidTransportDeviceLabel .

    ?fluidTransportDeviceType
      iohn6model:subClassOfTransitive
        iohn6:ScheduleFlowline ;
      iohn6model:classHasInnerDiameter
        ?innerDiameter ;
      iohn6model:classHasOuterDiameter
        ?outerDiameter ;
      iohn6model:classHasWallThickness
        ?wallThickness .
 }
 UNION
 {
   [] a iohn6tpl:RadiusOfCurvature ;
      iohn6tpl:hasArtefact
        ?fluidTransportDevice ;
      iohn6tpl:hasDiameterType
        ?diameterType ;
      iohn6tpl:valFactor ?factor .

    ?fluidTransportDevice
      a ?fluidTransportDeviceType ;
      rdfs:label
        ?fluidTransportDeviceLabel .

    OPTIONAL { ?fluidTransportDevice
               iohn6model:hasInnerDiameter
               ?innerDiameter }

    OPTIONAL { ?fluidTransportDevice
               iohn6model:hasOuterDiameter
               ?outerDiameter }

    OPTIONAL { ?fluidTransportDevice
               iohn6model:hasWallThickness
               ?wallThickness }

    ?fluidTransportDeviceType
       rdfs:subClassOf ?supertype .
    FILTER (?supertype !=
            iohn6:ScheduleFlowline) .

    FILTER (BOUND(?innerDiameter) ||
            BOUND(?outerDiameter) ||
            BOUND(?wallThickness))

 }
}
ORDER BY ?fluidTransportDeviceLabel
```

6.4.8 Integration with maintenance data

Data are one of the major challenges in the oil industry today. It is difficult to get an overview over what data that exist and where and how it is stored. By making both production and maintenance data available through the Integration Layer it becomes a domain independent data interface. Queries for retrieving choke change events and inspection data are given in chapter 6.4.8.1 and 6.4.8.2, respectively. Information about choke changes are currently used in erosion calculations, while inspection data are new information.

> The Integration Layer- a domain independent data interface

6.4.8.1 Choke change events

The Integration Layer provides information on what choke types are installed at each choke position, and the time the choke was last changed. This information is expressed as instances of the template iohn6tpl:ChokeChangeEvent, and stored in the graph http://iohn.org/activity-6/data-snb. Below is a query that lists choke change events for the chokes on the Snorre B platform. Note that due to limited resources, it was decided to represent choke types as strings, rather than a more time-consuming proper semantic modelling of a hierarchy of choke types.

Prefix	Namespace
rdfs	http://www.w3.org/2000/01/rdf-schema#
iohn6	http://iohn.org/activity-6/rdl/
iohn6model	http://iohn.org/activity-6/model/
iohn6tpl	http://iohn.org/activity-6/tpl/

```
SELECT ?choke ?chokeLabel
  ?dateTime ?chokeType
FROM <http://iohn.org/activity-6/model>
FROM
<http://iohn.org/activity-6/data-snb>
WHERE {
  [] a iohn6tpl:ChokeChangeEvent ;
     iohn6tpl:hasChoke ?choke ;
     iohn6tpl:valDateTime ?dateTime ;
     iohn6tpl:valChokeType ?chokeType .

  ?choke rdfs:label ?chokeLabel .
}
ORDER BY ?chokeLabel
```

6.4.8.2 Inspection events

The template nomenclature provides a template iohn6tpl:InspectionEvent to record the wall thickness of a pipe during a pipe inspection. The template takes four arguments; a pipe object, a time stamp (recording when the inspection was made), a wall thickness measurement (in mm), and a uncertainty measure (in mm). For a pipe inspection with measured wall thickness X and uncertainty Y, the actual wall thickness is to be interpredted as X +/- Y. Note that due to lack of proper inspection data, the Integration Layer contains no inspection events. However, if inspection events were present, they could be queried as follows.

Prefix	Namespace
rdfs	http://www.w3.org/2000/01/rdf-schema#
iohn6	http://iohn.org/activity-6/rdl/
iohn6model	http://iohn.org/activity-6/model/
iohn6tpl	http://iohn.org/activity-6/tpl/

```
SELECT ?pipe ?pipeLabel ?dateTime
  ?thickness ?uncertainty
FROM <http://iohn.org/activity-6/model>
FROM
<http://iohn.org/activity-6/data-snb>
WHERE {
  [] a iohn6tpl:InspectionEvent ;
     iohn6tpl:hasChoke ?pipe ;
     iohn6tpl:valDateTime ?dateTime ;
     iohn6tpl:valThickness ?thickness ;
     iohn6tpl:valUncertainty
?uncertainty .

  ?pipe rdfs:label ?pipeLabel .
}
ORDER BY ?pipeLabel
```

6.4.9 Discussion

The project's work with data integration shows that there is a large unutilized potential in improved interoperability. However, the processes of providing domain information and turning it into semantic representations are resource demanding and time consuming. The question is whether or not one large integration framework is cost efficient

ISO 15926 Oil and Gas Ontology is an overlying framework. The nature of standardization requires that information already within the framework is static. Users of the framework must therefore create their own realization of information and link this to the appropriate classes in the standard. The nomenclature and templates developed in this project is an example of such a local realization. The obvious benefit of this approach is that local modifications can be included without changing the standard, but for it to become an industry standard it requires that a company or other entity owns and maintains the realization. Companies asking vendors for ISO-15926 Oil and Gas Ontology compliance should therefore supply the relevant local representations. Unfortunately, the lack of clear ownership in research projects like the IOHN makes it difficult to reuse these results effortlessly.

The nomenclature and templates that have been developed in the project are of high quality, and have been used by four different applications to successfully retrieve data through the Integration Layer. These data include production time series, equipment information, events and inspection data. The platform model of Snorre B covers the project's requirements, but must be expanded and quality assured before it is suited for industry use. The only limitation on performance encountered in the current implementation of the Integration Layer is on data transfer rates, which ought to be improved.

The greatest benefit for the participants in the Production Pilot has probably been the learning experience. Familiarization in working with standards, nomenclatures, templates and models will be a great asset in future projects. In this process the participants have also uncovered several areas where interoperability can have a severe impact. Of the most prominent are enabling applications to extract all configuration data from the model and retrieval of a priori validated data from the Integration Layer.

There is undoubtedly a potential within data integration, but the question of benefits vs costs is still unanswered. We would like to suggest further investigations and preferably in projects where ownership of local representations are clearly defined.

6.5 Sand detection Case

The production of sand and solids is a common challenge and may constitute severe safety, environmental and economic risks. Good measurements of produced sand volumes are crucial in ensuring facility integrity and avoiding accidents. All new wells are therefore either equipped with acoustic sand detectors or erosion probes. These sensors are in most cases linked to alarms in the control room, which are set to go off if predefined threshold values on sand rates are exceeded. However, sand measurements can be influence by several factors and alarms can often be sounded erroneously, which lowers the trust and confidence of the control room operators in these warnings. To fully understand variations in sand measurements and improve the quality of alarms other operational data types, like pressure and choke opening, must also be examined.

In the Sand Detection Case a methodology for the following is suggested.

- detection of significant variations or events in sand measurements
- analyze whether the events are caused by sand or other factors
- maintain a set of calibrated sand measurement incorporating the results from the analysis

The analysis requires data from different data providers, and it is therefore a good case for testing the Integration Layer. The work related to interoperability and the Integration Layer is described in chapter 6.4.4, while the reminder of this chapter is devoted to sand detection. In the chapters 6.5.1 and 6.5.2 calculations of sand rates and detection of events are described. Chapter 6.5.3 provides an overview of identified categories of sand events and the actual examples used for analysis. The analysis methodology and results are presented in chapter 6.5.4, and how the results are used to maintain calibrated sand measurements is described in chapter 6.5.5.

The two systems applied in the Sand Detection Case are Epsis Decide and Siemens' PIMAQ Information Management System. Epsis Decide is a prototype implementation developed for the IOHN project to analyze sand events, while PIMAQ is both a provider of measured data and in charge of calculating sand values, detecting events and maintaining the calibrated results. The latter is described in more detail in appendix 6.B.

6.5.1 Sand rate calculations

To calculate the actual sand production for facilities can be a challenging task. In the case of acoustic sand sensors, calibration of the so-called ”zero-level” needs to be maintained to produce an accurate value. This calibration table describes the standard noise level at different fluid velocities. One of the motivations for implementing calculations in the IMS system, was in order to have a centralized system for all different sand sensors. When storing the calibration tables at the collecting data source, one can generalize and apply calculation to all sensors in a uniform manner, in contrast to having each sensor vendor implementations on stand-alone systems.

In order to calculate sand rates and provide the most correct estimation as possible; a centralized, autonomous approach was suggested. The Siemens Information Management System (PIMAQ) would calculate this value, do a detection and as a result, create an event that will be updated in the integration layer. This would be used to alert the decision support system (see 6.4.4).

The way the calculation formula is implemented in the PIMAQ system, inputs to functions can be changed based on the availability of data from other sources. This can be values like flow measurements, velocity estimates or other values calculated or allocated by different data providers.

The following values are calculated in the current implementation, and is available for general consumption by other data consumers:

- sand rate $[g/s]$
- sand rate with possible corrections $[g/s]$
- fluid velocity $[m/s]$
- flow rate $[m^3/day]$

Note that two sand rate time series are stored. One that was originally calculated based on the current parameters, the calibration tables and other dependent data. A second series exists that originally contains these calculated sand rates. The difference is if correction events has been stored in the Integration Layer as a result of the decision support tool.

In order to calculate sand production rates, the following external data is needed:

- Well test data polynomial
- Initial calibration tables for the sand sensors
- Piping inner diameter

The calibration tables are based on the formula:

$$SandRate = \left(\frac{Raw - Zero(v)}{Step(v)}\right)^{Exp(v)}$$

This implies that the velocity also needs to be calculated. This is done with the following formula:

$$Velocity = \frac{G' + O + W}{(24 \times 3600) \times \left(\frac{\pi \times d^2}{4 \times 1000000}\right)}$$

The flow is implemented using the polynomial:

$$c_2 \times y^2 + c_1 \times y + (c_0 - x) = 0$$

The polynomial is then required as a result from the Well tests. These are currently available from the EC data present in the integration layer. The current implementation does not read well test data from the integration layer as original planned, but it is possible to manipulate the well test polynomials in the GUI of the PIMAQ Application. In future implementations, the table can easily be populated from an external data source instead. Chapter 6.B.3 describes the calculation formulas in more detail.

The IMS system will perform a periodic, scheduled calculation where all the current raw data from the sensors and other collected data like pressure and temperature will be used to find the estimated sand rate. The schedule interval is configurable, but can typically be done every 5 or up to 60 minutes, based on how fast a sand detection event is to be created and acted upon. After performing the sand rate calculation, the detection algorithm for sand rates runs and produce potential new sand event results, as exemplified in figure 6.19.

6.5.2 Automatic detection

In the Siemens IMS system, PIMAQ, there are implemented a number of new components to handle the Sand Detection Case. The PIMAQ calculations and the applications have been used in a test environment using existing data from the Snorre A and Snorre B platforms.

When an IMS data source has data available from all sand sensors, detection can be done across several systems with fitted sensors. The current implementation uses the calculated sand rate and not the raw values from the sensors to detect sand events. The detection is done by first doing a temporary low-pass filter on the current data to eliminate noise and spikes. Then, candidate intervals that contains values that exceeds a certain threshold in the calculated sand rate for a minimum time window are stored. A post-processing step will combine the candidate intervals into larger intervals if they are close to each other in time, and remove singular smaller events. The parameters for the low pass filters, time windows and criteria for combining close intervals are adjustable, and further described in appendix 6.B.

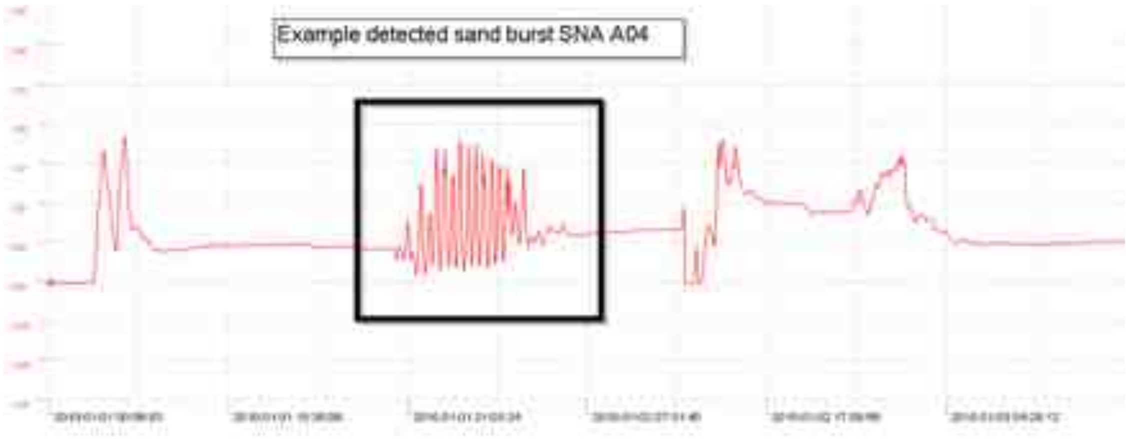


Figure 6.19: Sand detection example from calculated sand rate

The algorithm is currently tested running four times in one hour. This means that the detection is somewhat slower than an alarm at an HMI operator station. The nature of sand production means that it may ramp up more gradually. The scheduling interval can be re-configured to detect events quicker, thereby decreasing the time taken before the notification is actually stored in the integration layer and notified via other channels such as e-mail. Through the PIMAQ Application Server, it is possible to review and adjust these functions through a Web interface.

The current implementation is based on calculated sand rate, which could be misleading due to possible errors in the calibration tables. This is a potential for application improvements in the future. The challenge lies in defining normal and stable periods. This will be subject for a possible future project. The applications and algorithms implemented for automatic detection is described more in chapter 6.B.2.1.

During the course of implementation of the detection algorithm, the uncertainty of what thresholds to use for sand detection was considered to be a challenge. As described in 6.B.3.4, there are a number of different parameters that determines the actual intervals returned from the function. However, the most important thing is that the operator is given a rough idea of the interval. The decision support system can be used to further define the interval manually and adjust the start and end times of the event if necessary.

6.5.3 Sand events

An acoustic sand detector measures noise, but there is no absolute guarantee that the noise originates from sand particles. Even though the vendors of sand detectors have done a great effort in isolating the frequencies where noise from sand is most prominent, the noise can be caused by several other reasons. To identify the true origin of the noise more data and preferable all related production data must be taken into account. Domain experts from Statoil participated in the conceptual discussions, and some of the feedback from one of the workshops are shown in figure 6.21. The initial guesses were made by Epsis and Siemens, and the feedback is from Statoil. Based on this work the following 6 categories of sand events where identified:

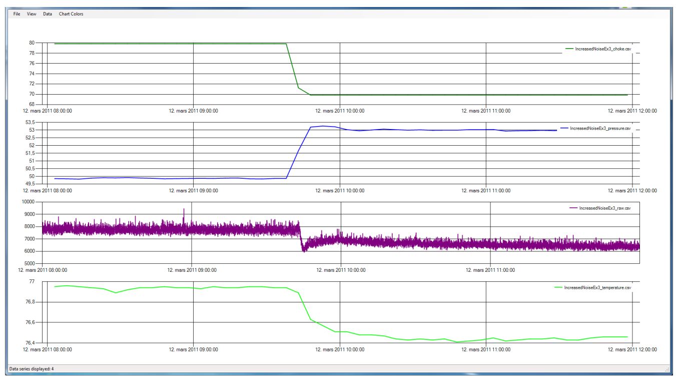


Figure 6.20: An example of a decreasing noise level due to reduction in choke opening.

- Proportional noise reaction to choke changes
- Inverse proportional noise reaction to choke changes

Well	Start Time	Stop Time	Initial Guess	Feedback
P-04B	*31.01.2010 14:32:25*	*31.01.2010 15:00:52*	*Choke opening?*	Choke opening.
P-04B	*16.02.2010 20:08:24*	*16.02.2010 20:47:58*	*Choke opening?*	Pressure ramp up, possible phase change.
P-04B	*16.02.2010 23:26:00*	*16.02.2010 23:35:55*	*Choke opening?*	Possible choke opening.
P-09A	*17.04.2010 00:01:30*	*17.04.2010 01:50:57*	*Real sand production?*	Slugging, initial wrong calibration.
P-09A	*28.05.2010 17:37:29*	*28.05.2010 18:28:08*	*Real sand production?*	Unstable production during startup.
P-09A	*29.05.2010 02:43:52*	*29.05.2010 04:10:20*	*Real sand production?*	Sand production.
P-09A	*02.06.2010 02:21:48*	*02.06.2010 21:24:48*	*Real sand production or choke increase?*	Combination - startup and some production.

Figure 6.21: Detected sand events with initial guesses and feedback

- Water breakthrough
- Slugging
- Actual sand production
- Normal production with minimal sand

Proportional noise reaction to choke changes means that as the choke is opened, more fluids flow and the noise increases, while when it is closed the flow is decreased and so is the noise. The noise is in other words proportional with the energy of the flow. In figure 6.20 an example of a proportional choke reaction is given. The choke opening is decreased, which results in an increase in pressure, and decreasing noise and temperature.

The inverse proportionality can occur as an increased noise level even though the choke opening is decreased. This can for instance be caused by changes in flow direction, and that the flow hits the pipe wall in a more direct angle

When water breaks through into the well the fluid composition is dramatically changed and most likely causing an increase in the noise level. Sand has often an affinity to water, and it is therefore also common that some sand is pulled along in a water breakthrough.

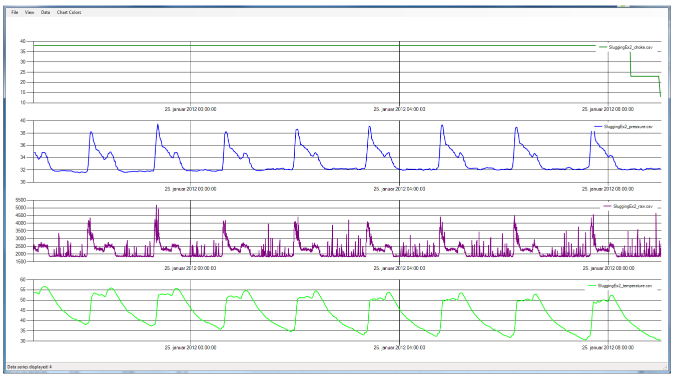


Figure 6.22: An example of a slugging well

Slugging means that a well's flow regime allows heavier particles to settle at some places in the pipes. Pressure increases as masses gather and the pipe is clogged. When the pressure reaches a dislocation level it is high enough to drive the clog, like a slug, through the production facility. In cases with high pressure buildups slugging may also constitute a severe safety hazard.

Sand production is the event when sand is actually produced. Most wells produce some sand, but when the levels get high so does the noise. Sand can also come in bursts when the well is suddenly exposed to sand loosening from the reservoir. This is dependent on whether the consolidation of sediments in the reservoir is loose or firm. In figure 6.23 continuously sand is shown with a potential burst around 06 AM.

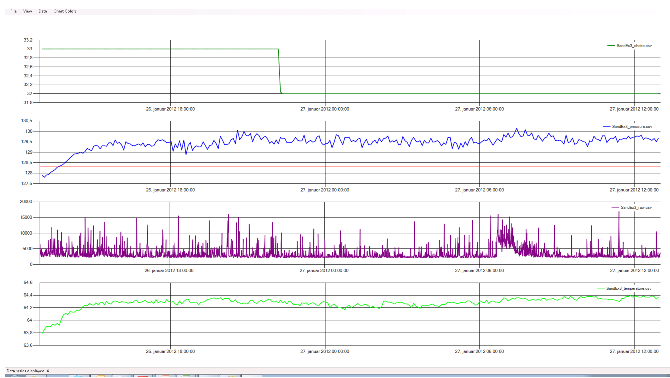


Figure 6.23: An example of real sand, with the sand measurements in the third row

It has been project policy to disturb the operational environments as little as possible. This decision made the task of collecting examples of sand events challenging as considerable experience in interpreting sand data was lost to the project.

In the category of proportional noise effect due to choke changes 8 good examples were identified. No examples where however found show-

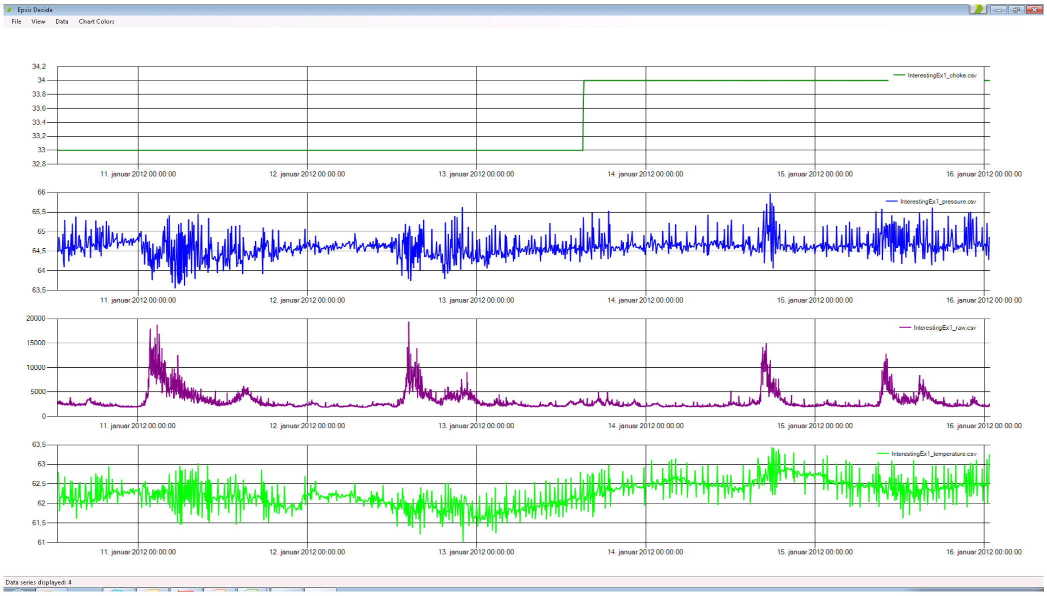


Figure 6.24: Sand burst or slugging?

ing inverse proportional noise effect due to choke changes none examples were found, and this category was removed. The third category was water breakthrough, which is expected to show in temperature. However, rates are needed to indicate when water breakthroughs have occurred and this information was unfortunately unavailable. The third category was therefore also removed. 3 good examples of slugging were detected. Real sand production seems to appear in two different forms, namely bursts and continuous, and the 5 identified examples should perhaps have been divided into two separate categories. There are 4 examples from the final category, normal production with little sand. This category works as a reference to how much variation that is normal for the different operational data when the production system is left alone. The last example, shown in figure 6.24, is particularly interesting as it is uncertain whether it is caused by slugging or a series of sand burst. An overview of the 21 sand event examples is given in figure 6.25.

6.5.4 Sand detection analysis

The analysis is performed using Machine Learning [9], which is a branch of Artificial Intelligence. By providing a set of cases of how different data types behave during different circumstances a machine can evaluate new events, give predications to the cause, and learn and improve on its prediction capabilities.

The data's characteristics used in the learning schema are described in chapter 6.5.4.1, and in chapter 6.5.4.2 the training and testing of the machine are discussed.

6.5.4.1 Data characteristics

The characteristics of timeseries are most often related to trends and noise. Trends can for instance be fitted as polynomials, exponential functions or trigonometric (periodic) functions, while noise are regarded as stochastic or random and represented by a statistical distribution.

Choke changes seem to introduce a relatively linear reaction in the other timeseries, and an Ordinary Least Square (OLS) method is used to fit a straight line. The results are given in figure 6.25, where β_0 is the intersection and β_1 is the slope. The choke change itself could probably be more precisely represented as a discrete step from one plateau level to another, but the straight line is used for simplicity.

Another prominent characteristic is the periodicity in the slugging examples. The sand measurements have been Fourier transformed to investigate whether any underlying frequencies can be detected. In figure 6.26 the power vs fre-

			Noise			Pressure		Choke		Temperature	
	#	Duration	Beta0	Beta1	Period	Beta0	Beta1	Beta0	Beta1	Beta0	Beta1
Proportional	1	8	10912	-0,206	Inf	44,5	0,0004	105,5	-0,0018	77,7	-0,0001
	2	8	5903	0,080	Inf	53,3	-0,0001	67,6	0,0005	76,2	0,0000
	3	4	8014	-0,131	Inf	49,4	0,0003	81,2	-0,0010	77,0	-0,0001
	4	4	4236	-0,092	Inf	19,4	0,0005	48,1	-0,0010	88,9	-0,0005
	5	3	2643	0,233	Inf	27,6	-0,0010	34,7	0,0014	87,9	0,0005
	6	6	4425	0,139	Inf	18,8	-0,0003	40,1	0,0020	88,6	0,0000
	7	12	1765	0,128	Inf	38,4	-0,0007	0,0	0,0026	61,1	0,0009
	8	4	2636	0,313	Inf	38,6	-0,0013	14,1	0,0011	68,5	0,0008
Slugging	1	5	5568	-0,005	948	48,3	0,0000	83,7	0,0000	77,2	0,0000
	2	12	2232	-0,003	5395	33,3	0,0000	39,4	-0,0001	48,5	-0,0002
	3	12	7152	-0,011	3595	32,5	0,0000	100,0	0,0000	54,6	0,0000
Sand	1	48	3259	0,000	Inf	34,3	0,0001	13,3	0,0001	53,4	0,0001
	2	48	2704	0,000	Inf	58,0	0,0000	21,5	0,0000	66,0	0,0000
	3	24	3572	-0,004	43159	129,1	0,0000	33,0	0,0000	64,2	0,0000
	4	8	4478	0,250	Inf	139,3	-0,0011	29,2	0,0003	61,8	0,0001
	5	2	4949	0,070	171	111,9	0,0000	37,5	0,0000	63,0	0,0000
Normal	1	48	3025	-0,001	Inf	12,9	0,0000	0,0	0,0000	8,4	0,0000
	2	20	4726	0,002	Inf	18,1	0,0000	58,1	0,0000	59,7	0,0000
	3	22	1888	0,000	Inf	64,1	0,0000	31,1	0,0000	61,0	0,0000
	4	14	1880	0,000	709	13,0	0,0000	0,0	0,0000	7,0	0,0000
Unknown	1	108	3490	-0,002	237590	64,5	0,0000	32,7	0,0000	62,0	0,0000

Figure 6.25: Properties of the data in the sand events

quency is plotted, and in this example the primary peak is at approximately 5400 seconds/-cycle. A new pressure buildup occurs in other words every 90 minutes. Pressure and temperature also experience periodicities in the slugging cases, but these characteristics have not been included to keep the total number of characteristics down.

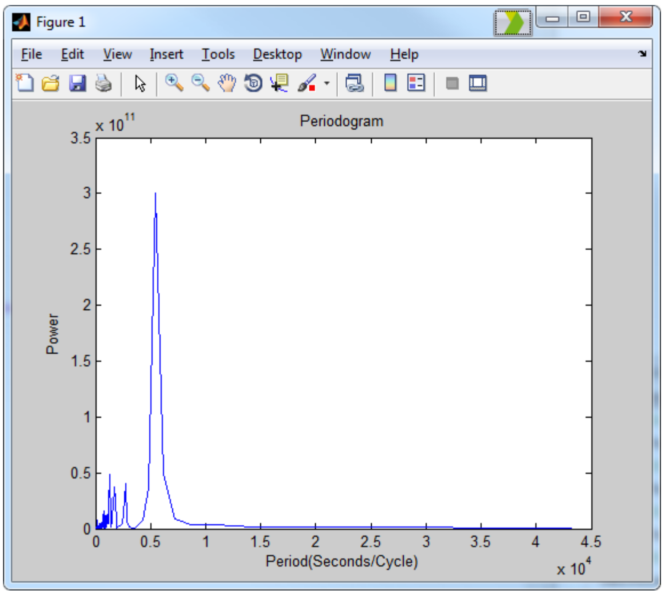


Figure 6.26: Detecting periodicity in measurements

Little effort has been made to fit measurement noise to a statistical distribution. It is assumed that the measurements' noise distribution are independent of sand event category. This is not necessarily true and particularly if the sand measurements are influenced by noise from external sources.

6.5.4.2 Training and testing the machine

The learning scheme used in this work is a Naive Bayes Classifier [14]. It is a simple probabilistic classifier based on Bayes' rule and it naively assumes independence between the data characteristics. The assumption of independence is seldom found in real life, and the intercept and slope of a fitted straight line are, for instance, clearly dependent. However, the Naive Bayes has turned out to work very well when tested on actual datasets. The project has used the Naive Bayes Classifier implemented in WEKA [4], which is a collection of machine learning algorithms implemented at the University of Waikato, New Zealand.

17 of the sand events displayed in figure 6.25 were used to train the learning scheme, while the remaining 4 were used to test it. The sand events used for testing were "proportional #3", "slug-

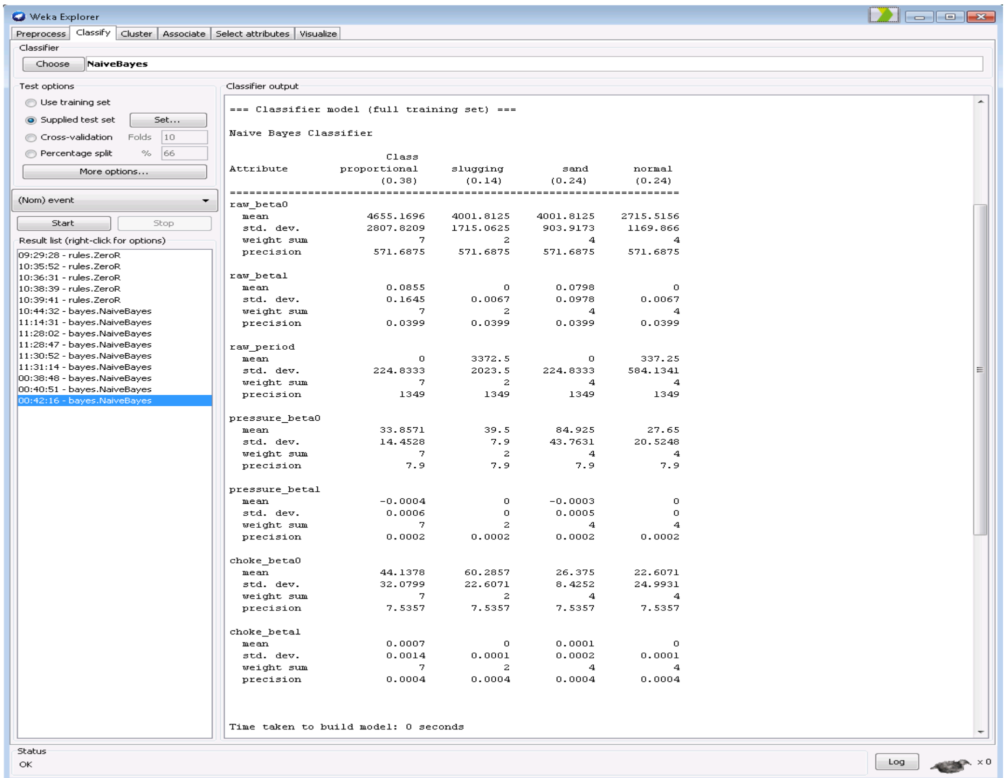


Figure 6.27: The statistics of the Naive Bayes Classifier applied to the training set

ging #3", "sand #3" and the unknown. In figure 6.27 the statistics of the Naive Bayes Classifier applied to the training are shown, and the classification of training examples to categories is shown in figure 6.28

=== Confusion Matrix ===
a b c d <-- classified as
8 0 0 0 | a = proportional
0 3 0 0 | b = slugging
0 0 3 2 | c = sand
0 0 0 4 | d = normal

Figure 6.28: Classification of sand events used for training.

The learning scheme classifies 2 of the 5 training events originally categorized as sand as normal. These events could of course be wrongly classified initially, but it is more likely that characterization by a straight line is insufficient to separate these two categories. All the parameters are in fact fairly equal with the exception of pressure. The 3 events classified as sand have a significantly larger pressure than the other categories. This is probably related to differences in the production conditions of the wells rather than a large pressure increase due to sand.

