Digital Chemical Engineering 2 (2022) 100015

Contents lists available at ScienceDirect

Digital Chemical Engineering

journal homepage: www.elsevier.com/locate/dche

Research Protocols

The Digital Design Basis. Demonstrating a framework to reduce costs and improve quality in early-phase design

David B. Cameron[a,*], Arnljot Skogvang[b], Mihaly Fekete[c], Henrik Martinsson[c], Morten Strand[c], Marcel Castro[d], Kirsten Helle[d], Anders Gjerver[e], Christian Mahesh Hansen[e], Kristin Høines Johnsen[f]

[a] *University of Oslo, Department of Informatics, Gaustadalleén 23B, Oslo 0373, Norway*
[b] *Lundin Energy Norway AS, Norway*
[c] *Aize AS, Norway*
[d] *TechnipFMC, Norway*
[e] *Aibel AS, Norway*
[f] *AkerBP ASA, Norway*

ARTICLE INFO

Keywords:
Digital twins
Standards
ISO15926
OPC
Industry 4.0
ISO/IEC81346
Systems
Concept design
Review
Semantic technologies

ABSTRACT

As a joint initiative, Aker BP, Lundin-Norway, Aker Solutions, TechnipFMC, Aibel and Aize, together with Sirius (the oil and gas research centre at the University of Oslo) have developed and demonstrated a common digital model representation of the information in early-phase design bases for oil & gas field developemnts. The scope of the project was to develop a proof of concept for a Digital Design Basis that supports data-centric rather than document-based engineering.

The project established a standards-based data model that holds data about both the design basis and functional requirements decided by an operator. This model that can be implemented in any relevant software tools in a concept study, to ensure that information shared between operators and EPC vendors, with their different software tools, have the same meaning and understanding. The model is based on a common digitalized language for communication along the field development supply chain.

1. Introduction

1.1. Scope and purpose of this paper

Energy developments on the Norwegian Continental Shelf are under unprecedented pressure with regard to cost, efficiency and sustainability. After the price crash of 2015, all parties in the Norwegian industry: operators, suppliers, government and trade unions, formed a KonKraft (competitive strength) initiative to maintain competitiveness (KonKraft, 2018). This paper presents the Digital Design Basis (DDB), a joint project that arose out of this work.

In early-phase design, operating companies prepare design basis documents that they send to vendors as a starting point for design. Each operator uses their own formats. This means that most information is supplied as PDF files and Excel spreadsheets. This means that a contractor needs to handle data in many different formats, from many operators, about essentially the same things. This results in tedious manual work, transcription errors and increased cost.

These problems can be resolved if all parties in early-phase design can agree on a shared data model for the information in the design basis. We call this a Digital Design Basis (DDB). This model provides a template into which operators and vendors can enter data and share it with each other. Engineering applications and life-cycle information systems can also then automatically ingest and use the data, without manual data entry. Through using this model, we move from a document-based to a data-based workflow.

The DDB project brought together three operating companies (Lundin Energy, AkerBP and Equinor), three engineering vendors (Aker Solutions, TechnipFMC and Aibel) and SIRIUS, a collaborative research centre at the University of Oslo.

The first phase of the project was completed in early 2021. This proof-of-concept prepared a shared model of the information and demonstrated its use in data transfer. The model captured the information needs of the involved operating companies. It is based, pragmatically, on current and developing standards, such as ISO/IEC81346, CFIHOS (CFIHOS, 0000), DEXPI (DEXPI, 2021) and ISO15926 (Kim et al., 2020) (especially part 14). The model itself is expressed using semantic

---

[*] Corresponding author.
E-mail address: davidbc@uio.no (D.B. Cameron).

https://doi.org/10.1016/j.dche.2022.100015
Received 31 December 2021; Received in revised form 3 February 2022; Accepted 9 February 2022

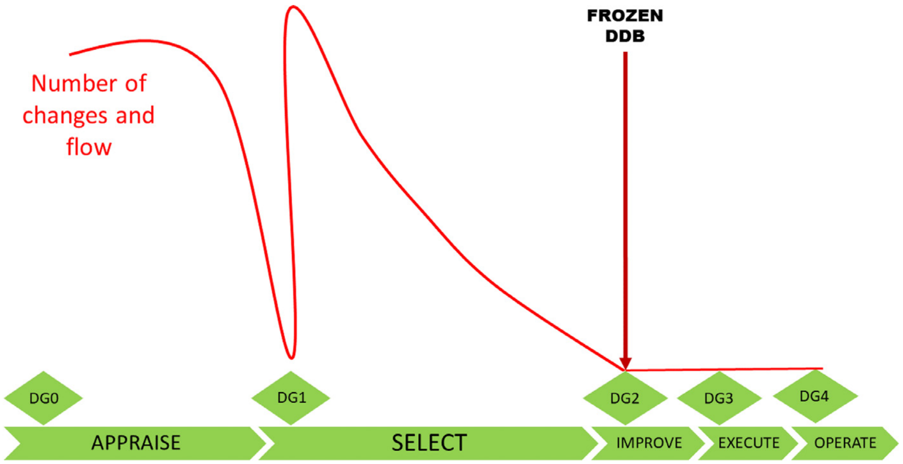


Fig. 1. Overview of the staged-gate field development process, showing the volume and frequency of changes in information during the early phases of the project.

web standards (OWL (Hitzler et al., 2012) and RDF (Cyganiak et al., 2014)). This choice of technology allows us to use reasoning methods to check consistency of data, requirements and assumptions.

The proof-of-concept shows that it was possible to load operator information into a data set defined by the model. This data was then ingested into the proprietary engineering tools that are developed by Aker Solutions and TechnipFMC. We also demonstrated how an engineering application for pipe simulation accesses this data.

This work is yet another in the long and complex effort to achieve efficiency in engineering through shared data models. We believe that recent advances in database and semantic technologies allow us to move faster and more directly towards this aim. We also demonstrate the power and potential of bringing an industrial ecosystem to work together on reducing transaction cost. Our findings are relevant for further work on later phases of engineering. They are also directly applicable to engineering activities in other process and heavy industries.

This paper is structured as follows. First, we will describe the design basis, its role in field development and the benefits that can be expected from moving to a DDB. We then briefly place this work into the broader context of research on interoperabilty, in particular semantic interoperability, in process engineering. We then present the technologies we have used to implement the DDB and present details of the model. Experience from two proof-of-concept examples is then presented, before we conclude and point towards on-going further work.

1.2. The role of the design basis in field development

The design basis in engineering is usually a set of documentation that defines, according the US Department of Energy (Design Basis, 0000), ”The set of requirements that bound the design of systems, structures, and components within the facility. Those design requirements include consideration of safety, plant availability, efficiency, reliability, and maintainability”. Sloley (2014) divides this information into a project-specific design basis and a basic engineering data document. The latter contains information that is specific to a site or company. He notes that these documents overlap. This was also our experience.

We collected and analysed typical design basis documents from the three operating companies. The documents used were either from early-phase projects or templates for the same. We saw that the design basis contained both requirements, site-relevant information and corporate data. Fig. 1 shows how the design basis grows as a project passes through its various stages and decision gates (DG). The diagram shows how the design basis grows with the project. Results of one phase form the design basis for the next. If this information is exchanged as documents, there is an in-built loss of information at each stage.

The project reference group discussed the potential benefits at each stage of adopting a digital design basis. These are listed in Table 1. The reference or customer group consisted of experienced project engineers from each operating company and vendor.

In this project we looked at the design basis from DG0 to DG2. In this phase of the project, an operator has selected an oil field for development and is interacting with vendors to identify, define and estimate the cost of a field development concept.

In this phase, the operator prepares a set of information that contains a mixture of *facts* and *requirements*. Here we use an idiosyncratic definition of the term "fact" to include the operators best estimates of the properties of the oilfield and the environment in which it is to be developed. The EPC supplier works to develop a concept that produces oil and gas based on these facts in a way that meets the supplied requirements. These requirements may come from corporate policies, industry standards or regulatory obligations. Specifications on the functional performance of the concept are also requirements.

2. The DDB in the context of engineering data interoperability

The interoperability of engineering data has been a research and implementation challenge for nearly as long as computers have been used in engineering. Progress has gone in waves (rather like Artificial Intelligence) where enthusiastic progress has been followed by periods of disillusionment and consolidation.

The first wave of work on the computerised management of engineering design began in the1980′s with the advent of the minicomputer and the first electronic CAD tools. An initial vision was to build engineering databases as a hub for information, but as Benayoune and Preece noted in 1987 (Benayoune and Preece, 1987), relational databases ”are not designed to handle the complex static and dynamic nature of process engineering design data”. From the first adoption of simulation tools, the need has been seen to support exchange of design data. This led to collaborative effort to develop standards, such as PDXI (Goldfarb and Dealmeida, 1996) in the1990′s. Wozny et al. (1992) drew up visions for this decade and a review of status around 2000 is given by Marquardt (1999). They both build a vision where an integrated design environment ties together engineering data and simulation.

A summary of this progress is given by Mayer et al. (2013), who, writing in 2013, look back on the two decades of work that resulted in the ISO15926 (Leal ,2005) and MIMOSA (Gilabert et al., 2012) standards. The Norwegian continental shelf was an area where many of these methods were developed and tried out (Gulla et al., 2006a; 2006b). It was in this period also that the OntoCAPE ontology was developed and applied to interoperability (Marquardt, 2010; Wiesner et al., 2011; 2010).

