INFORMATION MODELLING FRAMEWORK (IMF)

Asset Information Modelling Framework

Editors
Erlend Fjøsna, Arild Waaler

Contributors
Torleif Saltvedt, Per Øyvind Øverli, Idar Ingebrigtsen,
Jann Kåre Slettebakk, Bjørn Roar Ydstebø, Trond Karlsen,
Johan Wilhelm Klüwer, Martin Georg Skjæveland
And the community around

DECEMBER 17, 2020

https://readi-jip.org

READI

Change log

Document Version	Date	Change	
Draft	03.07.20		
Release 0	17.12.20	Complete IMF concept description	For review / approval

Contents

List of terms and their interpretation in the context of this document

Term	Description	Explanation
Asset	Physical or logical object owned by an organization, having either a perceived or actual value to the organization	In this document, asset is primarily used to refer to an O&G facility or subsystems thereof
Type	A set of information which specifies the common attributes shared by all instances of the Type	E.g., a valve of Type 'Ball Valve' has the attributes that are common to all valves of this Type of valve
RDS	Reference Designation System	The conceptual world of ISO/IEC 81346
RDL	Reference Data Library	The conceptual world of ISO 15926
RDF	Resource Description Framework	A standard way of writing triples
OWL2	Web Ontology Language	A standard way for modelling ontology
SKOS	Simple Knowledge Organization System	A standard that provides a way to represent controlled vocabularies
URI	Uniform Resource Identifier	For linking to an online resource
IRI	Internationalized Resource Identifier	For linking to an online resource
OTTR	Reasonable Ontology Templates	A tool for representing and instantiating ontology modelling patterns
IMF	Information Modelling Framework	The essence of this document
SCD	Scope Condition Demand	Structured requirement information
Digital Twin	Virtual representation of an asset	A concept for information model-based decision support
Digital Thread	Flow of information along life cycle	A concept for lossless flow of information between value adding steps
Digitizing	Creating a digital information format	e.g., PDF, CSV, etc.
Digitalizing	Creating a digital information process	e.g., Digital Product Configurator

PCA	POSC Caesar Association	The organization responsible for content development. of ISO 15926
MMD	Material Master Data management	AIBEL Industrial ontology
READI	Requirement Asset Digital Lifecycle Information	The JIP that has published this document
JIP	Joint Industry Project	A collaboration between operators and suppliers to solve a common problem.
EPC(I)	Engineering Procurement Construction (Installation)	Oil & Gas main contractors
TIRC	Technical Information Requirement Catalogue	Digitized NORSOK Z-018 standard
Ontology	The concepts used to model an asset	A model of (a relevant part of) an asset, listing the types of object, the relationships that connect them, and constraints on the ways that objects and relationships can be combined
ETL	Extract, Transform, Load	Copying data from a source, to a destination which represents the data differently or in a different context. This is a key concept in Data Warehouses.

1 Introduction [leaders, all]

1.1 Objectives [leaders, all]

The overall objective of this document is to bring an understanding of *how to move forward* towards creating the intended impact. As such, the target is not to provide a final recipe and complete set of libraries, but to define a clear framework based on decisions made and directions set, enabling an efficient transition into the implementation phase. The specific objectives of the document are to:

- Describe the elements of the IMF from the perspective and needs of the users throughout the life cycle of a project, i.e., the perspectives of the Client, the EPC(I) Contractor, Supplier, and the Operator
- Specify how the IMF shall be developed and extended
- Specify how the IMF shall be implemented in terms of integration with legacy information infrastructures
- Specify how the IMF shall be supported on a continuous basis
- Deliver use cases for implementation and demonstration

1.2 Background

1.2.1 Value proposition

Today, manual processing of requirements and technical information has a huge cost. Owners, operators, EPC contractors, and suppliers in the Oil and Gas Industry spend massive amounts of expert man hours to specify, implement and verify requirements versus design. Yet despite this we continue to see quality deviations in project execution and operation. Many expert man hours are also used to interpret, combine, and re-enter technical information in order to understand and explain how a facility is designed to function. This expert attention to requirements and technical information is needed in all phases of the lifecycle, from early engineering to operation and subsequent modification. There is a strong incentive to reduce cost, and an increasing recognition that digital technology is a powerful enabler.

The current flow of information is designed for a manual workflow. Technical information is presented in document-oriented formats that are easy to read by humans - usually in the form of documents and drawings. Content is largely descriptive, with less emphasis on formal information structure. To some extent, documents contain properties, e.g., in data sheet documents, but although documents may contain a range of properties, the information richness is low and is restricted by a fixed format.

The structure of technical information in current practice is almost flat. Typically, the structure is built around a proprietary Engineering Numbering System, a fixed set of system codes and a list of properties. This lack of richness in standardized information structures impedes the exchange of structured data. For instance, when data from databases is exchanged as a comma separated file with property values, the structure of the database is lost. Even when the digital thread is claimed to be intact, such as when the same application is situated at both sides of a data exchange, there is still an irrevocable loss of information in the transfer.

As a consequence, both requirements and technical information are, to a vast extent, fragmented. Each fragment gives just one piece of information and is structured according to particular needs. It typically adopts the specific perspectives of, for example, a discipline or a stage in the project lifecycle. Two fragments can contain information with widely different levels of granularity Since this granularity is so different, the fragments cannot easily integrate as a whole. This is why expert knowledge is always needed to interpret the data correctly.

The Information Modelling Framework is introduced to bring about a change in the way of working. The goal is that the industry should share a single model of an asset with a rich, multidimensional structure.

The Information Modelling Framework is such designed as to be able to proceed incrementally, modelling fragments of information step by step, with focus on where the largest gains lie, and such that each step gives incremental value without the need for a huge upfront investment.

The Information Modelling Framework also provides a collection of reusable **model blocks** that allow users to model individual fragments of information. A case in point can be design codes used by subject matter experts. These model blocks could be combined into larger models and be gradually extended with more detail, exploiting mechanisms for inheritance and propagation of information across the resulting models.

The resulting models serve to show how information is interrelated and support moving through the data using the perspective of choice. They enable richer data to be exchanged or shared, they increase the breadth and quality of data, and open up for automation of repetitive and tedious data processing tasks that today can only be done by human experts.

1.2.2 Enabling Technology

There have been previous attempts to modernize how this industry creates information models to represent the assets that we build and operate. These attempts have been driven by an early understanding of the value to be gained. So far there has been no success in scaling the solutions in size and across the industry. What is different today is that information science, technology, and methodology have advanced, while the readiness for change has reached a tipping point, as pointed out below.

The results of international standardization efforts (ISO15926 and ISO/IEC 81346) are now providing a unified way of structuring and codifying objects and systems, such that they can be identified accurately and unambiguously.

This standardisation enables the industry to align on one way of codifying, and it allows identification to use any relevant aspect, be it in a functional hierarchy, a location hierarchy, a product hierarchy, or other aspects which can be implemented as needed.

This standardisation also allows replacement of the numerous legacy coding manuals that are incompatible, incomplete, and a cause of significant cost.

The World Wide Web Consortium (W3C) has driven the development and demonstrated the value of semantic technology as a means to build and exploit rich information models in continuously evolving configurations. Other industries, such as biotech, have also furthered its development and use.

This technology is a vital tool for the information modelling needed to model an asset or facility such that it can be accurately and unambiguously described.

Other international standardization efforts (ISO 15926) are providing a unified way of formulating, classifying, relating and connecting information - and building libraries that are made available as industry-shared resources.

This standardisation allows the construction of what can be likened to 'digital catalogues' that serve to standardise information for types of objects and systems that are common building blocks across the industry.

Data storage and data processing power has decreased by magnitudes in cost and is increasingly available. This means that IT capacity is much less a limiting factor, both as regards power as well as scalability. The availability of novel database systems and other, fit-for-purpose, software for modelling and harvesting asset information reduces the need for development of specific software.

This computing opportunity can enable automation of much of the knowledge work that today is done manually, and thereby reduce time, cost, and quality issues.

1.2.3 Reusable information models

An asset or facility is usually built as a system configured from a set of building blocks that have at some level the same design across many facilities. These building blocks can be simple commodity items - such as valves – or they can be a more complex subsystem – such as an air compressor package. This approach is considered best practice in the industry, as it helps drive down cost and execution risk, as well as reducing operational and maintenance uncertainties. To support and further promote this approach, the Information Modelling Framework incorporates template functionality and the ability to model fragments of an asset, so that these can subsequently be integrated into a complete model.

Example: an air compressor package can be represented by a fragment asset model which can be re-used from project to project, with only a few characteristics being project specific. It can seamlessly integrate into the overall asset information model specific to the project whenever required, significantly reducing interface- and integration cost.

Example: a commodity valve of a given type can be represented as a set of characteristics particular to that type by means of using template functionality. Such a template function supports multiple reuses of the information developed to represent a commodity type, reducing the associated engineering cost towards zero.

1.2.4 Users

This document not only serves to describe an information modelling framework, it also brings together the expertise from several, very different, disciplines and domains. Failing to address the challenge of cross-discipline understanding would be a mistake, as it is often a significant barrier to success. Therefore, the information modelling framework is described from different perspectives, with emphasis on how the framework is going to be used, and by whom. Some of the chapters in the document are intended for specific users, but all users are encouraged to read the whole document. The following types of users have been identified:

Leaders, and All

Emphasis on gaining understanding of the value proposition, obtaining confidence in the short term and long-term feasibility, and enable messaging towards stakeholders and the wider audience, to promote the importance and industry ownership this undertaking has.

Client and requirements setter

Emphasis on gaining understanding of how to produce, issue, maintain and manage requirements on a digitalized format, and how to navigate the requirements sources.

Engineer

Emphasis on gaining understanding of the information modelling framework as a toolkit that can significantly increase the power of digitalisation in the industry, aimed at knowledge workers, from early idea conception and requirements setting, through design, building, operation and decommissioning.

Operators and Contractors

Emphasis on gaining understanding of how to operate the asset information model, how to utilize the model, and how to navigate the model and harvest information, including employing Digital Twin-type applications.

Commented [ØE1]: Savner vendors/suppliers i listen her for å få med hele supply-kjeden. Mer et spm: Burde mundigheter være nevnt? De er premissgivere gjennom sine lover og regler

Regulators and Authorities

Emphasis on gaining understanding of how the information-modelling framework can be used to improve compliance with their regulations and requirements. They can also gain insight into how they can digitalize their processes to exploit the possibilities offered by this framework.

Digital SME

Emphasis on gaining understanding of how to implement the information modelling framework, how to perform the actual asset information modelling, and how to navigate the model and harvest information, including employing Digital Twin-type applications.

Ontology SME

Emphasis on gaining understanding of the mechanisms of the information modelling framework, and how to further develop and extend the framework and its mechanisms.

IT SME

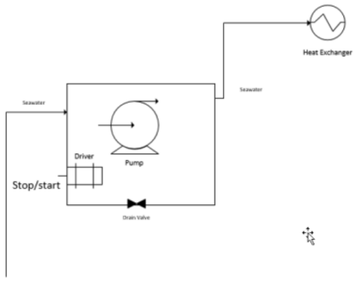
Emphasis on gaining understanding of how to implement the technology of the information modelling framework and integrate or co-exist this technology with legacy IT systems, both as regards computing technology as well as regards different paradigms of information modelling.

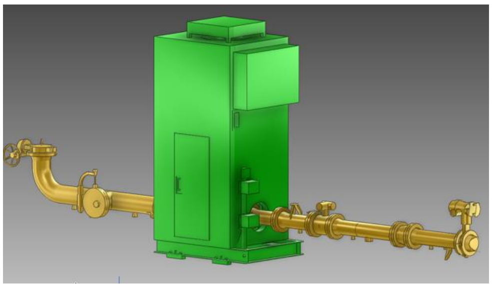
1.2.5 Industry example use case

A specific use case is followed through the steps of information modelling, such that concrete examples and solutions are demonstrated throughout the document. The use case is chosen from the oil & gas domain to be easy to relate to real world situations. For this purpose, a seawater-based cooling function is selected. The *System 50, Seawater System,* is a recognised subsystem in the industry. It shall be included in the modelling, as shall what we are describing as the *cooling function.* One level down – when thinking *System of Systems* - is the pump function and the cooler function. Still lower down are the functions that can be realised with available products: the pump units, the cooler(s), as well as specific individual catalogue products such as pump (hydraulic part), drive motor, valves, etc.

Modelling of the use case can be performed at any level of granularity desired, and in a top-down fashion. The higher level is the domain of the systems engineer, whereas the lower level is the domain of the subject matter experts and product suppliers.

Below, the flow diagram outlines the cooling function, and the 3D rendering illustrates how the pump function may look like as a package.





The IMF allows any level of granularity (how many levels down to do the modelling), and the use case shall also serve to demonstrate this capability.
The use case that is selected is mainly representative for the functional disciplines. An example which represents the area disciplines could be considered, if experience shows that this is needed.