In figure 6.29 a similar classification matrix is shown for the test cases. Initially all test cases were categorized as normal, and the 3 examples gathered from the proportional, slugging and sand categories are all classified accordingly. The unknown case are classified as sand.

=== Confusion Matrix ===
a b c d <-- classified as
0 0 0 0 | a = proportional
0 0 0 0 | b = slugging
0 0 0 0 | c = sand
1 1 2 0 | d = normal

Figure 6.29: Classification of tested sand events.

The operational measurements are all related to the same physical system. The occurrences of similar events are therefore expected to give similar reactions in the measurements. Sand detection or sand event classification is therefore a good candidate for machine learning, and the Production Pilot has illustrated how such a methodology can be applied. To create a fully functional learning scheme more data and domain expertise are needed to improve upon the description of categories and data characterization.

6.5.5 Corrected calculated data

As described in 6.5.1, the result of the analysis from the decision support system is used to correct both calculated data, and to improve the calibration tables for the acoustic sand sensors. This will be triggered implicitly and manually by a operator by interacting with the decision support system. When receiving a negative case of sand detection, the production will then set the corrected production to zero and calibrate the sensor tables accordingly. Even though the actual production seldom is a flat zero, an ideal situation would be if the values are re-calculated based on the new settings. This would capture the underlying fluctuations based on the new calibration, even though this can be considered as noise. In meetings with Statoil in the project, the need for automatically calibrating the sensors based on stable periods was suggested. The criteria of the stable period was to be defined further, but was not implemented during the IOHN project. Some vendors of sand sensors provide similar functionality already. This will reduce the time spent on maintaining configuration, which can be a considerable challenge for different data systems.

As described there will be two series with sand rates in grams per second. The actual calculated value and the calculated values with optional feedback. The current implemented work is further described in appendix 6.B.

6.5.6 Results

The following sections summarized some of the key results for the Sand Detection Case.

6.5.6.1 Siemens IMS System (PIMAQ)

In the Siemens IMS system, PIMAQ, there are implemented a number of new components to handle the requirements from the Sand Detection Case. The PIMAQ calculations and the applications have been used in a test environment and verified to pick up a number of events.

- Centralized calculation of sand rates using calibration tables and well tests.
- Automatic detection of sand events based on sand rate.
- Automatic correction based on feedback from the decision support system.
- Enable the PIMAQ application configuration to include (manual) semantic annotation of objects.

6.5.6.2 Epsis Decide

Epsis Decide is a prototype implementation aiming to identify the underlying cause of significant variations in acoustic sand measurements.

- sand event categories have been defined based on the way data behave under certain conditions.
- a learning scheme has been trained using the sand event examples available to the project.
- the test examples was correctly classified.

6.6 Erosion management Case

6.6.1 Introduction

Solids like sand are an inevitable by-product which is experienced from most reservoirs throughout the world. The volume of solids produced may vary significantly from one field to another and will also in general vary throughout the field lifetime; i.e. depending on the reservoir characteristics, the well completion and sand exclusion techniques applied as well as the production characteristics from the reservoir. Sand production may be continuous or intermittent. In many cases, sand production is associated with pressure depletion of the reservoir and/or water breakthrough and may thus be experienced at a late stage in the lifetime of the reservoir.

Typically, sand from the NCS is in the order of 200 – 350μm in size and sand volumes may vary from a few grams per day (sand free) to several tons per day (catastrophic sand production which requires process shut down). Sand screens are common technology used to limit the sand production from a reservoir. However, even if sand screens are used, some production of smaller sand particles are to be experienced, generally fines with size < 50μm. Sand screens may also fail resulting in huge production of sand and gravel which may thus represent a critical situation both with respect to safety and the availability of the production system.

Until the 1990s, most fields were operated in accordance with the MSFR (Maximum Sand Free Rate) strategy. In the MSFR strategy, wells were choked back if sand collected in a sand trap exceeded a pre-set value - typically a few grams of sand in the sand trap over a period of two hours. The same criteria were utilised irrespectively if it was a low producing oil well or a high producing gas well. It was emphasized that this strategy resulted in choking back wells with a low erosion potential and thus had a detrimental impact on the income from the field.

Operational experience from several fields operating according to MSFR also showed significant sand production. Due to the misperception that MSFR was equivalent to "zero" sand production, several incidents occurred, leading to excessive erosion damage to either piping or piping components as well as process upsets caused by the actual solids production.

The concept of Acceptable Sand Rates (ASR) was introduced by DNV during the 2nd half of the 1990's. As opposed to the more traditional approach of attempting to operate according to the so called MSFR strategy, the ASR strategy is based on controlled sand production which is closely linked to and limited by the production system design and operation.

The ASR strategy has successfully been applied by several operators to control sand production in a safe and cost efficient manner. Implementing an ASR strategy reduces the potential negative risks associated with a perceived "zero sand production" strategy, and often offers opportunities in terms of increased production or more efficient inspection and maintenance.

6.6.2 Objectives and scope of work

The objectives for the Erosion Management Case were originally defined as:

Successfully demonstrate the value of automated erosion management

Statoil has emphasized the importance of erosion management in controlling sand production, and the goals achieved by introducing and automating this are:

- Simplified work process
- Efficient access to relevant data
- Status overview and decision support
- Analysis functionality (online and offline data)
- Economic benefits

Successfully demonstrate the value of standards and data interoperability

Data interoperability is a main focus area for the IOHN project, and therefore a main goal from the use case. We want to achieve the following:

- Define nomenclature (extension of ISO 15926) related to sand, erosion and equipment characteristics needed by models.
- Be able to access data from data sources offering the data on these standards.
- Make data available for other applications/services using the same standards.
- Demonstrate how standards enable enterprise-wide deployment at minimal deployment costs.

While ISO 15926 has been a valid and useful starting point for ontology work, the main focus has been to describe a nomenclature that could be used in the IOHN project without being a direct extension of ISO 15926.

Originally, a point about the use of agent technology was also included in the objectives, but this point has later been excluded.

6.6.3 Sand management strategy

In order to cope with sand production in a safe and controlled manner, a Sand Management Strategy has to be in place and implemented within the organisation. A general Sand Management Strategy has been developed within the present project; ref [6]. The document defines requirements to a Sand management Strategy; i.e.:

- On-line sand monitoring/detection systems
- On-line erosion monitoring software

- Clearly defined responsibilities and requirements to personnel within the organisation dealing with both daily operational follow up activities as well as inspection and maintenance activities
- Clearly defined criteria for acceptable erosion rates and accumulated erosion
- Clearly defined criteria for operation of chokes with respect to minimum and maximum opening (Cv) during normal operation, as well as maximum acceptable deviation between measured Cv (based on travel and the Cv curve) and predicted Cv (based on the operational condition) before inspection of the choke should be performed

6.6.4 On-line erosion monitor software tools requirements

On-line erosion monitoring software tools are vital parts of a Sand Management Strategy both with respect to closely follow up the operation and status on a daily basis, and as basis for planning and optimisation of production, inspection and maintenance over a longer period of operation.

Specifications of the main features and functionalities which should be available in an erosion monitoring software tool suitable to support the over-all sand management strategy has been developed within the present project.

The detailed requirements are given in appendix 6.D and cover:

- **System**
  The applications should cover the system from wellhead to 1st stage separator as a minimum.
- **Data flow**
  The software should have the functionality to access operational data, well test data, inspection and event data and topology data from the Integration Layer developed within the IOHN project. The software should also have the functionalities to store data from analysis back into the Integration Layer for easy access by other applications. Additionally, the software should have functionality of data check and identification of erroneous data and rules for handling of erroneous data.
- **Critical components and erosion models**
  The software should have the functionality to estimate erosion rates and accumulated erosion for all critical (erosion exposed) components (chokes, bends, Tee's, pipes, manifolds, etc.) identified between the wellhead and separators. The erosion predictions should be performed utilising erosion models from DNV-RP-O501 [10] or similar models if required by Company. Details about erosion modelling and erosion model requirements are described in appendix 6.C.
- **Erosion indicators and operational parameters**
  The software should have the functionality of presenting erosion indicators like erosion rate, accumulated erosion, choke Cv, flow velocities etc. as trend plots/time series.
  The software should also have the functionality of presenting trend plots of operational data like production rates, pressure, temperature and sand rates etc.
- **Model calibration/tuning**
  The software should have the ability to utilise results from inspection and maintenance for calibration and tuning of the erosion models.
- **Decision support**
  The software should have functionalities that enable easy identification that the system is operated within specified acceptance criteria with respect to erosion rates, flow velocities, Cv difference, sand production rates etc.

6.6.5 Erosion monitoring software

In the present project two different software tools; i.e. ABB ErosionInsight and DNV's Erosion Monitor Application (DNV-EMA), were selected. Both these software tools have been developed in cooperation with Statoil over a period of time and have been used within Statoil for several years for various sand management and erosion predictions services.

Detailed description of the software tools are given in Appendices 6.F and 6.E. Further, more details regarding functionality and how the application can be used as part of erosion management, production optimisation, as well as inspection and maintenance optimisation are presented in connection with the specific work described for the Erosion Management Case and for the Condition based Inspection and Maintenance Case.

6.6.6 Work performed and results achieved – ABB

The main activities of ABB for the Erosion Management Case have been the following:

- ABB has been actively involved in the process of defining a nomenclature for the Erosion Management Case in the Integration Layer. Numerous examples have been provided on the current use of input data as well as results, including detailed description of all data access from external sources used by the ABB product ErosionInsight. ABB has also participated in the review sessions of the model and commented on the design, as well as providing examples for specific model queries.
- ErosionInsight has been set up for online erosion management for the Snorre A, B and UPA installations. This activity is described in detail in sections 6.6.6.1 and 6.6.6.2, and appendix 6.G.
- Development and demonstration of a prototype application to retrieve input data for online erosion management from the Integration Layer. This work is also described in section 6.4.6.
- At the beginning of the IOHN project (2008 – 2010), a considerable amount of work was done by ABB to interface the ErosionInsight application with the IBM commercial integration layer named IIP (later renamed to IIC). A short overview of this work is given in section 6.4.6.6.

6.6.6.1 Set-up of Erosion Management Case for Snorre A, B and UPA

6.6.6.1.1 Software and functionality

To demonstrate the value of automated online erosion management, ABB has set up ErosionInsight for the Snorre A, B and UPA installations as part of the present project. This has been an important activity in the project, and the main part of the work was performed in 2010.

ErosionInsight has been set up in two versions, one online version within Statoil, and one demo/test version set up within ABB using input data from the Integration Layer accessed via SOIL.

Within Statoil, ErosionInsight is actually an enterprise-wide application. It is accessible from a web page on the Intranet, and adding a new oil installation is only a question about adding the configuration for a new field. The input data are read from standardized database views defined in the Energy Components (EC) or Prosty production data bases. This means that there is only one installation of the ErosionInsight software within Statoil. It also means that the software is not tailor-made for each field, and that changes and improvements implemented for one field will also be available for the other fields. The software is set up to read input data and perform calculations every night, and it also offers mechanisms to update input data and perform recalculations after reallocation of the production rates.

Only minor changes in the ErosionInsight software have been implemented in the present project, as much of the required functionality was already available. However, improvements have been made to the ErosionInsight software recently, and this functionality has been applied to the Snorre set-up now in 2012. The most important new functionality is the use of sand rates in erosion calculations, improved and more flexible PVT models, new and more flexible erosion models, as well as improvements for enhanced user experience.

A thorough description of background, system architecture and functionality of ErosionInsight can be found in appendix 6.F. See also [2] for a background on Statoil's case study for Gullfaks on erosion management, and [5] for ErosionInsight cases within Statoil.

6.6.6.1.2 Configuration and set-up

In the present project, configuration of the system has been based on a pre-study [8] performed by PTI in 2005 (PTI was acquired by ABB in 2006.) This means that some of the information might be out-of-date. In addition, the Snorre organization has provided choke configuration data in Excel work sheets, data sheets etc, but not all necessary data have been provided, which means that in some cases best guess data or dummy data have been used in the configuration. The selection of erosion nodes has been made based on the pre-study in [8], but not all the erosion nodes described in the report have been configured.

Note that the project has been executed with a minimum of involvement from the Snorre organization. This was also the intention, and it means that some of the data used to configure the system might be outdated and that they should be revised by the Snorre organization before drawing too firm conclusions from the results.

When the work started in 2010, the implementation of Energy Components (EC) in the Snorre organization was quite new, and some necessary input data were missing, either in EC or in the EC database views used by ErosionInsight. Over the time the situation has improved, but there are still some challenges with respect to data availability:

- Choke types and choke change history are not monitored in the Snorre implementation of EC.
- The well test data available in the EC database views are not complete, e.g. there are no well test data available for Snorre A platform wells after 2010.
- Sand rate data are not available before February 2009.
- Sensor data for flowlines are not available before January 2011.

In the present project, ErosionInsight is set up to monitor the following:

- **Snorre A platform wells**
  Set up to monitor erosion in bends and chokes, Cv and Cv difference for chokes, and flow velocity in bends, chokes and pipes. Erosion rate is calculated with measured sand rate since February 2009 and also with fixed sand rate of 0.1157 g/s, corresponding to 1 kg/day. Well P-14 B is back calculated to October 2003 with fixed sand rate to look at erosion monitoring results when there was a serious incident in November 2003 with erosion and loss of containment.

- **Snorre UPA subsea wells**
  Set up to monitor erosion and flow velocity in bends since April 2009 with a fixed sand rate of 0.1157 g/s (1 kg/day). A fixed sand rate is used as the quality of the sand rate data in EC is uncertain. No choke monitoring as there are no chokes.

- **Snorre UPA flowlines**
  Set up to monitor erosion in bends and flow velocity in bends and topside chokes. Calculated since January 2011 when sensor data are first available in the EC views for flowlines.

- **Snorre B subsea wells**
  Set up to monitor erosion in bends, Cv and Cv difference for chokes, and flow velocity in bends and chokes since April 2009. Wells do not have acoustic sand detectors, so erosion is calculated with fixed sand rate of 0.1157 g/s, corresponding to 1 kg/day. Choke types and choke change history is uncertain, which also means that the results from the Cv monitoring might be uncertain.

- **Snorre B flowlines and manifolds**
  Set up to monitor erosion and flow velocity in bends for the test lines, to monitor erosion and flow velocity in bends and flow velocity in topside choke for the production lines, and finally flow velocity in the production manifold. Calculated back to January 2011 when sensor data are first available in the EC views for flowlines.

In addition, the Snorre EC views comprise data for the Vigdis subsea installation. Vigdis could also have been easily included, but this was not prioritized as we did not have detailed information about topology, configuration data and critical components with respect to erosion etc.

Appendix 6.G contains a detailed description of the configuration and calculation status for the Snorre field.

6.6.6.2 Snorre erosion monitoring results

In general, the following erosion indicators can be used to define acceptance criteria for safe production with respect to erosion:

- Flow velocity: A maximum limit, like e.g. 10 m/s might be used. The flow velocity should be monitored at locations with maximum flow velocity. The maximum limit should also be related to the sand rate.
- Sand rate: If sand rate measurements from acoustic sand detectors are available, a max allowable sand rate can be set, e.g. 0.1 g/s. The

sand rate measurements should be followed continuously to immediately detect events with high sand production.

- Erosion rate: If sand rate measurements are available and the erosion models have been qualified, maximum erosion rate can also be used as an acceptance criterion for safe production. Typical values might be 0.1 or 1 mm/year.
- Choke opening: Chokes operated at small choke openings are more prone to erosion than chokes operated at larger openings. Therefore a choke type should be used with a flow capacity that is adapted to the actual production from the well.

The other erosion indicators, accumulated erosion and Cv difference are well suited to support decisions about inspection, so they will be discussed more in detail in section 6.7.6.

In the following will be shown examples how the Snorre set-up can be used to support the operation of a field within the acceptance criteria for safe production. First, is shown an example for a Snorre A well, and then an example for the whole Snorre field.

6.6.6.2.1 Snorre A results

Sand production is a potential problem for the Snorre A platform wells. In 2003 there were serious incidents at Snorre A with considerable erosion and loss of containment for e.g. the P-14 B well, but also in wells like P-4 and P-9. After these incidents, the Snorre organization had to revise their production strategy to a much stricter regime, which in practice did not allow erosion.

The basic acceptance criteria tell that the sand production should not be higher than 0.1 g/s if the total flow velocity is 10 m/s. If the flow velocity is higher, e.g. 25 m/s, the sand production must be lower. These criteria are well established, and they have worked fine with no serious incidents since they were established around 2006. However, Snorre A allows ASR (Acceptable Sand Rate) tests. This means that the wells are can produce with higher sand production and higher degree of supervision for a short period, e.g. 6 hours.

Figure 6.30 shows an overview of the total flow velocity in m/s for the Snorre A platform wells from a day in April 2012. No wells have a total flow velocity above the max limit of 10 m/s which is indicated by the horizontal line, but the P-23 well is the well with the highest total flow velocity.

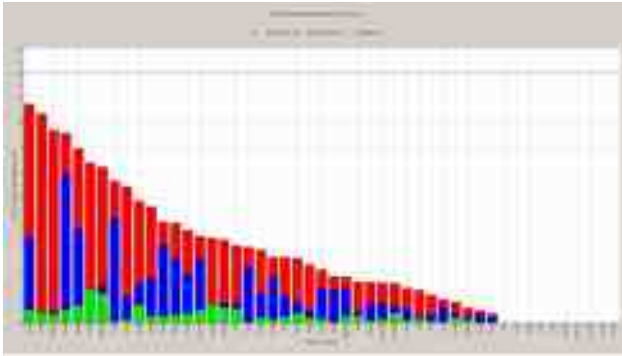


Figure 6.30: Overview of total flow velocity in m/s for the Snorre A platform wells from a day in April 2012

Figure 6.31 shows the time series plot for the flow velocity in m/s in the 4” bend upstream choke for the P-23 well. The upper plot shows the flow velocity for oil (red) and water (blue), and the lower plot shows the flow velocity for gas (red). By adding the flow velocities for oil, gas and water one get the total flow velocity, and the total flow velocity has been as high as 20 m/s in long periods for this well.

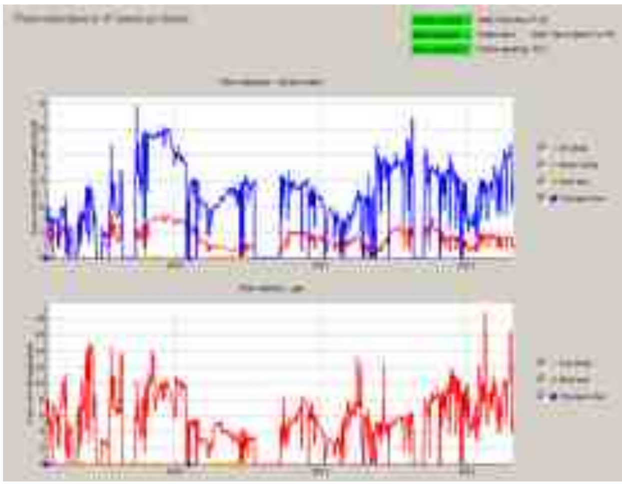


Figure 6.31: Time series plot for the flow velocity in m/s in the 4” bend upstream choke for the P-23 well

Figure 6.32 shows the time series for sand rate in g/s for the P-23 well. Even though the sand production rate is below the limit of 0.1 m/s (red horizontal line) most of the time, the combination of high flow velocity and sand rates just below the max limit over a considerable period of time might be a bad combination.

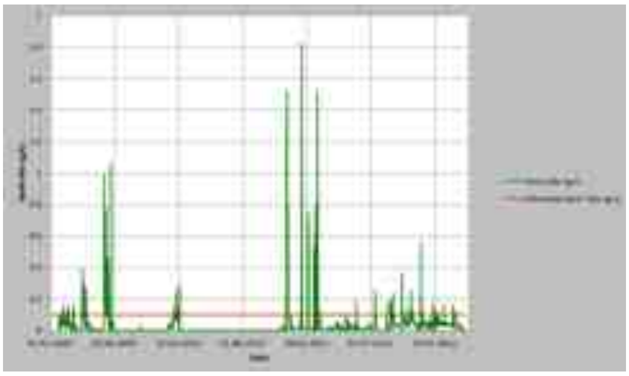


Figure 6.32: Times series for sand rate in g/s the P-23 well

Figure 6.33 shows the time series plot for erosion rate in mm/year calculated from measured sand rate for the P-23 well. The horizontal lines indicate the limit of 0.1 mm/year that has been set for erosion rate. Note that in the choke sleeve (upper plot) the erosion rate is well above the limit of 0.1 mm/year in short periods, while for the 4" bend upstream choke (lower plot), the erosion date is below the limit of 0.1 mm/year most of the time, but still there are quite a long periods in 2010 and 2011 when the calculated erosion rate is just below the limit of 0.1 mm/year.

Note that the choke configuration parameters and the choke model should be revised before drawing too firm conclusions about the results, but well P-23 is probably a candidate for inspection, which will be discussed more in detail in section 6.7.6.

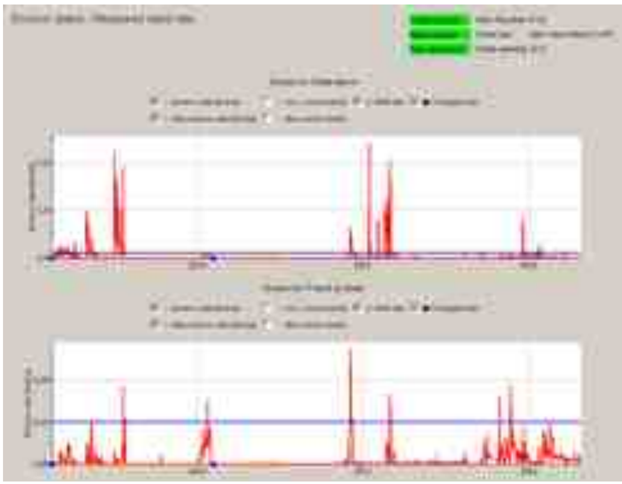


Figure 6.33: Time series for erosion rate in mm/year calculated with measured sand rate for the P-23 well

6.6.6.2.2 Snorre results overview and implications

Figure 6.34 shows the "traffic light" status overview for the Snorre A, B and UPA installations.

The left column shows the status for flow velocity. For Snorre the "red light" criterion for flow velocity has been set to 10 m/s. As one can see, on this day in April 2012 no wells or flowlines produced with a flow velocity above 10 m/s for any of the locations that are monitored.

The second left column shows the status for erosion rate calculated with fixed sand rate of 0.1157 g/s. Only a few wells (P-30, P-42 A from Snorre A and A-1 AH from Snorre UPA) produce with an erosion rate above the "yellow light" criterion of 0.1 mm/year on this particular day in April 2012. The "red light" criterion for erosion rate for Snorre is set to 1 mm/year.

The third column shows the status for the accumulated erosion calculated with a fixed sand rate of 0.1157 g/s. Only one well (A-1 AH from Snorre UPA) reports accumulated erosion above 1 mm, which is the "red light" criterion for Snorre. We will look more into well A-1 AH in the Condition based Inspection and Maintenance Case chapter. Other wells reporting accumulated erosion above 0.1 mm (which is the "yellow light" criterion for Snorre) are P-13, P-23, P-26, P-30, P-42 A and P-9 from Snorre A, and A-5 AH from Snorre UPA.

Results for erosion rate with measured sand

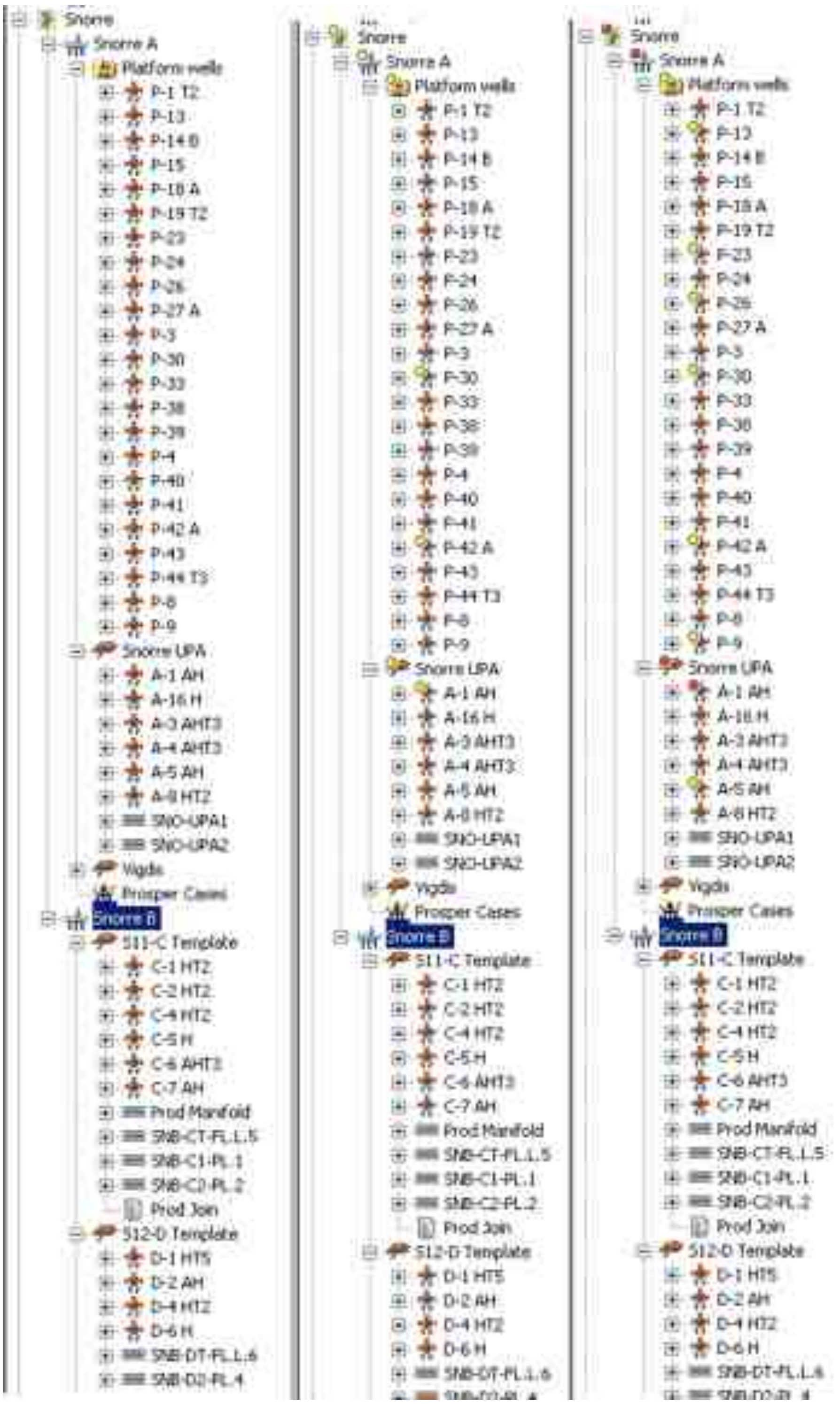


Figure 6.34: “Traffic light” decision support for the Snorre A, B and UPA installations from left to right: Total flow velocity, erosion rate and accumulated erosion.

rate are not shown here. ABB have recently added the use of measured sand rate in ErosionInsight. Calculations are performed with measured sand rate, but not all the facilities for plotting and decision support have been updated to include the measured sand rate. The ErosionInsight graphical user interface will be upgraded to include this in May/June 2012.

From the results, it might be concluded that with the exception of some Snorre A and Snorre UPA wells, the production at Snorre is quite prudent with respect to erosion. This is understandable, given the serious incidents from 2003. But it is possible that the production can be increased for some of the wells that do not show indications of erosion.

But before doing so, it would be very useful to qualify the models and results by comparison with inspection data. This will be discussed more in detail in the Condition based Inspection and Maintenance Case chapter.

6.6.6.3 Use of data from the Integration Layer

Use of data from a fully working Integration Layer will add considerable value to Erosion Management. The benefits of this and the work that still remains is described under Future work in section 6.8.3. Section 6.4.6 describes the ABB work to retrieve data from the Integration Layer.

6.6.7 Work performed and results achieved – DNV

The DNV-EMA software was developed as an integrated part of the development of ASR strategy. This development was performed in close cooperation with Statoil. The initial tool was solely programmed in Excel spreadsheet and required input data were manually transferred into Excel format.

Later on, the software has been updated and modified several times. In the latest revision, the code is programmed in Visual Basic for Applications using object-oriented models. This makes the code more flexible and simplifies the adaptation of the software to new developments.

Some of the development of the latest version of the software and adaption of the software to the Snorre A and B developments has been performed during the present project. Snorre A has, however, not been updated to the latest revision as no data has been available in the integration layer for Snorre A.

Additionally, an automated 'Status Report' functionality has been developed. The status report includes information/summary of operation of the field with colour codes showing how the field is operated with reference to specified acceptance/operational criteria; e.g.:

- Velocity
- Erosion rate
- Accumulated erosion
- Sand production

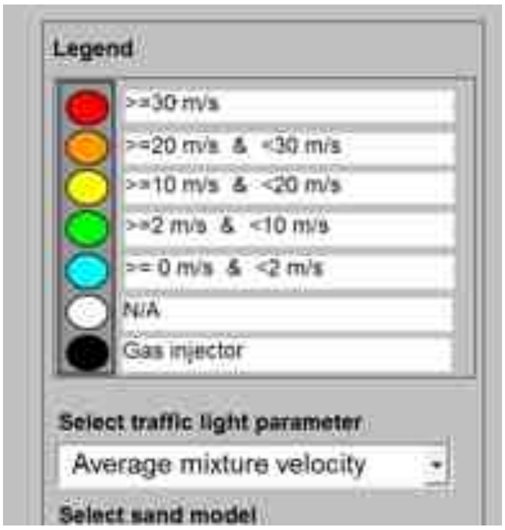


Figure 6.35: "Traffic light" legend and settings

The status report can also give information regarding next inspection, prioritising between different wells and components, etc. This functionality is demonstrated in more detail in the chapter addressing the Inspection and Maintenance Use Case.

The status report may also be customized to report other information; like instrumentation not functioning as they should, spurious operational data, etc.

The DNV-EMA software still requires operational data to be given in a spreadsheet format. For the applications in use, this is done manually (on weekly or monthly basis) or through some customized arrangements which allow for more automatic transfer of operational data into the software.

The main objective of the present project is to demonstrate that data may be transferred from the integration layer to the various applications. This objective has been successfully demonstrated within the project and is described in more detail in section 6.4.5.

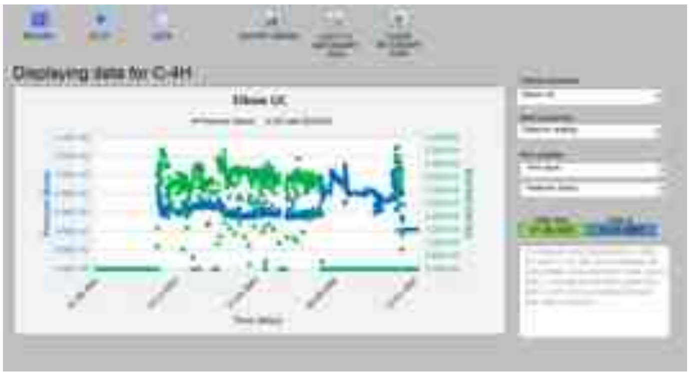


Figure 6.36: Example showing plot of production conditions

The 'front page' of the Software; see figure 6.38 will give an immediate overview of the status where colour indicators (traffic lights) show the status of various limiting parameters. Standard parameters that can be selected at the 'front page', which is critical parameters with respect to erosion management and production control, are velocity, erosion rate, accumulated erosion, and choke Cv drift. In case more detailed evaluation of a critical well or component are required, all operational data and results from the erosion calculations can be plotted as time series and used as input to the evaluation; see figures 6.36 and 6.37.