The ISO15926 standard dates back to 1990, with the EU-driven ESPRIT initiative ProcessBase. This led to the formation of several consortia for standardization of process plant life-cycle data modelling, notably

Table 1
The shared and individual benefits and motivators for implementing a digital design basis.

Phase	Operator	Supplier
DG0-DG1 Appraise Concept	• Reduced number of engineering hours. • Increased quality of data supplied to supplier. • Standardized formats and better revision control. • Reduced need to follow-up suppliers and resolve queries.	• Continual updating of data allows better responsiveness. • Better communication with operator around common understanding. • Automated generation of conceptual design.
DG1-DG2 Select Concet	• Simple revisions and extension of the design basis. • Standardized formats and better revision control	• Standardized solutions to reduce cost. • Reuse of existing solutions. • Faster planning and preparation. • Better use of capacity.
After DG2 FEED	• More precise information leads to better quality in project and deliverables. • Shared understanding and transfer of knowledge between supplier and operator. • Reuse project information to enhance operator's performance and suppliers' products. • Faster throughput and project execution leading to first production sooner.	

EPISTLE and POSC/Caesar. The first parts of ISO15926 were eventually published in 2003, with a data model based on the entity relation-based language EXPRESS (part 2). Later, the standard was extended with a reference data library containing terms used in the process industry (part 4), a template methodology (part 7).

By that time, W3C recommendations for OWL had been published, and it was decided to move ISO15926 in the direction of semantic web standards. Part 8, published in 2011, was the first effort by ISO15926 towards embracing the semantic web standards, describing how the template-building mechanism of part 7 may be implemented in OWL. In 2018, part 12 was published, providing a life-cycle integration ontology expressed in OWL, thus taking the full step into the semantic web.

The part 12 ontology has, however, features in the use of OWL reasoning that prevent potential users taking advantage of one of the most important novel features of OWL. To remedy this, ISO15926 part 14 is under development. This provides an ISO15926-based upper ontology in a way which also fully supports OWL reasoning. There is an ongoing process which aims to publish the ISO15926 part 14 ontology (not to be confused by the part 14 standard document) by POSC Caesar and the READI project (READI, 0000).

The currently most updated version of the ISO15926 part 14 ontology is used as upper ontology in our DDB data model.

Unfortunately, this activity did not result in a radical change in how the industry builds and maintains facilities. Braaksma et al. (2011) identified three types of challenge to adoption of interoperability:

Business Environment The financial and regulatory constraints or pressures that either inhibit or promote work interoperability. Presence or lack of support and implementation by tool vendors for CAD, simulation and engineering.

Organization Methods for interoperability are complex and require skills that are not present in companies. These new methods also demand new work practices. This provokes resistance to change and management scepticism.

Standards The standards proposed are complex, unstable, slow to develop and could not be scaled up (Gielingh, 2008). Standards, in particular ISO15926 (Smith, 2006), made ontological commitments that complicated use. The standards also lacked tool support and used an outdated language, EXPRESS.

Since around 2013 we have seen a renewed interest in interoperability, driven by a broader business interest in digital transformation. If we use the same structure as Braaksma et al. above, the following things have changed in ways that make it once again promising to work with standardisation for engineering interoperability.

Business Environment The oil industry suffered price shocks in 2013 and 2020 and is under constant pressure to respond to the challenges of global climate change. The process industry faces similar challenges. Top management supports digitalization initiatives that can help with these challenges. Tool vendors are adopting platform and service-based business models that require interoperability. Regulators and governments recognise the benefits of interoperability and encourage semantic data sharing as a tool to improve competitiveness.

Organization Organizations are working to introduce digital competence and implement new digital methods in organizations. This can lead to lower resistance to novel methods and new work practices. Project execution models are adapted to allow tighter integration of engineering data between customers, vendors and equipment suppliers.

Standards Industry consortia, notably CFIHOS (CFIHOS, 0000), READI (READI, 0000) and DEXPI (DEXPI, 2021), have worked to develop standards with limited scope and simple ontological commitments. Semantic standards have moved from EXPRESS to modern languages, notably OWL (Hakkarainen et al., 2006). This leads to better tool support and the ability to build on the programming skills of the wider semantic web community.

This recent work is summarized by Wiedau et al. (2021, 2019), who build on the DEXPI representation of piping and instrumentation diagrams to support a broader asset life-cycle. They see this approach as being an enabler for AI applications and requirement management in engineering and operations. They also introduce the ISO/IEC81346 standard (ISO/IEC81346-1, 2009) for aspect-based breakdown of complex systems. They propose that a combination of this standard with ISO15926 is desirable. We agree with this and have used this as a guiding principle in this work.

Fillinger et al. (2017, 2019) have also worked on interoperability across the asset life-cycle and implemented this as a prototype plant design tool (Fillinger et al., 2021). They note that even now, there are serious challenges to this type of interoperability (see also Doe et al. (2021)). We, once again, organize these under three headings:

Business Environment Vendors and customers are locked into proprietary systems. According to Fillinger et al. (2019), there seems that ”few software vendors and users (of customized software) are currently willing to accept some level of compromise regarding the standardization.”

Organization Unstructured documents (PDF, Excel, Word) are still the primary exchange formats.

Standards Lossless format conversions are needed when exchanging data across diverse, legacy systems (Doe et al., 2021). There is still lack of agreement on reference data. There are still few or no standards for information *exchanges*. Existing standards lack maturity and coverage. Excel, PDF and existing schemas are not scalable.

Our work seeks to complement the efforts described above by addressing an important, but limited, problem, namely interoperability of concept data in early engineering phases. The approach we describe in the next section builds a combination of ISO/IEC81346 and ISO15926 modelling principles and uses a modern, effective set of tools that support IT methods such as semantic reasoning.

3. Technical principles and tools

3.1. Technical principles

In this section we present and discuss the technology choices and modelling decisions we have made in building the DDB model. This section contains the computer science and semantic technology foundations upon which we build the model.

3.1.1. Top-down, functional modelling

We are representing design of an oil & gas field concept, not an as-built asset. This means that we choose to use a top-down systems breakdown, as described in ISO/IEC81346. Our modelling is based on the Reference Designation System (RDS) for Oil & Gas, issued in 2020 (READI, 2020). This classification allows complex oil and gas systems to be broken down into sub-systems that focus on specific *aspects* of interest. The three primary aspects defined in ISO/IEC81346 are function, product and location. We follow the approach taken by the READI Information Modelling Framework project (Fjøsna and Waaler, 2021), where we assign information to objects, blocks, in an RDS breakdown. In the early phase of a field development, most of our modelling focuses on the *functional* aspect of the system. The design process consists of proposing functional blocks that meet the functional requirements and constraints (what we have called facts) expressed in the design basis. In early phase, these blocks correspond to top-level systems in the RDS for Oil & Gas: Production blocks represent wells and well-heads, Transport blocks represent piping and riser and Processing blocks represent manifolds and the production facility.

3.1.2. Use of generic modelling patterns

Generic and reusable modelling patterns have been developed to capture the meaning and nature of a design basis. The aim is to reflect what is seen in the real world as precisely as possible, in such a way that complex relationships are captured and can easily be retrieved. Rather than modelling to support a specific use of the information, we have tried to model in a generic way, thereby making the information available to serve many purposes.

3.1.3. Open semantic web standards: OWL and RDL

The model is based on open, semantic web standards. Today, there are many large ontologies in active development and use, both open community efforts and proprietary industrial ontologies. During the last decade, we have also seen a migration of STEP and ISO15926 to these standards. Many software tools are available, both open source and commercial. This technology landscape serves the intentions of a common DDB data model and exchange format well. We hope to encourage uptake and use in the industry by basing the data model and formats on open and well-defined standards.

The Web Ontology Language (OWL) (Hitzler et al., 2012) is a family of knowledge representation languages for writing ontologies. An ontology is, in the context of computer science, a formal description of the concepts of a domain of discourse, and the relations between them (Man, 2013). It usually consists of a (1) taxonomy or vocabulary of concept *types*, with general concepts at the top and more specific concepts towards the leaf nodes, as well as (2) defined relation types between the concepts. Usually, concept membership is regulated with *axioms*, that is, logical expressions that define the requirements for being a member of a concept.

The original intention of OWL was to provide a web-friendly language to help computers understand the meaning of web documents and information. Thus, OWL builds on commonly used web technologies like XML and RDF. Today, OWL is used in a much broader context, for organizing and formalizing knowledge and domain vocabularies.

What sets OWL apart from most other languages for defining vocabularies, is its model-theoretic formal semantics. This means that for every well-formed OWL expression there is precise, mathematical way to assess its truth value, purely based on its intrinsic structure. These formal semantics allows for automated computer-based analysis of ontology properties. We can tell whether an ontology is consistent, or whether a concept is subsumed by another. This analysis process is called ontology reasoning. It is worth noting that the reasoning process is completely independent of how the concepts relate to the domain of discourse (“the real world”).

OWL has several profiles, defining subsets of the full OWL expressivity, with desirable computational properties with respect to reasoning. Our DDB data model uses the OWL 2 DL profile (Motik et al., 2012), which includes most of the expressiveness of OWL 2.

Finally, it is worth noting that OWL is an open standard, free for anyone to use to build ontologies or compatible software systems. Using OWL ensures that one is not tied to any one specific proprietary platform. This is essential when the intention is to build a common data model to be used across organizations in a supply chain.