Seawater System

The Seawater System is used to reference all functions that have to do with handling seawater, irrespective of its use. As such it does not necessarily reflect the functional breakdown of the asset. Typically, seawater is provided for functions such as Fire Fighting, and Cooling. In the context of the Use Case, it is the function of providing a supply of cooling water (water from the sea) as well as disposing of the water after use (discharge to the sea) that is of interest.

Cooling function (subsystem)

The Cooling function subsystem has the purpose of transporting heat away from a heat source. At the highest level the main parameter for this function is amount of heat per unit of time, e.g., as MW/hr. With more detail, this can be translated into rate of flow of cold liquid, e.g., m3/hr @ degC seawater. Typical sub-functions are: Supply of seawater, pumping of seawater, Heat Exchange, and Return and disposal of seawater. The Cooling function depends on an input of energy, a supply of cooling fluid, and a sink for spent cooling fluid. It delivers cooling.

Pump function (subsystem)

One of the essential subfunctions of the Cooling function is the creation of a movement of the cooling fluid – or in simpler terms Pumping of seawater. The main parameter here is the flow rate to be provided at the back pressure created by the downstream system piping and connected equipment. Main subfunctions are a Drive function (electrical motor) and Fluid mover function (pump hydraulic unit).

Cooler function (subsystem)

The other essential subfunction of the Cooling function is the Cooler function: A Heat Exchanger with the purpose of transporting heat from one fluid to another. In the Use Case this is done by thermally connecting the cold flowing fluid with the hot flowing fluid by means of a dividing surface made from heat conductive material. The main parameter is Heat transfer capacity, e.g., MW/hr.

Commodity items

Only the main functions have been described above. Adding to these are functions of fluid transport (piping), fluid isolation and control (valves), process control, safety, monitoring, and other system functions. Also, drilling further down into the model we find functions such as Fixation (e.g., of motor to the skid), and Connection (of e.g., of cable to the motor terminals).

1.2.6 Leveraging previous work

The Information Modelling Framework draws upon and further develops more than 20 years of experience and developments, ranging from industry initiatives to research programs and development projects. These are the most important sources of knowledge and experience.

PCA and ISO 15926

The pioneering work of PCA started more than 20 years ago. With the clarity of hindsight, it is fair to say that the ambitions at that time were high compared to the maturity of the available tools and methods. However, important steps were taken. ISO 15926 was established as an international standard. Key ideas were developed including those of a reference data library (RDL), core classes and templates. Valuable experiences from projects were gained. The Information Modelling Framework sets out to deliver on the intentions of the original work, building on experience and a radically more powerful technology toolbox. PCA now proposes a new Part 14 to the ISO 15926 community to accommodate these new innovations, bring them into wider use, and further promote them within ISO.

W3C based technology

The Web Ontology Language OWL is a recommendation (i.e., a standard) from the World Wide Web Consortium (W3C). The current version of OWL, also referred to as "OWL 2", was published in 2009, with a second edition published in 2012. OWL 2 is an extension and revision of the original version published in 2004. One reason for the success of OWL is availability of advanced technology to implement the standard, ranging from internet protocols to powerful reasoning engines used for consistency checking. These features are key enabling technologies for IMF. The W3C languages and associated technology are developed within an international innovation cluster comprised of research institutions and software vendors.

Material Master Data (MMD)

MMD is a standards vocabulary (i.e., an ontology) for piping bulk material that was first deployed for engineering for procurement at Aibel (an EPCI company) in April 2015. MMD is a very large ontology, comprised of more than 100.000 classes. A novelty of the MMD project is its successful uptake of artificial intelligence; MMD deploys reasoning engines to check consistency and support advanced queries. The lessons learnt from MMD have been leveraged in the development of the IMF.

The Z-TI pre-project and READI Phase 1

In 2017 the NORSOK Sector Board Petroleum commissioned a pre-project targeting methods for digitalization of the NORSOK standards for technical information. The SCD method for requirements was first proposed in this project and also principles for building asset models that deploy ontologies. These results shaped the subsequent READI Joint Industry Project. READI Phase 1 gained experience in adapting the SCD method and significantly matured the tool suite that was prototyped in the Z-TI project. The learnings have been leveraged in the development of IMF.

Commented [ØE2]: Det opprinnelige READI verktøyet var utviklet gjennom et R&D prosjekt i DNV GL . Det var betegnet DREAM og ble gjort tilgjengelig for READI JIP'en av DNV GL for videre utvikling. Dette var detaljer og ikke viktig. Burde SIRIUS og Optiqe være nevnt i denne sammenheng? Syntes deter Ok å vise at Norge har gode fagmiljøer og ligger langt fremme på området.

Commented [AW3R2]: Jeg skriver gjerrne litt om Optique/SIRIUS. Ang DREAM, så var jo input til DREAM det vi laget i Z-TI . Jeg ungikk å referere til DREAM fordi det ikke brukes lenger. Men skal det inn i fotnote?

The DRIPP asset model for Johan Sverdrup

Aibel delivered an asset model for parts of System 18 (Flowlines) and System 20 (Separation and stabilization) to the Equinor Johan Sverdrup project in 2019. This was the first project to deliver an asset model integrated with a rich ontology, in part exploiting and further developing Z-TI pre-project results. IMF will incorporate successful decisions and know-how from the DRIPP project.

Aker Solutions/ Aize Information Model

Aker Solutions Aize (Aker Solutions is an EPCI company, Aize a software company started by Aker Solution, now owned by Aker ASA) is implementing and managing a cloud-based digital infrastructure for operating digital twins. The Information Model underlies the infrastructure. This is an ontology library, developed by Aize, comprised of more than 100 OWL 2 ontologies organized in a strict dependency hierarchy. Below domain-independent top-level ontologies is a level of ontologies describing generic concepts in the engineering domain and mappings between system codes from Aker Solutions and from its customers. The lower level of ontologies represents oil and gas assets and related technical documentation. Many of these ontologies are populated using ETL processes that extract, contextualize and harmonize data from production systems. Current core model holds approximately 4,15 million triples. The Information Model experience of Aker Solutions Aize is particularly important for the information architecture part of the IMF.

ISO/IEC 81346

ISO/IEC 81346 is published jointly by IEC and ISO and establishes general principles for the structuring of systems. Based on these principles, rules and guidance are given for the formulation of unambiguous reference designations for objects in any system. Reference Designation systems (RDS). Originating as a standard in 1971 it now contains basic structuring principles (Part 1, new edition expected in 2021), codes for component systems (Part 2, 2019) and codes for power plants (Parts 10, 2019-21) and the building industry and (Part 12, 2018). The standard IEC 61355 is a close cousin of ISO/IEC 81346 describing rules and guidelines for classification and identification of documents.

ISO/IEC 81346 complements ISO 15926 by its focus on top-down system breakdown, which is an essential feature of the Information Modelling Framework. IMF makes novel contributions by adapting semantic methods and technologies to the standard. In June 2020 READI launched the first version of codes for systems that are particular to the Oil & Gas industry (RDS-O&G) in the Standard.

1.3 Intended impact [leaders, all]

1.3.1 Overall impact

The overall intended impact of the IMF is to make feasible as well as accelerate the digitalisation of the industry's value chain, in order to release the significant efficiency gains available. The overall desired impact includes:

- The transition from document-based documentation to exchange of structured, model-based information
- Automated and unbroken digital thread of the industry, capturing the complete life cycle of requirements in the System Engineering V-model, and in Operation
- Improved information quality, by managed redundancy of information, traceability, and accountability
- Interoperability across the industry, including between different digital twins / asset information models by using shared and open resources, methods and formats

1.3.2 Incremental, industry-wide, low-risk transition

Resulting from the concept of the IMF and the way it is constructed, and the fact that it builds on a foundation of international standards and shared resources, the industry can move forward at its own pace, yet in a coordinated fashion, and consequently at little risk. The rate of implementation and transition can be as careful and incremental as is deemed necessary in each case.

1.3.3 Digitalised, automated requirements processing

By using a digitalized flow of requirements, experts can spend less time on repeated implementation and verification of requirements, and minds are freed up to innovate solutions and improve formulation of design rules. Investment into digitalizing an increased part of existing requirements pays back by means of reduced manhours expenditure and better utilisation of scare expert skills.

1.3.4 Improved quality, precision, and consistency

The application of best available and proven technology and science, allows accurate, unambiguous and complete formulation of requirements, function and design. This means that, the delivered assets, as well as asset information models will benefit from an increase in quality (meets requirements), precision (no excessive margins), and consistency (without undesired variance). Improved efficiency and reduced consequential damage cost are the immediate impacts of this.

1.3.5 Interoperability of digital twins / asset information models

Through alignment on a common 'language' and by authoring information in a semantic data format referencing a common shared library resource, the requirements and the asset information models created will be independent from the tools and systems in which they reside. This brings about seamless transfer and integration of complete or fragment asset information models or requirement sets, enabling interoperability between different actors, as well as lossless transfer of such information from one project phase to the next, or from one application to another.

1.4 Implementation method and strategy [all]

The strategy for achieving a robust and sustainable industry-wide implementation and application of the Information Modelling Framework rests on four fundamental principles:

1. Leverage shared resources
2. Enable cross-discipline orchestrated workflow that builds on existing processes
3. Establish and maintain strong connection to recognised standards
4. Active involvement of international stakeholders for shared ownership and cooperation.

1.4.1 Shared resources and services

Access to shared resources, such as common vocabularies, type definitions, and ontologies, is made possible by the use of standard Internet communication protocols. Using the same mechanisms as for the WWW, the HTTP protocol and IRIs (International Resource Identifiers) are used access IMF shared resources. To sustain such shared resources an organisation must be available to provide or to manage the maintenance and continued development of content. A typical example is the POSC Caesar Association (PCA).

1.4.2 Workflow for building, maintaining, extending and deploying models

The IMF is based on three main pillars:

1. Asset Model Specification
2. Integrated Asset Model
3. Asset Application Model.

This structure reflects the need for accommodating different domains and work processes. Engineers and O& G domain experts need a format in which the asset can be specified (Asset Model Specification), experts in asset modelling and semantic technologies need a format in which the model can be implemented (Integrated Asset Model) based on input from the Asset Model Specification, and lastly, experts in IT/Digital need a format in which to implement the applications (Asset Application Model) that leverages connecting to the Integrated Asset Model for providing functionalities such as operational digital twin.

1.4.3 Built on Standards

Essential to the implementation strategy is that the IMF shall be founded on international standards as much as possible. It is therefore a strategic investment to further develop and propose extensions to or revisions of existing standards when required. The two main standards in this respect are ISO/IEC 81346, which is foundational to the IMF Asset Model Specification framework, and ISO 15926, which is foundational to the IMF Integrated Asset Model framework. The IEC61355 which is about classification and designation of documents for plants, systems and equipment is also a valuable resource as a means to standardize purpose-driven information.

1.4.4 Alignment across industry and internationally

Although the effort of the READI JIP at the outset had a Norwegian perspective and scope, it was soon realized that this would limit the value of the result for the industry. Therefore, it is a strategic decision to strive towards full international adaption of the IMF, and to actively seek synergies and harmonization with other related international initiatives. The prime partner for such cooperation is the CHIFOS initiative sponsored by IOGP. Cooperation to explore synergies and possible alignment has begun and is likely to continue in parallel with the IMF‘s continued development and implementation.

2 Designing, specifying and modelling an asset [All]

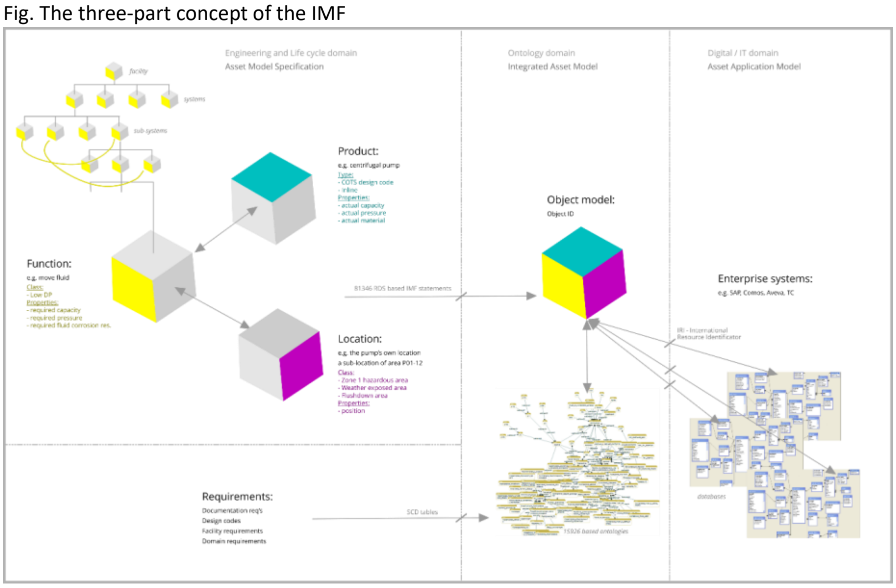
2.1 The three-part concept of the IMF

The IMF divides the concept of an Asset Information Model into three frameworks, each reflecting and accommodating the work process and domain expertise that applies:

- **Asset Model Specification**, for specifying the asset in such a format that it can be implemented as an information model, leveraging shared resources such as Type- and Model Block specifications
- **Integrated Asset Model**, for implementing the specified asset as an information model, leveraging shared resources such as ontologies, vocabularies, and Model Blocks
- **Asset Application Model**, for building functionality that interacts with the information model, leveraging the information integration and contextualisation made available.