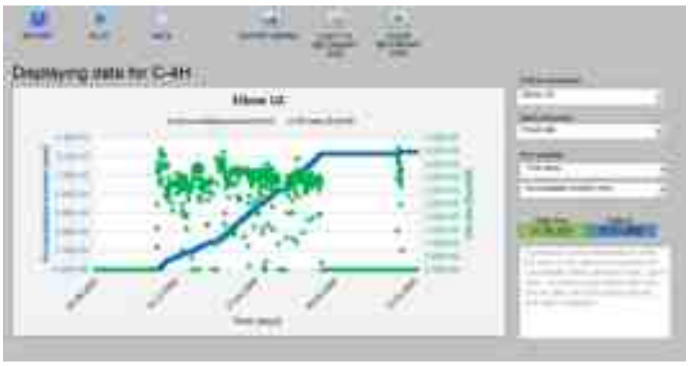


Figure 6.37: Example showing plot of erosion rate and accumulated erosion

6.6.7.1 Retrieving data from the integration layer

Retrieval of production data from the integration layer is handled through a separate dialog box where the user selects the time period and number of records (wells/pipelines) to be refreshed. The request is sent to the integration layer through a Java agent which returns the result in XML format. This XML format is then converted to a format compatible with the architecture of the Erosion Monitor. Finally, the data is merged with the existing production data records and an error check is performed to remove spurious values. Any changes are documented in text files for later reference.

6.6.7.2 Production optimization – ASR strategy

The software has been widely used as basis for production optimization in accordance with the ASR strategy.

The ASR strategy is based on allowing for a controlled sand production within specified acceptance criteria developed for the field. The acceptance criteria generally relate to specified maximum allowable fluid velocities, erosion rates, sand production rates and accumulated sand production over a period of time. Production optimization requires closely follow up of the production conditions (flow rates, pressure, temperature and sand production) with reference to the specified acceptance criteria.

The results and output from the software will give valuable input to simplify work processes, to enable system status overview, and for decision support.

6.6.7.3 Economic benefits/potential

The economic implications of utilizing the ASR strategy/software may vary significantly from field to field dependent on field characteristics and the operational limitations applied.

As an example, a pilot project was conducted for Statoil for the Gullfaks field in 2001. The Gullfaks field was at that time operated in accordance with the MSFR strategy. At that time 50 wells was restricted due to the specific MSFR acceptance criteria. Implementing the ASR sand

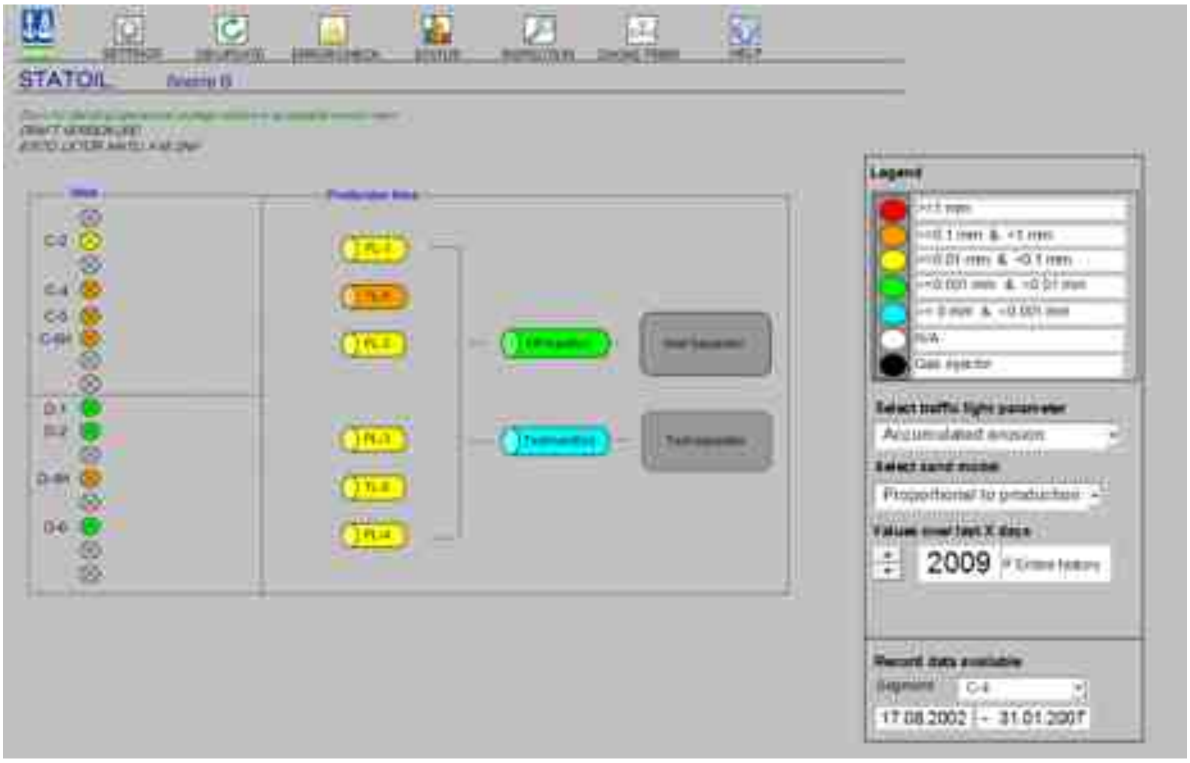


Figure 6.38: Geometry model with colour indicators showing accumulated erosion throughout the system

strategy it was demonstrated that the production could be increased with 15000bbld amounting to 1.5Million USD pr. day assuming an oil price of 100USD pr. bbl; Ref. [7].

6.7 Condition based Inspection and Maintenance Case

6.7.1 Introduction

Inspection and maintenance as well as production optimisation are key issues when operating a field in a sand production regime. Maximum production under safe conditions is a common goal for both the production engineers and the inspection and maintenance engineers.

Traditionally, inspection of topside piping and system components like chokes is performed at regular/specified intervals independent of operational characteristics and experience. The piping system is typically monitored using ultrasonic inspection, whereas chokes must be opened to be inspected.

Consequently, inspection of piping/components not being exposed to erosion may be executed unnecessarily, resulting in extra cost and possible loss of production. On the other hand, inspection at specified intervals may result in severe erosion not being detected in time. This may result in reduced safety margin and potential loss of integrity and controllability.

Inspection of subsea systems is normally not performed according to a similar plan as is used for topside systems due to limited access and lack of inspection methods suitable for subsea inspection. As a consequence, inspection of subsea systems normally requires shut down of the production and retrieval of the choke bridge and/or the XMT. Such an operation is associated with high cost and loss of production. Due to the excessive cost for inspection of subsea systems, such operations are often delayed or skipped until strong evidence of erosion is obtained and may thus result in reduced safety margins and potential loss of integrity and controllability.

Instead of performing inspection and maintenance as described above, integration of different sources of information, such as operational data, results from on-line erosion monitoring software, information from sand detection units, as well as inspection data could be used to support decisions in order to optimise:

- Inspection
- Maintenance
- Safety
- Production

As an example, wells and piping systems that show indications of high erosion rates based on results from the erosion monitoring tools can be selected for ultrasonic inspection. This information should be compared with information from the last inspection (Was it long ago? What were the inspection results then? How were the erosion rates? How has the production been in the last period? Has there been high sand production?) to make a final decision about inspection.

Similarly, for some chokes types, the flow capacity of the chokes will increase as the choke is eroded. Cv monitoring will detect such erosion, and Cv monitoring can be used to select chokes that need to be inspected and possibly replaced. The results from Cv monitoring are particularly valuable when compared with information about earlier choke changes. Cv monitoring is very well suited for Multiple Orifice Valve (MOV) chokes, and it is in active use to make decisions about inspection at different Statoil operated fields.

Traditionally, it has been the production engineers who have been in charge of the erosion management, but there is a trend that the inspection and maintenance engineers are getting more interested in the subject, as they see how it can be used to optimise the inspection and maintenance.

There are actually two aspects that make integration of erosion results and inspection/maintenance data particularly useful:

1. The results from the erosion monitoring can be qualified or calibrated by comparison with inspection results
2. Erosion results (accumulated erosion or Cv difference) can be used as indicators for inspection and possibly replacement, and this information is especially useful when compared with the inspection and maintenance history

By integrating inspection and maintenance data with erosion monitoring results and sand data, and also by adapting the software tools to the way the inspection/maintenance engineers work, we also think it will be possible to improve the collaboration between the domains of:

- Sand Detection
- Erosion Monitoring
- Operation and Production Optimisation
- Inspection and Maintenance

Figure 6.50 illustrates the concept with an example from Statoil.

6.7.2 Objectives and scope of work

The objectives for the Condition based Inspection and Maintenance Case were originally defined as:

- Define a Condition based Maintenance strategy for chokes and piping systems based on existing strategies applied today, while also bringing in additional requirements.
- Define improved work processes facilitating collaboration between the production and maintenance domains by evaluating existing work processes in use today and changing / extending / refining them as needed to support the defined maintenance strategy.
- Extend the sand management nomenclature developed in the project, with relevant terms from within the maintenance domain
- Supporting the defined strategy and work processes with software tools. This goal should result in a demonstration for maintenance personnel where the tools already in use in the erosion management pilot, have been extended to support at least parts of the defined strategy/-work processes, and where tool shortcomings and missing functionality wrt to the strategy/-work processes are also identified.

Later, the point about defining work processes has been taken out of the scope of work.

6.7.3 Condition based Inspection and Maintenance strategy

A strategy for Condition based Inspection and Maintenance has been defined within the present project ref. [6]. The document describes minimum requirements and conditions to be in place to adapt a condition based inspection and maintenance strategy for a field operated in a sand production regime.

For Condition based Inspection and Maintenance, the strategy describes how clear criteria for maximum accumulated erosion or choke Cv difference can be used to select when to inspect and possibly replace a component.

6.7.4 Special requirements for Condition based Inspection and Maintenance

In general, the requirements for the on-line erosion monitoring tools to support Condition based Inspection and Maintenance are very much the same as for the Erosion Monitoring use case. An overview of these requirements is given in chapter 6.6.4, while a more detailed description of the requirements is given in appendix 6.D.

There are especially two additional requirements that are important for Condition based Inspection and Maintenance, and they are described in the next two sections.

6.7.4.1 Support for tag names

The software should support identification of components (pipe, bend, choke) and sensors (pressure, temperature, choke opening, valve position) by tag names. Each tag name is unique, and the tag system is used to identify components in a process system. Tag names are the identifiers most familiar to the inspection/maintenance engineers. This is opposite to the production engineers who often do not relate to tag names to the same degree.

6.7.4.2 Inspection and Maintenance data

The applications should have the functionally to access and display inspection and maintenance data. Such data may cover (but not be limited to):

- Inspection history
  - Inspection date
  - Measured wall thickness for pipes and bends
  - Photos for choke
  - Textual description, e.g. estimated erosion for different locations in a choke
- Maintenance history
  - Date for maintenance work
  - Change of pipe component (bend, T-bend), including description for the new component in case there have been changes
  - Textual description
- Choke change history
  - Date for change of choke and/or choke trim
  - Choke type
- Sand event history
  - Date
  - Special event with high sand production
  - Amount of sand in sand trap

The applications should always have access to updated data. One challenge with respect to inspection and maintenance data is that these data normally reside in completely different data bases and computer systems than the other input data used for the erosion monitoring. While operational data (production rates, sensor data) are available in production data bases like Prosty, Energy Components (EC) or PI, the inspection and maintenance data can be work permits in SAP or in Excel work sheets at some person's computer. This means that inspection and maintenance data might be currently less available for use in a data provider like the Integration Layer.

6.7.5 Demonstration software

As for the Erosion Management Case study, the ABB ErosionInsight and the DNV Erosion Monitor Application have been used to demonstrate the case. Both applications had some functionality to support condition based inspection and maintenance prior to the project, and a defined work process already existed.

Detailed description of the software tools are given in appendix 6.E and 6.F.

6.7.6 Work performed and results achieved – ABB

ABBs activities in the Condition based Inspection and Maintenance Case have been the following:

- Together with DNV, ABB has specified requirements for erosion monitoring software and input data to support Condition based Inspection and Maintenance. These requirements are described in appendix 6.D.
- Demonstrate how inspection and maintenance data can add value to results from erosion monitoring tools and how such results can be used to support condition based inspection and maintenance. This has been demonstrated through examples from Snorre using the ErosionInsight online erosion monitoring system.

6.7.6.1 Set-up for Condition based Inspection and Maintenance for Snorre A, B and UPA

6.7.6.1.1 Software and functionality

The ErosionInsight software has been used for the Condition based Inspection and Maintenance Case. This is the same software as was used for the Erosion Management Case as described in section 6.6.6.1. The objective of this work has been to demonstrate how integration of data from erosion monitoring and inspection and maintenance data can add value:

- How results from erosion monitoring can be used to support decisions about inspection and maintenance
- And how inspection and maintenance data can be used to qualify and improve the results from erosion monitoring

Also for the Condition based Inspection and Maintenance Case, only small modifications were made to the ErosionInsight software during the present project.

ErosionInsight currently supports use of the following Inspection and Maintenance data:

- Choke types and choke changes: The application reads information about choke type and choke changes from the EC/Prosty production data base. This means that the choke type used in the calculations will always be up-to-date. Choke changes are also marked in time series plots as blue dots, and the name of the choke type will show up as a tool tip.
- Reset of accumulated erosion: If a choke is replaced, the new choke should have zero accumulated erosion. Depending on how much of the choke that is replaced (choke disc, complete choke) the accumulated erosion in different parts of the choke (inlet, outlet, sleeve etc.) should also be set to zero. Information about which parts are replaced is not available in the EC data base, so reset of the accumulated erosion is therefore not done automatically. But the software allows manual reset of the accumulated erosion.
- Inspection results: Similarly, the accumulated erosion for any erosion node can be set manually to a value obtained from inspection.
- Sand, inspection and maintenance data: ErosionInsight also has support to store various sand, inspection and maintenance event data, but as these data are not read from an external data source, but have to be entered manually; this functionality is currently not in active use. See figure 6.70 in appendix 6.F.

It is very important for the usability of inspection and maintenance data that these data are read automatically from a data source, so that they are always kept up-to-date. It cannot be expected that the end-users of the software will enter such data manually into the application.

ErosionInsight also supports the identification of components and sensors by their tag name, a requirement that is much wanted by the inspection/maintenance engineers. Currently, the tag names must be set up manually during the configuration of the system, but this is normally done only once. Figure 6.74 in appendix 6.F shows an example how the tag names can be used to identify components.

ErosionInsight should be extended to generate automatic status reports. Currently, status reports are written manually. Status reporting could also include generation of warnings sent by either e-mail or SMS, but in this case it is very important that any false alarms are eliminated.

6.7.6.1.2 Configuration and set-up

The configuration and set-up of ErosionInsight for Condition based Inspection and Maintenance Case is the same as for the Erosion Management Case, and it is described in detail in section 6.6.6.1 and appendix 6.G.

Like for the Erosion Management Case there have been some challenges with respect to input data:

- Data about choke types and choke changes were not available in the Snorre implementation of EC, so these data have been configured manually based on data from Excel work sheets and data sheets received from the Snorre organization. However, this information was not complete and not necessarily up-to-date, which means that dummy data or best guess data for choke types and choke Cv characteristics have been used in some cases.
- The well test data available in the EC database views are not complete.
- Sand rate data have only been available in the EC database views since February 2009.
- Sensor data for flowlines have only been available in the EC database views since January 2011.
- Information about tag names has not been provided by the Snorre organization, so components and sensors have not been configured with tag names.
- Production and sensor data for Snorre B production line 3 are currently not available in the EC database views, so calculations are not performed for this flowline.
- Only limited inspection data have been available, limited to measurements of wall thickness from Snorre B flowlines. Most of these data are from before January 2011, so the only observation relevant for this case was a measurement of wall thickness for Snorre B multipurpose line 2 from 2011, which showed zero accumulated erosion.

6.7.6.2 Snorre results

6.7.6.2.1 Inspection criteria

Potentially the following criteria could be used to make decisions about inspection:

- Accumulated erosion:
  - Accumulated erosion can be used as a criterion for inspection, especially when sand rate data from sand detectors are available. Typical limits could be e.g. 0.1 mm or 1 mm.
  - For the pilot installations of ErosionInsight (Sleipner and Snorre), sand rate data will be available for some of the wells and flowlines, and for these examples accumulated erosion is very well suited as a criterion for inspection.
  - For the fields where ErosionInsight is in active use (Gullfaks and Statfjord), accumulated erosion is currently not used as an inspection criterion, as measured sand rate data are not available. Accumulated erosion will then be calculated using a fixed sand rate, and this might give false alarm for wells with very little sand production.
- Cv difference:
  - The Cv difference describes the difference between the actual flow capacity of a choke and the theoretical flow capacity of a choke. When the choke is eroded, the flow capacity of the choke increases, which means that the Cv difference will increase gradually.
  - Cv difference is actively used as a criterion for inspection and possible replacement of chokes for the Gullfaks and Statfjord fields. For Gullfaks, the choke should be inspected when the Cv difference passes 7. This limit has been set to make a simple criterion, but this solution is not ideal, as the limit should be higher for a large choke (e.g. max Cv = 250) than for a small choke (e.g. max Cv = 15). For the latter a Cv difference of 7 means that the choke must be really heavily eroded.
  - For the chokes in use at Snorre (and Sleipner) it is not expected that erosion will affect the Cv difference to the same degree as for the MOV chokes in use at Gullfaks and Statfjord, but it is not impossible that there might be an increase in Cv difference, especially if the choke is heavily eroded.
  - The Cv difference results require high quality input data to be reliable. Especially, when the pressure drop over the choke is low, only small errors in the input data might have a

large effect on the results from the calculations. Therefore the engineer cannot use the Cv difference limit uncritically, but also has to check the quality of the input data and the magnitude of the pressure drop over choke before making a decision about inspection. ErosionInsight offers different mechanisms to support this quality check.

6.7.6.2.2 Snorre A results

In the following will be shown examples for the well P-23 from Snorre A that was also studied in section 6.6.6.2.

The upper plot in figure 6.39 shows the Cv plot for the well P-23:

- The green dots connected with a line show the Cv difference based on well test data. As can be seen, no well tests have been available in the EC database views after June 2010.
- The purple dots connected with a line show the Cv difference based on allocated production rates (monthly average).

The plot shows that the Cv difference based on allocated production rates increases gradually, which might indicate that the choke has been eroded and that it should be inspected.

Further, the upper plot in figure 6.39 shows:

- The red line shows the calculated Cv, which is calculated based on production rate, PVT properties and the pressure drop over choke. In ErosionInsight this Cv value is calculated using the Perkins choke model [12], which also takes care of critical flow through the choke.
- The blue line shows the theoretical Cv, which is determined from the choke Cv characteristics, which is actually a look-up table that gives the expected Cv for a given choke opening.
- The dark green line shows the maximum Cv for the actual choke type.
- The blue dots show choke changes (choke type is available as tool tip).
- The orange dots show well tests (well test date is available as tool tip).

The lower plot in figure 6.39 shows:

- The red line shows pressure drop over choke from production data.
- The red circles connected with a red line show the pressure drop over choke from well tests.

Since well test data are not available after June 2010, the decision about inspection must be based on the Cv difference from allocated production rates. Before June 2010 there was a relatively good match between the results from well tests and allocated rates, so there is a chance that this still is the case. So the results indicate that this choke should be inspected. The pressure drop over the choke is also high, which means that the results should be reliable in case the production rates used in the calculations are reliable.

To make the decision about inspection it would have been very useful to have earlier inspection data easily available to answer questions like:

- When was the last choke inspection and how were the inspection results?
- What about the choke that was changed in 2010? Was the choke eroded? Were there any sign of erosion in the cage that might have affected the Cv?
- What about other chokes of similar type? Have there been signs of erosion in the cage for other chokes of the same type? In that case, how is the erosion in the cage compared to other locations in the choke?

Also note that there might have been choke changes after 2010, but updated information about choke changes has not been provided after 2010.

Figure 6.40 shows the times series plot for erosion rate and accumulated erosion calculated from measured sand rate for the choke sleeve (upper) and the 4” bend upstream choke (lower) for well P-23.

- The green line shows the accumulated erosion in mm (right value axis), while the purple horizontal line shows the max limit for accumulated erosion, set to 0.1 mm in this case.
- The red line shows the erosion rate in mm/year (left value axis), while the blue horizontal line shows the max limit for erosion rate, set to 0.1 mm/year in this case.

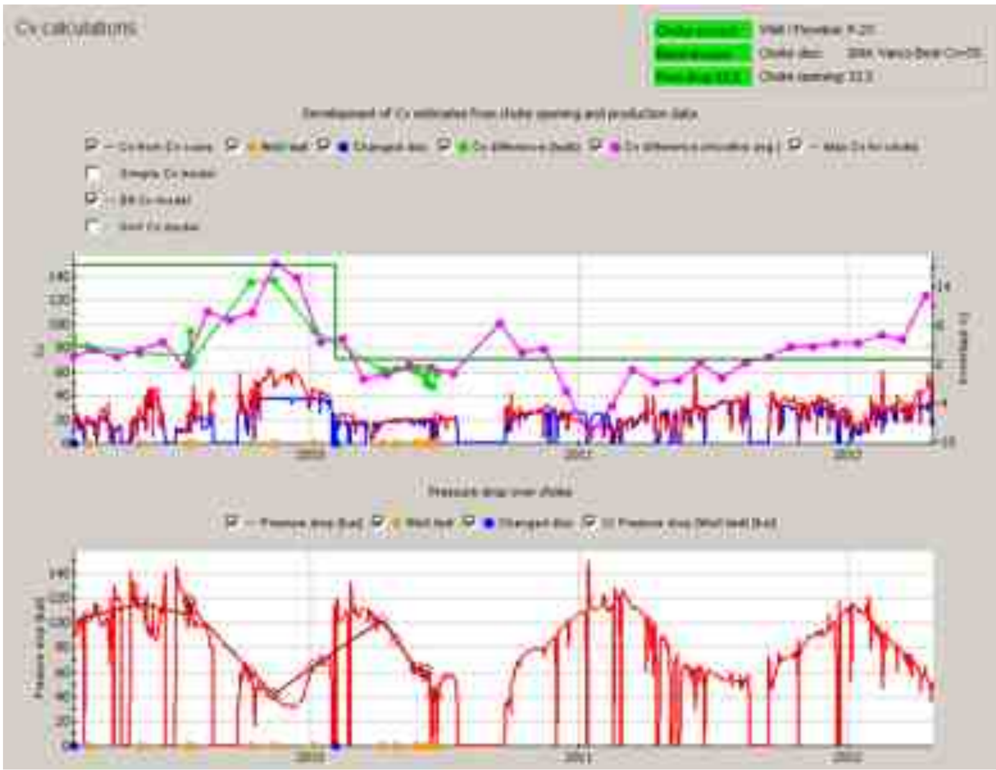


Figure 6.39: Time series plot for Cv and Cv difference (upper) and pressure drop over choke (lower) for well P-23

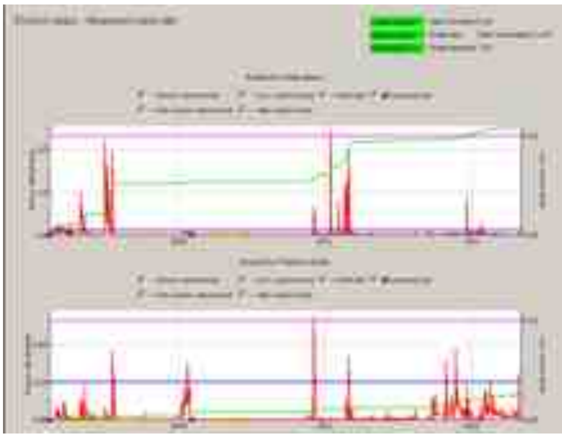


Figure 6.40: Times series plot for erosion rate and accumulated erosion from measured sand rate for the choke sleeve (upper) and the 4” bend upstream choke (lower) for well P-23

The accumulated erosion in the choke sleeve is higher than the maximum limit. Also note that the accumulated erosion is calculated since February 2009 when measured sand data were first available. This means that the accumulated erosion could actually be even higher. According to the limit that has been set, the choke should be inspected.

The accumulated erosion for the choke sleeve could have been set to zero after the choke change in 2010, but it is not known if the choke sleeve was changed at that time. This is another example where maintenance information about the choke change would have been useful.

6.7.6.2.3 Snorre UPA results

In section 6.6.6.2 was demonstrated mechanisms for decision support to decide which choke/bend that should be inspected next. The only well that showed up with accumulated erosion above the “high high” limit of 1 mm was the Snorre UPA A-1 AH subsea well.

Figure 6.41 further demonstrates these mechanisms. The upper left figure shows an overview of the Snorre UPA wells using traffic lights to show which wells pass the “high high” and “high” limit for accumulated erosion. The upper right figure shows an overview plot of accumulated erosion for the Snorre UPA wells sorted after value. And the lower figure shows the total flow velocity for the Snorre UPA wells sorted after value.

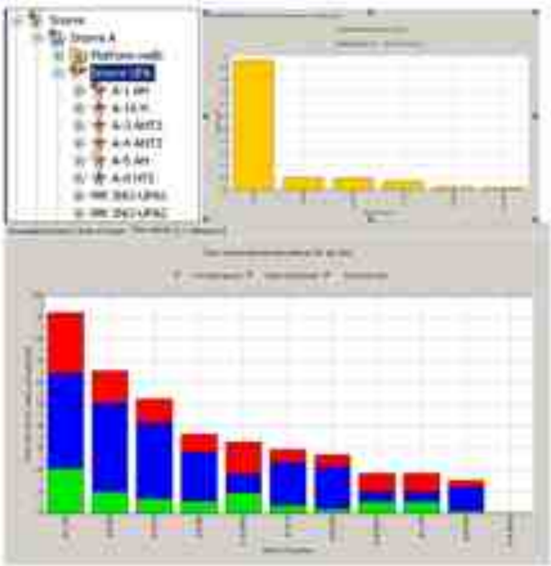


Figure 6.41: Overview plots of the accumulated erosion and flow velocity for the Snorre UPA subsea wells

Sand rate data are available for the Snorre UPA wells in the EC views, but the quality is uncertain, so the erosion calculations have been calculated assuming a sand rate of 0.1157 g/s, corresponding to 1 kg/day. Note that these wells do not have chokes, and that the results are based on the accumulated erosion since the calculations started in April 2009.

The results show that the accumulated erosion for the 3” bend in the A-1 AH well has passed 1 mm, which is the “high high” limit defined for accumulated erosion. Well A-5 has also passed the “high” limit of 0.1 mm for the 3” bend.

Before making a decision about inspection (this is not an easy task, as the well is a subsea well) the results from the erosion calculations should be qualified. Questions to ask are:

- Does this well produce sand? Are there any estimates about the amount of sand? Is the well supposed to produce more or less sand than the 0.1157 g/s that are used in the calculations?
- Has the well been inspected earlier? In that case, were there any signs of erosion?

The questions above could have been easily answered if other sand and inspection data had been easily available from the user interface by accessing data from the Integration Layer.

The flow velocity for the 3” bend in the A-1 AH well is also below the criterion of 10 m/s that has been defined for safe production for the Snorre field.

Figure 6.42 shows the erosion rate and accumulated erosion time series for the A-1 AH well, and it shows that the erosion rate (red line) has been above the limit of 0.1 mm/year (blue line) most of the time. This example shows that keeping the flow velocity below the acceptance criterion is not always sufficient. In this case the erosion rate is still considerable because of the small diameter (3”) for the bend.

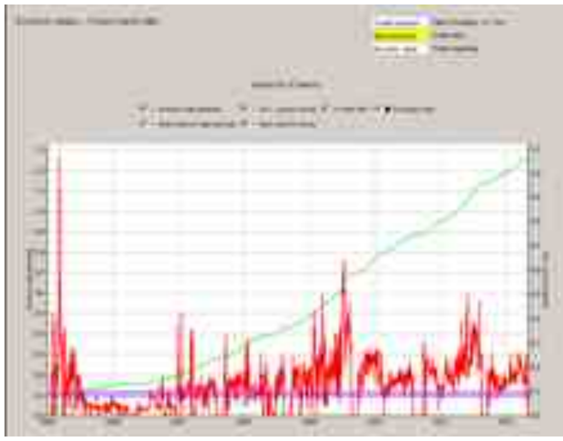


Figure 6.42: Erosion rate and accumulated erosion time series for the A-1 AH well

6.7.6.2.4 Snorre results and implications

The results have demonstrated how accumulated erosion and Cv difference might be used as criteria for inspection.

Accumulated erosion is particularly interesting when sand rate data are available. In cases where sand rates measurements are not available the results from fixed sand rates must be compared to other sand data (e.g. sand trap data) or earlier inspection data to give an idea whether sand is available or not.

Cv difference might also be used to support decisions about inspection, even though erosion was not expected to change the Cv difference for the chokes in use at Snorre. The increasing Cv difference for well P-23 indicates that the choke should be inspected, and if the choke really is eroded, it proves that Cv monitoring might also

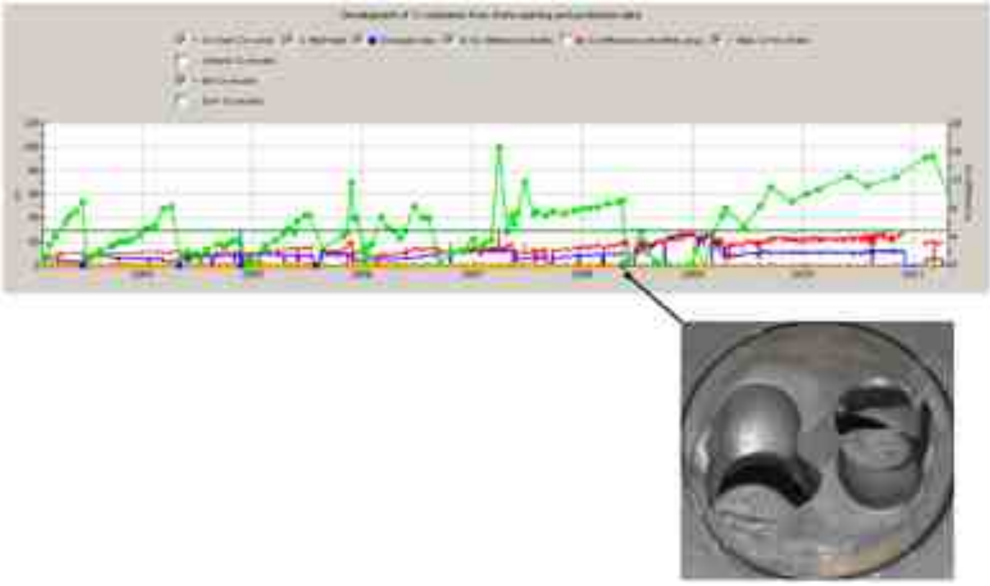


Figure 6.43: Example how inspection data can be made easily available from the ErosionInsight user interface

give valuable information for these choke types. Cv monitoring can also be used to reveal e.g. scaling or blocking of the chokes.

The results show that the P-23 well should probably be inspected, and that the A-1 AH well should be investigated more in detail (sand and inspection data etc) to see if the erosion rate and accumulated erosion estimated with a fixed sand rate reflect the actual situation. For both wells inspection data could have been used to qualify and improve the erosion model and configuration data. For the P-23 well both Cv difference and accumulated erosion can be used to make decisions about inspection. And for the A-1 AH well accumulated erosion could be used to make decisions about inspection if sand data and earlier inspection data confirm the results from the monitoring.

Based on accumulated erosion some more of the Snorre A platform wells should also be inspected. The Cv difference has not been analysed in detail for all the Snorre A platform wells, since well test data have not been available after June 2010.

As the results are based on models and configuration data that are not necessarily up-to-date, the configuration should be revised by the Snorre organization.

6.7.6.3 Use of inspection and maintenance data in ErosionInsight

Use of data from a fully working Integration Layer will add considerable value to Condition based Inspection and Maintenance. The benefits of this and the work that still remains is described under Future work in section 6.8.3. Section 6.4.6 describes the ABB work to retrieve data from the Integration Layer. Below is discussed some of the benefits from this.