3.1.4. Shared data models with private data

The DDB data model is based, where possible, on existing ontologies. This speeds up our development as we reuse others’ work rather than developing everything from scratch.

In practice, this meant that the model is based on several imported OWL ontologies. We import a stack of ontologies implemented by the READI Joint Industry Project (JIP), starting with an ISO15926 part 14 foundational ontology. We then used adapted versions of the ontology for tracking Provenance, Authoring and Versioning[1] and the SKOS Simple Knowledge Organization System RDF Schema[2]. We also import ontologies based on of IEC61355 (defining classification of documents) and ISO/IEC81346.

The data model serves as public and common authoritative resource for the vocabulary terms needed to define a digital design basis. Its purpose is to be shared between all stakeholders in the industry. This open approach to the data model must not, however, be understood as a “share-everything-with-everyone” approach. The design basis document contains business-critical and sensitive information, only to be shared between an operator and its chosen vendors. Thus, privacy also applies to the digital design basis. Although the data model is shared and freely available, each digital design basis is still private.

We achieve this by broadly separating the types in the model from the axioms. The common, shared model provides the shared grammar that we use to structure the axioms that define each specific design basis. This approach is illustrated schematically in Fig. 2. Here an operator uses an API or user interface to enter design basis information structured using the model. In our case we show an engineer from Lundin Energy entering data from their Alta Gotha field development. The stored model is then readable by Aker Solutions and TechnipFMC in a way that allows it to be imported into their proprietary design tools. The data can also be extracted for use in common design applications for well-head equipment and pipelines.

3.1.5. Use of templates to simplify modelling

A challenge when using OWL as a modelling languages is that it is a very low-level language. This means that many RDF statements are required to implement data models and that there are many repeated patterns that are needed for modelling. This can make the models difficult to understand and maintain. To use an analogy: building a model in OWL is like programming in Assembler. Higher level constructs are needed to abstract away this clutter and allow us to concentrate on the information to be encoded.

---

[1] http://purl.org/pav

[2] https://www.w3.org/TR/2008/WD-skos-reference-20080829/skos.html

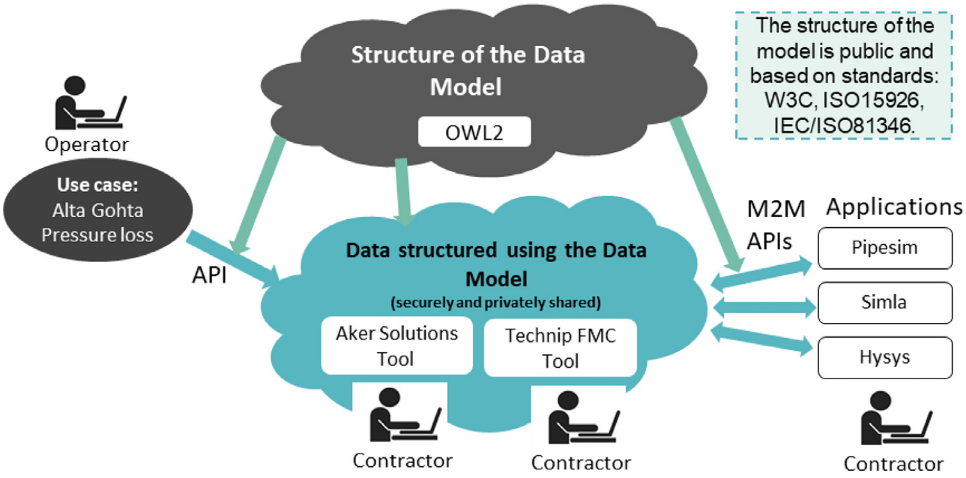


Fig. 2. How we structure and use the DDB.

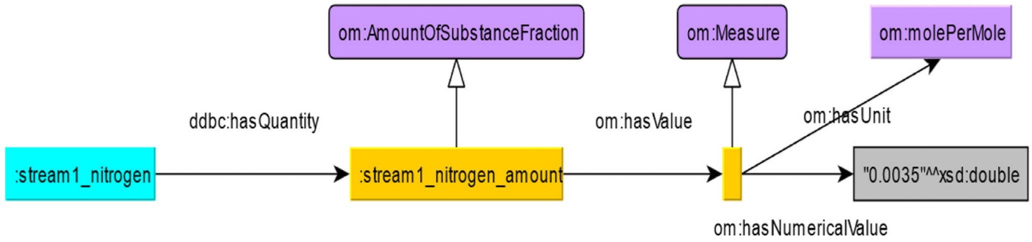


Fig. 3. Example of the modelling pattern used to represent a quantity of an object. The cyan and yellow objects are OWL named Individuals, while the rounded magenta objects are OWL classes. The magenta object with square corners is an OWL individual defining the units for the value. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

For example, each numerical value in the model is represented by a simple, but verbose pattern of OWL statements. Thus, the nitrogen content of a stream is represented by the pattern shown in Fig. 3. Here, the individual `:stream1_nitrogen` (cyan) has a `ddbc:hasQuantity` relation to an individual `:stream1_nitrogen_amount`, which is of `rdf:type` `om:AmountOfSubstanceFraction`. The individual `:stream1_nitrogen_amount` represents a quantity of the individual `:stream1_nitrogen`, namely, the amount of substance as a fraction, represented by the classifier `om:AmountOfSubstanceFraction`.

We have found that Reasonable Ontology Templates (OTTR) (Skjæveland et al., 2018) are a valuable tool for building the model and data. OTTR is a language for representing ontology modelling patterns. It is designed to support interaction with OWL or RDF knowledge bases at a higher level of abstraction, using modelling patterns rather than OWL axioms or RDF triples. In one sense, OTTR can be viewed as a macro language for RDF and OWL.

Take, for instance, the quite complex RDF pattern required to represent a quantity value shown above. Instead of having to manually write the whole pattern every time we want to express a quantity value, we define an OTTR template over the pattern. An OTTR template consists of a signature, which defines the number and type of input parameters to the template, much like a function definition in a programming language, and a template body, which defines how the input parameters are to be translated into OWL expressions or RDF patterns. The body may include calls to other OTTR templates, allowing a hierarchy of templates. In the case of a quantity value, we may define a template `Measure`, with the following signature:

`Measure(ottr:IRI ?phenomenon, owl:Class ?quantity, ottr:IRI ?unit, rdfs:Literal ?value)` The arguments are written as a list of `<type>` `<parameter_name>` pairs, and the parameters are (through the definitions in the body, which we omit here) interpreted as follows:

- `?phenomenon` - the object to which the quantity value applies.
- `?quantity` - the quantity which is expressed, a subclass of `om:Quantity`.
- `?unit` - the unit of measure for the quantity value, an OWL individual.
- `?value` - the numerical value of the quantity, usually a decimal number.

We can the express, e.g., that an object `:object` has a length of 3.5 metres by the template instance expression

`Measure(:object, om:Length, om:meter, ”3.5”^^xsd:double)`

This abstracts away the complexity of OWL quantity value modelling pattern, allowing us to focus on the parameter values. OTTR provides mechanisms for extracting template instances from tabular data sources, like Excel files and databases.

3.2. Units of measurement

There is currently no standard ontology for quantities and units of measure for the oil & gas industry. Since measurements and numeric values are essential in a design basis, DDB needed an ontology to regulate how such values are represented. The patterns shown above implements an approach to modelling units of measurement that builds on systematic review (Martın-Recuerda et al., 2020) of existing ontologies.

We chose ”The Ontology of Units of Measure (OM)2.0″[3] as our basis ontology. There was, however, a need to modify the original OM ontology slightly. Firstly, we needed to make OM fit into the ISO15926 part 14 upper ontology stack. Secondly, we needed to address some performance issues of the original OM ontology with the HermiT OWL reasoner. In addition, there were some units missing, e.g., volumetric units based on standard conditions for pressure and temperature.

---

[3] https://github.com/HajoRijgersberg/OM

3.2.1. Tools and integration

The model and demonstration was implemented using an open-source technology stack, where the Protégé ontology editor (https://protege.stanford.edu/) was used as authoring tool and Apache Jena Fuseki (https://jena.apache.org/documentation/fuseki2/) was used as a triple-store database and SPARQL.

Integration requires us to be able to offer the model through an API that supports queries from client systems. Our proof-of-concept offered a SPARQL end-point.

The model also needs to be serialized. We can write the model in all common graph serialization formats: Turtle, OWL/XML, RDF/XML and JSON-LD.

4. The design basis model

Having described the technology used, we are now in a position to describe the DDB model in detail. This section first presents the DDB core ontology, followed by discussions of the ISO/IEC81346 system breakdown. The section concludes with a discussion of the modelling of topology and fluid properties.

4.1. The DDB core ontology

The DDB model is built around a core ontology that contains concepts that represent the facts, constraints, system blocks and materials in the design basis.

Facts `OilandGasProductionProfile` with sub-classes `ProductionProfile-P10`, `ProductionProfile-P50` and `ProductionProfile-P50`, represent the operator's estimates of production from the field. `Agreement`, `Company`, `FieldOperator`, `FieldOwner`, `Petroleum ProductionLicense` and `License` are facts about the commercial status and identity of the development. Composition and properties of the well fluids are also represented, as described below.