For those familiar with software, there is a loose analogy between these three frameworks. Source code is analogous to the Asset Model Specification, Platform and Compiled code is analogous to the Integrated Asset Model, and Applications are analogous to the Asset Application Model. Throughout this document each of these three frameworks will be referred to and described individually, and when the functionality of the totality is the subject, it is referred to simply as an asset information model, which is an implementation in accordance with the Information Modelling Framework (IMF) described herein.

Fig. The three-part concept of the IMF



2.2 The life cycle of an asset information model

The life cycle of an asset information model logically supports four distinct project stages:

1. *Requirements' specification,*
2. *Design creation,*
3. *Asset fabrication*,

4. *Information utilization*.

2.2.1 Requirements' specification

Requirements' specification involves establishing requirements and the high-level design basis. Usually this takes place mostly during a project's Concept development and continues into the Front-End Engineering. To support this work, the IMF provides a structure for holding requirements such that these can be mapped to, and verified with the model of the design. The IMF also enables specifying and modelling the design in a top-down fashion, starting with a high-level design, typically a Design Basis.

2.2.2 Design creation

Design creation involves developing the input requirements and design data – as well as relevant background requirements and design (industry standard, best practice, internal) - into an increasingly detailed asset information specification and model, translating requirements into solutions and detailed design. Usually this begins in the Front-End Engineering, but mainly takes place during Detail Engineering. The IMF enables specifying and modelling the design at an increasing level of detail, verifying that requirements have been met.

2.2.3 Asset fabrication

Asset fabrication – which in this text includes manufacturing, construction and assembly – implies further developing the asset information specification and model such that it supports the process of building the physical asset. Usually this takes place after Detail Engineering have had some progress. The IMF enables an asset information model that can be extended and adapted to provide such support, but the scope of this document does not include this functionality. For illustration, examples of such extended functionality are: shop floor design (e.g., weld locations), construction work flow, assembly sequences, etc.

2.2.4 Information utilization

Information utilization implies using the asset information model to support operation and maintenance of the physical asset, and also support later changes and modifications to the physical asset. This takes place during the whole life cycle of the physical asset. The IMF serves to integrate information from a range of connected source systems, and enables information harvesting, contextualisation and a range of other functionalities typically described as Digital Twin functionalities. Specific operational needs can be supported by utilizing specific aspects of the model, e.g., by having a dedicated Maintenance aspect, the products can be mapped to the organisation of the maintenance processes, such that each product 'knows' its maintenance. The IMF also provides a foundation for information management of documentation not contained by the model itself, but held in dedicated documents, e.g., manufacturing records, operation manuals, etc.

2.3 Building a specification of the model

The first step when designing a facility or an asset following the systems engineering approach is to define the function at the top level: the main system(s). Then the function of the main system is broken down into functions of the sub-systems that again may be broken down further. The IMF Asset Model Specification framework mirrors this approach by building on the structuring and coding capabilities of ISO/IEC 81346, which allows the modelling of main function, sub-functions, further sub-functions, and so on, whilst not mixing up different information aspects such as function and location. Improving on conventional systems engineering, the ISO/IEC 81346 brings the concept of clearly segregated aspects - that are different views of an object, covering different kind of information perspectives.

Logically, an object is assigned values beginning with the 'Function' aspect. Depending on the type of object, the modelling may proceed to include the 'Location' aspect, then the 'Product' aspect.

One way to visualize this is to think of something as being represented by an information cube with several sides, each side representing one aspect. As an example, a 'pumping' system may have a Function aspect (about what it does), a Product aspect (about what it is), and a Location aspect (about which location it occupies).

To design and specify the asset model, the properties of the relevant aspects must be given values. E.g., the pumping **Function** specification could include (required) 'Flow capacity' having a value of 300 m3/h, and the pump **Product** specification (actual) 'Flow capacity' having a value of 500 m3/h, whereas the pump **Location** specification 'Environmental condition' has a value 'Naturally ventilated'.
At a higher system level, say a 'Fluid circulation sub-system', the specification cube would only need one aspect; **Function**, since no **Product** or **Location** information associated with it can be specified.

Instead of spelling out the full requirements by means of properties, it is often better to refer to one or several 'Types' that contain requirements that are common across many individual objects. This is done by assigning a value to the **Type** parameter. As an example, assigning the particular location to be `Type= 'Weather exposed'` ensures it 'inherits' all the requirements of a 'Weather exposed', such as coating, materials, sealing, shielding, etc. In the product aspect similarly, **Type** is used to refer a standard set of specifications, e.g., a motor may have the Type= 'IC411', and this then implies a set of specifications (IC411) that describes all such motors of this type.

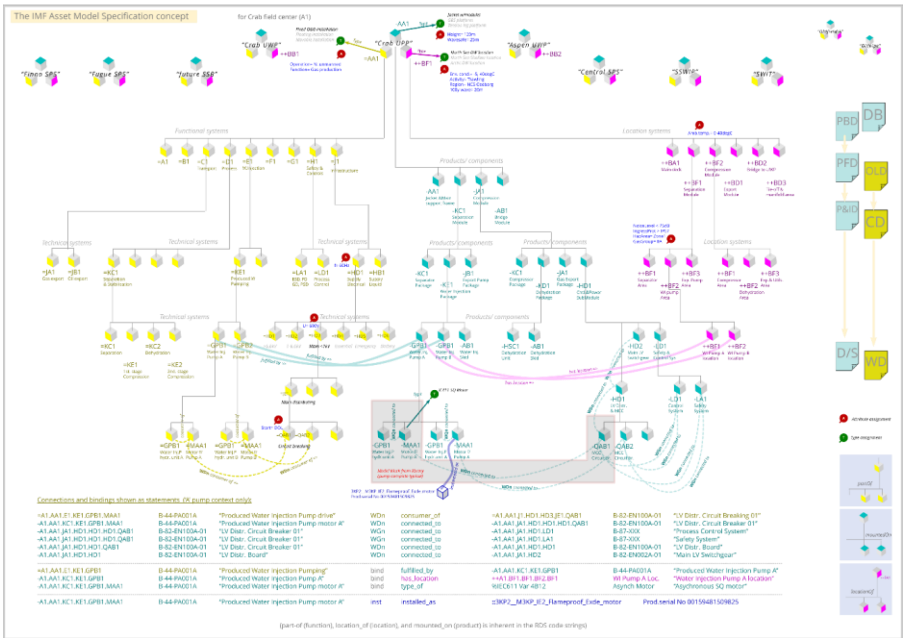
Thus, a full system-of-system hierarchy can be structured and coded, and serve as the specification of, and input to, the Integrated Asset Model. Note that the information of requirements to-, location of-, and specification of- the actual products/components will thus be preserved throughout the life cycle, and be accessible from a single source of truth.

Beyond the strictly hierarchical model, there is a need to define, classify and assign codes to relations or interfaces between systems and between components. Connections that have properties are specified as **Streams** between objects. An example of such **Streams** is where a pump is driven by a shaft connected to an electrical motor which is powered by an electrical supply connected to a switchboard.

When such connections are implemented in the Asset Model Specification, not only the structure of the system is specified, but also its topology. The concept of **'Streams'** is employed in the IMF for specifying such connections *within the same aspect*.

There is also a need to specify the integration *across* aspects, e.g., how a specific *function* is fulfilled by a specific *product* and is has location a specific *location.* This is done by the use of specifying a **Bind** statement. The format of such statements is '=GBP1' *fulfilled_by* '-GBP3' to bind function to a product, and '–GBP3' *has_location* '++BF2' to bind product to a location.

Fig. Illustration of the totality, but with focus on the example pump case:



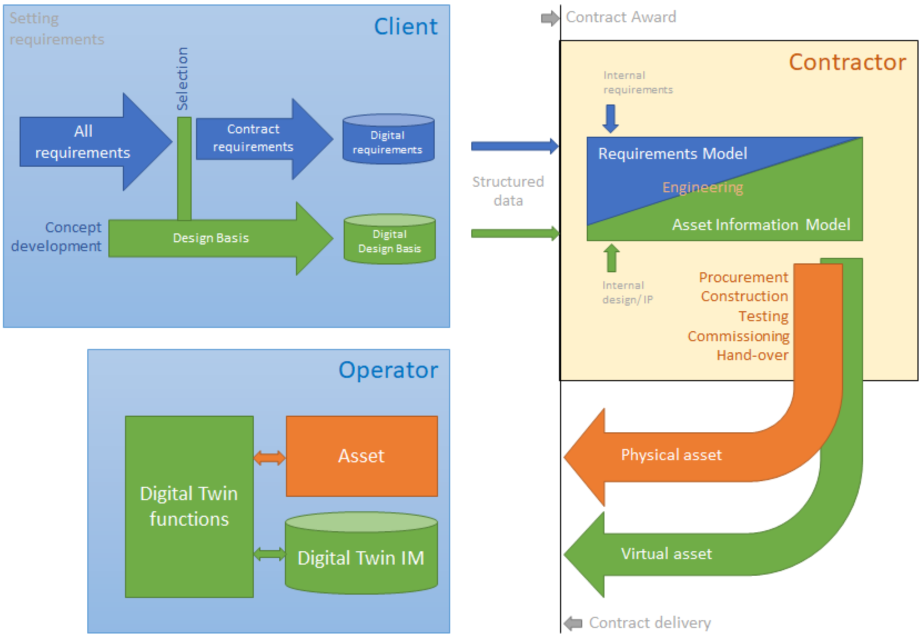
When specifying an asset with the IMF, extensive re-use is made possible. Typically, when a subsystem needs to be specified, it can be *inserted* into the asset model specification in the form of a **Model Block** fetched from a pre-existing library of pre-tested and verified model blocks. These are in the form of model fragment specifications. Due to the way codification is organized, model blocks can be pre-coded once and for all, and dropped 'plug & play' fashion into the project asset model specification, with no need for re-coding of identifiers.

3 Requirements Digital Thread [All]

3.1 Unbroken flow of requirements

An unbroken digital flow of requirements - from setting of requirements, to implementing and verifying requirements, is illustrated below.

Fig. Requirements Digital Thread



A prerequisite to quality is a consistent set of requirements that are applicable to the desired design: no more and no less. Traditionally, requirements, contained in many documents and standards, have been issued by the Client/Owner/Operator to the Contractor. These have contained many inherent inconsistencies and contradictions, as well as superfluous information, thereby incurring inefficiency costs. Such inefficiency costs are also significant during operation, due to the amount of manual Information Management work required.

A significant step towards a Requirements Digital Thread is to digitalize the requirements and the requirements setting process. By digitalizing the requirements in accordance with IMF, self-consistency is enforced and supported, and by providing a mechanism for selecting only applicable requirements, contractual accuracy and execution efficiency is improved, as is the information management efficiency during operation. The selection mechanism can be implemented by means of automated criteria-checking with design information, or it can be by manual input of selection criteria – or a combination of both.

As indicated by the figure above, issuing requirements is only the first step, and it begins at the Client/Owner/Operator. Once the requirements are issued and received, an engineering process at the Contractor produces an Asset Information Model that is verified towards the requirements at a higher and higher level of detail, as the project reaches completion.

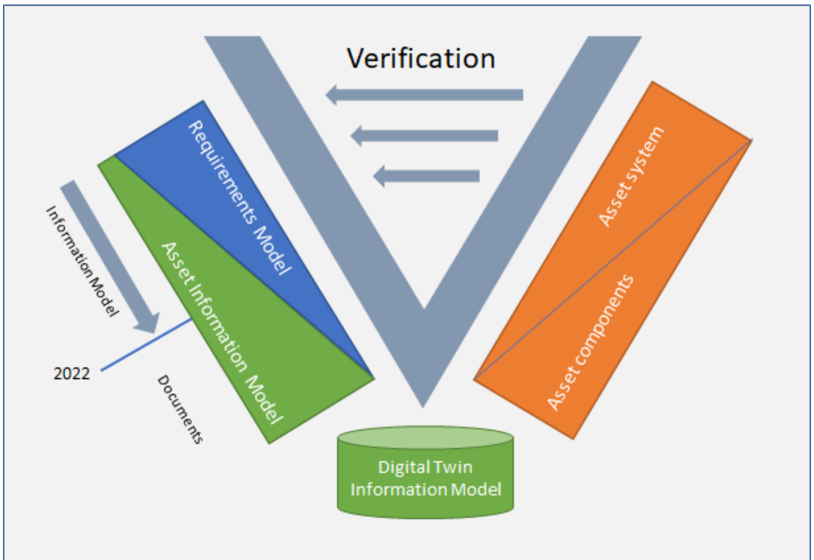
The IMF provides the framework for the Asset Information Model, which contains the design, and it provides a format for specifying, transferring, and computing requirements in a wholly digital fashion. Together this enables an unbroken flow of requirements from the originator (Client) to the implementer (Contractor), such that requirements are 'translated' into design, and such that implementation and verification can be largely automated.