6.7.6.3.1 Benefits and economic potential

Making inspection and maintenance data available from the ErosionInsight graphical user interface can add much value. The most appealing solution would be that these data are made available by right-clicking on different symbols in the time series plot. A drop-down menu shows up where the user can select inspection data types, e.g. a photo of the choke, like in the example in figure 6.43. Different event types (choke change, inspection, sand event, maintenance etc.) should have different symbols.

The very large benefit from this will be that those responsible for sand and erosion management for production optimisation can compare the erosion results with observations, and the observations can also be used to qualify and calibrate the erosion models and configuration data. The erosion models will then be more accurate and reliable. And better models combined with a closer supervision can facilitate more use of ASR (Acceptable Sand rate) for the Snorre installation and increased production.

Those responsible for Inspection Optimisation will have all information from the inspection and maintenance history, as well as erosion monitoring results easily available to support decisions about Condition based Inspection and Maintenance instead of inspection and maintenance following a fixed schedule.

6.7.7 Work performed and results achieved – DNV

As part of the 'Condition based Inspection and Maintenance' use case activity new functionalities have been implemented in the DNV-EMA and other functionalities have been upgraded. The main purpose of introducing the new functionalities has been to demonstrate how inspection and maintenance activities can be optimised by combing results obtained from the erosion calculators and data from various sources.

A summary of special features which have been developed/updated as part of the 'Condition based Inspection and Maintenance' use case are given below.

6.7.7.1 Colour codes

The functionality of colour codes 'traffic lights' has been updated and extended. The 'traffic light' now includes 5 categories where the limits can easily be changed to adapt to special requirements for a development; an example is shown in figure 6.44.

6.7.7.2 System overview

The 'front page' shows a schematic overview of the development/model and shows immediate status with respect to, for example, accumulated erosion for the various pipe segments; see figure 6.45 utilising the colour codes shown in figure 6.44.

Also other critical parameters can be selected; such as mixture velocity, erosion rate and difference between measured and estimated Cv.

6.7.7.3 Status report

The functionality of automatic generation of a Status report has been developed. An example of a status report for Snorre B is shown in figure 6.46. The status report utilises the same colour codes for the traffic lights as used for the front page (figure 6.45) and lists the main results and operational conditions.

The status report may easily be extended and/or tailor made to cover specific requirements from different developments

The status report can also give indications on time to next inspection based on for example present erosion rates. Alternatively, time to next inspection may be determined utilising forecast production profiles.

6.7.7.4 Inspection data

The present version of the software accommodates only inspection data manually inserted into the software; see figure 6.47. Some minor modifications will be required to allow for inserting inspection data retrieved from the Integration layer into the Inspection data sheet.

6.7.7.5 Reset of erosion data/replacement of a component

The software has the functionality to reset calculated erosion results either due to results from inspection or due to replacement of a component/choke. This is done by the 'new value' functionality shown in figure 6.47.

6.7.7.6 Case example

Figure 6.45 shows accumulated erosion in the range between 0.1-1mm for TL-5. TL-5 is a 5" multi- purpose line which is also used for routing of production from the C-template wells to the Snorre B platform. An indication of accumulated erosion in the range 0.1-1mm would have triggered inspection in case inspection had been planned utilizing erosion monitoring software as decision support.

Utilizing production data information (retrieved from the Integration layer) and results from the erosion software a more detailed investigation of the erosion development and root

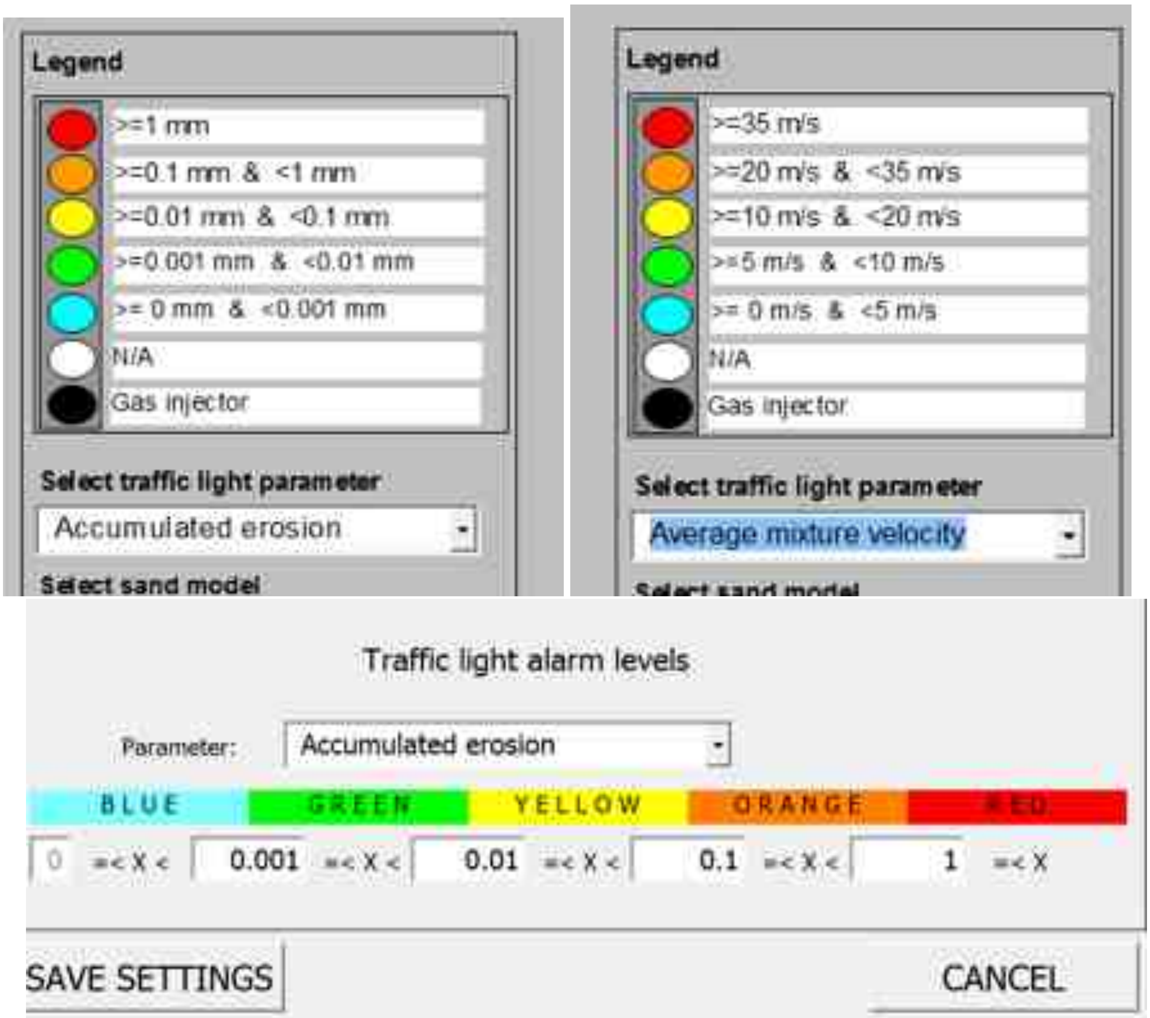


Figure 6.44: Example of traffic light functionalities and functionality for changing traffic light set points

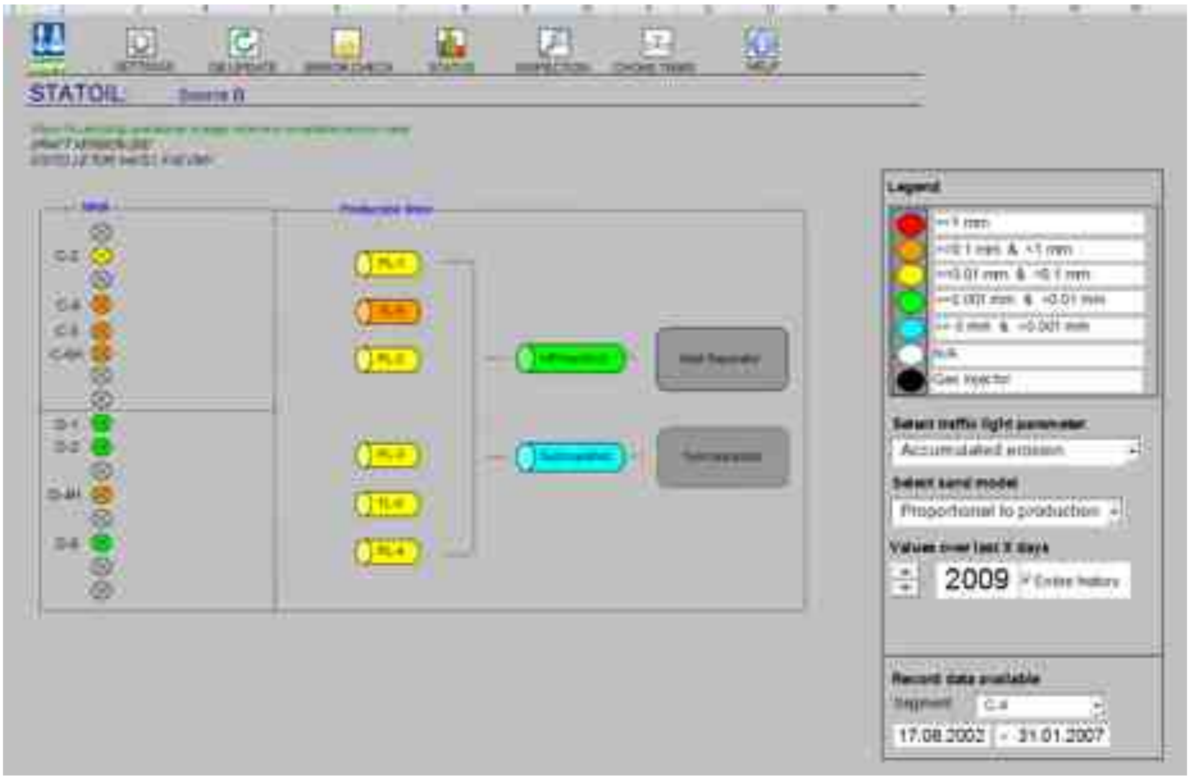


Figure 6.45: 'Front page' showing system layout of Snorre B and status with respect to accumulated erosion for critical components

Figure 6.46: Example of automatically generated status report

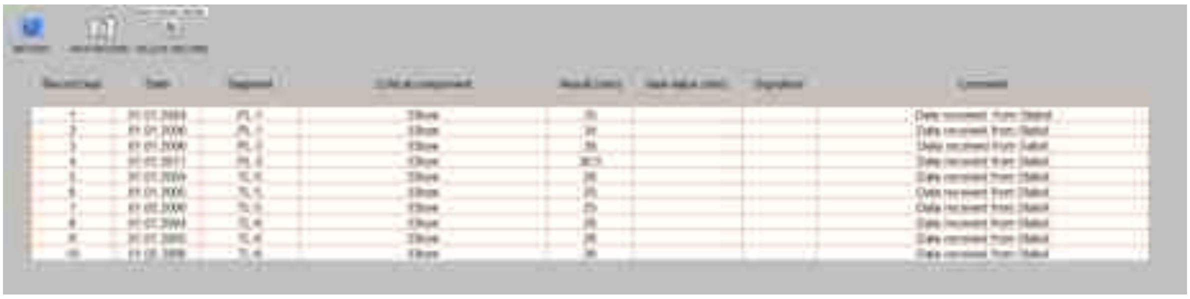


Figure 6.47: Example of Inspection data sheet with inspection data for the Snorre B flowlines - received from Statoil. (Only year inspection performed known. 1st January set as default)

cause of the erosion can be performed. Figure 6.48 shows that high erosion rates are predicted during the first 6-7 months of 2004 resulting in an increase in accumulated erosion by 0.5mm within 9 months.

Further investigations of available data; figure 6.49 shows that the high erosion rates during early 2004 are caused by very high velocities (>30m/s), which is again attributed to high GOR; i.e. due to gas break through. The increased gas production can further be linked to specific well(s) by looking at the production data/routing for the wells.

Inspection data received from Statoil for TL-5; see figure 6.47, show reduced wall thickness by 1mm from 2004 to 2005 supporting the results from the prediction. It should be noted that inspection has been performed utilizing ultrasonic wall thickness (UT) measurement technology. Normally, such a technology is experienced to have accuracy between 0.5 and 1mm.

6.7.7.7 Economic potential

Utilization of erosion simulations as basis for inspection planning may have significant economic impacts. For topside systems the volume of inspection work may be reduced by performing inspection in accordance with the estimated erosion from the software. Especially for inspection of chokes this will have large impact as inspection of chokes requires shutdown of the well and dismantling of the choke.

For subsea systems, the implications of optimization of inspection will be significantly larger than for topside. Inspection of subsea systems generally require shut down of the well and retrieval of the well-head/choke bridge. Normally, inspection of subsea systems is performed as part of other activities like planned well work-over, during Maintenance Shut Down, etc. and the cost for performing the extra inspection is not huge. However, in case an inspection of a subsea system has to be performed as a stand-alone operation, the cost will be high resulting from loss of production for some period of time, hire of vessel for retrieval and installation of subsea components/system and possible waiting time due to weather conditions. Typical cost for an extra intervention is assessed to be in the order 10-20MNOK.

6.8 Conclusion

6.8.1 Successes

Several successes have been achieved in the Production Pilot. This chapter summarizes results as seen from the different case studies as well as from an data integration point of view.

6.8.1.1 Integration Layer and data model

The Production Pilot has successfully described the sand management domain. This work is gathered in a domain nomenclature extended with information templates and a topological model of Snorre B. The descriptions are standardized by linking the domain classes to the relevant counterparts in ISO 15926 Oil and as Ontology. Some of the experiences working with the Integration Layer are:

- Applications have increased operability due to the defined model and the data access methods through the Integration Layer.

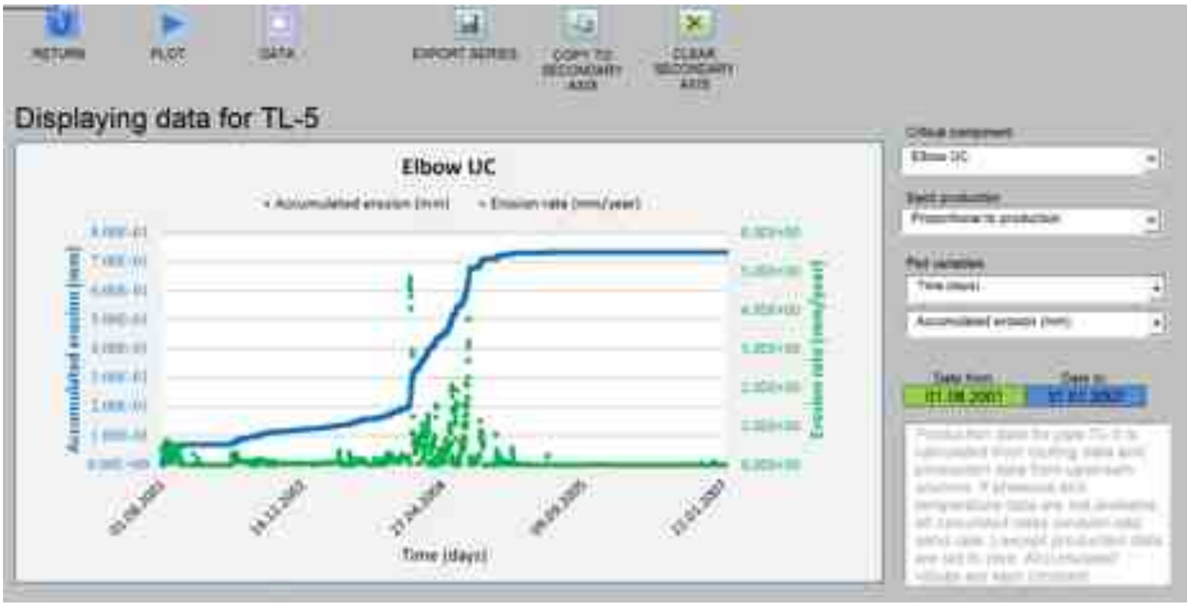


Figure 6.48: Details of erosion rate (mm/year) and accumulated erosion (mm) for TL-5 for the period 2001 to 2007

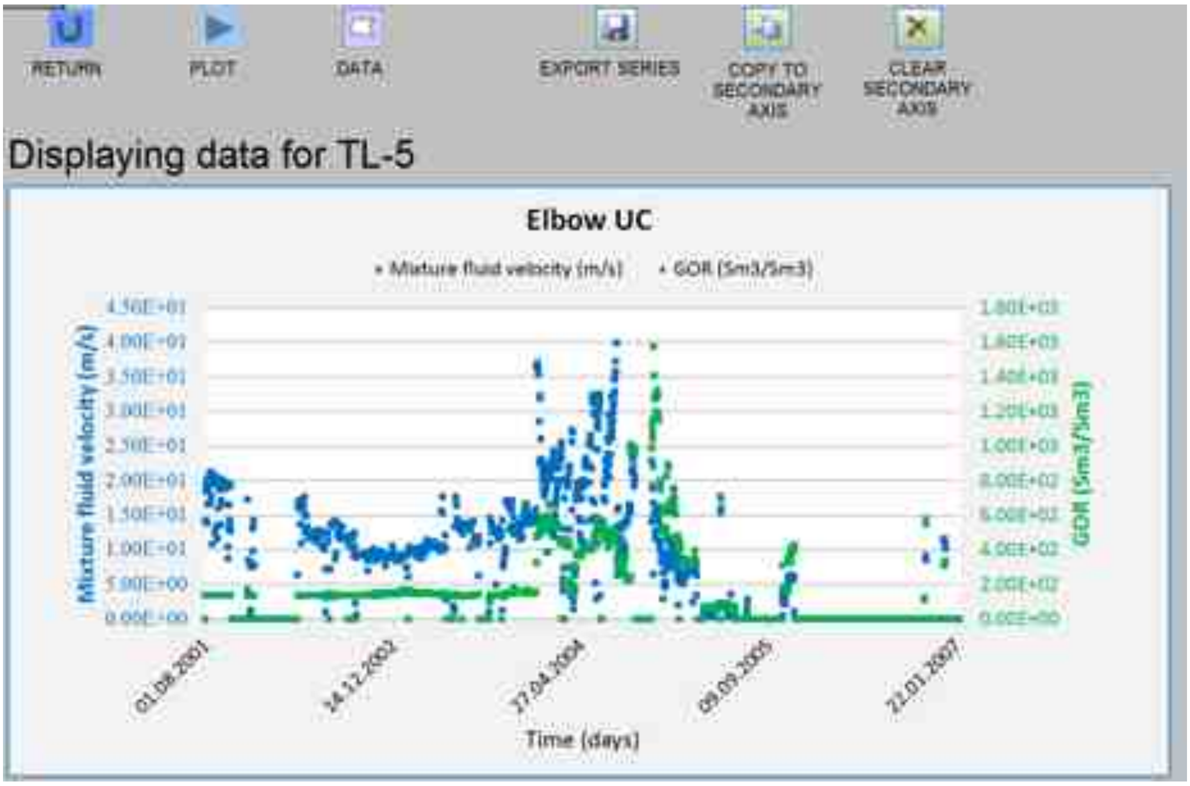


Figure 6.49: Details showing fluid velocity (m/s) and accumulated GOR (Sm3/Sm3) for TL-5 for the period 2001 to 2007.

- The time spent on integration when the model is defined and ready is perceived to be less than when having no predefined standard.

- Participants have gained first-hand experience by the use of Semantic standards and technologies.

- The clear definitions decrease the level of confusion when talking across different domains like for vendors and suppliers.

- Results are made available to other applications that can take advantage of the information.

6.8.1.2 Sand Detection

For the Sand Detection Case, sand rates were calculated in a centralized application, sand bursts were detected and made available in the Integration Layer and a prototype for identifying different cases of sand production were presented. Lastly and most importantly, methods and a successful concept for collaborating actively between two and more applications were established.

The overall results can be summed up as:

- It is feasible to calculate sand rate at an IMS system level. This also means that only a single point is used for configuring and maintaining calibration is required. Based on system setup, parts of formulas can be configured based on different measurement configurations.

- Detection of sand events can be performed at an IMS system level and removing it from the HMI. This can potentially remove some disturbances for the operators.

- Potential sand events can be categorized and the underlying cause for increased noise in acoustic sand measurements can be determined. A Naive Bayesian classifier for categorizing sand events has been tested with promising results.

- The concept of communication and updating the same data set in the Integration Layer can be reused for other applications and is not limited for exclusive use by the current IOHN implementations.

6.8.1.3 Erosion Monitoring

Figure 6.50 summarises how the interoperability of different data sources and data flow from the Integration Layer can be applied in combination with use of erosion monitor systems to optimise production, inspection and maintenance activities.

The erosion monitoring software systems have up until present been extensively applied for production optimisation, while the services with respect to inspection and maintenance optimisation have been formalised and more streamlined as part of the IOHN project.

The integration of operations and inspection data improves the decision support the erosion applications can provide. This would be valuable to several disciplines and areas of responsibilities within an oil company. In Statoil the relevant departments would be Operasjon og Produksjon Styring (OPS), Petroleumsteknologi (PETEK) and Process Data Management (PDM) for the production area, AnleggsIntegritet (AI) for the inspection and maintenance area and the central Production Support Centre (PSC). We believe the extended information could also facilitate increased collaboration between these departments.

The work described in chapters 6.6 and 6.7 has demonstrated the potential strength of erosion monitoring software systems when utilised in combination with access of data from the Integration Layer:

- To assess/follow up the status of the system with respect to operational acceptance criteria (velocity, erosion rate, sand rate, etc.) and status/planning with respect to inspection and maintenance (accumulated erosion, Cv drifting).

- Readily access to all relevant production data in case of ‘Root cause’ evaluations are required.

- Closely follow up of production and erosion development from each well to enable production optimisation (ASR strategy).

- Enable identification of most critical components with respect to erosion rates and accumulated erosion.

- Enable prioritizing and optimisation of inspection and maintenance activities.

- Readily get access to inspection data from previous inspections and corresponding operational data.
- Improve system overall safety as a more close follow up of the status of the system is feasible.
- May have a large economical potential both with respect to production optimisation and inspection and maintenance optimisation.

6.8.2 Lessons Learnt

During the work with the Integration Layer, modeling and interoperability, some of the lessons learnt are summarized as follows:

- There is beyond doubt a need for well formulated and clear definitions of data in the Oil & Gas Industry.
- The work of describing the domain was more time-consuming than originally expected.
- It might be more beneficial to use data addresses to "raw data" through the integration layer instead of instancing it in the integration layer.
- It might be beneficial to use Open Linked Data or similar to annotate data points in Web Services in order to clarify beyond doubt what kind of data that is transferred between systems.

6.8.3 Further work

This chapter describes potential further work for the integration layer and the systems involved in the Cases for the Production Pilot.

6.8.3.1 Integration Layer

The Integration Layer by today includes operational data, inspection data and topological and design data. Additionally, the following 'data' should be available from the Integration Layer:

- Failure investigation reports including photos, recorded values, textual description
- Maintenance and replacement history/documentation
- Sand event history
- Tag names for sensors and components
- Results from application systems to be made available to other users via the Integration Layer; for example by implementing the Status reports in the Integration layer

It is critical that the Integration Layer always offers complete and updated information. It must integrate the last updated information from all different data sources used to store data in the company. Also, the need for data access to the potential large amounts of raw data that is present in IMS systems has its own challenges. As an alternative to the present implementation, one could have considered only providing URL's or similar to refer to the actual data source and data field name for separate processing. A protocol like OPC-DA, OPC-HDA or OPC-UA could then have been used "as-is". The scenario for a data source provider would be to provide input for the data mapping from the semantic term (like "Wellhead Pressure for Well X on Field Y") and give an OPC-address and a tag number for query. This would greatly offload the integration layer processing, and could potentially ease the manual configuration if it were maintained "automatically" by the vendors that has systems close to the data source and within the company's domain. The maintenance of system configuration is a real challenge, so the redirect functionality would allow some of the data sources in the underlying systems be able to updated itself and potentially decrease the need for manual configuration updates.

When considering work to improve the Integration Layer, one also need to consider some questions around processes for establishing and working with these models. Many of these subjects need further investigation, some of them being:

- How to define if an application is compliant with ISO-15926.
- Ownership of created models.
- Cost-resource analysis of implementation of semantic models.
- Maintenance strategies for models and their instances.
- Interaction should also work without a separate integration layer. Check the feasibility of this approach.

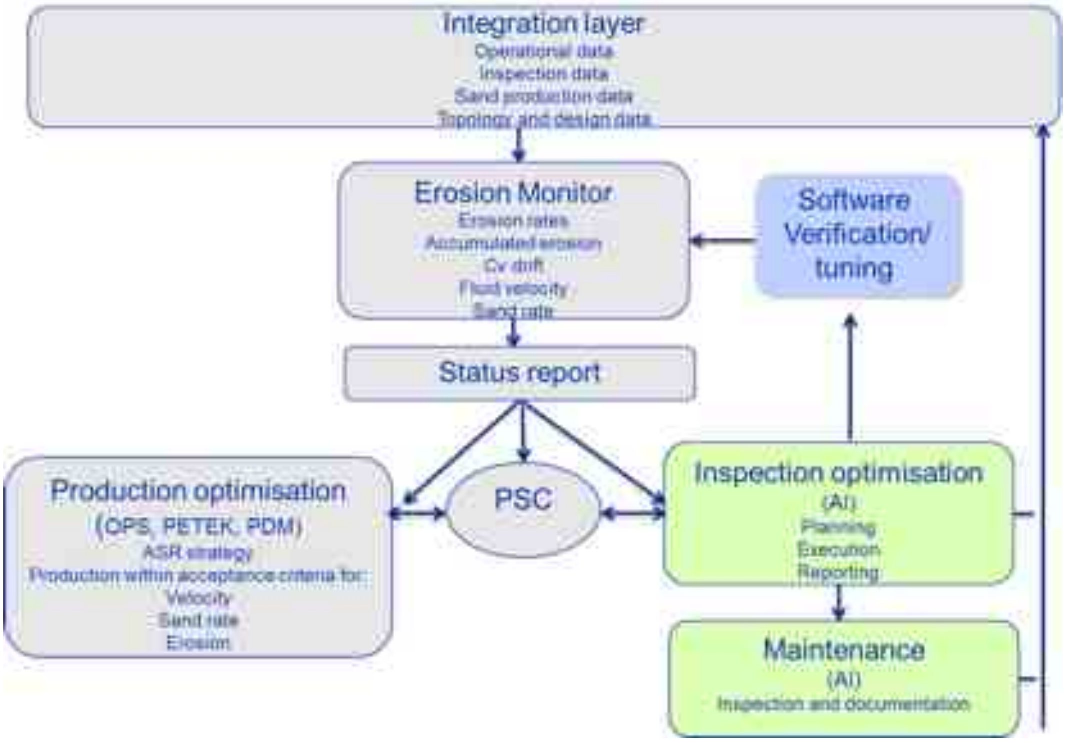


Figure 6.50: Schematic layout demonstrating data interoperability, data flow and use of erosion monitor systems as input to enhance collaboration between various disciplines within an oil company (Example from Statoil)

6.8.3.2 Sand detection

For the sand detection, the two different systems used was a prototype for sand detection from Epsis and implementations in the Siemens PIMAQ Framework. For the Siemens PIMAQ formulas and applications, the following items would be required for improvement and further usage of the Integration Layer:

- Create the SPARQL adapter for the PIMAQ database. This was not completed during the IOHN project.
- Perform better detection of sand using raw values. The current implementation uses calculated sand rates that has more error sources.
- Do automatic calibration on stable periods to minimize the need for manual maintenance.
- Expose parts of a complete model that can be queried by a central integration layer and keep up-to-date information.

The Epsis Decide implementation is currently a proof of concept, but improvements can be made by

- further investigate the statistically significant characteristics of different timeseries
- extending the number of sand event categories and improve the boundaries between them
- investigate more sophisticated classification or learning schemes.

6.8.3.3 Erosion Monitoring

Further work for Erosion Monitoring are listed for the different products used in this activity.

6.8.3.3.1 ABB ErosionInsight

In sections 6.6.6 and 6.7.6 are described set-up, configuration and results for the use of the ErosionInsight software for erosion monitoring at Snorre A, B and UPA. To fully take advantage of the work that has been done and the results obtained, the configuration and set-up should be revised by the Snorre organization to assure that everything has been set up correctly and with data that are up-to-date. It must also be verified that the configuration is set up according to the Snorre sand and erosion management strategy. An effort should be made to make data missing in the EC database views used by ErosionInsight available. Examples are: Well test data, choke types and choke history, sand rate measurements from before February 2009 and sensor data for flowlines before January 2011

In addition, for ErosionInsight to take advantage of the new data types that should be provided by the Integration Layer, the following improvements should be made to the application:

- The Integration Layer can offer more operational data than the Energy Components (EC) database views currently in use, and these data can be used to monitor erosion for new locations. The current ErosionInsight SPARQL module prototype currently collects the same data from the Integration Layer that were already available from the EC database views, but ErosionInsight should be extended to benefit from the flexibility offered by the Integration Layer.

- ErosionInsight should be extended to write results back to the Integration Layer – or – the Integration Layer should be able to collect results from the application. In this way, the results will be available for other applications accessing the Integration Layer.

- During the proposed Auto-generation of erosion model from data in the Integration Layer (see below), tag names should also be automatically imported from the Integration Layer to identify components and sensor values to make the application more useful to the inspection/maintenance engineers.

- ErosionInsight should be extended to make inspection and maintenance data from the Integration Layer easily available from the graphical user interface. The most appealing solution would be that these data are made available by right-clicking on different symbols in the time series plot. A drop-down menu shows up where the user can select inspection data types, e.g. a photo of the choke. Different event types (choke change, inspection, sand event, maintenance etc.) should have different symbols. Examples are shown in section 6.7.6.

- ErosionInsight should be extended to generate automatic status reports. Currently, status reports must be written manually. Status reporting could also include generation of warnings sent by either e-mail or SMS, but in this case it is very important that any false alarms are eliminated.

6.8.3.3.2 DNV-EMA

Below is given a summary of additional development required to fully make the DNV-EMA software to exploit the potential of the Integration layer.

- The present version of the software support only manual transfer of inspection into the Inspection data sheet. Some minor programming is required to fully support inspection data transfer from the Integration layer.

- The software does not have the functionality to utilise design data; e.g. pipe dimensions and radius of curvature, from the Integration layer as input in the software. Some minor programming is required to fully support design data transfer from the Integration layer to software.

- The software has a model for calculating choke Cv. However, in the present version there is no functionality that utilise the difference between measured and estimated Cv to a tool to assess erosion in the chokes. The reason is that the chokes installed at Snorre B are not prone to erosion of the ports in the trim. This functionality can, however, be implemented with some minor work.

- The software has no general functionality for calibration and tuning of the model based on inspection data. The functionality of re-setting of values, for instance due to replacement of a component or based on inspection result, is however, available.

6.8.3.3.3 Auto-generation of erosion model from topology model in Integration Layer

Ideally the Integration Layer could provide all necessary configuration and topology information, so that the erosion monitoring software applications themselves do not need to store this information. It is also a requirement that this information is complete and always up-to-date.

Ideally, the optimal functionality would be that the applications could be able to auto-generate a model based on the topology and configuration information found in the Integration Layer. It is, however, assessed that the latter functionality would require extensive work and programming to be up running.

6.8.4 Dissemination

The presentations were in general held in collaboration with the Integration Platform and Semantic Model, activities. The IOHN project has been presented at more conferences than listed below, but these are venues where the Production Pilot had a major part.

- Semantic Days, 2010, 2011
- Intelligent Energy 2010
- IFEA: Integrerte operasjoner i olje- og gassindustrien, 2010

6.9 References

[1] IOHN Activities Production Pilot and Semantic Model. Snorre uml, December 2011. Available at https://www.posccaesar.org/browser/projects/IOHN/Activity%206%20%28R%26P%29/Sand%20production/SnorreUML.vsd.