Constraints Constraints are expressed as values allocated to variables at points in the system model. For example, an operator may specify a minimum allowable pressure drop along a production flowline. This is realized by a specification of pressure at the outlet of the flow-line.

System blocks We have defined types for `Well`, `SubseaTemplate`, `SubseaPipelineEndTermination`, `Pipeline`, `PipeEnd`, `Outlet`, `Inlet` `OilAndGasField` and `FPSO` model the functional systems that can be used to build the concept and the design basis.

Materials The types `GaseousCompound`, `Hydrocarbon Compound`, `LiquidCompound`, `NaturalGas`, `Produced Water` and `RawCrudeOil` are used to model the materials processed in the concept. Detailed modelling of the composition of the fluids to be processed is done using the ChEBI ontology. We describe this in more detail below.

4.2. System breakdown structure

Systems in the DDB model are modelled based on the IEC/ISO81346 oil and gas system structure definitions and its ontology representation provided by READI JIP. The ontology contains an oil & gas system and component classification hierarchy based on the following four main categories: facilities, main systems, technical systems and component systems.

In this work the object property `lis:functionalPartOf` is used to express that a system is a functional part of another system, as shown in Fig. 4. The figure also shows RDS designators in parentheses.

4.3. System topology

By system topology, we mean how the various components of a system are interconnected, in order to control the process flow through the system. The systems specified in the proof-of-concept case are functional representations of the would-be technical asset of the oil field. The topological relations between those systems are implemented by the object property `ddbc:functionallyConnectedTo`.

Figure 5 shows how the components are connected. The arrows are instances of the object property `ddbc:functionally ConnectedTo`, and the yellow rectangular boxes are individuals representing the components which together make up the production line. The components are not directly connected, but are rather connected via individuals of the class `ddbc:ConnectionPoint` (blue diamonds in the diagram). This provides a logical handle on which to attach further facts or constraints.

Some components, like the production pipeline, have features, i.e., distinctive attributes or aspects, modelled as subclasses of lis:Feature, as shown in Fig. 6. For the production pipeline, the feature is an instance of the class `ddbc:Outlet`, indicating that it is the outlet end of the pipeline. There is also an instance of the class `ddbc:ProcessPoint`, placed at the production pipeline outlet feature with the object property `ddbc:locatedAt`. We attach constraints, maximum and minimum values for pressure and temperature, at these points.

4.4. Materials and fluid properties

We have modelled streams as instances of the class `lis:Stream`, without defining or constraining streams, except for indicating which pipeline a stream flows through and its chemical composition. The composition is described as a set of quantity values which apply to a given stream at a defined process point, which is modelled as an individual of the class `ddbc:ProcessPoint`.

As a vocabulary for chemical entities, we use (a modified variant of) the ChEBI ontology (Degtyarenko et al., 2008), in which we only include those parts relevant for an oil and gas context.

The chemical components can be divided into two categories: specific molecules or hydrocarbon molecules grouped by their number of carbon atoms (so called ‘hypothetical components’). The latter category may be specified by (possibly unbounded) intervals of carbon atoms, e.g., C10-12 (containing arbitrary hydrocarbons with 10, 11, or 12 carbon atoms) and C10+ (containing arbitrary hydrocarbons with 10 or more carbon atoms).

A stream individual has `rdf:type` `lis:Stream`, and it is related to its chemical components with the object property `lis:hasPart`, as shown in Fig. 7. Each component has an `rdf:type`. For specific molecules, the type is the ChEBI class for that molecule. However, for hypothetical components, the type is an OWL class axiom stating that in addition to being a hydrocarbon (`obo:CHEBI_24632`), the component as a regulated interval of carbon atoms, as shown for the component `:stream1_c10plus` in Fig. 7.

Now that we have established how to represent the chemical components of a stream, we may attach, to each component, values for amount of substance, for molecular weight, for density, and other relevant properties. This is done following the pattern described in the OM2 ontology.

We have created a set of OTTR templates for stream components with amount of substance values, with optional fields for molecular weight and density. We have also created OTTR templates for creating hypothetical components, with different carbon atom count intervals. We also include SPARQL queries for retrieving chemical component data per stream from a DDB payload; see appendix on SPARQL queries. This has made rather complicated OWL patterns accessible to non-specialist engineers.

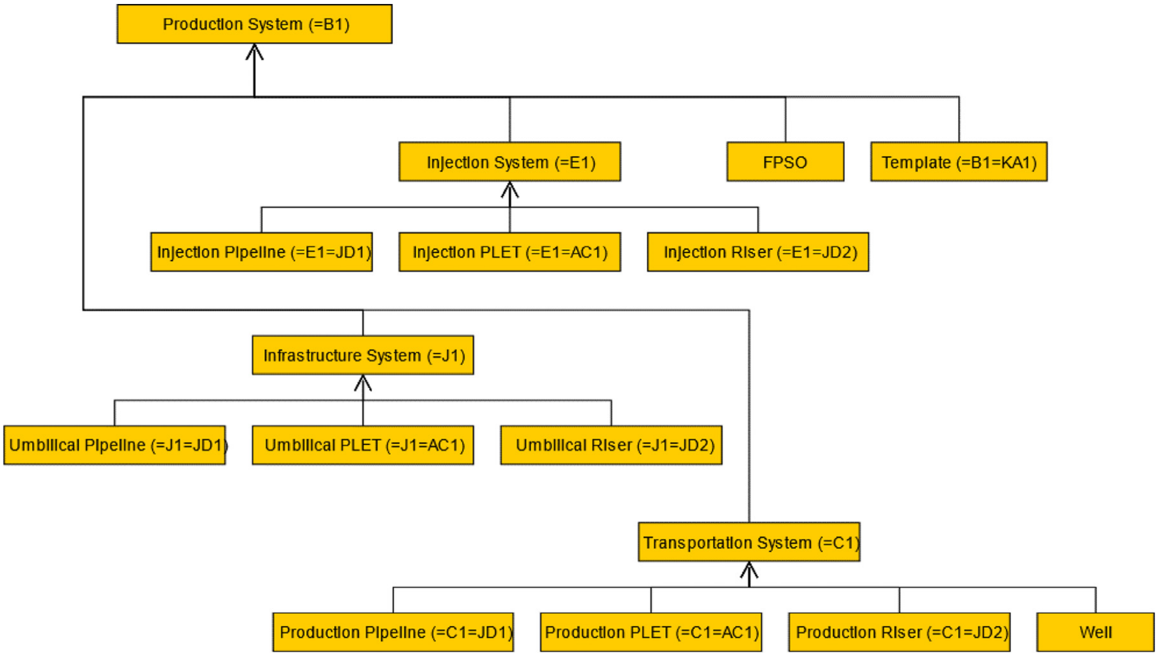


Fig. 4. Functional breakdown of the functional objects of the demonstration case. The arrows encode the object property `lis:funcionalPartOf`. RDS designators are shown in parentheses.

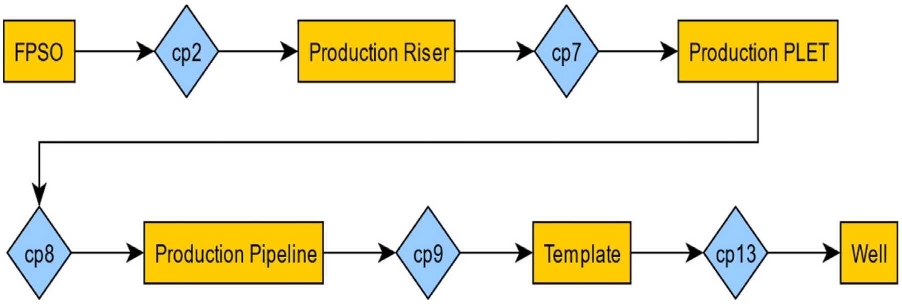


Fig. 5. The connections of the various components of the production line of the DDB demonstration case.

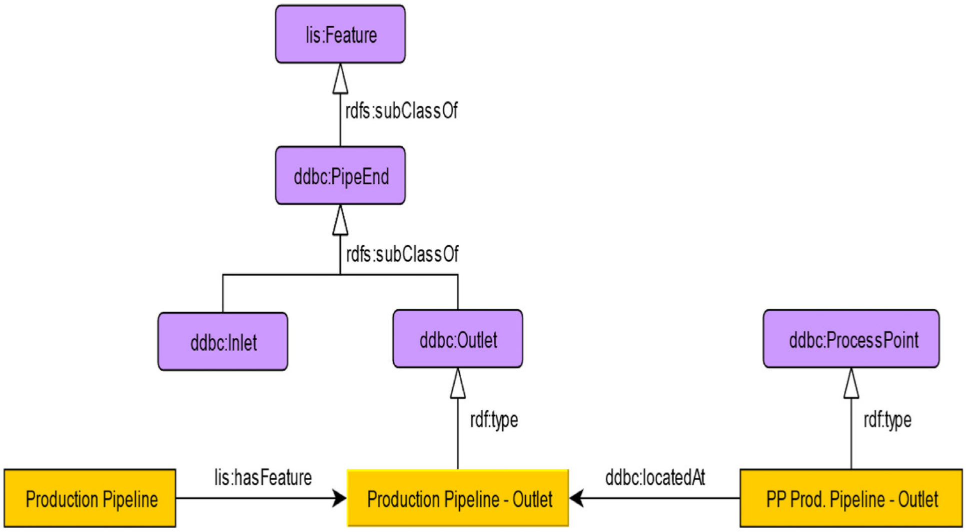


Fig. 6. The pipe outlet feature of the production pipeline and a process point located at the outlet feature. The rectangular yellow boxes are individuals, while the rounded magenta boxes are OWL classes. The arrows are named with the type of relation they implement. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