Ideally the resulting Asset Information Model holds all data relevant to the asset, such that it can be engineered, procured, constructed, tested, operated and maintained. This includes supplying the information model required to build Digital Twin functionality. However, in a more pragmatic perspective some information related to the asset will for a long while continue to reside in conventional documents, e.g., manufacturing records, where the value of 'data-centric' is lower.

3.2 System Engineering approach

At the Contractor, the creation of design, and the implementation and verification of requirements, usually follows the V-approach of Systems Engineering.

Fig. V-method of system engineering.



System Engineering view on development of Asset Information Model and Digital Twin

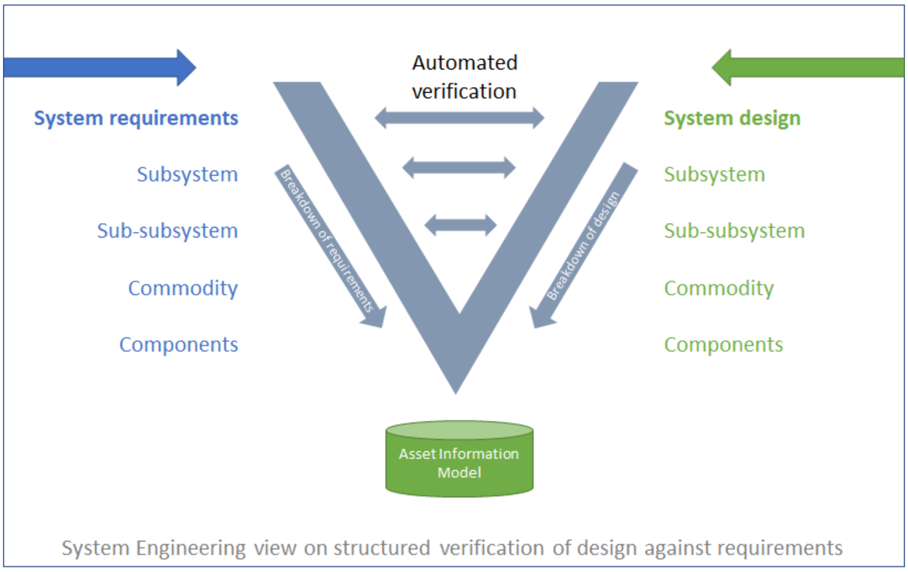
This model describes a process in which requirements are translated into design in a top-down fashion, resulting in a detail design that can be implemented by means of manufacturing or procurement, then in a bottom-up fashion the components, products and sub-systems are integrated until at the top level the system is complete. At each level of integration, a verification is performed of implementation being in accordance with requirements.

Where the model illustrated differs from a conventional System Engineering V-figure, is that for modern assets or facilities the process must also produce a Digital Twin information model. By doing this in accordance with the IMF, ideally the Digital Twin information model will be identical to the completed Asset Information Model, but as the figure illustrates there is likely to be a progressive transformation in which design implementation goes from being document based to become increasingly information model based. The year 2022 in the figure is arbitrary, and only serves to illustrate the point. In the intermediate a Digital Twin will rely partially on a structured data information model, partially on information contained in conventional documents and databases.

3.3 Automated verification

Automated verification is enabled when both requirements and design are in the form of structured information, instead of in documents.

Fig. Structured verification in a V-model.



The verification takes place in a top-down fashion, beginning with verifying that the high-level design conforms with the high-level requirements. Contrary to the classic V-model where the process moves down the V and then up, this process in parallel moves down, verifying at an increasing level of granularity until complete, or until remaining details are manually verified based on conventional document-document verification.

4 Minimum Viable Product

To enable early demonstration of the IMF, the approach of developing Minimum Viable Products (MVPs) is employed, with fast cycles of develop-test-adjust. The target of the MVPs is to demonstrate the concept, the feasibility, the scalability, and the value created.

4.1 Use Case Asset Model

This is a proof of concept of the Model Block feature of the IMF, and at the same time a demonstration of a small-scale Asset Information Model. The Use Case is the pump package, and the electrical motor part of this package is modelled in such detail that it is ready for procurement. The model is first developed as an Asset Model Specification (the 81346 world), and then implemented as an Integrated Asset Model (the 15926 world).

4.2 Z-018 TIRC tool/service

A tool to provide selection and generation of sub-sets of requirements on a digitalized form is developed by the READI JIP, and will be offered to the market as a service. Initially the scope of this tool/service is the NORSOK Z-018 Standard, governing documentation requirements. The functionality of the tool/service will rest on the framework specified in this document (IMF), including the libraries of ISO 15926-14. The service offers semi-automated selection and submittal of NORSOK Z-018 requirements in a digital format such that verification of requirements versus design can be automated.

4.3 Digital Design Basis

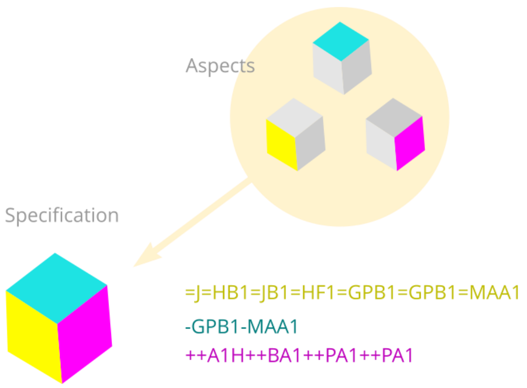
This is a digital format and template for a standardized way of providing Design Basis information to the next step in the project execution life cycle, typically from the Client to the Contractor, such that this information can be transferred to the next target system without loss of information or data quality. The Digital Design Basis is outside of the scope of READI JIP, but builds on the same foundations, and this description is therefore included here for completeness.

5 IMF Asset Model Specification [Engineers]

5.1 Building the IMF Asset Model Specification

It is recommended that the reader is somewhat familiar with the ISO/IEC 81346 as a background for this chapter, and in particular the O&G part and guideline. In-depth knowledge is, however, not required. Note that the standard intends to give room for interpretation and variant applications, and therefore needs domain specific guidelines, part of which is in this IMF document.

Fig. Aspects envisioned as different sides of the same thing



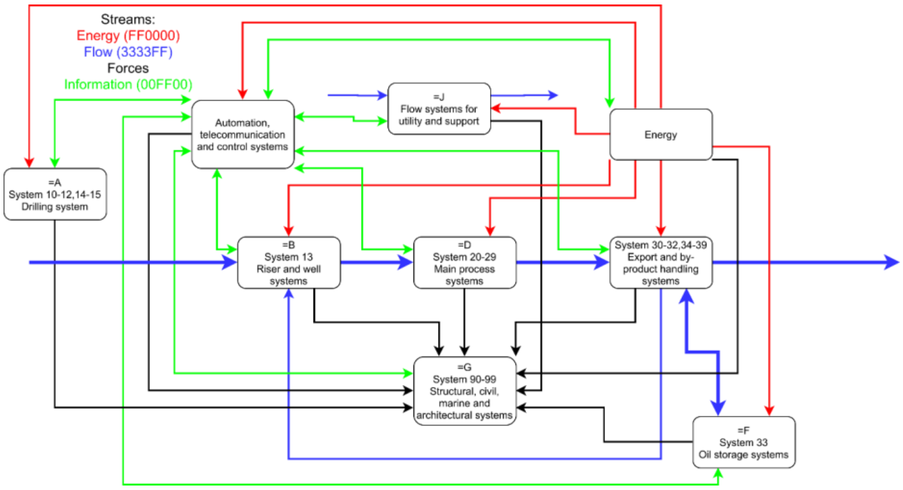
The use of aspects is fundamental to both the ISO/IEC 81346 standard, and to the IMF. There are strong benefits throughout the life cycle of keeping information about objects clearly segregated by means of aspects. For example, not to mixing function and location. Function and location are part of very different break-down hierarchies, and by mixing the two aspects valuable contextual information is lost.

5.1.1 Functional aspect

(Yellow cube side, RDS designation prefix: '='.)
The purpose of developing an IMF Asset Model Specification is to capture first a functional description of the facility or asset – the way it is required to work – in such a format that is familiar and accessible to the engineers, so that knowledge of semantic technologies or IT shall not be required.

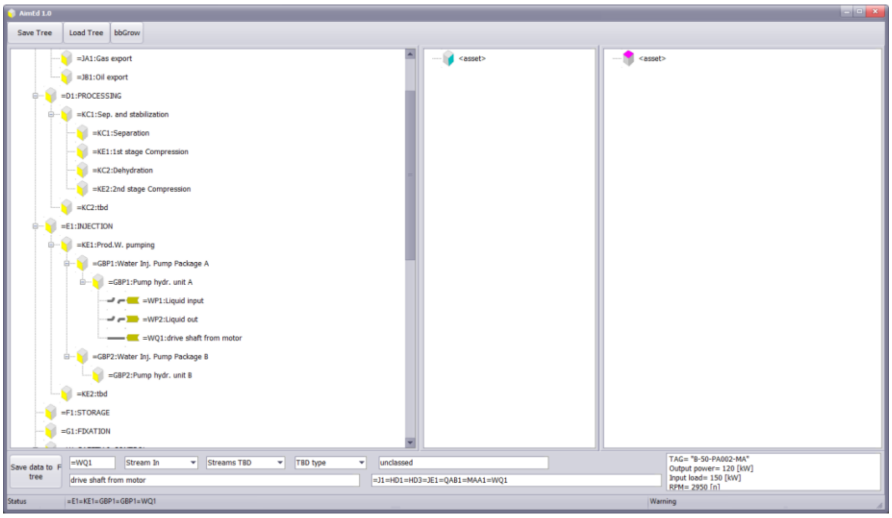
The first step, creating the functional description (functional aspect) is enabled by the clear split of ‘aspects’ given by ISO/IEC 81346, and the way the IMF defines the framework for an Asset Model Specification. This aspect is shown in ISO/IEC81346 as well as in IMF illustrations as yellow-sided cubes. The yellow cube-diagrams are intended for *showing the functional structure*, with both the *part_of* hierarchy and the streams/connections, as well as key properties and type classifications - all being essential to specifying the functional requirements.

Fig. Field Block Diagram.



Typically Block Diagrams are more suitable for mapping out the streams between main functions (see above), whereas file directory (tree-view) style diagrams are good for working with the hierarchy (see below). An editor could be designed to offer both these formats, since the underlying information model specification is the same.

Fig. Demonstrator of an editor for building Asset Model Specification.



Building up the functional part of the Asset Model Specification should begin at the top, with reference to the scope of the model. If the scope is an offshore platform, the modelling starts by defining the platform as a **Functional System Block** with **Streams** in and out, then break that down into functional systems, e.g., Injection System (as seen in the middle of the editor screen picture above), and then begin breaking down into Technical Systems, sub-systems, and so on, until reaching desired granularity with respect to defining functional requirements.

The desired level of granularity would reflect the project execution phase where this takes place. Thus, during early phase concept study the specification would only go 2-3 levels down, for example to the level of 'Pumping Package'. In a later engineering phase, a further 2-3 levels are needed in order to set requirements to for example pump-pipe connections (flanges), and so on - with no real limit of how detailed the specification can be made.

A complete functional Asset Model Specification is a complete set of functional requirements for the facility or asset - at the desired level of granularity or detail.

Throughout the specification, functions and streams are characterized by means of assigning to them properties: **Type** is a property which declares that the function is of some type of predefined function (industry standard), and attributes are values that are appropriate for such a *type of function.* They are specific for the actual asset or facility. One such attribute can be a Tag Number, an identifier primarily intended for human use. It is not the true model identifier.