[2] J. Andrews, H. Kjørholt, and H. Jøranson. SPE 94511 Production enhancement from sand management philosophy. A Case Study from Statfjord and Gullfaks. In *SPE 6th European Formation Damage Conference, Scheveningen, Netherlands*, May 2005.

[3] dotNetRDF Website. An open source .net library for rdf, 5 2012. Available at http://www.dotnetrdf.org.

[4] Machine Learning group of University of Waikato. Weka. Available at http://www.cs.waikato.ac.nz/ml/weka/.

[5] K. Hovda and K. Lejon. Effective Sand Erosion Management in Chokes and Pipelines. In *Sand Control and Management 2008, Kuala Lumpur*, January 2008.

[6] O. Kvernvold and K. Hovda. IOHN Activity 6 Condition Based Maintenance Strategy. IOHN project internal report, 2011.

[7] O. Kvernvold, L.E. Torbergsen, R. Eriksen, and H. Kjørholt. New strategy for sand management for safely improving production performance, 2002. Presented at DOT 2002.

[8] L Nøkleberg and K Hovda. Specification: PTI Insight online erosion management system, Snorre A, B and UPA. Technical report, PTI report, 2005.

[9] Tom M. Mitchell. *Machine learning*. McGraw-Hill, 1997.

[10] DNV Recommended Practice RP O501. Erosion wear in piping systems. revision 5.0-2011. Technical report, Det Norske Veritas AS, 2011.

[11] Oddmund Kvernvold and Lars Even Torbergsen. DNV 2004-1181, Snorre B - Design and Operational Data. Technical report, DNV, December 2004.

[12] T. K. Perkins. Critical and subcritical flow of multiphase mixtures through chokes. *SPE Drilling & Completion 8(4)*, 8(4):271–276, December 1993.

[13] Trond Unneland and Mike Hauser. Real-time asset management: From vision to engagement – an operator's experience. *Journal of American Statistical Association*, 90:928–934, 1995.

[14] Ian H. Witten, Frank Eibe, and Mark A. Hall. *Data Mining: Practical Machine Learning Tools and Techniques*. Elsevier, 2011.

6.10 List of deliverables

In the following list, preliminary or intermediate versions of deliverables have been left out.

The deliverables are available from the password protected project-internal WIKI-site: http://www.posccaesar.org/wiki/IOHN/Internal.

Title	Date	Contributors	Type
Production pilot description	2009/09	Bård Henning Tvedt(Epsis), Jennifer Sampson(Epsis) and Randi-Helene Halmøy(Epsis)	Report
Sand detection use case description	2009/09	Bård Henning Tvedt(Epsis), Jennifer Sampson(Epsis) and Randi-Helene Halmøy(Epsis)	Report
Erosion management use case description	2009/09	Oddmund Kvernvold(DNV) and Knut Hovda(ABB)	Report
Maintenance Extension Description	2010/05	Bård Henning Tvedt(Epsis), Knut Hovda(ABB), Oddmund Kvernvold(DNV) and Kjell Lejon(Statoil)	Report
System requirements Reservoir and Production	2010/10	Knut Hovda(ABB), Espen Breivik(Siemens) and Bård Henning Tvedt (Epsis)	Report
Ontology assertions Reservoir and Production	2010/12	Bård Henning Tvedt(Epsis)	Report
Condition Based Maintenance Strategy	2011/01	Oddmund Kvernvold(DNV) and Knut Hovda(ABB)	Report
Production Nomenclature	2011/11	The Production Pilot participants	Excel spreadsheet
Snorre B model	2011/11	Bård Henning Tvedt(Epsis) and Christian M. Hansen (DNV)	Visio model

6.11 Public available material with contributions from the Production Pilot

The Production Pilot contributed to the following material that is publicly available. Links to external public sources are provided where available. Note that this material is not considered being a part of the project deliverables.

Title	Date	Contributors	Type
A demonstration of interoperability based on open standards	2010/03	Knut Hovda (ABB) and Øystein Haaland (IBM)	Presentation Intelligent Energy 2010
Demonstrating the IOHN architecture in the sand and erosion managment domain	2010/06	Knut Hovda (ABB) and Øystein Haaland (IBM)	Presentation Semantic Days 2010

Appendices: Production Pilot

6.A Domain descriptions

6.A.1 Nomenclature

6.A.1.1 Facility components

The terms in this category describe the equipment in the topological model.

6.A.1.1.1 iohn6:AcousticSandDetector

Class name: *Acoustic sand detector*
Definition: *A sand detector using acoustic signalling.*

6.A.1.1.2 iohn6:AdjustableChokeValve

Class name: *Adjustable choke valve*
Definition: *A choke valve that is made with adjustable flow/pressure restriction (i.e. has a valve operator by which it can be adjusted.)*

6.A.1.1.3 iohn6:Artefact

Class name: *Artefact*
Definition: *An inanimate physical object that is made or given shape by man.*

6.A.1.1.4 iohn6:ContinuousFluidTransportDevice

Class name: *Continuous fluid transport device*
Definition: *Superclass for all classes of objects through which fluid can flow continuously.*

6.A.1.1.5 iohn6:DataSource

Class name: *Data source*
Definition: *Superclass for all classes of objects that can provide data of some sort.*

6.A.1.1.6 iohn6:ExpansionJoint

Class name: *Expansion joint*
Definition: *Flexible joint between two pipes to receive expansion from the pipes.*

6.A.1.1.7 iohn6:FlowMeter

Class name: *Flow meter*
Definition: *A detecting instrument intended to measure and indicate flow rate and/or produce a signal which represents the measured flow rate.*

6.A.1.1.8 iohn6:Flowline

Class name: *Flowline*
Definition: *A pipeline carrying oil, gas or water that connects the wellhead to a manifold or to production facilities, such as heater-treaters and separators*

6.A.1.1.9 iohn6:Flowline_10inch_Sch120

Class name: *10" Flowline - Sch 120*
Definition: *Class of flowlines of 10 inch diameter that follows the specifications of Schedule 120.*

6.A.1.1.10 iohn6:Flowline_10inch_Sch140

Class name: *10" Flowline - Sch 140*
Definition: *Class of flowlines of 10 inch diameter that follows the specifications of Schedule 140.*

6.A.1.1.11 iohn6:Flowline_18inch_Sch120

Class name: *18" Flowline - Sch 120*
Definition: *Class of flowlines of 8 inch diameter that follows the specifications of Schedule 120.*

6.A.1.1.12 iohn6:Flowline_5inch_Sch120

Class name: *5" Flowline - Sch 120*
Definition: *Class of flowlines of 5 inch diameter that follows the specifications of Schedule 120.*

6.A.1.1.13 iohn6:Flowline_5inch_Sch160

Class name: *5" Flowline - Sch 160*
Definition: *Class of flowlines of 5 inch diameter that follows the specifications of Schedule 160.*

6.A.1.1.14 iohn6:Flowline_6inch_Sch120

Class name: *6" Flowline - Sch 120*
Definition: *Class of flowlines of 6 inch diameter that follows the specifications of Schedule 120.*

6.A.1.1.15 iohn6:Flowline_8inch_Sch120

Class name: *8" Flowline - Sch 120*
Definition: *Class of flowlines of 8 inch diameter that follows the specifications of Schedule 120.*

6.A.1.1.16 iohn6:Manifold

Class name: *Manifold*
Definition: *A fluid container that contains a chamber with several openings. Typically an arrangement of pipes used to redistribute the flow of a fluid or gas, typically from a single inlet to a number of outlets or vice versa.*

6.A.1.1.17 iohn6:OffshorePetroleumField
Class name: *Offshore petroleum field*
Definition: *A petroleum field where oil and gas activities and discharges take place at sea.*

6.A.1.1.18 iohn6:OilAndGasPlatform
Class name: *Oil and gas platform*
Definition: *An oil and gas platform is a large offshore structure used to house machinery needed to facilitate the production of oil and natural gas.*

6.A.1.1.19 iohn6:PigTrap
Class name: *Pig trap*
Definition: *Pig traps are used for inserting pigs into a pipeline then launching, receiving, and finally removing them without flow interruption. Pipeline pigs are devices that are inserted into and travel throughout the length of a pipeline driven by a product flow.*

6.A.1.1.20 iohn6:PlatformName
Class name: *Platform name*
Definition: *The full name of an oil and gas platform.*

6.A.1.1.21 iohn6:PlatformShortName
Class name: *Platform short name*
Definition: *The short name of an oil and gas platform.*

6.A.1.1.22 iohn6:PressureElement
Class name: *Pressure element*
Definition: *A detecting element intended to sense pressure.*

6.A.1.1.23 iohn6:ProductionLine
Class name: *Production line*
Definition: *A subsea to surface line used for production purposes. Can consist of several continuous fluid transport device segments.*

6.A.1.1.24 iohn6:Riser
Class name: *Riser*
Definition: *A physical object that is a vertical or near vertical pipe for the upward flow of liquid or gas.*

6.A.1.1.25 iohn6:RiserHangOff
Class name: *Riser hang-off*
Definition: *The purpose of a riser hang-off system is to provide a structural support between the riser or umbilical and the outer j-tube.*

6.A.1.1.26 iohn6:ScheduleFlowline
Class name: *Schedule flowline*
Definition: *Superclass of all flowline classes following the Schedule specifications.*

6.A.1.1.27 iohn6:SubseaHeader
Class name: *Subsea header*
Definition: *A subsea manifold branch piping for transferring fluids from the branch piping to the flowline(s).*

6.A.1.1.28 iohn6:SubseaTemplate
Class name: *Subsea template*
Definition: *A subsea fixed structure with slots for several wells.*

6.A.1.1.29 iohn6:SubseaToSurfaceLine
Class name: *Subsea to surface line*
Definition: *Superclass of all classes of subsea to surface lines.*

6.A.1.1.30 iohn6:TemperatureElement
Class name: *Temperature element*
Definition: *A detecting element intended to sense temperature.*

6.A.1.1.31 iohn6:TestLine
Class name: *Test line*
Definition: *A subsea to surface line used for test purposes. Can consist of several continous fluid transport device segments.*

6.A.1.1.32 iohn6:ThreePhaseSeparator
Class name: *Three phase separator*
Definition: *A separator intended to separate three fluids in different phases, typically gas and liquid and water.*

6.A.1.1.33 iohn6:Valve
Class name: *Valve*
Definition: *A fluid regulator that can be used to control (i.e. permit, obstruct, regulate or divert) a fluid stream.*

6.A.1.1.34 iohn6:Well
Class name: *Well*
Definition: *A device that is arranged to obtain, produce, store or inject a material into the earth.*

6.A.1.1.35 iohn6:WellHeadModule
Class name: *Well head module*
Definition: *A module that is the assembly from the upper well casing to the production tubing hangers*

6.A.1.1.36 iohn6:Wellbore
Class name: *Wellbore*
Definition: *A 'borehole' drilled in the earths crust to form part of a well.*

6.A.1.2 Measured data types

The terms in this category describe the available measured and derived operational data.

6.A.1.2.1 iohn:DataType
Class name: *Data type*
Definition: *A class of all the different measurement types used in the domain, e.g. Sand rate, Inside pressure, etc. Not to be confused with types as used in computer programming.*

6.A.1.2.2 iohn:GasOilRatio
Class name: *Gas Oil Ratio*
Definition: *A ratio that is the ratio between the produced volume of gas and the produced volume of oil at standard conditions defined to be at pressure of 1atm=1.013bar and 60DegF=15.6DegC.*

6.A.1.2.3 iohn:WaterCutRatio
Class name: *Water Cut Ratio*
Definition: *The ratio of water produced compared to the total liquids produced at standard conditions defined to be at pressure of 1atm=1.013bar and 60DegF=15.6DegC.*

6.A.1.2.4 iohn:VolumeFlowRate
Class name: *Volume Flow Rate*
Definition: *A flow rate which is the volume of matter that crosses a given surface divided by time. (instantaneous rate of volume flow due to a sound wave, ISO 31-7:1992(E))*

6.A.1.2.5 iohn:OilVolumeFlowRate
Class name: *Oil Volume Flow Rate*
Definition: *A volume flow rate of oil given standard conditions defined to be at pressure of 1atm=1.013bar and 60DegF=15.6DegC.*

6.A.1.2.6 iohn:GasVolumeFlowRate
Class name: *Gas Volume Flow Rate*
Definition: *A volume flow rate of gas given standard conditions defined to be at pressure of 1atm=1.013bar and 60DegF=15.6DegC.*

6.A.1.2.7 iohn:WaterVolumeFlowRate
Class name: *Water Volume Flow Rate*
Definition: *A volume flow rate of water given standard conditions defined to be at pressure of 1atm=1.013bar and 60DegF=15.6DegC.*

6.A.1.2.8 iohn:InsidePressure
Class name: *Inside pressure*
Definition: *Force perpendicular to the area divided by area.* $p = dF/dA$ *where dF is the perpendicular component of the force acting on the area element dA. The pressure is positive when the force is acting into the area. The inside pressure is the force acting on the inside area of a container.*

6.A.1.2.9 iohn:WellTestPressureSeparator
Class name: *Well Test Pressure Separator*
Definition: *The pressure recorded in a separator during a well test.*

6.A.1.2.10 iohn:WellTestPressureWellHead
Class name: *Well Test Pressure Well Head*
Definition: *The pressure recorded at the well head during a well test.*

6.A.1.2.11 iohn:WellTestStroke
Class name: *Well Test Stroke*
Definition: *The choke opening used during a well test.*

6.A.1.2.12 iohn:InsideTemperature
Class name: *Inside temperature*
Definition: *The degree or intensity of heat or cold as measured on a thermometric scale, and a measure of whether two systems are relatively hot or cold with respect to one another. Two systems brought into contact will, after sufficient time, be in*

thermal equilibrium and will have the same temperature. Inside temperature refers to the temperature on the inside of a container.

6.A.1.2.13 iohn:SandRate

Class name: *Sand Rate*
Definition: *The amount of sand particles that crosses a given surface divided by time.*

6.A.1.2.14 iohn:SandRateRawAcoustic

Class name: *Sand Rate Raw Acoustic*
Definition: *The raw data produced by an acoustic sand detector.*

6.A.1.2.15 iohn:ValveState

Class name: *Valve state*
Definition: *The state of an open/close valve; open or closed.*

6.A.1.2.16 iohn:Stroke

Class name: *Stroke*
Definition: *Choke opening.*

6.A.1.3 Configuration data types

The terms in this category describe the available configuration data in the topological model.

6.A.1.3.1 iohn:DiameterType

Class name: *Diameter type*
Definition: *A diameter type; inner or outer diameter.*

6.A.1.3.2 iohn:InsideDiameter

Class name: *Inside diameter*
Definition: *A constant representing the inside diameter of a container.*

6.A.1.3.3 iohn:OutsideDiameter

Class name: *Outside diameter*
Definition: *A constant representing the outside diameter of a container.*

6.A.1.4 Value aggregations

The terms in this category describe whether the data are raw measurements or grouped in some way.

6.A.1.4.1 iohn:AggregationType

Class name: *Aggregation type*
Definition: *A class of all types of data aggregation, e.g. averaging, accumulation, etc. over a given time period. Also includes a representation of raw values, i.e., no aggregation.*

6.A.1.4.2 iohn:RawValue

Class name: *Raw Value*
Definition: *Raw unprocessed data values.*

6.A.1.4.3 iohn:AccumulatedValue

Class name: *Accumulated Value*
Definition: *Data values accumulated over a time period.*

6.A.1.4.4 iohn:AveragedValue

Class name: *Averaged Value*
Definition: *Data values averaged over a time period.*

6.A.1.4.5 iohn:DailyAveragedValue

Class name: *Daily Averaged Value*
Definition: *Daily averaged data values.*

6.A.1.4.6 iohn:WeeklyAveragedValue

Class name: *Weekly Averaged Value*
Definition: *Weekly averaged data values.*

6.A.1.4.7 iohn:MonthlyAveragedValue

Class name: *Monthly Averaged Value*
Definition: *Monthly averaged data values.*

6.A.1.4.8 iohn:YearlyAveragedValue

Class name: *Yearly Averaged Value*
Definition: *Yearly averaged data values.*

6.A.1.4.9 iohn:AllocatedValue

Class name: *Allocated Value*
Definition: *Allocated data values.*

6.A.1.4.10 iohn:AllocatedDailyValue

Class name: *Allocated Daily Value*
Definition: *Allocated daily data values.*

6.A.1.4.11 iohn:WellTestDailyValue

Class name: *Well Test Daily Value*
Definition: *Value obtained during a well test and adjusted to daily values.*

6.A.1.4.12 iohn:WellTestDailyAveragedValue
Class name: *Well Test Daily Averaged Value*
Definition: *Average value obtained during a well test.*

6.A.1.5 Units of measurements

6.A.1.5.1 iohn:UnitOfMeasure
Class name: *Unit of measure*
Definition: *A class of all types of unit of measure.*

6.A.1.5.2 iohn:Hour
Symbol: *h*
Definition: *Number of hours.*

6.A.1.5.3 iohn:StandardCubicMetrePerDay
Symbol: *Sm3/d*
Definition: *Standard cubic meter per day.*

6.A.1.5.4 iohn:StandardCubicMetrePerHour
Symbol: *Sm3/h*
Definition: *Standard cubic meter per hour.*

6.A.1.5.5 iohn:CubicMetrePerHour
Symbol: *m3/h*
Definition: *Cubic meter per hour.*

6.A.1.5.6 iohn:StockTankBarrelOilPerDay
Symbol: *Stbopd*
Definition: *Stock tank barrel oil per day.*

6.A.1.5.7 iohn:MillionStandardCubicFeetPerDay
Symbol: *mmscfd*
Definition: *Million standard cubic feet per day.*

6.A.1.5.8 iohn:StockTankBarrelWaterPerDay
Symbol: *Stbwpd*
Definition: *Stock tank barrel water per day.*

6.A.1.5.9 iohn:PoundPerSquareInch
Symbol: *psi*
Definition: *Pound per square inch.*

6.A.1.5.10 iohn:BarAbsolute
Symbol: *bara*
Definition: *Bar absolute.*

6.A.1.5.11 iohn:BarGauge
Symbol: *barg*
Definition: *Bar gauge.*

6.A.1.5.12 iohn:Celcius
Symbol: *deg C*
Definition: *Degrees Celcius.*

6.A.1.5.13 iohn:GramsPerSecond
Symbol: *g/s*
Definition: *Grams per second.*

6.A.1.5.14 iohn:KilogramsPerHour
Symbol: *kg/h*
Definition: *Kilograms per hour.*

6.A.1.5.15 iohn:Count
Symbol: *counts*
Definition: *Counts.*

6.A.1.5.16 iohn:MeterPerSecond
Symbol: *m/s*
Definition: *Meter per second.*

6.A.1.5.17 iohn:Percentage
Symbol: *%*
Definition: *Percentage.*

6.A.1.5.18 iohn:Degrees
Symbol: °
Definition: *Degrees.*

6.A.1.6 Logical types

6.A.1.6.1 iohn:Boolean
Class name: *Boolean*
Definition: *A binary variable taking the values true/false.*

6.A.2 Templates

This section contains an alphabetical listing of the ISO 15926 Templates developed by the IOHN activities Semantic Model and Production Pilot. The following prefixes are used.

Prefix	Namespace
iohn	http://iohn.org/rdl/
iohn6	http://iohn.org/activity-6/rdl/
iohn6tpl	http://iohn.org/activity-6/tpl/
iohn6model	http://iohn.org/activity-6/model/
owl	http://www.w3.org/2002/07/owl#

6.A.2.1 iohn6tpl:ArtefactHasPart

Description: *States that one artefact is logically a part of another artefact.*

Derived property: iohn6model:isPartOf

#	Role	Type
1	iohn6tpl:hasPart	iohn6:Artefact
2	iohn6tpl:hasWhole	iohn6:Artefact

6.A.2.2 iohn6tpl:ArtefactName

Description: *States that a string is a name of an artefact.*

#	Role	Type
1	iohn6tpl:hasArtefact	iohn6:Artefact
2	iohn6tpl:valName	xsd:string

6.A.2.3 iohn6tpl:ChokeChangeEvent

Description: *States that an iohn6:AdjustableChokeValve or the internal parts of an iohn6:AdjustableChokeValve were replaced at a given time (as xsd:dateTime). The type of the new choke is given as an xsd:string value in the property iohn6tpl:valChokeType.*

#	Role	Type
1	iohn6tpl:hasChoke	iohn6: AdjustableChokeValve
2	iohn6tpl:valDateTime	xsd:dateTime
3	iohn6tpl:valChokeType	xsd:string

6.A.2.4 iohn6tpl:ChokeOfWell

Description: *States that a choke is the main choke for a well.*

Derived property: iohn6model:hasChoke

#	Role	Type
1	iohn6tpl:hasWell	iohn6:Well
2	iohn6tpl:hasChoke	iohn6: AdjustableChokeValve

6.A.2.5 iohn6tpl:DataSourceValue

Description: *States that a given iohn6:DataSource has a given data value (in xsd:double) at a given time (as xsd:dateTime) for the given iohn: DataType, iohn:UnitOfMeasure, and .*

#	Role	Type
1	iohn6tpl:hasDataSource	iohn6:DataSource
2	iohn6tpl:hasDataType	iohn:DataType
3	iohn6tpl:hasUnitOfMeasure	iohn:UnitOfMeasure
4	iohn6tpl:hasAggregationType	iohn:AggregationType
5	iohn6tpl:valDateTime	xsd:dateTime
6	iohn6tpl:valDoubleValue	xsd:double

6.A.2.6 iohn6tpl:Downhole

Description: *States that a measuring device is located downhole of a well.*

Derived property: iohn6model:downholeMeasuringDevice

#	Role	Type
1	iohn6tpl:hasWell	iohn6:Well
2	iohn6tpl:hasMeasuringDevice	iohn6:DataSource

6.A.2.7 iohn6tpl:Downstream

Description: *Location of the measuring device relative to choke with respect to flow.*

Derived property: iohn6model:downstreamMeasuringDevice

#	Role	Type
1	iohn6tpl:hasChoke	iohn6: AdjustableChokeValve
2	iohn6tpl:hasMeasuringDevice	iohn6:DataSource

6.A.2.8 iohn6tpl:EndOfLine

Description: *States that a continous fluid transport device is the end of a subsea to surface line.*

Derived property: iohn6model:hasEnd

#	Role	Type
1	iohn6tpl:hasLine	iohn6: SubseaToSurfaceLine
2	iohn6tpl: hasFluidTransportDevice	iohn6:ContinuousFluid-TransportDevice

6.A.2.9 iohn6tpl:FlowConnection

Description: *Represents a connection which allows the flow of fluid through the connected parts from the first argument to the second.*

Derived property: iohn6model:flowsInto

#	Role	Type
1	iohn6tpl:hasUpstreamCFTD	iohn6:ContinuousFluid-TransportDevice
2	iohn6tpl: hasDownstreamCFTD	iohn6:ContinuousFluid-TransportDevice

6.A.2.10 iohn6tpl:InnerDiameterOfIndividual-Pipeline

Description: *States the inner diameter of a particular pipe in millimeter.*

#	Role	Type
1	iohn6tpl:hasPipeline	iohn6:ContinousFluid-TransportDevice
2	iohn6tpl:valDiameter	xsd:float

6.A.2.11 iohn6tpl:InnerDiameterOfPipelineType

Description: *States the inner diameter of a pipeline type in millimeter.*

#	Role	Type
1	iohn6tpl:hasPipelineType	owl:Class
2	iohn6tpl:valDiameter	xsd:float

6.A.2.12 iohn6tpl:InspectionEvent

Description: *states that an object for which the type is a subclass of iohn6:ContinousFluidTransport-Device has been inspected at a given date (iohn6tpl :valDateTime) in order to measure the wall thickness (iohn6tpl:valThickness) with a certain uncertainty value (iohn6tpl:valUncertainty). The wall thickness is given in millimeters, and the uncertainty is given as an absolute value in millimeters; if X is the measured wall thickness and Y is the uncertainty value, then (X +/- Y) is actual wall thickness.*

#	Role	Type
1	iohn6tpl:hasPipeline	iohn6:ContinousFluid-TransportDevice
2	iohn6tpl:valDateTime	xsd:dateTime
3	iohn6tpl:valThickness	xsd:float
4	iohn6tpl:valUncertainty	xsd:float

6.A.2.13 iohn6tpl:MeasurementDevice-Attachment

Description: *A connection of a measuring device to some fluid conducting artefact, which allows some measurement to take place.*

Derived property: iohn6model:attachedTo

#	Role	Type
1	iohn6tpl:hasMeasuringDevice	iohn6:DataSource
2	iohn6tpl: hasFluidTransportDevice	iohn6:ContinuousFluid-TransportDevice

6.A.2.14 iohn6tpl:OuterDiameterOfIndividual-Pipeline

Description: *States the outer diameter of a particular pipe in millimeter.*

#	Role	Type
1	iohn6tpl:hasPipeline	iohn6:ContinousFluid-TransportDevice
2	iohn6tpl:valDiameter	xsd:float

6.A.2.15 iohn6tpl:OuterDiameterOfPipeline-Type

Description: *States the outer diameter of a pipeline type in millimeter.*

#	Role	Type
1	iohn6tpl:hasPipelineType	owl:Class
2	iohn6tpl:valDiameter	xsd:float

6.A.2.16 iohn6tpl:ProductionData

Description: *Derived from instances of the Data-SourceValue template and combinestates values for gas/oil ratio, water cut, oil rate, gas rate, water rate, and on stream hours into a single template statement.*

#	Role	Type
1	iohn6tpl:hasWell	iohn6:Well
2	iohn6tpl:valDateTime	xsd:dateTime
3	iohn6tpl:valGasOilRatio	xsd:double
4	iohn6tpl:valWaterCut	xsd:double
5	iohn6tpl:valOilRate	xsd:double
6	iohn6tpl:valGasRate	xsd:double
7	iohn6tpl:valWaterRate	xsd:double
8	iohn6tpl:valOnStreamHours	xsd:double

6.A.2.17 iohn6tpl:ProvidesDataType

Description: *States that a given iohn6:DataSource provides a data type in a given unit of measure with a given aggregation type.*

#	Role	Type
1	iohn6tpl:hasDataSource	iohn6:DataSource
2	iohn6tpl:hasDataType	iohn6:DataType
3	iohn6tpl:hasUnitOfMeasure	iohn6:UnitOfMeasure
4	iohn6tpl:hasAggregationType	iohn6:UnitOfMeasure

6.A.2.18 iohn6tpl:RadiusOfCurvature

Description: *The radius of curvature of a given pipe is the product of the length of the given diameter type and the factor.*

#	Role	Type
1	iohn6tpl:hasArtefact	iohn6:Artefact
2	iohn6tpl:hasDiameterType	iohn:DiameterType
3	iohn6tpl:valFactor	xsd:float

6.A.2.19 iohn6tpl:SandEvent

Description: *States that a sand event with a given unique sand event id for the given sensor started at the given xsd:dateTime time stamp.*

#	Role	Type
1	iohn6tpl:valSandEventId	xsd:integer
2	iohn6tpl:hasDataSource	iohn6:DataSource
3	iohn6tpl:valDateTime	xsd:dateTime

6.A.2.20 iohn6tpl:SandEventEnd

Description: *States that the sand event with the given sand event id ended at the given xsd:dateTime time stamp. It is implicitly required that there exists an instance of the SandEvent template with the same sand event id, in which the sensor resource and start time stamp is recorded.*

#	Role	Type
1	iohn6tpl:valSandEventId	xsd:integer
2	iohn6tpl:valDateTime	xsd:dateTime

6.A.2.21 iohn6tpl:SandEventProcessing

Description: *States that the sand event with the given sand event id is processed in order confirm whether it is a true sand event, the result of which is represented by the provided boolean value. It is implicitly required that there exists an instance of the SandEvent template with the same sand event id, in which the sensor resource and start time stamp is recorded.*

#	Role	Type
1	iohn6tpl:valSandEventId	xsd:integer
2	iohn6tpl:valIsTrueSandEvent	xsd:boolean

6.A.2.22 iohn6tpl:StartOfLine

Description: *States that a continous fluid transport device is the start of a subsea to surface line*

Derived property: iohn6model:hasStart

#	Role	Type
1	iohn6tpl:hasLine	iohn6: SubseaToSurfaceLine
2	iohn6tpl: hasFluidTransportDevice	iohn6:ContinuousFluid-TransportDevice

6.A.2.23 iohn6tpl:Upstream

Description: *Location of the measuring device relative to choke with respect to flow.*

Derived property: iohn6model:upstreamMeasuringDevice

#	Role	Type
1	iohn6tpl:hasChoke	iohn6: AdjustableChokeValve
2	iohn6tpl:hasMeasuringDevice	iohn6:DataSource

6.A.2.24 iohn6tpl:WallThicknessOfIndividual-Pipeline

Description: *States the wall thickness of a particular pipe in millimeter.*

Derived property: iohn6model:

#	Role	Type
1	iohn6tpl:hasPipeline	iohn6:ContinousFluid-TransportDevice
2	iohn6tpl:valThickness	xsd:float

6.A.2.25 iohn6tpl:WallThicknessOfPipeline-Type

Description: *States the wall thickness of a particular pipeline type in millimeter.*

Derived property: iohn6model:

#	Role	Type
1	iohn6tpl:hasPipelineType	owl:Class
2	iohn6tpl:valThickness	xsd:float

6.A.2.26 iohn6tpl:WellEquipment

Description: *Derived property–states that the artefact is equipment for the well.*

Derived property: iohn6model:hasEquipment

#	Role	Type
1	iohn6tpl:hasWell	iohn6:Well
2	iohn6tpl:hasArtefact	iohn6:Artefact

6.A.2.27 iohn6tpl:WellForPlatform

Description: *States that a iohn6:Well is a well for a iohn6:OilAndGasPlatform.*

Derived property: iohn6model:hasPlatform

#	Role	Type
1	iohn6tpl:hasWell	iohn6:Well
2	iohn6tpl: hasOilAndGasPlatform	iohn6: OilAndGasPlatform

6.A.2.28 iohn6tpl:WellNameEvent

Description: *states that an well has changed its name at a given date/time.*

#	Role	Type
1	iohn6tpl:hasWell	iohn6:Well
2	iohn6tpl:valDateTime	xsd:dateTime
3	iohn6tpl:valNewName	xsd:string

6.A.2.29 iohn6tpl:WellTestEvent

Description: *States that a well test was conducted on a given well at a given date/time.*

#	Role	Type
1	iohn6tpl:hasWell	iohn6:Well
2	iohn6tpl:valDateTime	xsd:dateTime

6.A.2.30 iohn6tpl:WellTestValue

Description: *States that a well test produced a specific test value, but it is not known which data source was used to produce the value. The value is hence connected to the well directly. The first argument is the well that was tested, the second argument is the date/time for the test, and the third argument is the test value produced. The type of measurement (rate/pressure/etc) is given in the fourth argument, the unit of measure in the fifth argument, and the aggregation type in the sixth argument. Note that the value in the third argument may be adjusted/corrected from the actual readout of the data source.*

#	Role	Type
1	iohn6tpl:hasWell	iohn6:Well
2	iohn6tpl:valDateTime	xsd:dateTime
3	iohn6tpl:valDoubleValue	xsd:double
4	iohn6tpl:hasDataType	iohn6:DataType
5	iohn6tpl:hasUnitOfMeasure	iohn6:UnitOfMeasure
6	iohn6tpl:hasAggregationType	iohn6:AggregationType

6.A.2.31 iohn6tpl:WellboreOfWell

Description: *States that a well has a wellbore.*

Derived property: iohn6model:hasWellbore

#	Role	Type
1	iohn6tpl:hasWell	iohn6:Well
2	iohn6tpl:hasWellbore	iohn6:Wellbore

6.B Siemens IMS System (PIMAQ) and implemented modules

6.B.1 Overview and Architecture

The IMS system component from Siemens is called PIMAQ (Plant Information Management and Data Acquisition System). This consists of two main parts, namely the Historian which collects all process data for all tags in the field, and the application layer. Usually for the Siemens Oil & Gas segment, the historian is implemented using Aspentech Infoplus.21. This is the case for both platforms used for references in this project, namely the Snorre A and Snorre B platforms. All applications can use data for analysis. Basically the historians are set up to store all data that is populatating the operator HMI screens for life of field. This means that exposing the IMS system to other applications provide a comprehensive data source. The challenge is that the addressing for data points is still through field-specific tags, and by leveraging the IMS system or the integration layer, the usage of these types of data sources would be improved.