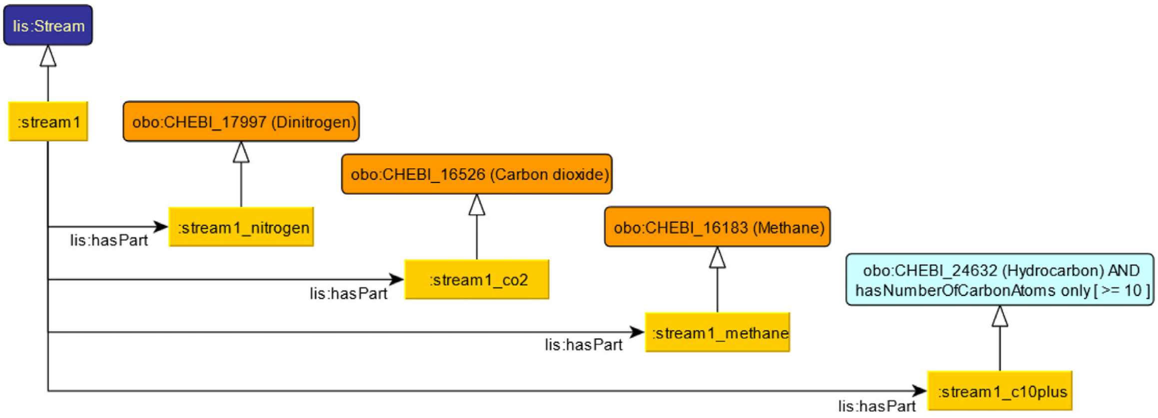


Fig. 7. Example of modelling of chemical components of a stream. The stream individual `:stream1` has four components, `:stream1_nitrogen`, `:stream1_co2`, `:stream1_methane`, and `:stream1_c10plus`.

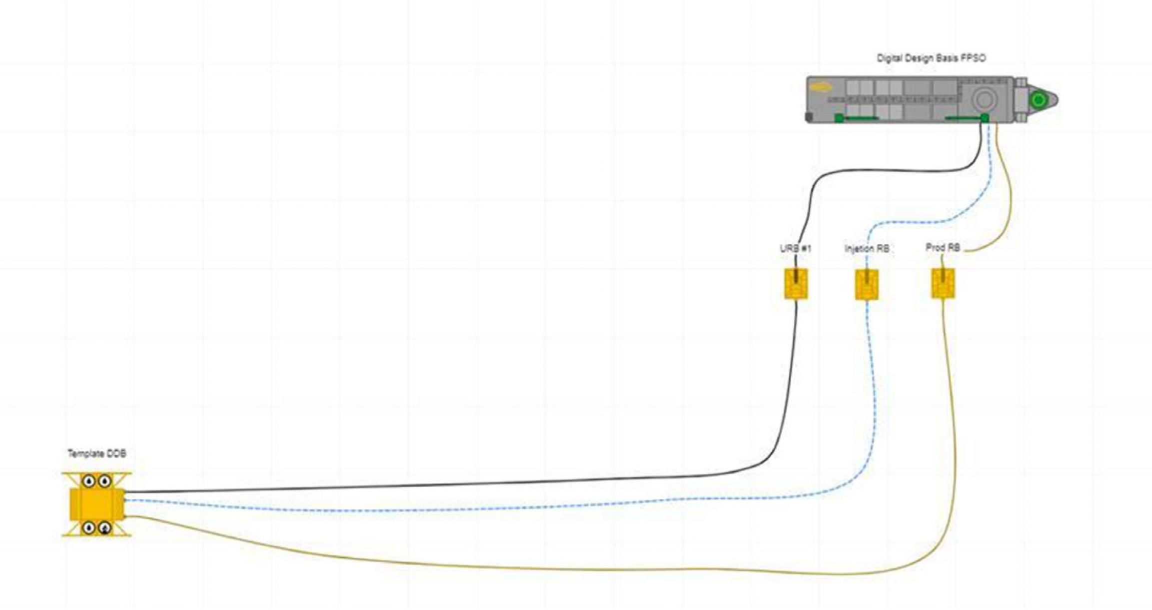


Fig. 8. The layout of the demonstration case. The user interface is a mock-up that represents the core elements in the model.

5. Demonstration with commercial engineering tools

We conclude the paper with a description of the demonstration cases. In all three cases we loaded a design basis description, derived from a real field development case, into the Fuseki triple-store database. We then used this data in three simple demonstration cases:

1. Import into the TechnipFMC Subsea Studio^{TM} tool.
2. Import into the Aize / Aker Solutions Subsea Configuration tool.
3. Extraction of data to run a pipeline sizing calculation.

The Digital Design Basis starts with a field layout from the operator. The model allows us to structure the rest of the DDB data and requirements around the technical systems in the layout. The case layout is shown in Fig. 8. It shows a well template with three lines: a production flow-line, an injection line and an umbilical for power and services.

We then enter a definition of the field development, including descriptive text. Next, we define the reservoir properties, using the stream modelling described above. We manage either full analyses or analysis with pseudo-components. Thermodynamic table files can be added as data blobs if desired. They can also be generated by the contractor.

We then define the wells, and use the concept of process points to design constraints at key locations. This is a flexible concept that allows simple linkage to the process flow sheet.

Then we define the technical systems in the concept. The types of technical systems are consistent with ISO/IEC81346. Once this is done, the operator can share a the Design Basis data with the engineering vendor. They can use this data in their applications by relying on the same common data model.

5.1. *TechnipFMC Subsea Studio^{TM}*

Subsea Studio^{TM} is a digital tool, developed by TechnipFMC, that provides a graphical framework for building field development concepts in the Concept, FEED and Tender phases of a project. It incorporates the vendor's subsea engineering knowledge and can be used to visualize, optimize and select the best field development. The proof of concept used Subsea Studio to evaluate design basis information for a fictitious operator and field.

The DDB model was populated with the field's design basis and functional requirements. These were transferred to Subsea Studio by mapping the DDB models and Subsea Studio's data model. Fig. 9 shows the

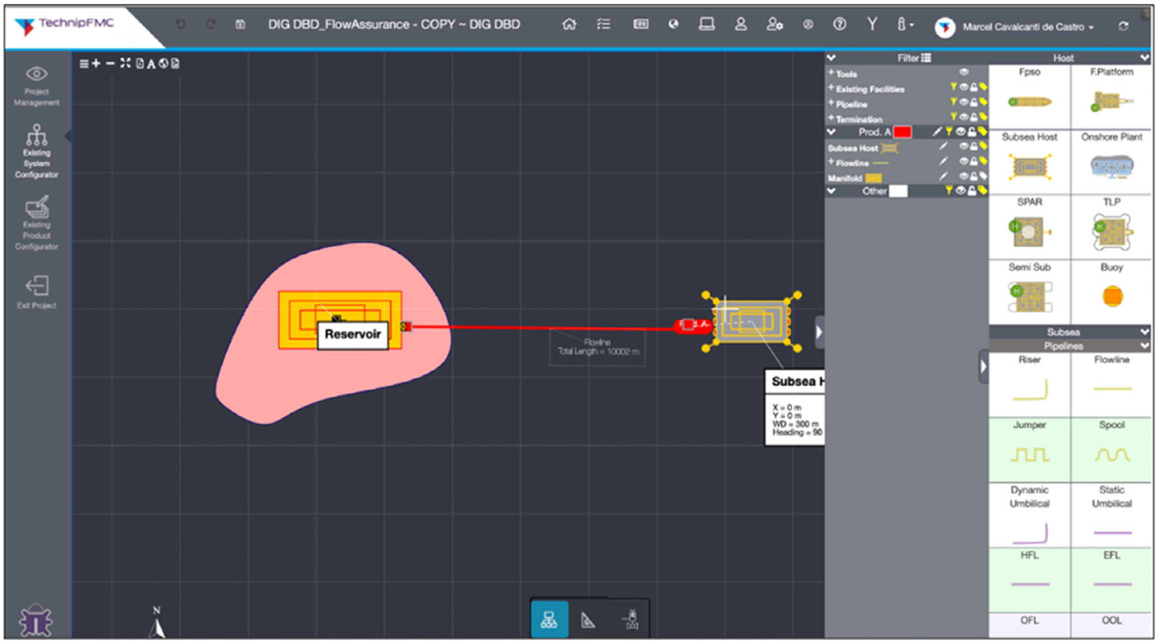


Fig. 9. The proof of concept in the TechnipFMC engineering tool.

field layout in Subsea Studio, which is composed by a reservoir, a template, pipeline and a PLET (Pipeline End Termination).

Subsea Studio consumes the DDB data from the Fuseki endpoint through a scripted interface which maps the two models. The goal with the integration is to calculate the outlet pressure and temperature at the end of a subsea pipeline for a single well scenario.

Data exchange of is done in real time by use of pre-defined SPARQL queries towards the Fuseki endpoint. The data is them imported in Subsea Studio as a new field layout. The following data is consumed:

- Composition of the reservoir fluids.
- Field water depth and ambient temperature
- The well production profile, giving time-series of water, gas and oil flow rates
- Surface roughness and thermal insulation for production pipeline

Figure 10 shows the values imported from the Fuseki endpoint to Subsea Studio for the Well production profile. This interface was specially built for this task and used exchange of JSON files. Subsea studio then used its interfaces to transfer the data further to simulation applications. Scaling this solution up requires programming of a mapped interface between the engineering tool and the model. We will come back to this challenge in the conclusions section below.