5.1.2 Product/component aspect

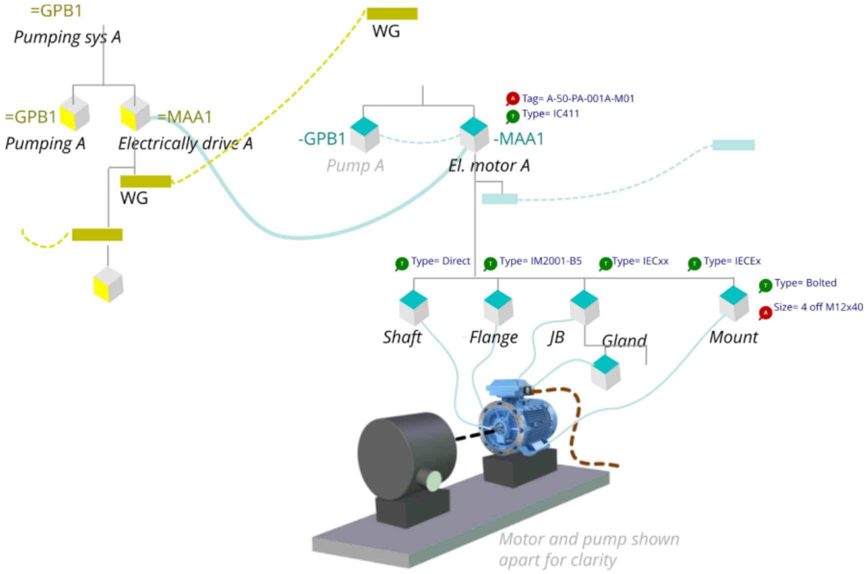
(Cyan cube side, RDS designation prefix: '-'.)

Note that the name Product/component aspect may be a bit misleading. Think of this as the *physical specification* aspect, not the actual tangible product/components aspect (which is the Installed aspect). The Product/component aspect is usually shown using the cyan colour (light blue).

After specifying the **Functional** aspect, i.e., the functional requirements, the logical flow is to specify the **Product**/component aspect, i.e., the physical specifications - often in interaction with specifying

the **Location** aspect, i.e., the locational arrangement with associated requirements. In practice, however, it is convenient to begin specifying these two next aspects even before the Functional aspect is finished.
Below is shown a close-up of an electrical motor model, illustrating a breakdown into detailed Product/component specification, with a few properties included.

Fig. Visualisation of partial Asset Model Specification for a pump motor.



Whereas the **Functional** aspect is understood as a 'function-part-of-function' hierarchy, the **Product**/component aspect is understood as a 'thing-mounted_on-thing' hierachy, but essentially all the hierarchies are *part_of* hierarchies. The 'mounted_on' should be read to also include mounted in, mounted around, mounted under, contained in, etc. It is a physical assembly hierarchy.

Whereas the motivation for building the specification in the **Function** aspect is to establish a functional requirements specification, the motivation for building the specification in the **Product**/component aspect is to establish the complete specification of how those functional requirements are to be fulfilled. In addition, requirements given by how things are arranged into locations, as established by the specification in the **Location** aspect, must be fulfilled by this specification. A complete **Product**/component specification should provide sufficient technical information required to procure the goods that are intended to meet the specification.

As for the **Function** aspect, throughout the specification the **Product**/component functions and streams are characterized by means of assigning to them properties. **Type** is a property which declares that the function is of some type of predefined specification (industry standard), and attributes are values that are appropriate for such a *type of specification* and are specific for the actual asset or facility. One such attribute can be a Tag Number to go on the nameplate.

5.1.3 Location aspect

(Magenta cube side, RDS designation prefix: '++' or '+'.)
The **Location** aspect is usually shown in the magenta colour (light purple). Specification of the **Location** aspect is done somewhat in parallel with specifying the Function- and then the **Product**/component aspects. It serves to specify the locational arrangement of things.

Typically, for an offshore platform this entails breaking the platform into modules, the modules into levels and rooms, and then sometimes into areas. For this the absolute location prefix '++' is used, whereas when relative location is needed, like grid location inside an electrical cabinet, the relative location prefix '+' is used.

The **Location** aspect hierarchy is understood to be a 'sublocation *has_location* location' breakdown but essentially is also a *part_of* hierarchy. The motivation for building the **Location** aspect specification is to specify the locational arrangement of things, together with the parameters and requirements that characterise these locations. One example is that when a location is classified as 'flush down area' it carries forward a requirement for all exposed things located there to have a specific degree of ingress protection, such as IP67.

As for the **Function** aspect, throughout the specification the Locations are characterized by means of assigning to them properties. **Type** is a property which declares that the location is of some predefined type of location (e.g., office area, High Voltage area, HC process area), and attributes are values that are appropriate for such a *type of location,* and are specific for the actual asset or facility. One such attribute can be a location name to go on the door sign-plates.

5.1.4 The Type "aspect"

(No specific cube side, RDS designation prefix: '%'.)
Is used in some industries, but is *<u>not</u>* applicable in IMF.

5.1.5 Installed aspect

(Dark blue cube side, RDS designation prefix: '::')
This aspect is specifically defined by IMF.
This aspect is different from the other aspects. Instead of serving as a specification, it serves as a template for later required information. The three primary aspects (**Function**, **Product**, **Location**) shall, together and completely, describe what is needed. However, when something later is delivered to fulfil that need, it has become real in the form of an actual thing. The **Installed** aspect is used to refer to this *real thing aspect*.

Information in this aspect only becomes available when the asset is real (operational), and therefore it is not a 'to be' specification. This aspect is illustrated conceptually in a dark blue colour. The information contained in this aspect is dictated by the data about the actual thing that is presently delivering to the need. E.g., a motor that has been installed may therefore have information about its actual shaft power, production serial number, and run hours.

The motivation for building the specification in this aspect is to establish placeholders for information that shall become available when the asset is built/operated. It serves to establish the information scope and context, including the relation between the actual thing and its required function, specification and location.

5.1.6 Additional aspects

Additional aspects may be defined as needed, at any stage. Such aspects can offer a specialized organisation or view of the asset information in order to support any particular organisation of the work. The IMF does not define any such aspect, but it could be beneficial to later define such extensions, if cross-industry standardisation is desired. Examples of such aspects are:

- **Procurement**, introducing the dimension of suppliers & contracts, and the relation between contracts and goods,
- **Maintenance**, introducing the dimension of maintenance activity breakdown, and the relation between tasks and equipment,
- **Documentation**, the relation between the document information structure and individual objects.

5.2 Model specification implemented in the engineering work process

Usually when designing a facility or an asset following the systems engineering approach, first the function at the top level is defined; the main system(s). Then the function of main system is broken down into functions of the sub-systems that again may be broken down further. This is the Systems Engineering method, and its prime objective is to achieve a *quality assured engineering of a solution*.

However, the objective of the method given by the IMF is to also *establish an asset information model* that ultimately shall provide value throughout the entire life cycle of the asset.

This means that the engineering process that incorporates the IMF also needs to capture additional information that is required to build a model over and above what is provided by the Systems Engineering method. This information is about how objects are related and how objects are connected, the topology of the system.

To give one example: in Systems Engineering language a pump is part of the Seawater System, a heat exchanger is part of the Lube Oil system. More information is needed to establish, e.g., the topology that describes how the pump and the heat exchanger are connected by means of a thermal energy stream, or about how the electrical motor and the pump are connected by means of a mechanical energy stream. In this perspective, the functions and the streams are objects with relations, and with properties specifying their characteristics.

Both the systems breakdown *and* the topology are needed to specify an asset information model. Legacy engineering work processes do this to some extent, particularly within the Process- and Automation disciplines, but a lot of the information is, in practice, abandoned as the project development progresses, or it gets 'lost' in volumes of documents.

The IMF provides a method and framework for continuously enriching the asset model, preserving topology information, system structure, and context. Instead of thinking of documents such as Block Diagrams, One Line Diagrams, and Process Schematics as 'recipes' for how to build the solution, a different mind-set is encouraged: These documents are a visual interface to the Asset Model Specification.

To serve this purpose the objects represented by a document must be assigned identifiers, properties, and relations as per the IMF. As an example, this means that when a pump is shown (an object) it may still be visualised as a pump symbol having a conventional tag number (A-50-PA-001A), but this is now only a 'Visual Tag' property, and the true identifier exists 'behind the scenes', as per the IMF.

One way to envision how to interact with the IMF type of information for the pump, including the true identifier, is to - in the design authoring tool - to 'inspect properties', 'right click' or similar, assuming such functionality is (made) available in the tool. Alternatively, or complementary to, a dedicated IMF Asset Model Specification authoring tool could be employed.

5.3 Support for conventional 'Tag numbers'

Tag numbers have been, and still are, used as identifiers for equipment in our industry. We know them as variants of the format '50-PA-001A' (seawater pump). There have been many attempts at developing the ultimate tag numbering manual, but with limited success.

Early in the history a tag number would map 1:1 with actual pieces of equipment, like pumps, valves and gauges, but then when automation and digitalisation took hold, there was an increasing need to tag also alarms, measured values, derived values, and calculated status. Different companies in different regions tended to do this differently. The issue of Tag numbers started to become problematic.

Project-specific tagging drives significant cost, and the tag format, intended for humans, does not serve the needs of information model-based systems such as Digital Twin.

The IMF provides a simple means of identification of the supplier's product *category* using designators from ISO/IEC81346, e.g., a pump is category '= GPB'.
For the product as part of the entire asset (pump as part of Injection Water system) the format '=E1=KE1=GBP1=GBP1' is prescribed, and it identifies/places the pump in the functional hierarchy. Similarly, such strings provide the means to identify/place the pump in any aspect, e.g., **Function**, **Product**, **Location**, etc.).

Obviously, these identifiers are rarely fit-for-purpose for humans to read and understand, with the possible exception for some infrastructure disciplines. Therefore, the conventional Tag numbers are fully supported by the IMF, not as true identifiers, but as a nameplate string contained in an attribute of the object.

5.4 Streams and Functional System Blocks

Industry assets consist of numerous systems. How systems are interfaced and how the various aspects are bound together is not very visible in a *part_of* structure. An electrical distribution system is easy to follow down the distribution system, but the connections to the consumer of the energy is not that easy to find. The concept of **Streams**, together with **Functional System Blocks** (FSB), provides a systematic way of specifying these connections between systems.

5.4.1 Principal statements

A few principal statements govern the arrangement of Streams and Functional System Blocks:

- Any **Functional System Block** that is part of an industry asset or product is there with the purpose of handling one or several of the defined Stream types. (see next paragraph)
- **Functional System Blocks** that are handling the same type of Streams shall be part_of the same Functional hierarchy.

- A **Functional System Block** is categorised based on the set of main Streams defined as input and output of the block.
- ***Main* Streams** are defined as those needed to fulfil the intended function of the block.
- ***Secondary* Streams** are created as consequence of how main **Streams** are processed. They become more relevant as the level of detail in the model increases, i.e., at lower levels of the hierarchy.
- There will always be a balance in **Streams** between inputs and outputs of **Functional System Blocks**.

5.4.2 Stream types

The following main **Stream** types are defined:

- Forces - such as transferred by a bracket, bolts, flanges, etc.
- Information - such as exchanged with a control system, sensor, etc.
- Energy (power) - see subdivision below
- Material flow - see subdivision below

Of these Stream types the following two are subdivided:

- Energy (power)
    - Electric - such as power from supply and to a motor
    - Thermal - such as cooling or heating
    - Solar - such as PV panels
    - Mechanical - such as a motor drive shaft
    - Sound (noise) - such as noise emitted from a machinery
    - Wind - such as that acting on wind turbine blades
    - Hydropower - such as that acting on a water turbine
- Material flow
    - Fluid (liquid) - such as water, oil, chemical fluids, low gas ratio multiphase fluids
    - Gas - such as methane, air, nitrogen, high gas ratio multiphase fluids
    - Dry granulated - such as sand, powder
    - Solid pieces - such as bricks, boxes, pieces

5.4.3 Functional System Block types

The following **Functional System Block** types are defined:

- **Functional System Blocks *providing a Stream***
    - Convert one type of **Stream** to another type of **Stream** (Example of products used to fulfil the function are Motors, Turbines)
    - Dividing one **Stream** into several of the same kind of **Stream** (Example of products used to fulfil the function are Separators, Centrifuges)
- **Functional System Block *transporting a Stream***
    - Transporting or moving one **Stream** from one point to another (Example of products used to fulfil the function are pipes, cables)
    - Storage of a **Stream** (Example of products used to fulfil the function are tanks, caverns)
- **Functional System Block *Receiving a Stream***
    - Collecting several **Streams** of the same kind into fewer **Streams** of the same kind (Example of products used to fulfil the function are manifolds, cable terminals, radio receiver)

- **Functional System Block *Controlling a Stream***
    - Controlling a **Stream** of the same kind by means of controlling the **Stream** rate (Example of product is a switch, regulator, control valve, isolation valve)
- **Functional System Block *Blocking a Stream***
    - Blocking a **Stream** of the same kind by means of obstructing the **Stream**
    - (Example of product is a blind flange, an enclosure, insulation)

5.5 Model Blocks, and reuse

Model Block is a concept where a fragment of an asset information model is encapsulated such that it can be reused in other asset information models. In the **Model Specification** this can simply be done by inserting the desired **Model Block** specification – as a specification fragment - into the appropriate location in the overall **Model Specification**. One example is when a pump package of some **Type** is needed, the **Model Block** specification (i.e. the RDS coding) for such a pump package can be inserted in the specified hierarchy, and then becomes part of the overall model specification. This mechanism is likewise part of the Integrated Asset Model.