For special analysis and for generating data, separate implementations that can be employed in different scenarios. For the use case in the IOHN project, some new algorithms and applications were needed. This appendix will describe what kind of implementations that were done as part of the project. Using a highly modular approach, components can be re-used in other projects and applications. Formulas are built as tree-structures to perform wanted results.

6.B.2 Implemented Applications

For the sand detection use case, the following applications were implemented using the PIMAQ Application Framework:

6.B.2.1 Sand Data Manipulation Function

This application is the main engine for calculating the sand data values, and also to provide an interface to control calibration tables and other parameters that is needed for this calculation. When running scheduled, the following functions are run and the following state is saved:

- Calculation: Last successful time of writing of engineering values for each sensor
- Data mining: Last run, status if the algorithm ended inside a burst. This will be reported at next run.

The application has the following main User Interface elements:

- Calculation parameters - For running calculations manually and adjust parameters.
- Calibration tables - Adjust calibration tables for sensors manually. *(See figure 6.52)*.
- Well Tests - Review and manipulate Well test parameters manually.
- Data Mining - Review and run search for sand events.

The function will get a list of all configured sand sensors that has been configured in the IMS system. It will read an interval from a specified field of the object that is configured to calculate the engineering values for the particular interval. This is read manually from the data provider and the resulting intervals are used for the purpose of saving it into a different field of the database. After calculating the engineering values, these will be written back to the historian. The calibration table is currently a master calibration table that contains lookup data for all sensors, separated by a key that is the equivalent of the sensor name. When manipulating the master calibration table for a specific value, the value will be updated and the calibration editing audit log will be updated. The calibration table is stored in IMS global relational database storage, with tag and velocity as the primary keys. The audit log is also stored in the global storage.

The data mining function will get a list of all configured sand sensors. It will use the configured formula tree to calculate all the actual values. The data returned from the formula described in section 6.B.3.4 are time intervals that are considered excitations for sand detection. If the data mining is run in scheduled mode, it will also generate a notification in the PIMAQ framework. This will allow subscribers like a remote Web Service to write or update notifications in an integration layer that a burst of sand data has occurred. If not scheduled and only invoked through the GUI, no notification should be generated unless explicitly invoked by the 'Notification' function.

When scheduling, the state of the data mining should be stored in case the routine is in the middle

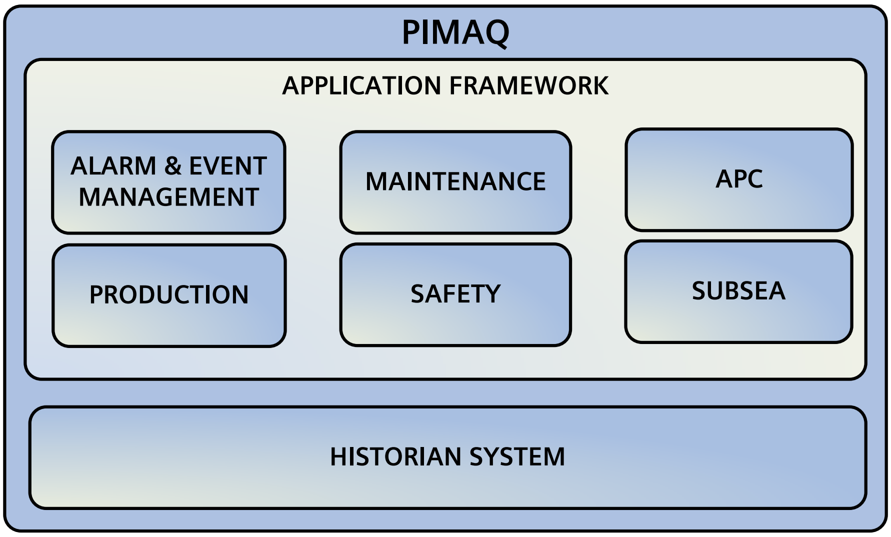


Figure 6.51: Siemens PIMAQ Components

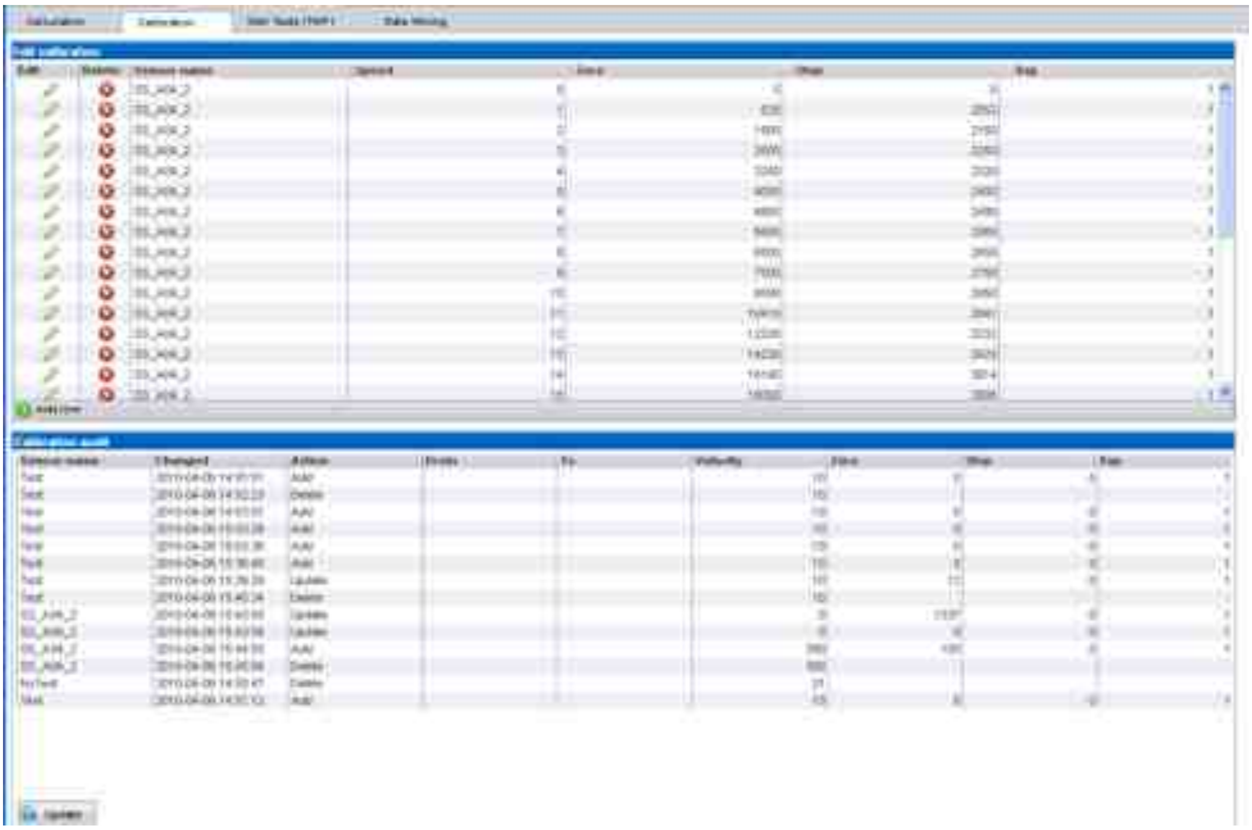


Figure 6.52: Sand Sensor calibration tables in PIMAQ

of the burst. The burst will not be reported until the next run. When running data mining on an interval which already has been searched through, the debug version will delete every interval in the data mining results table that has either start or end time in the new data mining interval. The release version will just leave the intervals in the table.

A correction of sand rates happens when on of the following two cases occur:

- A known external event has trigged high sand rate detection, e.g. slugging will produce fluctuations in pressure that will affect the calculated sand rates.
- The velocity-zero-step table is not calibrated correctly giving incorrect sand rates.

These two cases will be identified by an entity external from the sand data manipulation function (e.g. from a decision support application). A message is returned sand data manipulation function specifying which interval that should be corrected, which values that actual should have been in the interval, and if the correction is trigged by a known external event or that is due to erroneous calibration. The recalibration is trigged when it is calculated sand production when there is no sand production. For the current use case in the project, the entries are currently not updated automatically.

When correcting the values, the original value will be kept in a separate field, so that one can always see the raw values from the sensor, the originally automatically calculated sand rate and the calculated sand rate with possible corrections.

6.B.2.2 Sand Data Report

This application is mainly for visualizing results from the sand calculations and the data mining. It plots and presents the data that was scheduled and can be used as a stand-alone tool for investigating well tests and calibrations if needed.

The application has the following main User Interface elements:

- Values - Plot sand rate data from any interval or the detected sand events. *(See figure 6.53.)*
- Well Tests - View any Well Tests that is used for calculation of sand rate.
- Calibration - View calibration tables and audit log for any sand sensor.

The functionality that produce these results are described in section 6.B.2.1 and section 6.B.3.

6.B.3 Implemented Formulas

The following formulas were implemented in the PIMAQ Application Framework for supporting calculations:

6.B.3.1 SAND_RATE

This formula implements a sand rate in metres per second as described in [1]:

$$SandRate = \left(\frac{Raw - Zero(v)}{Step(v)}\right)^{Exp(v)}$$

Based on the raw value that the acoustic sand sensor picks up, for the number of particle counted per second. The algorithm has the following structure:

$$SAND\_RATE(R, Z_v, S_v, E_v)$$

Where R is the raw value, Z, S, E are the corresponding zero, step and exponential values from the calibration tables at the current velocity v. The entries from the calibration are linearly interpolated if an existing velocity lies outside an entry in the table.

6.B.3.2 FLOW_RATE

The algorithm calculates flow rate of a liquid based on the flow vs. pressure polynom:

$$c_2 \times y^2 + c_1 \times y + (c_0 - x) = 0$$

Where x is the pressure, y is the liquid flow and c represents coefficents from Well test results.

The structure is as follows:

$$FLOW\_RATE(x, c_0, c_1, c_2\,[, p_1, s_{\%}, l_{HH}])$$

Where x is the pressure, c are the coefficents from the Well tests, p is the pressure on the second bore if it exists, s is the pressure increment if there are two bores in the well and l is the high hysterises limit for identifying shut-in.

Figure 6.53: Plotting sand rates in PIMAQ

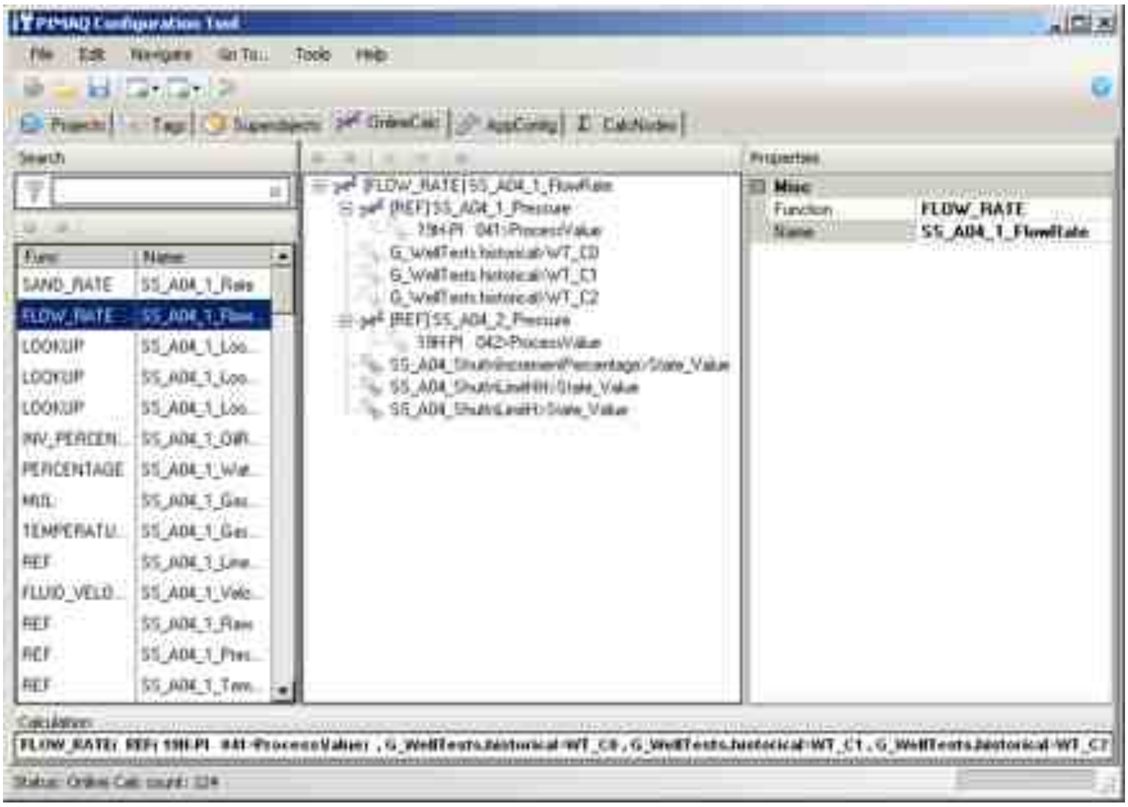


Figure 6.54: Configuring formulas in the PIMAQ Config Tool

6.B.3.3 FLUID_VELOCITY

This formula implements a fluid velocity as described in [1]:

$$Velocity = \frac{G' + O + W}{(24 \times 3600) \times \left(\frac{\pi \times d^2}{4 \times 1000000}\right)}$$

Where G is the temperature corrected gas rate, O is the Oil rate in standard cubics per day, W is the Water rate in standard cubics per day and d is the inner diameter of the pipeline.

The structure is as follows:

$$FLUID\_VELOCITY(G, O, W, d)$$

With parameters as described above.

6.B.3.4 EXCITATED_PERIOD

The following function is able to find intervals which a value exceeds a certain threshold. It is a generic formula that in this case is used for finding potential sand events. The following tasks is done in this algorithm:

- Low pass filter the values.
- Find the intervals of the bursts in the provided data set, based on the threshold value.
- Merge the intervals that lies close to each other.
- Remove intervals that are of a too small duration.

It has the following structure:

$$EXCITATED\_PERIOD(T, V_n\,[, f, g_{min}, d_{min}])$$

Where T is the threshold value, V are the values returned within the current timeperiod, f is the cut-off frequency for the low-pass filter, g is the minimum timegap between intervals and d is the minimum duration of the interval.

6.B.3.5 TEMPERATURE_CORRECTED_GAS_RATE

This formula implements a temperature corrected gas rate as described in [1] as:

$$G' = G \times \frac{(273 + T) \times 1.01325}{(273 + 15) \times (P + 1.01325)}$$

Where G is the gas rate in standard cubics per day, T is the flowlite temperature in degrees Celsius and P is the flowline pressure in bar G.

It has the following structure:

$TEMPERATURE\_CORRECTED\_GAS\_RATE(G, T, P)$

With parameters as described above.

6.B.4 Calibration table correction

If a recalibration is to be done, the zero value is the will be adjusted. That is; it is set equal to the value of the raw value such that the sand rate becomes zero from the calculation described in section 6.B.3.1.

The flow chart for the function ZeroCalibrate is given in figure 6.55.

The regression is based on a least square estimate of the regression coefficient a and b in the linear equation:

$$y = ax + b$$

The first coefficent is given by:

$$a = \frac{n\sum_{i=1}^{n} x_i y_i - \left(\sum_{i=1}^{n} x_i\right)\left(\sum_{i=1}^{n} y_i\right)}{n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2}$$

and the second given by:

$$b = \frac{\sum_{i=1}^{n} y_i - b\sum_{i=1}^{n} x_i}{n}$$

Where x is the velocity and y is the raw value.

6.B.5 References

[1] ClampOn. How to integrate the particle monitors into a control system. Technical report, 11 2005. Document Number 62.320.0013.00 rev 2.

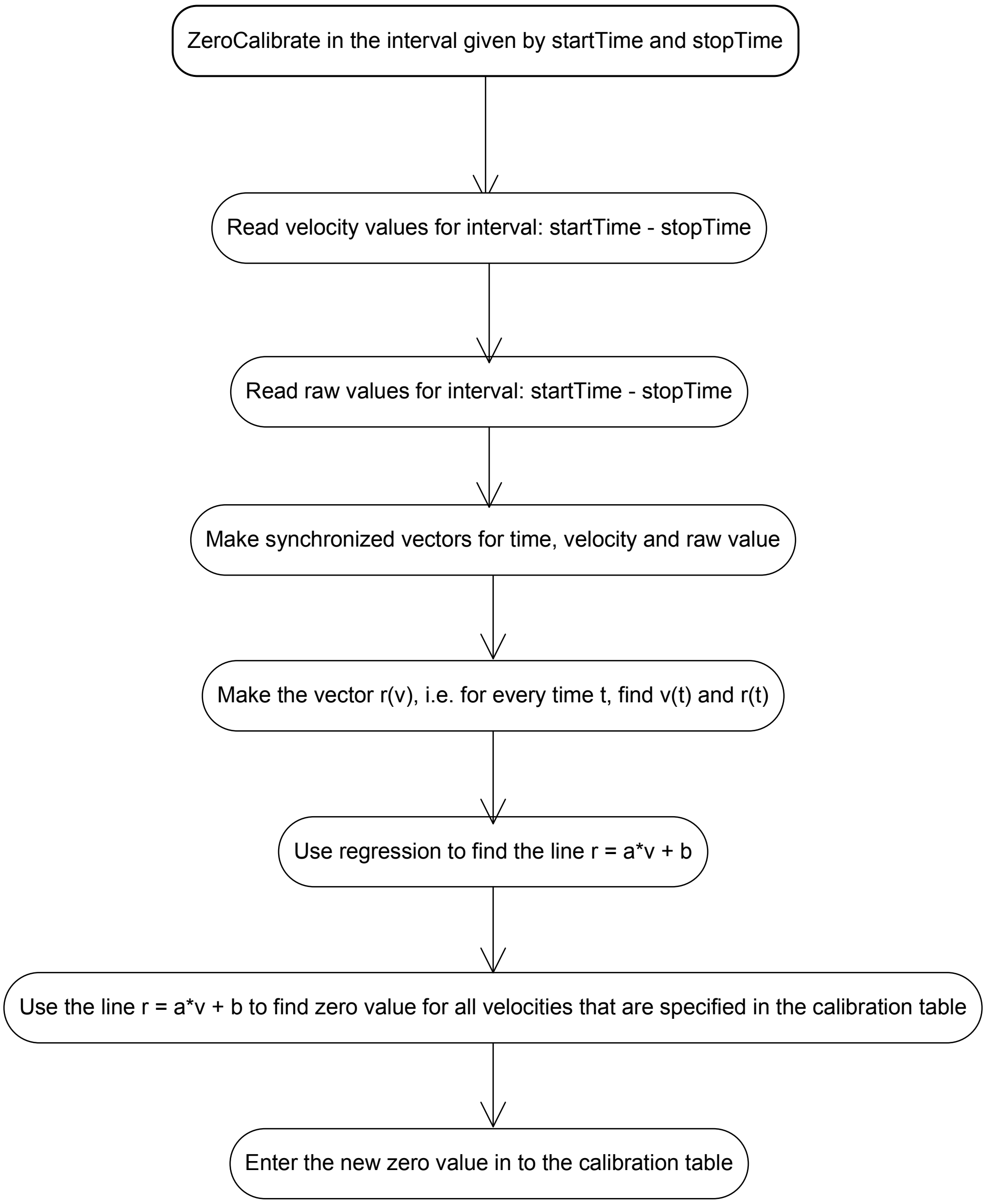


Figure 6.55: Flow chart for calibrating sand sensor zero values

6.C Erosion Modelling and Prediction

6.C.1 Erosion model

Erosion is due to particles (either droplets or solid particles) impacting at a surface. Solid/sand particles are the most critical/relevant for production from oil/gas reservoirs as erosion due to droplets requires significantly higher impact velocities (typically ≤150-200m/s) than solid particles to result in any erosional damage.

Erosion due to solids may result in severe damages to piping and components in a production system for oil and gas; figure 6.56 shows some typical examples of erosion damages experienced from various offshore developments.

Resulting erosion/material loss from particles impacting on a surface is dependent on; see figure 6.57:

- Sand particle impact velocity (V)
- Sand particle impact angle (α)
- Mass of sand hitting the surface (m)
- Material grade/ductility

When the impacting parameters and the material grade are known, the resulting material loss, E, can be determined from equations of the form:

$$E = K \cdot m \cdot F(\alpha) \cdot V^n \quad (.1)$$

where:

- K and n are material constants
- $F(\alpha)$ is a function of impact angle - see figure 6.57

K, n and $F(\alpha)$ are parameters characterizing different material grades. These parameters are determined from experimental investigations in a special test rig.

Erosion resistance may vary significantly between different material grades [1]. It is, however, worth noticing that the erosion resistance for standard steel grades is more or less the same, while solid Tungsten Carbides and some ceramics materials may have significantly higher erosion resistance; see figure 6.58.

6.C.2 Erosion predictions

In order to determine the resulting erosion rates for a specific system or component, the impact locations, velocities and angles have to be known in addition to the specific material grade. The impact conditions for the sand particles depend on the design/layout of the system/component, the operational conditions (flow rate of gas, oil and water, pressure and temperature) and sand production rate as well as particle size/size distribution.

Erosion predictions will generally include four steps:

1. Developing a geometrical model for the flow path
2. Calculating the flow field resulting from specific operational conditions
3. Determining the particle trajectories and the resulting impact locations and impact velocities and angles for the sand particles
4. Calculating the resulting erosion rate utilising Equation .1 given in section 6.C.1

The most general approach is to use CFD (Computational Fluid Dynamics) to perform erosion simulations/predictions. An example from a CFD analysis of the subsea template piping at Snorre B is shown in figure 6.59.

Utilising CFD is a time consuming/high cost approach that is not suited for the on-line simulations. In order to speed up simulation more simple models have been developed for standard pipe components; DNV-RP-O501 [2] has models for relevant pipe components like:

- Straight/smooth pipes
- Pipe bend/elbow
- Blinded Tee-bend
- Reducers
- Welded joints
- Intrusive erosion probes

DNV-RP-O501 has been validated against experimental investigations and CFD simulations and represents an industry standard technology utilised

Figure 6.56: Examples of erosion damages – three left cases represent erosion in control choke, no. 4 is erosion in a pipe bend and no. 5 is erosion in a diverter due to a gas blow out

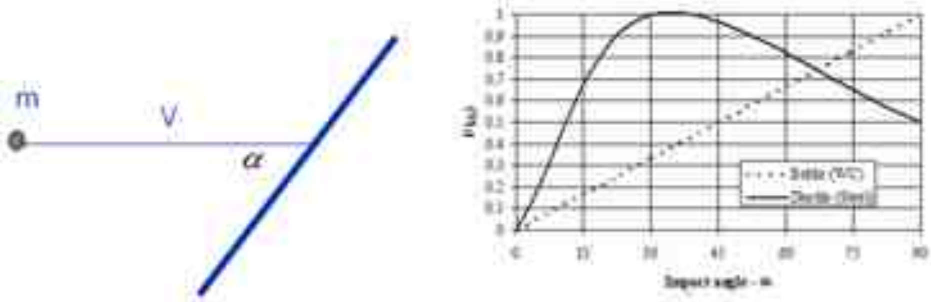


Figure 6.57: Main parameters influencing resulting erosion (left) and $F(\alpha)$ showing characteristic dependence on impact angle for ductile and brittle materials (right)

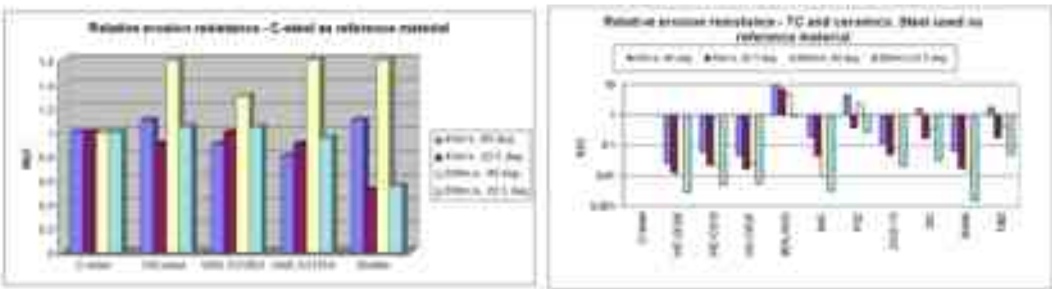


Figure 6.58: Relative erosion resistance with reference to C-steel. Steel grades (left) and some erosion resistant materials – Tungsten carbides and ceramics (right)

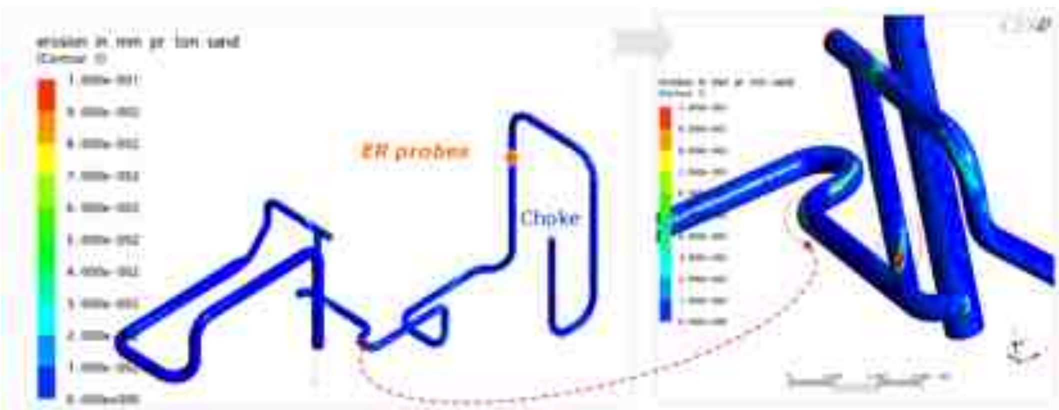


Figure 6.59: CFD model of subsea template piping for Snorre B and resulting erosion contours

world-wide as basis for estimation of erosion calculations.

Both ABB's ErosionInsight and DNV's 'Erosion Monitor' utilise the erosion models from DNV-RP-O501.

6.C.3 References

[1] K Haugen, O Kvernvold, A Ronold, and R Sandberg. Sand erosion of wear-resistant materials: Erosion in chokes. In *Proceeding of 8th International Conference on Erosion by Liquid and Solid Impact, Cambridge, UK*. Center for Integrated Operations in the petroleum industry, September 1994. Available at http://www.ioconf.no/2009/presentations/parallel6.

[2] DNV Recommended Practice RP O501. Erosion wear in piping systems. revision 5.0-2011. Technical report, Det Norske Veritas AS, 2011.

6.D Erosion monitoring software requirements

The features and functionalities that should be available in an Erosion Monitoring Software suitable for Erosion Management and Condition Based Inspection and Maintenance is described in this appendix.

It should be emphasized that the specified requirements and functionalities describe an ideal system, and it is not a description of the capacity/functionality of current Erosion Monitoring software.

6.D.1 System to be covered

The software should be able to model the system; i.e. piping, chokes etc. for the fluid/sand carrying parts from the wellhead to the 1st stage separator; see figure 6.3 in chapter 6.4 showing an example from Snorre B.

6.D.2 Data flow

The software should have the functionality to access operational data, configuration and topology data, and inspection data from a data provider like the Integration Layer developed within the IOHN project; see figure 6.60.

Ideally, the system should have the functionality to store the data from the analysis back into the data provider so that other applications may easily access the results.

6.D.3 Input data

6.D.3.1 Operational data

The applications need to get necessary operational data from the data provider as time series:

- Gas, oil and water production rates
- On-stream hours
- Pressure and temperature at critical locations
- Choke openings/Cv values
- Choke types
- Sand production data
- Etc

The production rates are often allocated rather than measured production rates. The allocated rates can be changed as a result of reallocations, which typically take place after a few days, after about one month, and also after up to one year. After reallocations the production rates in the production data bases are changed. The data provider and the applications should have mechanisms to update the data and results after reallocations.

6.D.3.2 Well test data

Well tests are performed to get better estimates about how much and what the wells are producing. During a typical well test, the well is routed to the test separator where the well production can be measured. Another type of well tests is deduction tests, where the tested well is shut down, and the production is estimated from the loss of total production. Deduction tests are often performed for subsea wells, as it is too expensive to shut down the other subsea wells to produce the tested well alone.

Well test data are actually only a variety of operational data. It consists of sensor data, flow rates etc. from the period of a well test. Well test rates are more reliable than the allocated rates, and well test data must be analysed and approved before they are entered (often manually) into the data bases.

These high quality data are needed for some of the calculations, e.g. calculation of Cv. The Cv estimate represents the flow capacity of a choke, and if the input data are of good quality, it might be used as an erosion indicator for chokes.

6.D.3.3 Topology and configuration data

Topology and configuration data necessary to set up an erosion monitoring system are typically:

- Topology of the system
  - Relations between e.g. wells, flowlines, manifolds, pipes, chokes, bends, erosion nodes and tags for sensor data used in the calculations at the different locations
- Pipe and bend data
  - Inner diameter for pipes and bends
  - Curvature of bend
  - Material for pipes and bends
  - Erosion model parameters

- Choke data
  - Choke type
  - Choke Cv characteristics i.e. theoretical Cv as a function of choke opening for different choke types
  - Choke dimensional data for erosion models
  - Choke material (e.g. steel, Tungsten Carbide)
  - Choke type (multiple orifice valve (MOV), single stage internal plug, single stage external sleeve, multi stage internal plug, multi stage external sleeve, labyrinth, etc.)
  - Erosion model parameters
- PVT data
  - Oil, gas and water density at standard conditions
- Sand data
  - Sand particle diameter or distribution
  - Sand rate to use in erosion calculations

The topology and configuration data are relatively static, but they might also change over time. Examples are:

- Bends, pipes and chokes can be replaced during maintenance work
- Topology can be changed if the process changes
- PVT data can change if the properties of the hydrocarbons change over time
- Sand data can change over time

Choke is an example of a component that might be changed quite often, and it would be beneficial if the data provider could offer information about choke type and choke changes, so that these data do not need to be updated manually, see next section.