5.2. Sizing a pipeline

We demonstrated that it was possible to extract data semantically from the model and use it to define a consistent data set for a pipeline sizing tool. We assumed for the case that we were sizing a single production pipe from the Template to the Pipeline End Manifold (PLEM).

The DDB model provided the following input to the simulator:

- Flow rates for gas, liquid and water.
- The flowing well-head pressure and temperature.
- The heat transfer coefficient for the pipeline.
- Ambient temperature.
- Pipeline bathymetry and geometry (distance and elevation).
- A thermodynamic table for calculations (supplied as a blob of data).

The contractor was then required to estimate values for pipeline inner diameter and roughness. The simulation then calculated outlet pressure and temperature. The contractor could then adjust inner diameter to match constraints at the pipe end.

Figure 11 shows the flow assurance simulation results for the field. The plot shows the pressure and temperature results along the length of the pipeline (10km).

5.3. Integration with Aker Solutions design tool

Aker Solutions also used the DDB data in a series of evaluations using their Subsea Configurator tool. Their purpose was to use the design basis to choose the best subsea components from their catalogue for the concept. This would then allow them to estimate the price of the concept.

These evaluations required the following data from the design basis:

- Number of template slots required (4).
- Number of wells (3).
- Water depth (200m).
- Minimum, maximum and operating temperatures (10.3, 119.8 and 104.2 °C)
- Minimum, maximum and operating pressures (156, 239 and 198 bara).

The contractor was then required to specify or assume a 25 year field life. They also applied their internal guidelines for installation standards.

The design basis data was loaded from the Fuseki endpoint into the configuration tool, as shown in Fig. 12.

This input data is then used to explore options for the well-head assembly (the drill centre), as shown in Fig. 13. Here the system determines that eight Christmas Tree (XT) assemblies, five template manifolds and three well-heads could meet these design requirements.

The engineer can then examine each of these options, evaluate any gaps and trade-offs needed and select the best products for the concept. The tool then documents this choice and estimates cost.

6. Lessons learned and observations

The demonstrations above showed that it was possible to implement a DDB model using semantic web standards (OWL, RDF), public reusable ontologies (e.g. ChEBI and OM2) and ISO15926. The model represented

Year	Nominal Oil Flow	Nominal Water Flow	Nominal Gas Flow	Nominal Water Cut	Nominal GOR	Nominal Tin	Nominal WHFP	Turndown Oil Flow	Turdow
1	10000 Sm3/d	1000 Sm3/d	100000 Sm3/d	9,091 %	10 Sm3/Sm3	40 °C	50 bar	1000 Sm3/d	500
2	9670 Sm3/d	1000 Sm3/d	300000 Sm3/d	9,372 %	17,58 Sm3/Sm3	40 °C	50 bar	1000 Sm3/d	500

Fig. 10. Import of values into the TechnipFMC engineering tool.

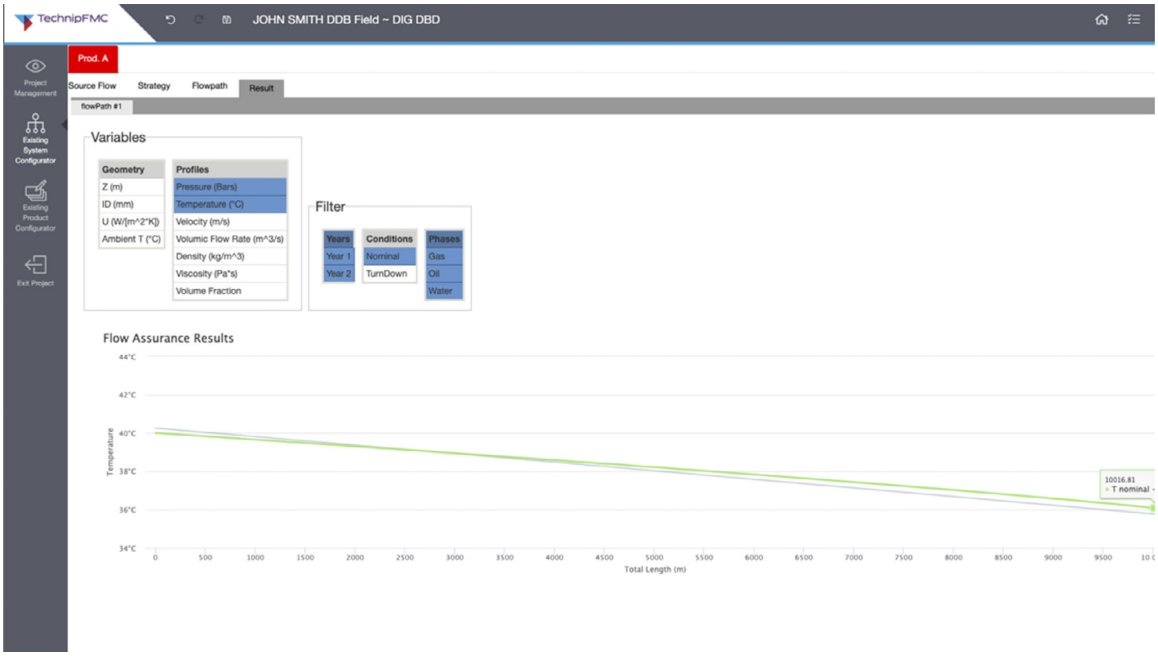


Fig. 11. Use of DDB data in a simulation tool.

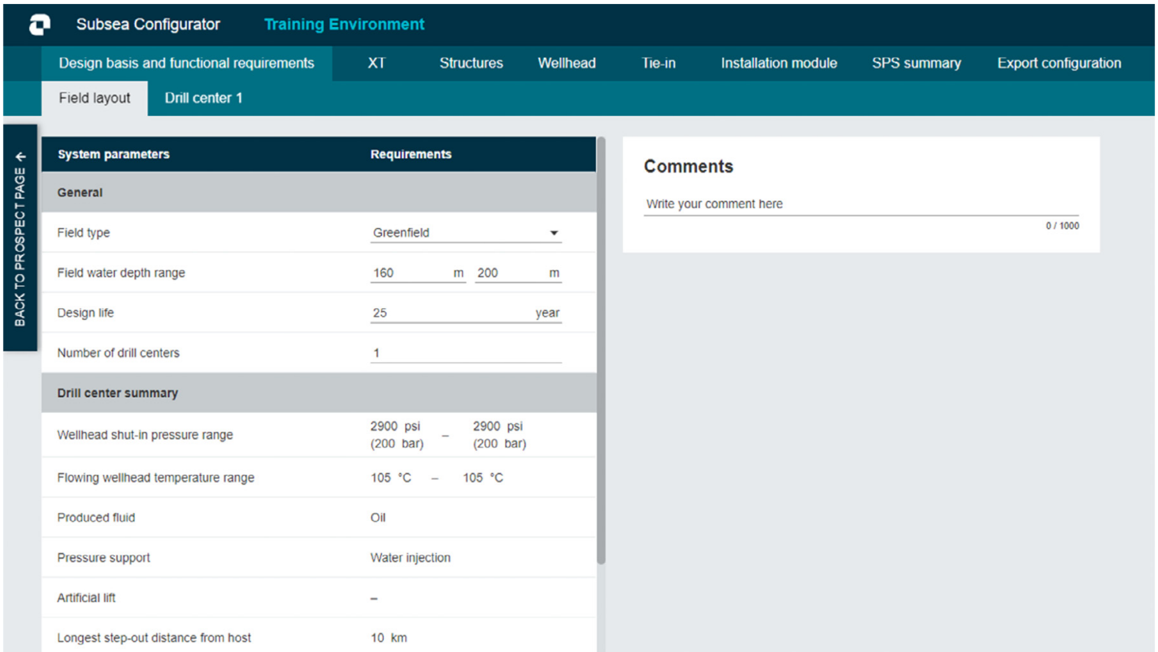
the facts and constraints needed for conceptual design in a machine-readable format, exposed by a SPARQL end-point. This allowed vendor's proprietary systems to extract data in a machine-readable format.

Tooling and software was all open-source. This is useful, but better performance and scalability may be obtained if we work with commercially-supported tools.

The data model is open-source and is published at the web-site of the SIRIUS Centre for Scalable Data Access (www.sirius-labs.no/ddb).