When used to some extent the **Model Block** concept can greatly simplify the work of establishing the overall asset model, as complexities and details at the low level need not be developed.

Such Model Blocks have the power to hold de facto standards - design typicals, such that they can be shared across the industry, by being made available as an online resource.

This concept also supports developing **Model Blocks** that encode designs that are Intellectual Property of a company, in which case their access would be limited. The mechanisms would be the same. It can also be envisioned that such **Model Blocks** could be offered to the industry on a commercial basis.

5.6 Commodities and Design Codes

At the lower levels of the **Asset Model Specificatio**n there is a need for specifying higher detail. Very often these details are already a given - when Commercial Off the Shelf (COTS) systems or products are chosen, or a standard design code is chosen. In these cases, it is beneficial to refer to them in a way similar to picking from a catalogue, be it a 'product catalogue' or a 'design code catalogue'. Thus, by selecting a 'catalogue item', all the details are available by means of reference. The mechanism for this is to use the **Type** property to hold such a reference.

This mechanism enables the reuse of *product and component* information, drawing on libraries of such 'catalogue' information, whereas the concept of **Model Block** enables the reuse of *system design* information, drawing on libraries of design typicals.

5.7 Reference Data

Whereas the **Asset Model Specification** is the *master* for the actual project- or asset, it draws on a shared resource of definitions such as types and classes, property vocabularies, and units of measure. For example, when referring to a **type** in the Asset **Model Specification**, the **type** must be from a list of allowed types, and the **type** itself must be defined. For this the shared resource is the *master*, and the reference data is available in a look-up fashion when building the model specification (mechanism tbd).

5.8 Proprietary libraries

The commercial benefits of deploying the IMF across industry, enabling shared resources and re-use are strong, but in some cases competitive advantage takes priority, with the need to protect Intellectual Property (IP). The IMF fully accommodates this need. Reference data, design codes, and model blocks can be proprietary, in which case they will be hosted by the owner, instead of hosted by an industry service (such as PCA).

6 IMF Requirements modelling [Engineers]

Digital requirements are in place when requirements are expressed unambiguously and represented digitally as structured data that machines can process.

Time consuming revision cycles are at present a significant impediment to common industry requirements. This situation can be remedied with requirements in structured form, since this allows a fine-grained revision and efficient distribution. Compare this to today's practice where a whole document is as a rule revised at the same time. Once a document is revised, there is no support for updating specialist applications in the companies that need the requirements; such update must be done manually in each company by inspecting documents with the revised specifications. Minimal tool support is available to detect errors and ensure consistency.

Requirements in structured form open for transformative changes in work processes by automation of work that is now performed manually by domain experts. However, this automation can only be achieved if the requirements are so precisely specified that they can be reasoned upon by computers. This is enabled by the **Scope Condition Demand (SCD)** format and method.

6.1 Scope Condition Demand – SCD format for expressing requirements

Requirement specifications are today, more or less exclusively, written as natural language text and distributed as text. Modern requirements management systems support methods to structure the documents, typically by giving each statement a unique identifier, but they do not change the fact that requirements are text. This section is built around some text from ISO 19001-5:

– `Equipment with a transport dry weight above 1000 kg shall be weighed by the manufacturer and weight certificate shall be issued.`

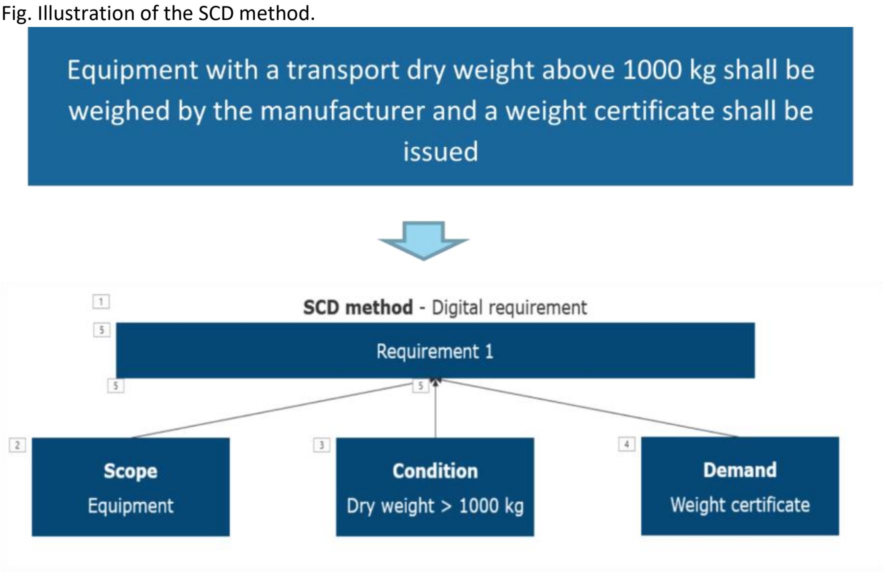
The rationale behind the requirement is that prior to lifting, the point of gravity of that which shall be lifted must be calculated. The threshold of 1000 kg is not sacrosanct and may potentially be changed, but the need for a weight certificate for heavy equipment is not likely to vanish.
We can say ISO 19001-5 is the source of the requirement and that the work process of lifting and transporting is the claimant. Here are some further observations: -

– The requirement statement contains the word “shall”. The SCD method is about such obligatory requirements. Often requirement specifications contain advices, references to best practice and other non-obligatory prescriptions. These are for the time being ignored. The central point about “shall requirements” is that they need to be verified. -
– The requirement statement is about equipment. We shall say that this is the scope of the requirement. If follows that the requirement is also about valves since a valve is equipment. In the SCD method we want computers to draw such inferences.

- The “transport dry weight” is a property that forms what we shall call the condition of the requirement. - It may seem easy to determine when the weight certificate shall be issued like, e.g., when a valve manufacturer delivers a valve that weights more than 1000 kg. But what if the valve is then mounted on a skid? Is it then sufficient to deliver a certificate for the skid only? We can say that such questions come from the context of the requirement. For simplicity we will ignore conditions from the context here, but point out that the SCD method is capable of dealing with this particular one and that context of requirements is a central point for future work.
- We shall refer to the weight certificate as the demand of the requirement. This is the part that has to be verified. A particular challenge for verification is that the requirement does not specify the name of the weight certificate. Can it be embedded in a document with a different name? If so, automated verification must get access to where the weight certificate is located. If the name of the certificate is defined upfront, it is in principle straightforward to check whether or not it has been issued.

Quite often the text in requirement specification documents is unnecessary complicated. One then has to split complex sentences up into a series of simple requirement statements that demand just one particular thing before the SCD method can be used.

Fig. Illustration of the SCD method.



SCD allows for formalized representations of requirement statements. More precisely this means that the expressions for Scope, Condition and Demand are classes and properties selected from a Reference Data Library (RDL).

One function of the RDL is to serve as a common dictionary of technical terms, a common language across the industry. However, the RDL is more than just a dictionary: - It contains a subclass hierarchy. For example, it “knows” that a valve is equipment and that an actuated control valve is a valve. It also contains data properties such as “dry weight”, ref. above example. RDL resources are sourced from existing industry standards wherever possible. ISO 14224 contains a candidate for asset breakdown structure. ISO/IEC 81355 identifies documents and information objects that are required in technical information deliveries. ISO 15926 is an upper ontology, i.e., it defines general and domain independent terms to facilitate interoperability of ontologies across multiple domains.

A key point about requirements is that they need to be verified. For example, a delivered valve with weight above 1000kg must be delivered with a weight certificate, and this is something that must be checked. Verification must of course be done with respect to actual objects in an asset. For each object in a model of an asset, the SCD method enables any requirement that “hits” the object (**Scope**) to be automatically identified, as well as the **Condition** that exist for it, and from this to determine the consequent **Demand** to it.

6.2 Modelling of Requirements

** Will be updated with how to transform conventional requirements into SCD requirements **

6.3 Elevation and inheritance of requirements

A requirement should be set as high up in the hierarchy as possible, such that it can cascade down to all the objects where it applies. This ‘cascading’ is done by means of classes and inheritance mechanisms. An example is that when a system is classed as type ‘Electrical 690V AC’, the requirement for a voltage level of 690V ±10% which applies to this system, is inherited by all systems or objects that are part of this system.
Thus, there is no longer a need to duplicate this requirement for every individual electrical object. By ‘lifting’ requirements as high up in the hierarchy as possible, the total number of requirements are drastically reduced. Test have indicated a reduction of 50-70%. To get this payback, an upfront investment is required in the form of reviewing requirements and reallocating them to the appropriate level in the hierarchy, such that they can be applied to the asset model in this fashion.

7 Implementing the Integrated Asset Model [Ontology]

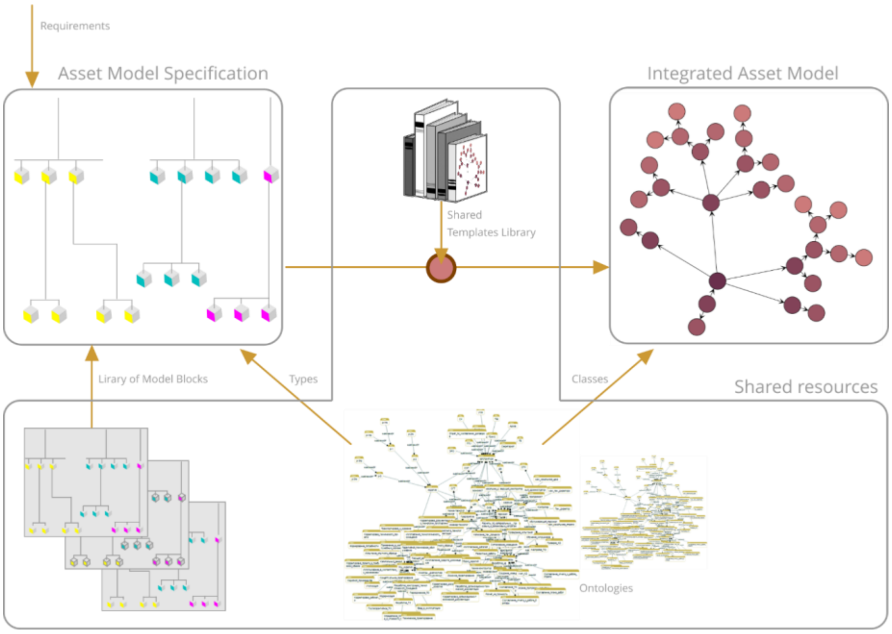
The Integrated Asset model is an object model implemented on open standards where data and information is represented and shared in an application independent manner.

The Integrated Asset Model is generated from an Asset Model Specification and reusable industry-wide library resources such that

- The Asset Model Specification specifies the project specific part of the Integrated Asset Model, based on ISO/IEC 81346
- Class libraries are designed in conformance with ISO 15926-14. They exploit and extend ISO 15926-4 and related reference data libraries including CFIHOS RDL and company developed libraries like Aibel’s MMD
- Ontology Templates libraries extend ISO 15926-7 and are defined in conformance with ISO 15926-14.
- The Integrated Asset Model is represented as an OWL 2 compliant ontology according to the W3C standards suite.

The sections below provide clarifications and details to this approach that are necessary for implementation.

7.1 The transformation of an Asset Model Specification into Integrated Asset Model



The **Asset Model Specification** consists of specification objects that represent a (partial) specification of an asset and its design (e.g., things, systems, signals, streams). One specification object specifies a single aspect of an asset, hence multiple specification objects must be used when specifying different aspects for the same asset.

The **Integrated Asset Model** consists of objects that represent the assets and its design. These objects are *generated* from the **Asset Model Specification** by transforming and merging all specification objects for the same asset into a single object in the **Integrated Asset Model**. An object in the **Integrated Asset Model** therefore captures all specifications about an asset. Objects in the **Integrated Asset Model** have strong identifiers that can be shared across applications.

The **Asset Model Specification** is described in discipline expert terminology—preferably using established discipline tooling, while the **Integrated Asset Mode**l is expressed as an ontology using the OWL ontology language. By representing the **Integrated Asset Model** in OWL2 the semantics captured by the **Integrated Asset Model** is made explicit and actionable. Sophisticated verification and consistency checks are available using established semantic technology tools. In the translation into the **Integrated Asset Mode**l Ontology, existing reference data libraries and ontologies, such as ISO15926-14, will also be integrated. This allows the **Integrated Asset Model** to exploit the semantics defined in these external resources and ensures standards compliance.

The transformation from the Asset Model Specification to the Integrated Asset Model is performed using a template-based method where the asset specifications are translated to ontology expressions using predefined and reusable templates published in shared template libraries. The use of the template-based method ensures that the translation is consistent, reproducible and automatable.