Ideally, the data provider could offer all necessary configuration and topology information, so that the applications can use these data directly. If these data are always up-to-date, the topology and configuration data used by the applications will also

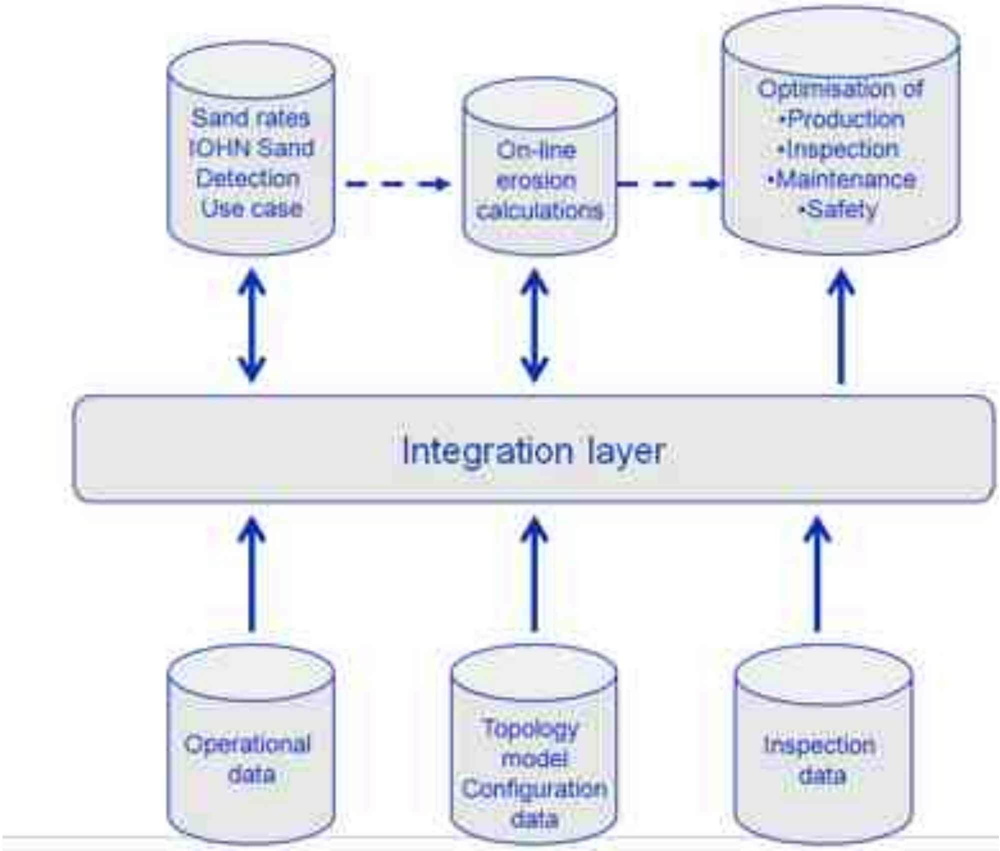


Figure 6.60: Schematic layout of data flow for erosion monitoring software

be up-to-date. It might, however, be more realistic that the applications could auto-generate a configuration based on the topology and configuration information offered by the data provider.

6.D.3.4 Maintenance and inspection data

Maintenance and inspection data can be considered as singular events occurring at irregular intervals. Such data may cover (but not limited to):

- Information from inspection; e.g. measured wall thickness for pipes and bends, or photos for chokes
- Maintenance history; e.g. change of pipe component (bend, Tee-bend)
- Change of choke and/or choke trim
- Sand events; e.g. special events with high sand production, or amount of sand in sand trap

Such data should preferably be available from the data provider as time series with required information of the singular events. However, in case such data are not available, the software should also have the feasibility of storing event data; e.g. results from inspection, change of chokes/choke trims/pipe components etc.

6.D.4 Functionality

6.D.4.1 Erosion indicators and erosion models

The software should have the ability of presenting time series of erosion indicators as trend plots, i.e.:

- Erosion rate
- Accumulated erosion
- Choke Cv difference (difference between Cv calculated from production data and theoretical Cv from choke curve)
- Flow velocities
- Sand rate
- Accumulated sand production

The software should have the functionality to estimate erosion rates for all critical (erosion exposed) components (chokes, bends, Tee's, pipes, manifolds, etc) identified from the wellhead to the separators. The erosion predictions should be performed utilising erosion models from DNV-RP-O501 [1] or similar models if required by Company.

Computational Fluid Dynamic (CFD) can be used for analysis of complex geometries like chokes, manifolds/headers, complex piping isometric, etc. The results from the CFD simulations can be used to derive correction factors to the erosion models in DNV-RP-O501 or to develop response models to be implemented in the erosion monitoring software.

The predictions should be based on daily operational data; i.e. flow rates of oil, gas and water, pressure and temperature and sand production.

In case information of sand production on daily basis is not available, the software should have the feasibility of prescribing sand production rates either as (g/s) or (kg/day) or (ppmW).

6.D.4.2 Decision support

The results from the erosion analysis should be presented in a way that enables easy identification of critical wells/piping/components with respect to accumulated erosion and other critical parameters like erosion rate, velocity, Cv difference, etc. The identifiers could be based on traffic light and colour codes. The 'set points' for the colour codes should be easily changeable.

The software should also have the ability of presenting summary or status sheets for all wells/piping/components. The status sheets should also display the values for which inspection is recommended.

The software should have the ability to produce summary reports. The summary reports should include all parameters relevant for inspection planning and prioritising; e.g. accumulated erosion, erosion rates, Cv/Cv difference, set points for colour codes, previous inspection data, recommended values for inspection etc.

The results from the erosion analysis are to be used to support planning and prioritising of inspection and maintenance. The final decision, planning and coordination must, however, be the responsibility of those responsible for inspection and maintenance..

6.D.4.3 Model calibration and tuning

The software should have the ability to utilise the results from inspection and maintenance for adjustment, calibration and tuning of the erosion models and accumulated erosion results.

One simple and very obvious option is to reset results for accumulated erosion based on results from inspection or replacement of a component.

Another more advanced alternative is to tune or calibrate the erosion models based on inspection data; e.g. in case inspection data generally shows a trend that the erosion predictions over-estimate the erosion rates. Such an option will, however, have to be handled with care and changes in the prediction models should not be performed without thorough analysis and evaluation of the main reasons for the discrepancy between predictions, inspection and maintenance data.

6.D.4.4 Data quality check

It is a general experience that operational data can be erroneous. Utilising erroneous input data in the erosion simulation model may result in false warnings that inspection is required or lead to critical components or conditions not being identified and included in the inspection plan.

Ideally, the data quality should be the responsibility of the operator, and the data offered by the data provider should have been quality checked and should be reliable and ready for use without any further requirements for quality check by the user.

The ideal situation that all data are reliable, will probably not be the case. In such situations, the applications should also have the functionality of performing a quality check of the data retrieved. Such a quality check may, however, be very exhausting due to the potential huge number of potential classes of erroneous data and will generally also be site specific. As a minimum, the software should have some general QA tests that would detect certain general classes of erroneous data; e.g.

- Pressure upstream a choke lower than downstream a choke
- Pressure subsea less than topside
- Pressure/temperature out of defined or possible ranges
- Flow rate out of defined or possible ranges
- Data from a sensor showing identical data over a period of time
- Data from different sensors (example pressure) installed at same location show different values
- Etc.

Identification and correction of erroneous data could, for example, be reported in the status report. Rules for how to handle erroneous data should be developed and agreed with the customer.

6.D.5 References

[1] DNV Recommended Practice RP O501. Erosion wear in piping systems. revision 5.0-2011. Technical report, Det Norske Veritas AS, 2011.

6.E Description of DNV Erosion Monitor Application

6.E.1 Introduction

The DNV Erosion Monitor Application (DNV-EMA) was originally developed in parallel with the ASR (Acceptable Sand Rate) strategy. The development of this strategy started in the 1990's with the focus mainly on production optimisation. At that time most of the fields were operated according to the MSFR (Maximum Sand Free Rate) strategy. In the MSFR strategy, wells were choked back if sand collected in a sand trap exceeded a pre-set value - typically a few grams of sand in the sand trap over a period of two hours. The same criteria was utilised whether it was a low producing oil well or a high producing gas well. It was emphasized that this strategy resulted in choking back wells with a low erosion potential and thus had a detrimental impact on the income from the field.

To address this DNV introduced a new strategy that was allowing for some sand production but where the acceptable sand production would be different from one well to the other dependent on the operational characteristics of the well. The term 'ASR strategy' was first introduced in a Pilot Project for Statoil for Gullfaks A in the period 1999 to 2001. The strategy has later on been applied to most of the Statoil developments; see examples ref. [1, 2, 3, 4, 5]. A schematic work process for ASR strategy investigations is shown in figure 6.61.

A schematic work process for ASR strategy investigations is shown in figure 6.61.

In order to implement and follow up the ASR strategy, DNV was asked by Statoil to develop an erosion monitoring software. DNV-EMA was initially an Excel spread sheet but has later been updated to a window based software programmed in Visual Basic for Applications (VBA). Most recently, the software has been further updated to accommodate for object modelling in order to simplify the adaption of the model to a new field.

Below is described in more detail the functionalities of the DNV-EMA software with special emphasis on the functionalities addressing Inspection and Maintenance optimisation.

6.E.2 Geometry model

DNV-EMA has the feasibility of modelling a field from the wellhead to the 1st stage separator. Normally the geometry model is divided into subsystems; e.g. wells/wellhead/subsea system, pipeline/risers and manifolds. The borders between the subsystems are at 'locations' where the flow is commingled.

Figure 6.62 shows an example of the geometry model for Snorre B as adopted in the DNV-EMA software. Snorre B is developed with two subsea templates each with 8 well slots. The production from the wells is routed to the platform in two production lines and one test line from each of the subsea templates. At the platform the production from all production lines/test lines is commingled in the manifold piping and routed further to the separator.

6.E.3 Critical components/software models

The procedure is to identify critical components throughout the system. Generally, components included in the analysis/evaluations are:

- Pipe bends/elbows upstream and downstream of chokes
- Tee- bends upstream and downstream of chokes

Other geometries may also be implemented; e.g.:

- Smooth pipe
- Reducers
- Welded joints
- Intrusive sand detectors

The models available in the "Erosion wear in piping systems" document cover the above mentioned components; see figure 6.63.

It is to be noted that the models in "Erosion wear in piping systems" are 'singular' models; i.e. no effect of pipe geometry/upstream conditions are accounted for. Such effects may, however, be implemented utilising safety factors or developing response functions by performing detailed CFD simulations covering the full 3-dimensional geometry of the system

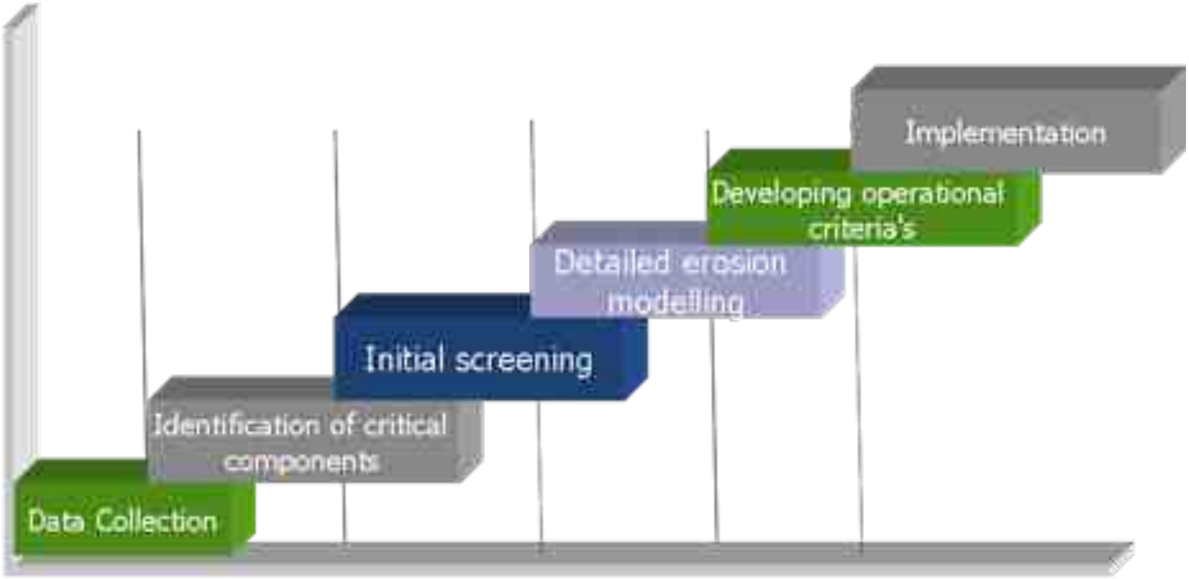


Figure 6.61: Schematic work process for DNV's ASR strategy

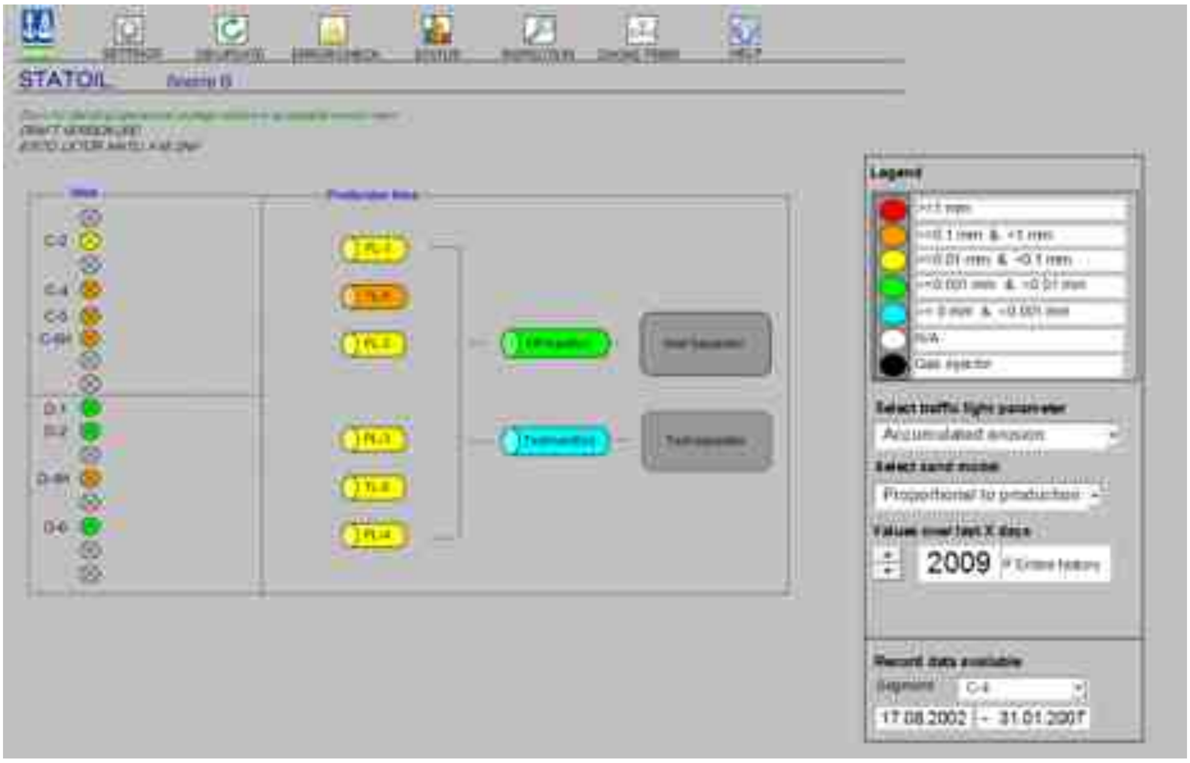


Figure 6.62: Model of Snorre B development – also showing colour codes and status for accumulated erosion for wells and production lines

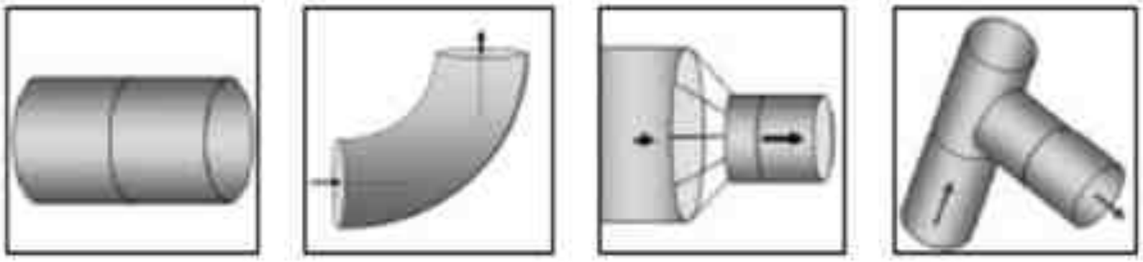


Figure 6.63: Erosion models available in DNV-RP-O501.

6.E.4 Required input data and input data quality

Data required as input to DNV-EMA are:

- Production rates (oil, gas, water) from each well
- Well routing
- Production rates for pipelines/risers/manifolds are determined by accumulating the production rates from the wells routed to the specific pipeline/riser/manifold
- Pressure and temperature at locations of critical components
- Sand production from each well. The sand production may either be specified in (g/s) or (ppmW) or may be obtained from sand detectors
- Sand production rates for pipelines/risers/manifolds are determined by accumulating the sand production rates from each of the wells routed to the specific pipeline/riser/manifold

Production rates are normally given in Sm3/day; i.e. flow rate of oil, gas and water at 1atm and 15.6°C. The software has a general PVT model (based on correlations developed utilising the PVT package HYSYS) implemented to determine the flow velocities and physical properties of the fluids at specified pressure and temperature. The PVT model takes into account phase change and effects of pressure and temperature on fluid properties like density and viscosity.

The PVT model can be updated/tailor made to cover a specific field.

Figure 6.64 shows an example of comparison between results from the PVT model in DNV-EMA and HYSYS.

6.E.5 Operational data

DNV-EMA requires operational data given in Excel spreadsheets in time series format; see example figure 6.65. Date tranfer from production data base like EC is normally done manually (on weekly or monthly basis). However, some special/custom made arrangements are made which allow for more automatic transfer of operational data into the software.

One of the main objectives in the IOHN project is to enable automatic data transfer from the integration layer to the software/Excel input spreadsheet.

6.E.6 DNV EROSON MONITOR functionalities

The DNV-EMA software has a number of functionalities which enables easy access and presentation of both operational data and results from the erosion calculations.

Results from analysis utilising DNV-EMA software cover the following main parameters:

- Erosion rates for the selected components (mm/year)
- Accumulated erosion over the period of production (mm)
- Velocities for critical components (m/s)
- Accumulated sand production over the period of production (kg)
- Cv for choke (-)

All input data and results from the erosion calculations can be plotted as time series. One or two series can be shown in each plot and it is possible to show combinations of both results from the analysis and operational data in the same plot; see example figure 6.66.

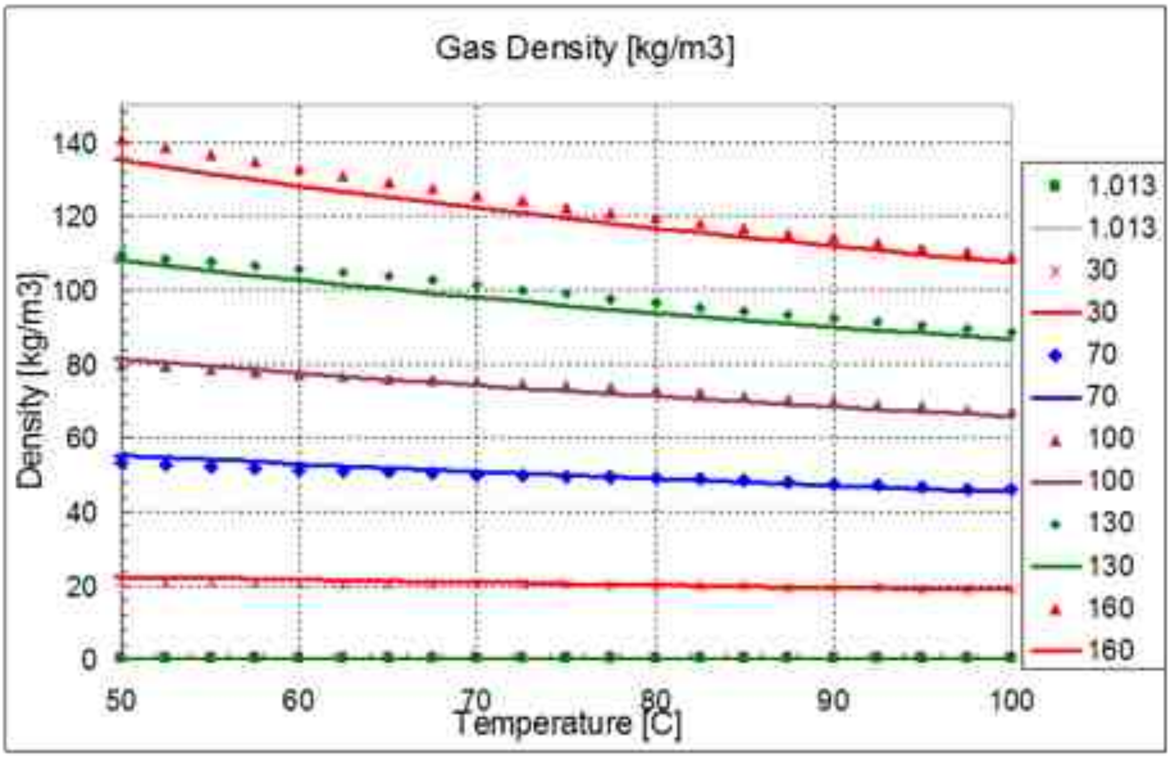


Figure 6.64: Comparison between PVT model in DNV-EMA and HYSYS

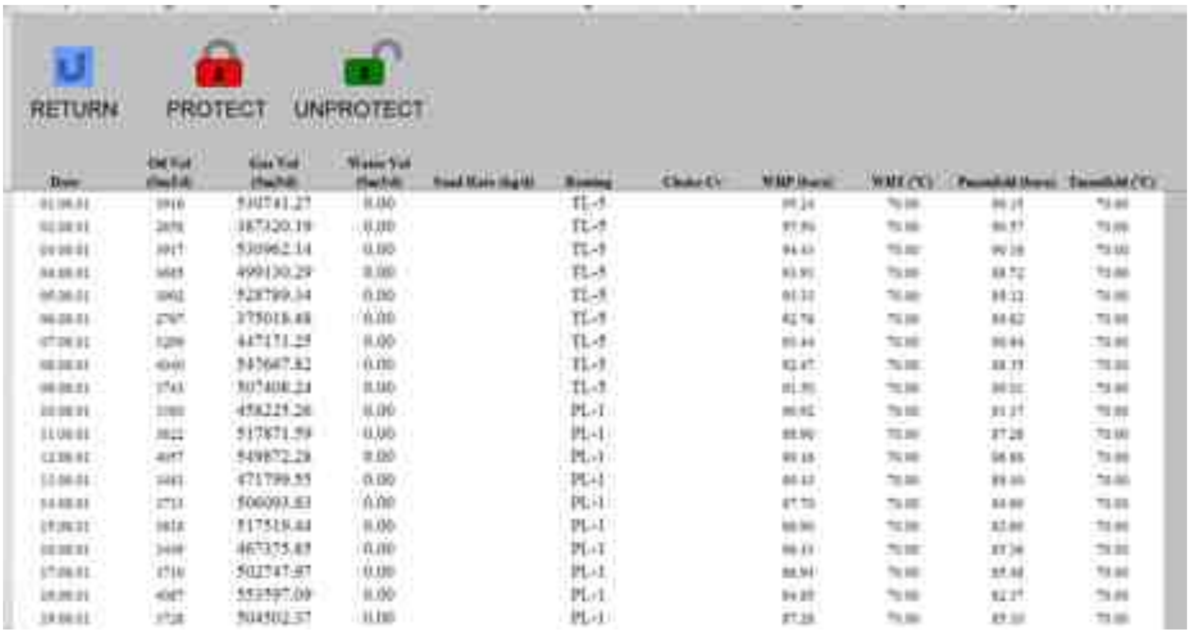


Figure 6.65: Example showing of raw production data for well C2 at Snorre B

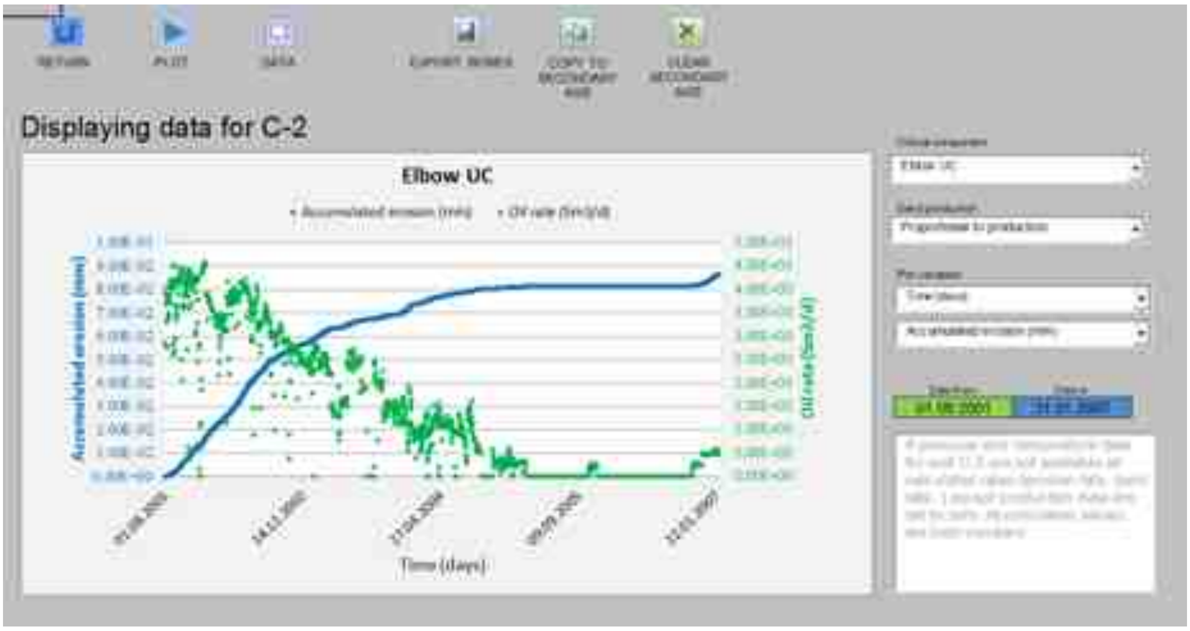


Figure 6.66: Example showing plot of accumulated erosion (mm) and production rate (Sm3/day oil) for well C-2 at Snorre B

Such a functionality may be of validity when, for example, the root cause for a detected high erosion incident is to be analysed.

Figure 6.62 shows the ‘front page’ for Snorre B where colour indicators (traffic lights) are used to give an overview of the status with respect to accumulated erosion for the various critical pipe components.

The ‘traffic light’ includes five ranges where the limits are easily changed to adapt to special requirements for a development.

The traffic light functionality is also used to present an overview of other parameters like:

- Average erosion rate over last period of time
- Velocities over last period of time
- Etc

DNV-EMA also has the functionality of automatic generation of a Status report. An example of a status report for Snorre B is shown in figure 6.67. The status report utilises the same colour codes for the traffic lights as used for the front page – figure 6.61 and lists the main results and operational conditions.

The status report may easily be extended and/or tailor made to cover specific requirements from different developments

The status report can also give indications on time to next inspection based on for example present erosion rates. Alternatively, time to next inspection may be determined utilising forecast production profiles.

6.E.7 What if - Forecast predictions

Forecast production profiles can be utilised to analyse the expected consequences of different future production scenarios and the impact of the production scenarios will have on the inspection and maintenance strategy.

6.E.8 Opportunities

The general features of the DNV-EMA software allow for a close follow up and evaluation of the status of a production system operating in a sandy service. By implementing the DNV-EMA software in combination with DNV’s ASR (Acceptable Sand Rate) strategy will enable the possibility of:

- Optimise production capacity
- Optimise inspection activities
- Optimise maintenance requirements
- Increase over-all system safety

Figure 6.67: Example of automatic generated status report – page 1 and page 4

6.E.9 Limitations

6.E.9.1 Choke Cv

The software has a model for calculating choke Cv. However, in the present version there is no functionality that utilise the difference between measured and estimated Cv to assess erosion in the chokes. The reason is that the chokes installed at Snorre B are not prone to erosion of the ports in the trim. This functionality can, however, easily be implemented.

6.E.9.2 Model calibration

The software has no general functionality for calibration and tuning of the model based on inspection data.

The functionality of re-setting of values, for instance due to replacement of a component or based on inspection result, is however, available.

6.E.10 References

[1] O. Kvernvold, L.E. Torbergsen, R. Eriksen, and H. Kjørholt. New strategy for sand management for safely improving production performance, 2002. Presented at DOT 2002.

[2] K. Lejon, A.B. Reme, R.B. Woster, O. Kvernvold, and L.E. Torbergsen. Sand production management for snorre b subsea development – lesson learned and actions taken, 2007. Presented at DOT.

[3] DNV Recommended Practice RP O501. Erosion wear in piping systems. revision 5.0-2011. Technical report, Det Norske Veritas AS, 2011.

[4] F. Selfridge, M. Munday, O. Kvernvold, and B. Gordon. Safely improving production performance through improved sand management, 2003. Presented at SPE Conference 2003.

[5] T.G. Torgersen, K. Lejon, O. Kvernvold, and L.E. Torbergsen. Snorre a sand production management. experience and solutions, 2006. Presented at TUV NEL Sand production Management Seminar.

6.F Description of the ABB ErosionInsight software

6.F.1 Overview

ErosionInsight is a software system for erosion monitoring developed by ABB. When Statoil changed their sand strategy from Maximum Sand Free Rate (MSFR) to Acceptable Sand Rate (ASR) around 2001, the development of the ErosionInsight software system was part of this strategy. The work with the software started in 2001, initiated by Halvor Kjørholt in Statoil, and in the beginning the main product functionality was specified by Statoil (Halvor Kjørholt, Jamie Andrews, etc.) See [1].

The software system was developed by PTI (Production Technology Integrated) from 2001 to 2006 under the name of PTI Insight. In 2006 the PTI Insight software system was acquired by ABB, and employees from PTI continued to work with the product in ABB. From 2011, the software system changed name to ErosionInsight. See [2].

The ErosionInsight system is typically set up and configured after DNV has made an analysis of the erosion potential for critical components in the system.

ErosionInsight is currently in use to control erosion in components like chokes and bends for the following Statoil fields:

- Since 2001: Gullfaks field (~ 90 wells)
- Since 2003: Statfjord field (~ 90 wells)
- Since 2005: Extended to the Gullfaks and Statfjord subsea system (~ 60 wells and flowlines)
- Since 2010: Pilot installation at the Snorre field as part of the IOHN project (~ 40 wells and flowlines)
- Since 2011: Pilot installation for the Sleipner field (~ 50 wells and flowlines)

Figure 6.68 shows examples of erosion in chokes and bends.

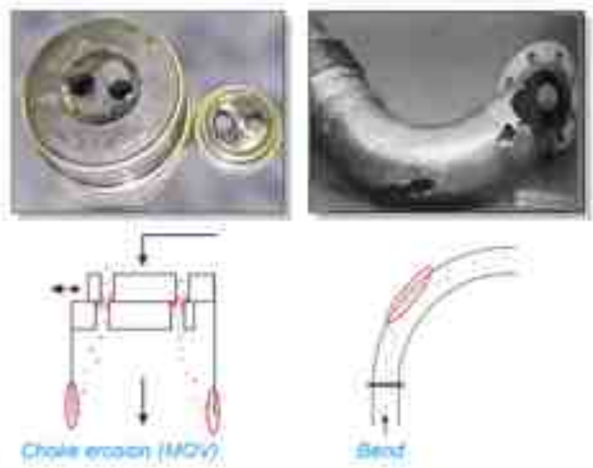


Figure 6.68: Examples of erosion in chokes and bends

6.F.2 Functional description

This section contains a functional description of the current version of the ABB ErosionInsight software system. In addition, we have focused on the functionality and limitations with respect to the requirements to the erosion management software described in appendix 6.D. We have also looked at the new opportunities the Integration Layer might provide.

The last part of the section contains a short overview of other functionality that the ABB ErosionInsight software system can offer.

6.F.2.1 System overview and data flow

Figure 6.69 shows an overview of the ErosionInsight software system and data flow. The boxes show the different components, while the arrows indicate the data flow.