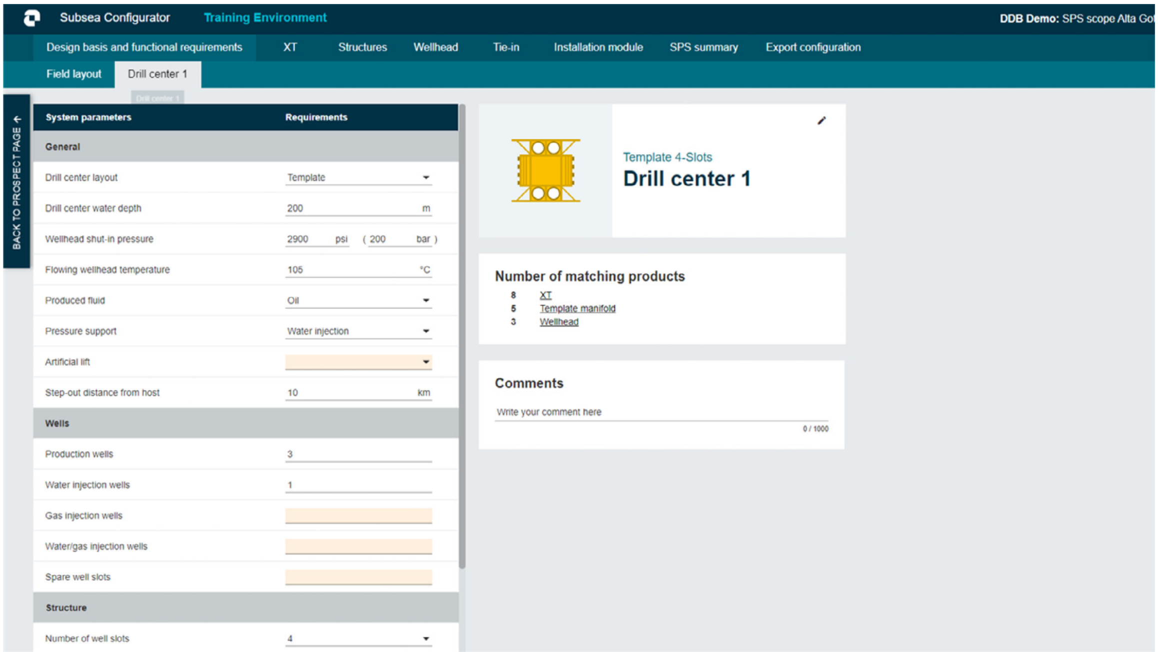
We found that the IEC/ISO81346 system breakdown approach from READI could be used and provided a valuable way of describing the field development concept in a top-down way. Organizing the design basis as attributes of functional blocks and the connections between them is conceptually easy to understand and provides a good upper structure for more detailed modelling later in the design process. Our work here is consistent with the READI Information Modelling Framework (IMF) approach (Fjøsna and Waaler, 2021). The practical experience here has been a valuable contribution to the further development of the IMF.

Consistency in units of measurement remains a challenge in all interoperability. We have built on recent evaluations of existing semantic models of units of measurement and chosen an approach that combines



System parameters	Requirements
General	
Field type	Greenfield
Field water depth range	160 m 200 m
Design life	25 year
Number of drill centers	1
Drill center summary	
Wellhead shut-in pressure range	2900 psi (200 bar) – 2900 psi (200 bar)
Flowing wellhead temperature range	105 °C – 105 °C
Produced fluid	Oil
Pressure support	Water injection
Artificial lift	–
Longest step-out distance from host	10 km

Fig. 12. Input data loaded into the Subsea Configurator tool.



System parameters	Requirements
General	
Drill center layout	Template
Drill center water depth	200 m
Wellhead shut-in pressure	2900 psi (200 bar)
Flowing wellhead temperature	105 °C
Produced fluid	Oil
Pressure support	Water injection
Artificial lift	
Step-out distance from host	10 km
Wells	
Production wells	3
Water injection wells	1
Gas injection wells	
Water/gas injection wells	
Spare well slots	
Structure	
Number of well slots	4

8	XT
5	Template manifold
3	Wellhead

Fig. 13. Proposal of options in the Subsea Configurator.

rigour with flexibility. However, all interoperability work needs to build on an agreement about the quantity types for numerical values and their unit representation.

We saw that some of the representations and modelling patterns in OWL were very verbose. This was a particular issue for the representation of fluid compositions, where we need to represent a vector of compositions. Our implementation resulted in around 8000 lines of code for exchanging a fluid composition with 25 components. Using templates allowed us to build models using tabular data and vectors of variables. This hid the verbosity from end-users. However, we are still concerned about the ability to scale up these patterns to large-scale engineering projects.

A possible approach is to move to a hybrid approach, where we mix object-oriented structuring of blocks, connections and parameters with an ontology that defines the meaning of all the blocks and parameters and allows us to reason across the object-oriented data set. We are working further on this in the context of the READI IMF.

Finally, the proof of concept looked only at parameters, i.e. requirements that can be reduced to a number, not functional requirements per se. We are working further on how we can link enumerated, qualitative or textual requirements to the modelling elements in the digital design basis. Transforming textual requirements to numbers, bits and bytes is hard. However, we believe that current work on semantics and natural language will help in this.

7. Conclusions

In this work we have reported how a consortium of operators and competing EPC companies demonstrated that a common data model can be used to capture and share design basis data in early phase development. This proof of concept was done to explore how recently matured semantic technologies could be used to move an important business work flow away from a document-based model to a data-centric model.

Semantic modelling made this representation possible and allowed data to be entered in a structured way and be consumed by engineering applications. We have validated the basic approach, which builds on reusing existing semantic models where possible. We have also demonstrated the feasibility of mixing the modelling approaches defined by ISO15926 and ISO/IEC81346.

We believe the industry needs to have more projects like this, where consortia along the supply chain work with academia and software vendors to agree on interoperability standards by working on real, non-trivial problems. Fortunately it appears that the European Union, World Economic Forum and International Organization of Oil & Gas Producers agree with this goal.

Our approach here is not restricted to the oil & gas industry. The system breakdown and modelling of fluid properties can be extended straightforwardly to chemical, fine chemicals and energy applications. A good first step would be an extension of the RDS for Oil & Gas to ensure that it covers the unit operations in these other domains. We are working further with the READI partners to do this.

This work was experimental, where we were seeking to prove that recent advances in system modelling and ontologies could be applied to a real design basis problem. This meant the integrations with tools tended to be pragmatic rather than user-friendly and scalable. Further work is needed to provide the tools that are necessary to integrate models like this into engineering work flows. Semantic technology tools are too low-level to be used by practising engineers. OTTR templates have addressed some of these usability challenges, but further work must focus on developing a set of tools to simplify configuration of the model and access to data.

A graphical tool is needed for building system-oriented models by selecting nodes and connecting these nodes with topological and semantic relationships. This tool should also allow the configuration of design basis data in a guided, but flexible sequence. This interface can exploit the semantic content of the model to provide flow and check consistency.

This interactive tool must be supplemented with tools that allow data to be entered into the DDB in bulk, using tabular data. OTTR provides some of this functionality, but this functionality needs to be lifted up into interactive tools.

Mappings need to be developed towards common process simulators and engineering design databases. We need to both read design variables from the DDB and write calculated results back to the DDB. Here we need to work together with the industry so that our models and vendor's models converge over time to an actual or de-facto standard. We are cautiously optimistic about the possibilities of this being successful. We see that many influential vendors are interested in exposing their data using semantic schemas and open formats. We should aim for work practices where a DDB harvests data from engineering tools without intervention from the engineer.

The proof-of-concept has also helped us to develop a more systematic approach to defining the digital design basis. Time and organizational constraints meant that this first effort was more inductive than deductive. The modelling was driven by the data we had to represent and we then drew systematic conclusions from the solutions developed.

The lessons from this project have been taken up in further, ambitious initiatives by each of the partners. In particular we are contributing to the revisions of READI IMF and RDS for Oil & Gas. This work aims to address the tooling challenges above in the context of several on-going field development projects. It is also developing a formal systematization of modelling and use of data in engineering projects. The results of this work have also been taken into the development of the forthcoming Part 14 of the ISO15926 standard. We hope that these initiatives together will provide elements for establishing a practical, scalable framework for sharing information in the process engineering sector.

8. Glossary and abbreviations

ChEBI	Chemical Entities of Biological Interest.
EPC	Engineering, Procurement and Construction.
IMF	Information Modelling Framework.
OM2	An Ontology of Units of Measure.
OWL	Web Ontology Language (a W3C standard).
RDF	Resource Description Framework (a W3C standard).
RDS	Reference Designation System: a set of codes used in ISO/IEC81346.
READI	Norwegian Joint Industry Project on semantic modelling for engineering applications.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was organized as an innovation project in the SIRIUS Centre for Research Driven Innovation at the University of Oslo The centre is financed by the Research Council of Norway, Project number 237898. Work was also financed by the participating companies.

References

Benayoune, M., Preece, P.E., 1987. Review of information management in computer-aided engineering. Computers & Chemical Engineering 11 (1), 1–6. doi:10.1016/0098-1354(87)80001-7. https://www.sciencedirect.com/science/article/pii/0098135487800017.

Braaksma, A.J.J., Klingenberg, W.W., van Exel, P.W.H.M.P., 2011. A review of the use of asset information standards for collaboration in the process industry. Comput. Ind. 62 (3), 337–350. doi:10.1016/j.compind.2010.10.003. https://www.sciencedirect.com/science/article/pii/S0166361510001387.

Cyganiak, R., Wood, D., Lanthaler, M., 2014. RDF 1.1 Concepts and Abstract Syntax. https://www.w3.org/TR/rdf11-concepts/.

CFIHOS, Standards. https://www.jip36-cfihos.org/cfihos-standards/.

Design Basis, DOE Directives, Guidance, and Delegations. https://www.directives.doe.gov/terms_definitions/design-basis.