7.2 Asset Model Specifications in precise terms

An **Asset Model Specification** consists of specification objects, relations between specification objects and requirement rules that define how properties are assigned to specification objects. The specification objects have unique identifiers and may have **type**. **Types** are triggers of both requirement rules and of model blocks, where the latter is the mechanism for reuse of design.

Aspects

A specification object has always one and only one **aspect**. **Aspects** hence partition the specification objects into distinct groups.

The purpose is to give *context* to specifications. Following ISO/IEC 81346 an **aspect** conveys a specific perspective or view, and hence serves to expose information that pertains to the same perspective and hide information in one perspective from the view of another perspective.

The core aspects of ISO/IEC 81346 naturally relate specification objects to different types of objects in an asset.

- In the **Function** aspect, specification objects typically specify systems, function blocks, streams
- In the **Product**/ component aspect, specification objects specify physical components and assemblies
- In the **Location** aspect, specification objects typically specify locations or elements in a 3D model

Other **aspects** can be introduced. In particular we may introduce more than one instance of the core **aspects.** We may have, for example, two **function aspects** and three **product aspects**.

As explained in Section 5.1.5 the IMF has introduced the **Installed aspect**. Specification objects of this **aspect** typically capture information from manufacturer, and from sensors during operation.

Relations

The idea that **aspects** group similar information and keep distinct information apart is reflected in constraints on **relations**. There are two kinds of relations: **intra-aspect relations** and **inter-aspect relations**.

Intra-aspect relations relate specification objects of the ***same*** **aspect**. Two such relations are especially important for the construction of **Asset Model Specifications**.

- The *part of* relation captures breakdown structures. Different aspects will typically have different breakdown structures. This is reflected in the constraint that *part of* can only relate specification objects within the same aspect. The breakdown structure of, say, a function aspect is hence never mixed with the breakdown structure of a product aspect.
- The *connected to* relation captures system interfaces. As a case in point, assume you want to model a stream from one system into another. The IMF modelling pattern will take the output stream of the former system as a specification object, with one set of properties, and the input stream to the latter system as another specification object with associated properties that may differ from those of the former specification object. Since the output stream and the input stream is fundamentally the same stream, there is an interface between the two stream specification objects; this interface is captured by a *connected to* relation.

Breakdown structures are the mechanism for modelling systems of systems abstractions. There is an interesting interplay between breakdown structures and system interfaces. Drilling down in a breakdown structure means getting more detail in the model specification. While doing so, finer system interfaces may be seen. This process is reminiscent of the zooming functionality in Google Earth where, say, two cities are displayed with a single line connecting them. When zooming in, smaller intervening cities are displayed and what seemed to be one connection between two larger cities is now split into several connections between smaller cities.

Intra-aspect relations of the **Asset Model Specification** are preserved in the Integrated Asset Model. For the Integrated Asset Model, the relation names are taken from the vocabulary of ISO/TR 15926-14. For instance, a *part of* relationship between two specification objects of the function aspect is recast into a *functionalPartOf* relationship in the **Integrated Asset Model**.

Inter-aspect relations relate specification objects across *different* aspects. We may think of two inter-related specification objects as counterparts of different aspects. The fundamental assumption is that two inter-related specification objects are two different specifications of what in the real asset is the same thing. We may, for instance, have one pump system specification object in the function aspect inter-related to a pump system specification object in the product aspect. These two specification objects are specifications of the very same pump viewed at similar levels of granularity from two different perspectives. Inter-aspect relations used in examples in this document include:

- A Function is: *fulfilled by* a Product
- A Product is: *installed as* an Installed

The relation between a Product (i.e., a specification object of **Product** aspect) and a **Location** is a little subtle. In the examples we state that a Product is: *has location* a Location because this is the property of main interest. Strictly speaking the Location inter-related to a Product is a specification object that specifies the locational *extension* of the Product and not where the Product is located. That is, the Location is the space that the Product occupies.

When the **Asset Model Specification** is transformed into an **Integrated Asset Model**, different inter-related specification objects may collapse into the same object in the Integrated Asset Model.

Naming of specification objects

For the sake of this paragraph, we will view all specification objects of a given aspect in a **Model Specification** as nodes in a graph where the *part of* relation defines the directed edges of the graph. This graph is a so-called spanning tree: it has a unique top node, all nodes can be reached from the top node through *part of* edges, and the path from the top node to any other node is unique. This means that we can index, or name, each node uniquely by indexing its unique path from the top node. The simplest way to do this is to introduce a numbering system that enumerates all *part of* children of a node; reference will then be a string of numbers encoding the step from a node to a child through the tree. Clearly, such an index will have no information beyond its instructions for reaching a node from the top node. But it will work as an identifier defined entirely within the frame of the Model Specification itself, no reference to external identifier systems is needed.

The identification system of ISO/IEC 81346 works essentially in this way, though making the identifiers more informative than just a sequence of numbers. What 81346 Reference Designation System (RDS) does is to introduce letter codes that give a high-level grouping of the specification objects. If we want to name one child of the top node, and this child node has code GAB, we can simply name this node GAB. We only need a numbering system if there are more than one GAB child, then we need to write, say, GAB1 and GAB2. If the next child we want to name is an AAB, the name of this is node is GAB AAB, encoding its path from the root, and so on. Note that the system of letter codes reduces the need for numbers significantly.

Provided that the letter codes have meaning for the discipline expert, the name of any specification object will carry information about the role of the specification object or, in other words, information about its context. It may even betray the sequence of design choices being made before the given specification object was identified. ISO/IEC 81346 provides several systems of letter codes, and the READI project has added a system of O&G codes to this collection.

The same scheme works for each aspect. RDS uses prefixes to distinguish one aspect from another, see Section 5.

It is also possible to use intra-aspect relations to flip between aspects. This is especially important for referencing specification objects of the Installed aspect, cf. Section 5.1.5.

Types

Note to the reader of review version. This paragraph will be spelled out with example illustrations.

Key points:

- Simple types: Ball valve, safety valve
- Complex types: Ball valves that are not safety valves, Valves that are both ball valve and safety valve, Valves that are either of them
- Type hierarchy in class libraries for the Integrated Model
- Types are not sensitive to aspect. We do not treat Type as an aspect
- Types are selected from class libraries

Properties and requirement rules

Note to the reader of review version. This paragraph will be spelled out with example illustrations.

Key points:

- Scope-Condition-Demand form of requirements
    - Scope: A Type
    - Condition: Relations and property values
    - Demand: Property values
- Rules evaluate top-down
- Reusable rule library (for standards' requirement specifications)

Model Blocks

Note to the reader of review version. This paragraph will be spelled out with example illustrations.

Key points:

- The top node of a model specification can be plugged into the leaf nodes of another and in this way seamlessly extend the latter. This is the basis for library resources that can reuse, say, a model of the design code API 610 OTH.
- A model block will also get the system interfaces right

7.3 Structure of the Integrated Asset Model

The transformation of the Asset Model Specification into the Integrated Asset Model is structure preserving as explained in section 7.2.2 Relations. Following ISO 15926-14, the Integrated Asset Model is constructed using best practise design patterns for ontologies represented as ontology templates. The ontology templates ensure that the translation of the Asset Model Specification's relations is uniform and allows for merging and enriching the specification objects into objects with rigorous semantic descriptions in the Integrated Asset Model. These semantic descriptions may in turn be used to verify the correctness of the specification and the model transformation, by for example checking that the intra-aspect relation *part of* only relates objects of the same aspect.

For the current version of the IMF document, we refer to READI deliverables:

- ISO 15926-14:2020(E) for modelling patterns of Part 14
- Documentation of READI Tool from READI Phase 1 for organisation of ontology modules

Note. ISO/IEC 81346-1 introduces a key distinction between an occurrence and an individual.

- In IMF terms an 81346-1 occurrence corresponds to an object in the Integrated Asset Model that integrates all specification objects except those of the Installed aspect. In ISO 15926-14 such an object is called a functional object.
- An object in the **Integrated Asset Model** corresponding to a specification object of the Installed aspect stands, in the **Integrated Asset Model**, in an installedAs relation to a functional object. This pattern is described in ISO 15926-14.

7.4 Semantic Technologies as the Recommended Technology Stack

We recommend the use of semantic technologies and practical logic-based tools and methods for the implementation of the Integrated Asset Model. In particular we recommend to use OWL as the ontology language and OTTR as the framework for translating Asset Model Specifications into Integrated Asset Models in a scalable manner.

Note on terminology: The languages of the discipline expert and the ontology expert are usually not the same which may cause confusion. The ontology expert will, when "speaking OWL", use these terms:

- "(OWL) Individual" for object
- "(OWL) Class" for type
- "(OWL) Data property" for property
- "(OWL) Object property" for relation
- "IRI" for strong identifier

Commented [MS4]: Er asset model objektene individider eller klasser?

7.4.1 OWL as Model Representation Language

We recommend the use of the Web Ontology Language (OWL) as the language for representing the integrated asset model. OWL is the de-facto standard ontology language for practical use. It is an open standard backed by the World Wide Web Consortium (W3C), it is founded on well-known and studied concepts and principles from formal logic, and builds on existing well-proven web technologies (such as Unicode, HTTP, IRI, XML) for its representation format and implementation. These fundamentals and technologies are actively supported and further developed by an innovative community comprising researchers, technology vendors and users.

The following features motivate the recommendation of using OWL as the chosen model representation language for the IMF.

Open and extendable (schema-less) model: As its underlying representation format, OWL uses a graph-like data model format called the Resource Description Framework (RDF), also a core W3C standard. A fundamental and inherent feature of RDF is data, since it is based on a simple graph structure, may be merged even if their vocabulary schemas differ. These features also carry over to OWL ontologies, making OWL ontologies easy, on a technical level, to extend and merge with other technologies. This is in contrast to for example relational database schemas were extending the schema with new tables or columns often require more elaborate redesign.

Global identification scheme: OWL uses the Internationalized Resource Identifier (IRI) internet protocol standard as its identification scheme, i.e., using the format which is best known as the format for webpage addresses also as the format to designate model objects. This brings with it the benefit that existing web architecture can be exploited for distributing and consuming information about the identified objects, also known as Linked Data.

Support for multiple identifiers: OWL does not abide by the unique name assumption, an assumption where different names, i.e., identifiers, by definition refer to different real-world entities. OWL provides explicit constructs for specifying that different model objects refer to the same real-world object. The schema-lessness of OWL also supports annotating any model object dynamically with arbitrarily many identifying names.

Shared and distributed vocabularies and data: Using existing well-proven web architecture and technology, OWL ontologies are easily shared in a distributed and de-centralised manner.

Declarative knowledge representation format: The OWL is based on formal logic that provides support for rigours semantic descriptions of the model objects. Using declarative specifications, powerful relationships between the model artefacts, such as inheritance of attributes between objects, may be succinctly expressed.

Verification and consistency: The formal logic that underlies OWL allows ontology models to be formally checked by tools for correctness and inconsistencies. In particular, this is a powerful mechanism for detecting duplicate classes, the existence of which causes huge challenges for data quality and correct answers to queries.

7.4.2 OTTR as Model Construction and Mapping Language

We recommend the use of the Reasonable Ontology Templates (OTTR) framework for expressing and performing the translation from the Asset Model Specification to the Integrated Asset Model.

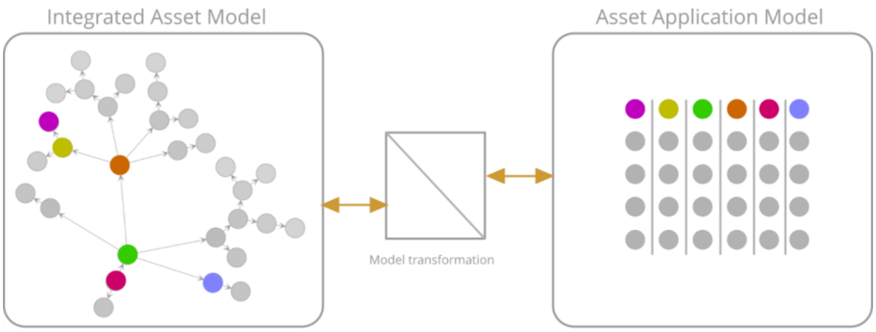
The OTTR framework is designed to improve the efficiency and quality of constructing and maintaining ontologies, and is built to fit with existing semantic technology languages and tools. The framework allows complex modelling patterns to be represented as reusable and instantiable templates, following many best-practice modelling practices and techniques, such as uniform modelling, modular patterns that encapsulate complexity, separation of concerns, and simple input formats.