The ErosionInsight software system consists of the following four components:

- **ErosionInsight online calculation module:** This module reads operational data, well test data and inspection/event data from standardized database views in the the Prosty or Energy Components (EC) production databases. Based on configuration and topology information read from the ErosionInsight application database, erosion calculations are performed, and data and results are saved as time series in the ErosionInsight application database. The calculations run automatically every night, but the calculation

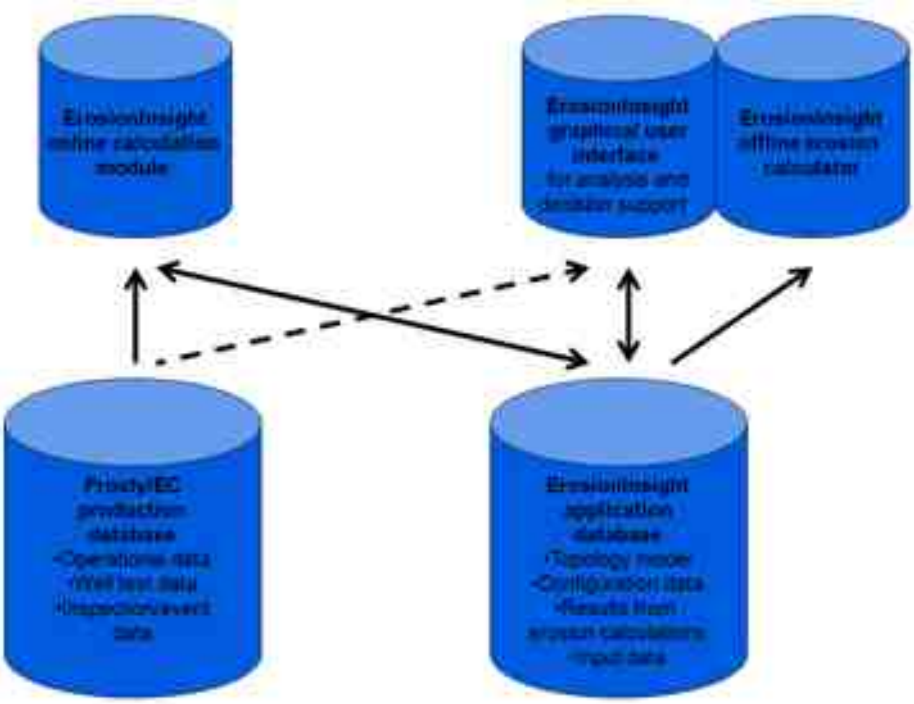


Figure 6.69: Overview of the ErosionInsight software system

module can also be run manually when necessary.

- **ErosionInsight application database:** The database contains all configuration and topology information necessary to perform the erosion calculations. The data base also contains all the input data and results from the online calculations.
- **ErosionInsight graphical user interface:** The user interface is available for all Statoil users from the office network. The users can start the application from a web page, and the application is used for decision support and analysis, e.g. that a choke or a choke disc needs to be inspected and possibly replaced. The user interface gets most of the data from the ErosionInsight database, but it also reads some sand data directly from Prosty. The graphical user interface is also used to configure the ErosionInsight system.
- **ErosionInsight offline erosion calculator:** The offline calculator is a part of the graphical user interface, and it can be used to perform what-if erosion calculations. It is possible to enter input data manually, copy data from Excel work sheets or to import data from the ErosionInsight database.

In the IOHN project, the Integration Layer can easily replace the Prosty/EC production database in figure 6.69. But as long as it is not possible to write data back to the Integration Layer, the ErosionInsight application database will still be needed to store results and configuration data. To some extent, the Integration Layer might also offer some of the configuration and topology information like which sensor tags belong to which erosion node or geometry data for pipes and bends. These data might replace some of the topology and configuration data in the ErosionInsight application database, but this has not been implemented in the IOHN project.

The ErosionInsight software system performs calculations for erosion nodes where necessary sensor data are defined in EC/Prosty (or default data can be used). With the Integration Layer, the application might get access to other sensor data, which means that the erosion monitoring can also be extended to other erosion nodes.

6.F.2.2 Input data

6.F.2.2.1 Operational data

The ErosionInsight software system reads the following operational input from the Prosty/EC database:

- Gas, oil and water flow rates
- On-stream hours
- Pressure and temperature at critical locations
- Choke openings
- Choke types
- Sand production data

If rate data for e.g. flowlines or manifolds are not available in Prosty/EC, such rates can be calculated from the rates of the wells feeding the flowline or manifold. Currently, routing information is not available in the Prosty/EC databases, so the routing information must be configured manually.

When the production rates are changed due to reallocations, the online calculation module will pick up the new rates and perform recalculations automatically.

Direct reading of operational data from the Integration Layer has not been implemented, but the SPARQL client prototype developed within the project can make the data available for the application.

6.F.2.2.2 Well test data

The ErosionInsight software system reads the well test data described in section 6.D.3.2 in appendix 6.D from EC/Prosty. The application uses the well test data to calculate Cv and Cv difference, which require the high quality well test data to give good results.

Direct reading of well test data from the Integration Layer has not been implemented, but the SPARQL client prototype developed within the project can make the data available.

6.F.2.2.3 Inspection/event data

The software reads dates for changes of chokes and choke discs automatically from the EC/Prosty database. This is very useful information for the analysis to decide whether a choke or choke disc needs to be inspected and possibly replaced.

In addition, inspection data can be entered manually as measured accumulated erosion for the components that are monitored. If such data are included, the accumulated erosion calculated from the erosion models will be set to the observed values.

The software also has the capability to store and display any event, e.g. events with high sand production, inspection, or change of choke disk, see figure 6.70. But this option is currently not in active use, as the events are not added automatically, but have to be entered manually.

Automatic reading of inspection and event data from the Integration Layer as described in section 6.D.3.4 in appendix 6.D would be very beneficial, since such data then can be updated automatically instead of manually. And it will be easier to make decisions about inspection and maintenance based on the combination of erosion monitoring results and the inspection and maintenance data.

Direct reading of inspection and event data from the Integration Layer has not been implemented, but the SPARQL client prototype developed within the project can make some of the data (e.g. choke type and choke changes) available for the application.

Figure 6.70: Example of inspection events that can be entered manually in ErosionInsight

6.F.2.2.4 Topology and configuration data

The geometry / topology of the system to be modelled, is configured as a hierarchy consisting of the following components:

- Oil field
- Platform
- Subsea template or well group
- Well, flowline or manifold
- Critical components like chokes, pipes and bends.
- Erosion nodes where erosion rate is calculated

The Prosty/EC databases do not provide any of the necessary topology and configuration data, and consequently, the ErosionInsight system does not read any of these data from Prosty/EC.

Reading of topology and configuration data from the Integration Layer would be very beneficial. The configuration might be simplified, and also changes in configuration and topology could be updated automatically when there are changes in the data in the Integration Layer.

6.F.2.3 Functionality

6.F.2.3.1 Erosion indicators and erosion models

The ErosionInsight application offers the following erosion indicators:

- Erosion rate based on fixed sand rate
- Accumulated erosion based on fixed sand rate
- Erosion rate based on measured sand rate
- Accumulated erosion based on measured sand rate (if available)
- Theoretical Cv from choke opening and choke Cv characteristics
- Cv calculated from production data
- Cv difference, i.e. difference between Cv calculated from the production data and the theoretical Cv
- Flow velocities
- Sand rate and accumulated sand (if available)

The calculations are based on daily operational data. For days when well test data are available, the Cv results are calculated from the well test data.

The erosion indicators are presented in time series and overview plots, see examples on the following pages.

The software system offers different PVT models, but the basic PVT model is a simple general PVT model developed by DNV. The PVT model is used to calculate actual flow rates, densities etc. at the different locations in the system, based on production rates of oil, gas and water at standard conditions, as well as pressure and temperature for the location.

For chokes the following erosion models are available:

- Gullfaks MOV chokes: Model for erosion the choke outlet for Gullfaks MOV chokes specified by DNV
- Statfjord MOV chokes: Model for erosion in choke outlet for Statfjord MOV chokes specified by DNV
- Single-stage choke with internal plug or external sleeve: The general "Guidelines for chokes" erosion model developed by DNV
- Multi-stage choke with internal plug or external sleeve: The general "Guidelines for chokes" erosion model developed by DNV

For bends the erosion rate is computed based on the DNV-RP O501 model [3].

The system can easily be expanded to include other components and erosion models. Cv from production data is calculated using the Perkins choke model [4]. This model handles critical flow through the chokes.

6.F.2.3.2 Time series plots

All erosion indicators and also input data can be plotted in times series plots. Figure 6.71 shows an example of a time series plot for Cv and Cv difference.

- The green dots connected with a green line show Cv difference calculated from well test data (right value axis)
- The red line shows the Cv calculated from production data (left value axis)
- The blue line shows the Cv calculated from choke opening an choke Cv curve (left value axis)
- The green horizontal line shows the max Cv for the choke type (left value axis)

More time series plot examples are shown in figures 6.77, 6.78 and 6.79. All time series plots show well tests as orange dots and choke changes as blue dots.

Figure 6.71: Figure showing Cv, Cv difference, pressure drop over chokes etc as time series

6.F.2.3.3 Decision support

ErosionInsight has multiple mechanisms for decision support.

The first mechanism is status visualization in the data tree. See figure 6.72. The different criteria that can be visualized are:

- Erosion rate (ES): Above a "high" limit (yellow) or "high high" limit (red)
- Accumulated erosion (AE): Above a "high" limit (yellow) or "high high" limit (red)
- Cv difference (CV): Above a "high" limit (yellow) or "high high" limit (red)
- Flow velocity (FS): Above a "high high" limit (red)
- Pressure drop over choke (PD): Below "low" limit (yellow) or "low low" limit (red)
- Choke operation (CO): Consider changing to a smaller or larger choke based on different criteria

The "high" and "high high" limits for erosion rate (ES), accumulated erosion (AE) and flow velocity (FS) are configurable, while the limits for Cv difference (CV), pressure drop over choke (PD) are fixed, based on Statoil specified values, but it should be quite easy to make these criteria configurable.

With this status visualization, it is easy to get an overview of the wells that should be followed up more closely, by drilling down into the time series plots for each well to make a final decision about inspection.

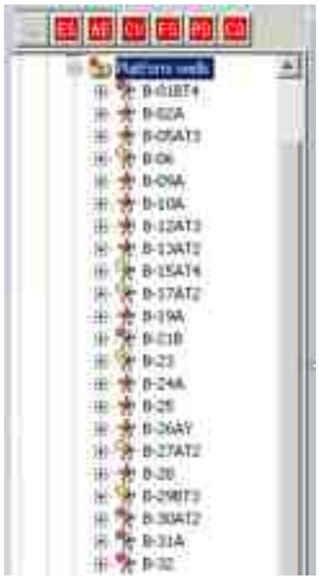


Figure 6.72: Status visualization in the data tree. Wells with red traffic has a "high high" alarm limit while wells with a yellow traffic light has a "high" alarm limit with respect to one of the criteria

The data tree can also be expanded to see which erosion node generates the alarm, see figure 6.73.

Figure 6.73: Expansion of the data tree to see which erosion node generates the alarm

Also note that the components like chokes and bends can be identified by their tag name, see figure 6.74. Inspection and maintenance personnel identify the components by the tag name, and therefore this information is important.

Figure 6.74: Expanded data tree with tag names

The second mechanism for decision support is summary plots for all erosion nodes for well groups or platforms.

Figure 6.75 shows an overview plot from ErosionInsight for the erosion rate for all chokes and bends for a platform. Choke erosion rates are shown as red columns, while the yellow ones show the bend erosion rate. The summary plot allows sorting based on e.g. bend erosion rate or choke erosion rate. Data series can be turned on and off.

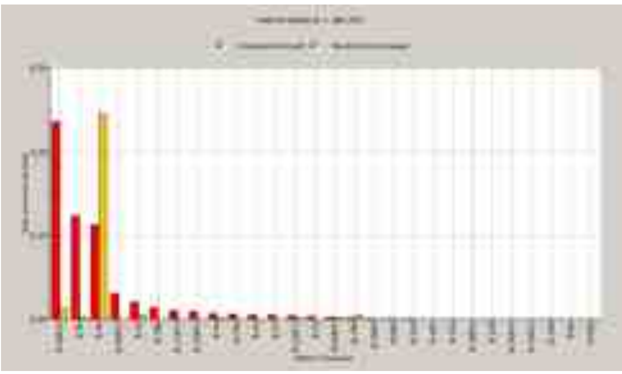


Figure 6.75: Overview of erosion rate for all chokes and bends for a platform

If the criterion for inspection is the same for all erosion nodes, we could have easily displayed the value in the overview plot, but the ErosionInsight software system allows different limits for each erosion node, so therefore the application displays these critera in the times series plots for each well, flowline, manifold or erosion node instead.

Similar overview plots are also available for accumulated erosion, for Cv difference from last well test and for flow velocities.

The plot in figure 6.76 shows the Cv difference from last well test (red bars - left value axis) for all wells for a platform. The plot offers different sorting criteria, as well as additional information like:

- Pressure drop over choke (yellow bars - right value axis): Cv difference value is more uncertain for chokes with low pressure drop.

- Choke opening (blue bars – not shown here): Chokes that produce with low choke opening (in % stroke or degrees) are often more prone to erosion than those that produce with high opening.

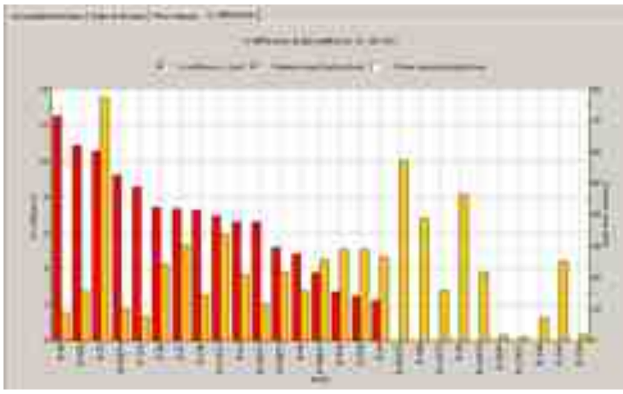


Figure 6.76: Overview of Cv difference and pressure drop over chokes from last well test for all chokes for a platform

The third mechanism for decision support is visualization of time series with limits for safe production or inspection displayed in the time series plot. As ErosionInsight currently uses alarm limits defined on a well/flowline/manifold or erosion node basis, it is better that the limits are shown in the time series plots than in the overview plots.

Figure 6.77 shows an example of a time series plot for erosion with inspection limits for the choke outlet for a well. The green line shows the accumulated erosion, while the pink line shows the maximum limit for accumulated erosion (right value axis). As we can see, the maximum limit has been exceeded for accumulated erosion, and the choke should be inspected. The red line shows the erosion rate (left value axis). In periods the erosion rate is higher than the max limit (blue line), and consequently, the well should probably be operated in another way to limit erosion.

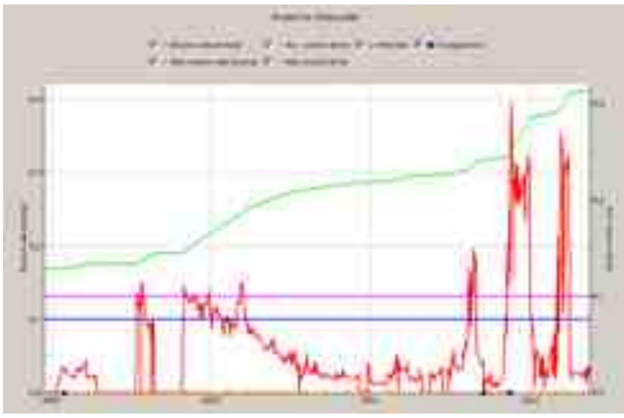


Figure 6.77: Figure showing erosion rate and accumulated erosion as time series

The ErosionInsight software system does not offer automatic summary reports as suggested in section 6.D.4.4 of appendix 6.D.

6.F.2.3.4 Model calibration/tuning

Currently, ErosionInsight offers some functionality for model calibration. The accumulated erosion can be reset to a measured value for a certain erosion node, e.g. based on inspection results, ref. section 6.F.2.2.

Other erosion model parameters can also be changed, e.g. the fixed sand rate and sand particle size used in the erosion calculations, or the geometry factor used in the DNV-RP O501 bend erosion model[3].

More advanced calibration mechanisms are not yet available.

6.F.2.3.5 Data quality check

The ErosionInsight software comprises different mechanisms to assure sufficient quality of the input data. Mechanisms are:

- Results will not be calculated if pressure sensors show that pressure drops in the wrong direction, e.g. pressure downstream choke is higher than pressure upstream choke.
- Sensor input parameters can have minimum and maximum values defined on platform, well/flowline/manifold or tag level.
- Sensor input parameters can have default values defined on platform, well/flowline/manifold or tag level.

The ErosionInsight software system also offers facilities to plot time series for all the operational input data:

- Pressure data
- Temperature data
- Choke opening
- Production rate
- GOR, WC and total liquid production

The plotting facilities also offer plotting of well test data. This functionality has proved to be very useful to identify well tests where the quality has not been sufficient.

Figure 6.78 shows an example of a time series plot for the pressure upstream and downstream choke for a well. The plot shows that in a period in April and May 2011, the pressure downstream choke has been missing, and the default pressure downstream choke has been used. The default value could be changed and the results could be recalculated for this period.

Figure 6.79 shows an example of a time series plot for the gas oil ratio (GOR), water cut (WC) and total production rate. It can be observed that the GOR has changed rapidly for the last well test.

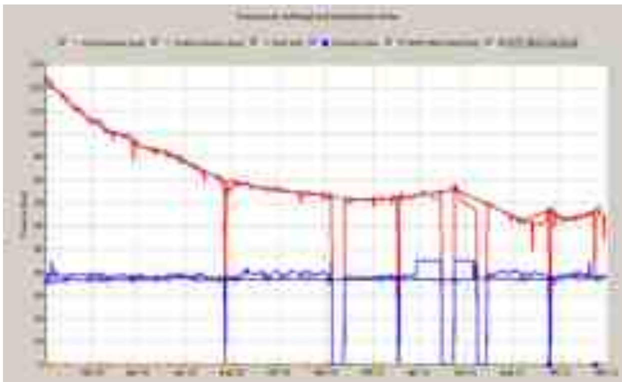


Figure 6.78: Times series for pressure upstream and downstream choke, also showing pressure values from well tests

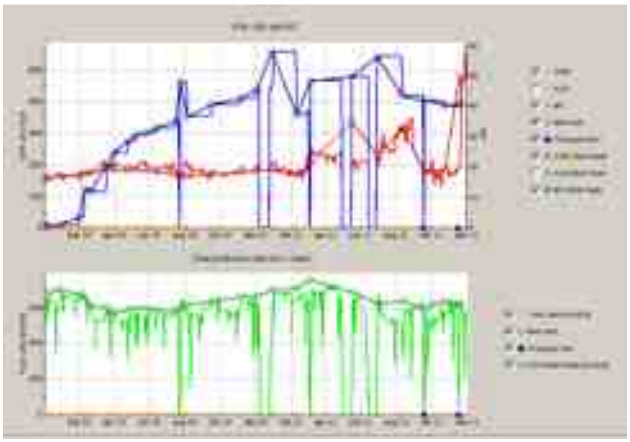


Figure 6.79: Times series for gas oil ratio (GOR), water cut (WC) and total production rate, also showing values from well tests

6.F.2.3.6 Auto-generation of erosion model from topology and configuration data

ErosionInsight does not offer any functionality for auto-generation of erosion model from topology and configuration data, but it would beneficial to have such functionality when the Integration Layer can offer such data.

6.F.2.4 Other functionality

This section gives a short overview of other functionality that ErosionInsight can offer, which are not described in the requirements for the erosion monitoring software described in appendix 6.D.

6.F.2.4.1 Support for selection of choke type

ErosionInsight can be used to give recommendations about selection of choke types to decrease the erosion in the choke. Choke Cv characteristics can be displayed for all choke types, and the offline erosion calculator can be used to determine which choke size is suitable for the current production and operating conditions.

6.F.2.4.2 Sand support

The plots in figures 6.80, 6.81 and 6.82 show examples of functionality available to display sand data for the well.

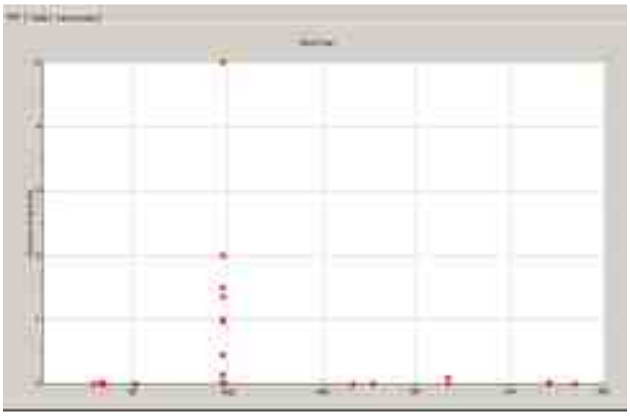


Figure 6.80: Figure showing the amount of sand in the sand trap

Figure 6.81: Log from sand trap inspection

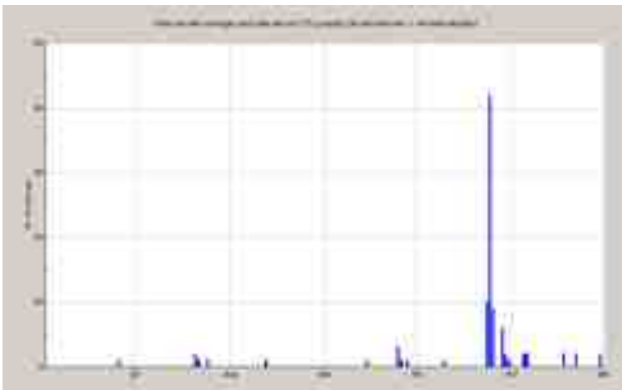


Figure 6.82: Plot showing number of 30 minute intervals per day with sand rate above 0.75 g/s

6.F.2.4.3 Well test diagnosis

ErosionInsight can also be used to support the decision about which well should be tested next. Figure 6.83 shows change in well head pressure since last test for all wells for a platform, and wells where the well head pressure has changed most since the last well test should probably be tested first. Similar plots can be made for change in liquid rate and change in choke opening since last well test.

Figure 6.83: Plot showing change in well head pressure since last well test for all wells for a platform

6.F.3 References

[1] J. Andrews, H. Kjørholt, and H. Jøranson. SPE 94511 Production enhancement from sand management philosophy. A Case Study from Statfjord and Gullfaks. In *SPE 6th European Formation Damage Conference, Scheveningen, Netherlands*, May 2005.

[2] K. Hovda and K. Lejon. Effective Sand Erosion Management in Chokes and Pipelines. In *Sand Control and Management 2008, Kuala Lumpur*, January 2008.

[3] DNV Recommended Practice RP O501. Erosion wear in piping systems. revision 4.2-2007. Technical report, Det Norske Veritas AS, 2007.

[4] T. K. Perkins. Critical and subcritical flow of multiphase mixtures through chokes. *SPE Drilling & Completion 8(4)*, 8(4):271–276, December 1993.

6.G ErosionInsight Snorre configuration

ABB has set up ErosionInsight for the Snorre A, B and UPA installations as part of the IOHN project. The main part of this work was performed in 2010. The configuration was based on information in a report from 2005 [1]. Some of the configuration data might therefore be outdated. The Snorre organization has also provided some updated choke configuration data during the present project, but this information has not been complete. Also note that a selection of the erosion nodes described in [1] has been configured.

ErosionInsight has been set up to read time dependent input data (production data, well test data, sensor data, choke data etc.) from the Energy Components (EC) production data base. The application reads input data from ErosionInsight specific database views that have been defined in the EC data base. Normally, these views contain all necessary data to perform erosion calculations, but as can be seen in the overview below, there have been some challenges with respect to the content of data in the database views.

To calculate bend erosion is used the DNV-RP-O501 model for bends [2]. To calculate choke erosion is used “DNV Guidelines for chokes models”, and to calculate choke Cv is used the Perkins choke model [3]. A fixed sand rate of 0.1157 g/s (corresponding to 1 kg sand/day) has been used in the erosion calculations with fixed sand rate.

In the next sections is given an overview of the current configuration and calculation status for the different parts of the Snorre A, B and UPA installations.

6.G.1 Snorre A platform wells

- Wells configured:
  - P-1 T2, P-3, P-4, P-8, P-9, P-13, P-14 B, P-15, P-18 A, P-19 T2, P-23, P-24, P-26, P-27 A, P-30, P-33, P-38, P-39, P-40, P-41, P-42 A, P-43, P-44 T3
- Erosion nodes:
  - 4” bend upstream choke (erosion rate, flow velocity)
  - 6” pipe upstream choke (flow velocity)
  - 8” pipe downstream choke (flow velocity)
  - Choke (Cv monitoring - theoretical Cv, calculated Cv, Cv difference)
  - Choke inlet and sleeve (erosion rate, flow velocity)
- Calculation status
  - All wells have been back-calculated from mid April 2009 and many wells even longer back. Well P-14 B has been calculated back to October 2003 to possibly predict erosion, as there was a serious incident with erosion and loss of containment for this well in November 2003.
- Challenges
  - Information about choke types and choke changes is not available in the Energy Components (EC) data base, and this information has been based on information from Excel work sheets provided by the Snorre organization. However, the choke type information in the work sheet was not complete, so the calculations have been based on a “best guess” for choke types.
  - The EC database views for well test data do not contain well test data after 2010. Well test data are typically used for the Cv monitoring, since the rate data from well tests are more reliable than the allocated production rates.
  - The two points above mean that Cv monitoring might be uncertain. On the other hand, erosion is not expected to affect the Cv (which represents the actual flow capacity for the chokes) for the Varco Best chokes installed for the Snorre A platform wells.
  - Sand rate data are available in EC from February 2009, which means that erosion rates with measured sand rate have calculated from February 2009.
  - For the P-14 B well production data were not available in EC before October 2003.
- Future opportunities
  - Online erosion monitoring will give good results for the Snorre A platform wells because of good data quality.
  - Use of the measured sand rate in the erosion rate calculations from February 2009 will provide especially valuable results.

- The configuration should be revised by the Snorre organization to possibly increase the selection of erosion nodes to be monitored and to assure that the configuration data are correct.

6.G.2 Snorre UPA subsea wells

- Wells configured:
  - A-1 AH, A-16 H, A-3 AHT3, A-4 AHT3, A-5 AH, A-8 HT2
- Erosion nodes:
  - 3” bend upstream choke (erosion rate, flow velocity)
  - 4” bend upstream choke (erosion rate, flow velocity)
  - No choke erosion monitoring as the wells do not have chokes
- Calculation status
  - Back-calculated from mid April 2009, some wells even longer back.
- Challenges
  - Very few well tests are available in EC, but this is not critical as Cv monitoring is not performed since the wells do not have chokes.
  - Sand rate data are available in EC, but the quality is uncertain, so erosion rates have been calculated with fixed sand rate.
- Future opportunities
  - The configuration should be revised by the Snorre organization to possibly increase the selection of erosion nodes to be monitored and to assure that the configuration data are correct.

6.G.3 Snorre UPA flowlines

- Wells configured:
  - SNO-UPA1, SNO-UPA2
- Erosion nodes:
  - 8” bend upstream choke (erosion rate, flow velocity)
  - Topside choke (flow velocity)
- Calculation status
  - Calculated from 6 January 2011 when sensor data were first available in the EC database views used by ErosionInsight.
- Challenges
  - Sensor data have only been available in EC since January 2011. This means that the flowlines were not prioritized during the initial setup in 2010.
  - Sand rate data are supposed to exist, but they are currently not available in the EC database views used by ErosionInsight. The quality of the sand data is also unknown. This means that erosion rate is calculated with fixed sand rate only.
  - Chokes have been configured with dummy choke characteristics as configuration data have not been available.
- Future opportunities
  - Sand rate data can be added to the EC data base views used for flowlines. If sand rate data are available and reliable, erosion rate calculations with measured sand rate can be calculated.
  - It is also possible to set up erosion rate and Cv calculations for the chokes if we get the choke configuration data.
  - The configuration should be revised by the Snorre organization to possibly increase the

selection of erosion nodes to be monitored and to assure that the configuration data are correct.

6.G.4 Snorre B subsea wells

- Wells configured:
  - C-1 HT2, C-2 HT2, C-4 HT2, C-5 H, C-6 AHT3, C-7 AH, D-1 HT5, D-2 AH, D-4 HT2, D-6 H
- Erosion nodes:
  - 5” bend upstream choke (erosion rate, flow velocity)
  - 8” bend downstream choke (erosion rate, flow velocity)
  - Choke (Cv monitoring - theoretical Cv, calculated Cv, Cv difference)
  - Choke (flow velocity)
- Calculation status
  - Back-calculated from mid April 2009, some wells even longer back.
- Challenges
  - Information about choke types and choke changes is not available in the Energy Components (EC) data base, and this information is not either easily accessible from other data sources. So the calculations are based on a “best guess” for choke types. Earlier, Kent Introl chokes were used for the Snorre B subsea wells. These have later been replaced by Mokveld and CCI chokes, but it has not been possible to get the exact choke change history.
  - The EC database views for well tests are missing well tests in some periods, but the views now seem to contain data for new well tests.
  - The two points above mean that Cv monitoring will be uncertain. With the possible exception of the new CCI chokes, erosion is not expected to affect the Cv (which represents the actual flow capacity for the chokes) for the chokes installed at the Snorre B subsea wells.
  - Sand rate data are defined in EC back to February 2009, but since the Snorre B subsea wells do not have acoustic sand detectors these data are most likely to be dummy data. Therefore the erosion rates are calculated with fixed sand rate.
- Future opportunities
  - The configuration should be revised by the Snorre organization to possibly increase the selection of erosion nodes to be monitored and to assure that the configuration data are correct.
  - It is also possible to set up erosion rate calculations for the chokes if the choke types and choke configuration data are made available.

6.G.5 Snorre B flowlines

- Wells configured:
  - Test lines: SNB-CT-FL.L.5, SNB-DT-FL.L.6
  - Production lines: SNB-C1-PL.1, SNB-C2-PL.2, SNB-D2-PL.4
  - Production manifold
- Erosion nodes:
  - 5” bend in test lines (erosion rate, flow velocity)
  - 8” bend upstream choke in production lines (erosion rate, flow velocity)
  - 10” bend downstream choke in production lines (erosion rate, flow velocity)
  - Topside choke in production lines (flow velocity)

  - 18” pipe in Production Manifold (flow velocity)

- Calculation status
  - Calculated from 6 January 2011 when sensor data were first available in the EC database views used by ErosionInsight.

- Challenges
  - Sensor data have only been available in EC since January 2011. This means that the flowlines were not prioritized during the initial set-up in 2010.
  - Sand rate data are supposed to exist with probably good quality, but they are not available in the EC database views for flowlines used by ErosionInsight. This means that for now erosion rates are calculated with fixed sand rate.
  - Chokes have been configured with dummy choke characteristics as configuration data have not been available.

- Future opportunities
  - Sand rate data can be added to the EC data base views used for flowlines. These sand rate data are probably of good quality, and erosion rate calculations with measured sand rate can be calculated.
  - It is also possible to set up erosion rate and Cv calculations for the chokes if the choke types and choke configuration data are made available.
  - The configuration should be revised by the Snorre organization to possibly increase the selection of erosion nodes to be monitored and to assure that the configuration data are correct.

6.G.6 References

[1] L Nøkleberg and K Hovda. Specification: PTI Insight online erosion management system, Snorre A, B and UPA. Technical report, PTI report, 2005.

[2] DNV Recommended Practice RP O501. Erosion wear in piping systems. revision 5.0-2011. Technical report, Det Norske Veritas AS, 2011.

[3] T. K. Perkins. Critical and subcritical flow of multiphase mixtures through chokes. *SPE Drilling & Completion 8(4)*, 8(4):271–276, December 1993.

Copyright information

Twenty-two companies have participated in the IOHN project. These are (in alphabetical order): ABB, Business Association of Norwegian knowledge- and technology based enterprises (Abelia), Baker Hughes, Cisco, Computas, Det Norske Veritas (DNV), ENI, Epsis, FMC, The Norwegian Defence and Security Industries Association (FSI), IO Centre, International Research Institute of Stavanger (IRIS), National Oilwell Varco, Norwegian Institute of Science and Technology (NTNU), The Norwegian Oil Industry Association (OLF), POSC Caesar Association, Petroleum Safety Authority Norway, Siemens, Statoil, The Norwegian Defence, University of Oslo and University of Stavanger.

Two sub-projects are also partly sponsored by the Research Council of Norway; GOICT under the VERDIKT programme (RCN nr. 183235) and AutoConRig under the PETROMAKS programme (RCN nr. 187473).

More information about the project may be found on the project website (http://www.iohn.org) or via info@iohn.org. The website also includes a password protected area where the project results are stored.