Degtyarenko, K., Matos, P.d., Ennis, M., Hastings, J., Zbinden, M., McNaught, A., Alcántara, R., Darsow, M., Guedj, M., Ashburner, M., 2008. ChEBI: a database and ontology for chemical entities of biological interest. In: Nucleic Acids Research 36. Publisher: Oxford University Press, p. D344. doi:10.1093/nar/gkm791. (Database issue) https://www.ncbi.nlm.nih.gov/sites/ppmc/articles/PMC2238832/.

DEXPI, 2021. DEXPI P&ID Specification Version 1.3. Technical Report. Dechema, Frankfurt am Main. https://dexpi.org/wp-content/uploads/2020/09/DEXPI-PID-Specification-1.3.pdf.

Doe, R.M., Kaur, K., Selway, M., Stumptner, M., 2021. A review of interoperability in the AECO sector compared with the Oil & Gas sectors. In: Proceedings of the 38th International Conference of CIB W78. Luxembourg, pp. 282–293. https://itc.scix.net/paper/w78-2021-paper-029.

Fillinger, S., Bonart, H., Welscher, W., Esche, E., Repke, J.-U., 2017. Improving interoperability of engineering tools - data exchange in plant design. Chem. Ing. Tech. 89 (11), 1454–1463. doi:10.1002/cite.201700032. https://onlinelibrary.wiley.com/doi/pdf/10.1002/cite.201700032.

Fillinger, S., Esche, E., Tolksdorf, G., Welscher, W., Wozny, G., Repke, J.-U., 2019. Data exchange for process engineering - challenges and opportunities. Chem. Ing. Tech. 91 (3), 256–267. doi:10.1002/cite.201800122. https://onlinelibrary.wiley.com/doi/pdf/10.1002/cite.201800122.

Fillinger, S., Seyfang, B., Wozny, G., Repke, J.-U., 2021. PlantDesign - prototypical implementation of an integrated engineering tool. Computers & Chemical Engineering 147 (147–2021), 107233. doi:10.1016/j.compchemeng.2021.107233. Saved from BrowZine: https://browzine.com/articles/434212808.

Fjøsna, E., Waaler, A., 2021. READI Information modelling framework (IMF). Asset Information Modelling Framework. Technical Report. READI Joint Industry Project, Oslo. https://readi-jip.org/wp-content/uploads/2021/03/Information-modelling-framework-V1.pdf.

Gielingh, W., 2008. An assessment of the current state of product data technologies. Comput.-Aided Des. 40 (7), 750–759. doi:10.1016/j.cad.2008.06.003. https://www.sciencedirect.com/science/article/pii/S0010448508001139.

Gilabert, E., Conde, E., Abaunza, J., 2012. A standard-based data model for configuration management and maintenance support. IFAC Proceedings Volumes 45 (31), 139–144. doi:10.3182/20121122-2-ES-4026.00038. https://www.sciencedirect.com/science/article/pii/S1474667015339094.

Goldfarb, S., Dealmeida, J., 1996. PDXI progress report. Computers & Chemical Engineering 20, S91–S96. doi:10.1016/0098-1354(96)00026-9. https://www.sciencedirect.com/science/article/pii/0098135496000269.

Gulla, J.A., Strasunskas, D., Tomassen, S.L., 2006. Semantic Interoperability in Multi-Disciplinary Domain. Applications in Petroleum Industry. In: Proceedings of the 2nd International Workshop on Contexts and Ontologies: Theory, Practice and Applications (C&O-2006). Riva del Garda, Italy, p. 5. http://ceur-ws.org/Vol-210/.

Gulla, J.A., Tomassen, S., Strasunskas, D., 2006. Semantic Interoperability in the Norwegian Petroleum Industry. In: Information Systems Technology and its Applications, 5th International Conference ISTA'2006. Klagenfurt, p. 15.

Hakkarainen, S., Hella, L., Strasunskas, D., Tuxen, S., 2006. A Semantic Transformation Approach for ISO 15926. In: Roddick, J.F., Benjamins, V.R., Si-said Cherfi, S., Chiang, R., Claramunt, C., Elmasri, R.A., Grandi, F., Han, H., Hepp, M., Lytras, M.D., Mišić, V.B., Poels, G., Song, I.-Y., Trujillo, J., Vangenot, C. (Eds.), Advances in Conceptual Modeling - Theory and Practice. Springer, Berlin, Heidelberg, pp. 281–290. doi:10.1007/11908883_34.

Hitzler, P., Krötzch, M., Parsia, B., Patel-Schneider, P. F., Rudolph, S., 2012. OWL 2 Web Ontology Language Primer (Second Edition). https://www.w3.org/TR/owl2-primer/.

ISO/IEC81346-1, 2009. Industrial systems, installations and equipment and industrial products - Structuring principles and reference designations - Part 1: Basic rules. Technical Report. International Organization for Standardization, Geneva.

Kim, B.C., Kim, B., Park, S., Teijgeler, H., Mun, D., 2020. ISO 15926-based integration of process plant life-cycle information including maintenance activity. Concurrent Engineering 28 (1), 58–71. doi:10.1177/1063293X19894041. Publisher: SAGE Publications Ltd STM.

KonKraft, 2018. Project Competitiveness - changing tide on the Norwegian continental shelf - Summary and recommendations from the Committee. Technical Report. KonKraft, Oslo. https://konkraft.no/wp-content/uploads/2020/03/Summary_and_recommendations.pdf.

Leal, D., 2005. ISO 15926 "Life cycle data for process plant": an overview. Oil & Gas Science and Technology 60 (4), 629–637. doi:10.2516/ogst:2005045. http://ogst.ifpenergiesnouvelles.fr/10.2516/ogst:2005045.

Man, D., 2013. Ontologies in computer science. Didactica Mathematica 31, 43–46.

Marquardt, W., 1999. Von der Prozeßsimulation zur Lebenszyklusmodellierung. Chem. Ing. Tech. 71 (10), 1119–1137.

, 2010. In: Marquardt, W. (Ed.), Ontocape: A re-usable ontology for chemical process engineering.RWTH. Springer, Heidelberg ; New York. OCLC: ocn449851783.

Martın-Recuerda, F., Walther, D., Eisinger, S., Moore, G., Andersen, P., Opdahl, P.-O., Hella, L., 2020. Revisiting ontologies of units of measure for harmonising quantity values - A Use Case. In: The Semantic Web - ISWC 2020. Springer, Cham, pp. 551–567. doi:10.1007/978-3-030-62466-8_34.

Mayer, W., Stumptner, M., Grossmann, G., Jordan, A., 2013. Semantic interoperability in the Oil and Gas industry: a challenging testbed for semantic technologies. In: AAAI Technical Report FS-13-04. AAII, Arlington, VA, p. 4. https://www.aaai.org/ocs/index.php/FSS/FSS13/paper/viewFile/7602/7550.

Motik, B., Patel-Schneider, P.F., Cuenca Grau, B., 2012. OWL 2 Web Ontology Language Direct Semantics (Second Edition). W3C Recommendation. W3C. https://www.w3.org/TR/2012/REC-owl2-direct-semantics-20121211/.

READI, 2020. Reference Designation System for Oil and Gas - READI. https://readi-jip.org/reference-designation-system-for-oil-and-gas/.

READI, Shaping the future of digital requirements and information flow in the oil and gas value chain. https://readi-jip.org/.

Skjæveland, M.G., Lupp, D.P., Karlsen, L.H., Forssell, H., 2018. Practical ontology pattern instantiation, discovery, and maintenance with reasonable ontology templates. In: The Semantic Web - ISWC 2018. Springer International Publishing, Cham, pp. 477–494. doi:10.1007/978-3-030-00671-6_28.

Sloley, A., 2014. Ponder the Process Design Basis. https://www.chemicalprocessing.com/articles/2014/ponder-the-process-design-basis/.

Smith, B., 2006. Against idiosyncrasy in ontology development. In: Formal Ontology and Information Systems. IOS Press, Amsterdam, pp. 15–26. http://ontology.buffalo.edu/bfo/west.pdf.

Wiedau, M., Tolksdorf, G., Oeing, J., Kockmann, N., 2021. Towards a systematic data harmonization to enable AI application in the process industry. Chem. Ing. Tech. 93 (12), 2105–2115. doi:10.1002/cite.202100203. https://onlinelibrary.wiley.com/doi/pdf/10.1002/cite.202100203.

Wiedau, M., von Wedel, L., Temmen, H., Welke, R., Papakonstantinou, N., 2019. ENPRO Data integration: extending DEXPI towards the asset lifecycle. Chem. Ing. Tech. 91 (3), 240–255. doi:10.1002/cite.201800112. https://onlinelibrary.wiley.com/doi/pdf/10.1002/cite.201800112.

Wiesner, A., Morbach, J., Marquardt, W., 2011. Information integration in chemical process engineering based on semantic technologies. Computers & Chemical Engineering 35 (4), 692–708. doi:10.1016/j.compchemeng.2010.12.003. https://www.sciencedirect.com/science/article/pii/S0098135410003649.

Wiesner, A., Saxena, A., Marquardt, W., 2010. An ontology-based environment for effective collaborative and concurrent process engineering. In: 2010 IEEE International Conference on Industrial Engineering and Engineering Management, pp. 2518–2522. doi:10.1109/IEEM.2010.5674186. ISSN: 2157-362X.

Wozny, G., Gutermuth, W., Kothe, W., 1992. CAPE In der Verfahrenstechnik aus industrieller Sicht: status, Bedarf, Prognose oder Visjon? Chem. Ing. Tech. 64 (8), 693–699.