OTTR Templates for a particular domain or purpose are intended to be collected in well-designed libraries that are published for reuse using the same techniques and mechanisms as for OWL ontologies. Such template libraries play a similar role in the construction of ontologies as programming APIs do in software development. Capturing modelling patterns as ontology templates prepared by ontology experts in cooperation with domain pattern experts is expected to significantly lower the time construct and maintain ontologies, while increasing the quality of the produced ontology. This is due to the fact that a few ontology experts can, by building a limited, but for practical purposes, complete set of high quality and carefully aligned modelling patterns, put a large number of domain experts in the position to actively contribute in constructing complex ontologies without the need for understanding the intricacies of the underlying logical languages. The ontology experts are (only) responsible for maintaining the template library (in cooperation with domain experts), while the ontology is generated from instantiations of these templates with data that is programmatically collected from the Asset Model Specification.

An open source reference implementation that provides tool support for using the OTTR framework is available. The tool can be used to extract data from external sources, instantiate templates, and expand the template instances to OWL ontology format. It is also an invaluable tool in the construction and management of template libraries.

8 IMF Asset Application Model [Digital/ IT]

This chapter will be developed to the next revision of this document.



How to integrate the IMF Integrated Asset Model with the legacy enterprise information systems. While the integration of an information model designed using the IMF with another model of the same type is straightforward, the integration with the information models of a legacy system is based on mappings, model transformations and system interfaces. This chapter addresses key aspects of the infrastructure that is needed.

8.1 ID management

Mapping of legacy tag numbers to RDS reference designators. Mapping between the multi-aspect and highly structured RDS - and the (legacy) single- or mixed-aspect and less structured models is less of a challenge than mapping between different types of legacy models.

8.2 Information architecture

Coexistence of ERP systems and semantic data systems. Relates to both ERP for EPC and ERP for operations. User experience during operations will mainly be founded on (legacy) tag numbers, whereas during EPC the RDS ID is more relevant. Solution must be IT-system agnostic.

8.3 Protocols for data exchange

W3C based protocols between semantic data systems and legacy systems, ETL procedures.

8.4 Navigating and utilizing contextualized information

Connecting Tag numbers with RDS strings, giving access to context information, utilizing the RDS designation to zoom in/out on the asset. Digital Twin functionality.

8.5 Mapping to legacy and custom vocabularies

How to enable coexistence with legacy systems by mapping between RDS/RDL naming and corresponding legacy system naming – for vocabularies such as attribute names.

9 Industrial and commercial application of the IMF [all]

On the path to an operational facility or asset, requirements are managed and design information is created and transferred in numerous steps. The main steps are the project phases that correspond to the incremental value creation leading up to, and into the Operation of the asset. These steps are usually also defined as distinct contracts carried out by different actors. Additionally, many similar steps occur within each phase. Relying on data on a document- or tabular format to provide the means for this information flow is hugely inefficient, which is why applying the IMF will provide value, even if implementation is initially only for parts of the total information flow.

9.1 Upgrading instead of disrupting existing work processes

Design of a facility such as an offshore platform is at the top level - in the early phase - formulated by means of various block schematics that illustrate the various systems and the main streams between them. By providing an information framework that supports a top-down approach, the IMF can be utilized to enhance such documents instead of replacing them, allowing such documents to be 'information reports' instead of information sources.
Further down into more details the same is true for such documents as Process Schematics and One Line Diagrams. This kind of 'enhancement' is achieved by assigning IMF (RDS) identifiers, attributes, and types to each system function block, and to the streams between them, thereby beginning to build an information model.
Thus, essential asset information is captured from day one, and represents a holistic view/ information model that can be enriched incrementally as the project matures, and as the value of such model is demonstrated. At any time, or level of detail, this approach can be finalized without losing value.

9.2 Foster development and availability of IMF modelling tools

The format of the IMF opens for a development of editing, authoring, and modelling tools - as well as more advanced applications or tools such as reasoning engines and generative engineering. Each of the parts of the IMF – the Asset Model Specification, Integrated Asset Model, and Asset Application Model - are powerful frameworks or platforms for developing such software functionality.

9.3 Applying IMF to save cost – un-break the information value chain

Frequently terms such as 'broken Digital Thread', 'lack of Data-centricity', 'Information fragmentation', and 'missing Context data' are used to describe a pain-point that is incurring significant cost as well as quality deterioration. By beginning to implement an IMF information model for the facility, the first steps are taken towards repairing this situation. The more extensive the model is implemented, the more value is returned by means of un-breaking the value chain.

9.4 Applying IMF to create value – providing a Digital Twin foundation

The IMF serves to integrate information from a range of connected source systems, and enables information harvesting, contextualisation and a range of other functionalities typically described as Digital Twin functionalities. A Digital Twin is a digital representation, sufficient to meet the requirements of a set of use cases.
A digital representation needs to have at its core a set of information that describes what is to be represented. Individual data is not sufficient, which is why today large efforts towards the end of a

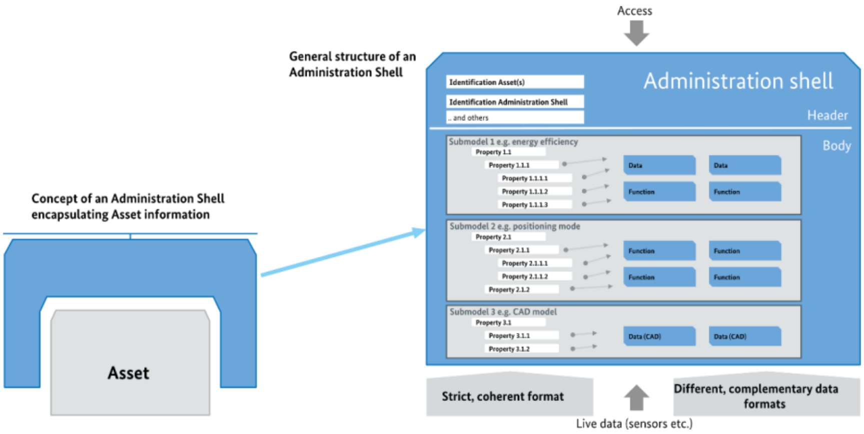
project goes into contextualizing and interrelating the produced data elements (documentation) such that it approaches an *information* model that is capable of powering a Digital Twin. The IMF instead offers this information modelling from day one.

10 IMF Integrated Asset Model in relation to AAS and OPC UA

This chapter is a brief discussion of the most relevant standards/initiatives that can be applied for deploying the IMF Integrated Asset Model for real facilities. In the present industry technology landscape Industrie 4.0 holds an increasingly prominent position.

10.1 Asset Administration Shell (AAS)

The Asset Administration Shell (AAS) is an important part of the Industrie 4.0 concept. It provides a framework for keeping and managing asset information, along the entire life cycle, across the entire value chain, and between all parties involved. The Administration Shell may be a logical representation of a simple component, a machine, or a plant at any level of the equipment hierarchy.



To this extent the AAS is somewhat similar to an Integrated Asset Model built as per the IMF, but has less emphasis on Digital Twin capabilities, and more emphasis on administration and management of asset information. In the IMF context the term 'asset' is primarily thought of as a facility (of some complexity) - with a holistic view - and usually comprising numerous functions/ products/ components, whereas in the AAS context the term 'asset' primarily refers to every function/ product/ component individually, which is intended to connect together on a network. In fact, AAS defines "asset" as "a thing with value".

As the name implies the AAS can be thought of as a shell or encapsulation of the asset data. It enables access to this asset data to be uniform and standardized, without needing to know the detailed format and structure of data in the underlying systems.

The AAS provides a controlled and secure information access to all assets represented in the network. Standard interface, unambiguously identification, syntax and semantics allows for a technology-neutral and open communication with the AAS.

This document does not specify how to encapsulate the IMF Integrated Asset Model such that it incorporates the AAS representation, but to do so would be the logical next step to make, in order to leverage the standardisation development work of Industrie 4.0, and to ensure compatibility with these standards.

10.2 OPC Unified Architecture (OPC UA)

OPC Unified Architecture is a series of standards that supports the vertical and horizontal connectivity between components, applications, and functions. OPC UA is applicable to components in all industrial domains, such as industrial sensors and actuators, control systems, Manufacturing Execution Systems (MES) and Enterprise Resource Planning (ERP) systems, etc.

OPC UA defines a common infrastructure model to facilitate an information exchange. It allows separating *instance* and *type* data in the AddressSpace, which enables components to add semantic rich self-describing information. The OPC-UA defines how properties should be formatted and organized.

Combining OPC UA with the Automation Markup Language (AutomationML / AML) enables connectivity and compliance with concepts and structure of the Asset Administration Shell.

This document does not discuss how an IMF Integrated Asset Model could leverage AML or OPC UA, but there are strong conceptual parallels that should be further investigated.

11 Standardisation and alignment with international initiatives [all]

The following is an overview of standards and standardisation activities that are relevant for the IMF development and use. The standards are listed loosely in order of perceived importance to our work.

11.1 ISO/IEC 81346 draft -O&G

Industrial systems, installations and equipment and industrial products — Structuring principles and reference designations — Part <new>: Reference Designation System for Oil and Gas

11.2 ISO/CD TR 15926 –14

Industrial automation systems and integration — Integration of life-cycle data for process plants including oil and gas production facilities — Part 14: Data model adapted for OWL2 Direct Semantics

11.3 W3C OWL2

OWL 2 Web Ontology Language, Structural Specification and Functional-Style Syntax

11.4 ISO/IEC 81346-1

Industrial systems, installations and equipment and industrial products — Structuring principles and reference designations — Part 1: Basic rules

11.5 ISO/IEC 81346-2

Industrial systems, installations and equipment and industrial products — Structuring principles and reference designations — Part 2: Classification of objects and codes for classes

11.6 ISO/IEC 81355 (EN/IEC 61355)

Collection of standardized and established document kinds

11.7 ISO/TS 15926-4

Industrial automation systems and integration — Integration of life-cycle data for process plants including oil and gas production facilities — Part 4: Initial reference data

11.8 ISO/TS 15926-7

Industrial automation systems and integration — Integration of life-cycle data for process plants including oil and gas production facilities — Part 7: Implementation methods for the integration of distributed systems: Template methodology

11.9 ISO 14224

Petroleum, petrochemical and natural gas industries — Collection and exchange of reliability and maintenance data for equipment

11.10 IEC 61360

Common Data Dictionary (IEC CDD) (a common repository of concepts for all industrial/technical domains)

11.11 IOGP JIP33

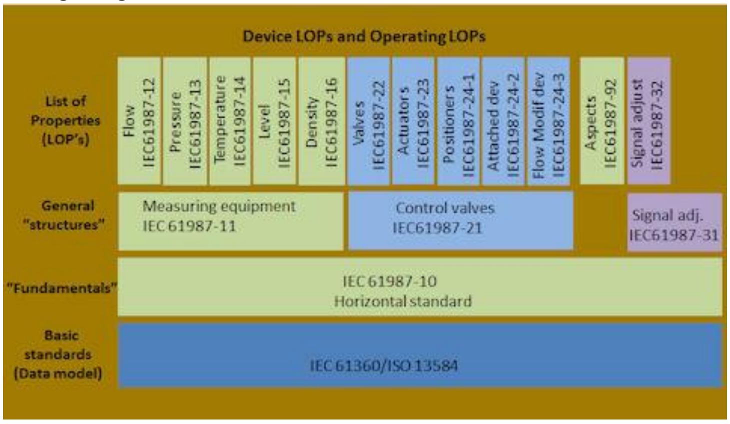
Joint Industry Programme 33: Standardizing Procurement Specifications

11.12 IOGP JIP36 / CFIHOS

Joint Industry Programme 36: Capital Facilities Information Handover Specification

11.13 IEC 61987

Industrial-process measurement and control – Data structures and elements in process equipment catalogue. Fig. Structure of IEC 61987.



11.14 DEXPI

A general data exchange standard for the process industry, covering all phases of the lifecycle of a (petro-)chemical plant, ranging from specification of functional requirements to assets in operation. Currently, the focus of the DEXPI initiative is the exchange of Piping and Instrumentation diagrams (P&IDs).

11.15 Standard Norge

The main standards organization of Norway. It claims responsibility for all standardization areas except for electrotechnical and telecommunication issues.

12 IMF maintenance and User Support services [all]

The IMF enables extensive use of shared resources. To support this, a service and organisation needs to be in place. Such organisation(s) shall ensure the availability of shared resources, as well as manage or orchestrate maintenance and development of content, as well as alignment with applicable standards.
The POSC Caesar organisation is an example of such an organisation, relevant for the ISO 15926 standard. Similarly, organisation(s) should be established to support the other standards/content that are foundational to the IMF, as well as to support the IMF in itself, possibly elevating the document to become a standard.

12.1 Maintain structures – ISO/IEC 81346

To be further detailed in later revision, as work progresses.

12.2 Maintain ontologies – ISO 15926, ISO/IEC 81355

To be further detailed in later revision, as work progresses.

12.3 Maintain NORSOK Digital Standards Z-001, -018

To be further detailed in later revision, as work progresses.