



33rd Annual **INCOSE**
international symposium
hybrid event
Honolulu, HI, USA
July 15 - 20, 2023

Common Language for Systems by the ISO/IEC 81346 Reference Model

The key to digital continuity across silos and enterprises

Henrik Balslev
Systems Engineering A/S
Østerbrogade 48, 2. floor
DK-2100 Copenhagen
+45 21 68 48 67
hb@syseng.dk

Thomas Barré
AIRBUS S.A.S
316 route de Bayonne
FR-31300 Toulouse
+33 6 27 44 50 82
thomas.barre@airbus.com

Copyright © 2023 by Henrik Balslev & Thomas Barré. Permission granted to INCOSE to publish and use.

Abstract. This paper focuses on the practical application of system architecture modeling and reference enabling in systems engineering, introducing the fundamental principles and proof of concept. The fundamental principles are defined by ISO/IEC 81346 standard series, which provides a common language for systems across all engineering disciplines through the Reference Designation System (RDS). RDS is used to create a reference model with unambiguous identifiers representing system breakdowns in different views called aspects, allowing for clear reference designations of system elements and their relationships, complying with the basic rules of ISO 704 – Terminology work. The practical use of this system reference model is demonstrated through a selected use case: the Digital Design, Manufacturing & Services (DDMS) transformation program at AIRBUS S.A.S. (Toulouse, France). This case includes practical experience and a brief introduction to tools that assist with the usage of ontologies and terminology work in daily life.

Introduction

Nowadays, modern systems engineers, work with the same fundamental principle by (i) organizing a set of interrelated objects into systems, (ii) structuring the comprised elements within a system into hierarchical relations, and (iii) defining non-hierarchical relations between elements across systems.

The International Organization for Standardization (ISO) defines these principles in the standard ISO 704 by establishing terminological clarification of the links between objects, concepts, and designations. The standard highlights the importance of clarifying relations among concepts with increasing complexity of a system concept (ISO 2009, pp. vi, 1). A system concept is hereby described as a “mental representation” of the actual system (INCOSE 2015; ISO 2009). Modelling such a system concept in compliance with ISO 704, 5.5, the standard defines that “**at least** the following relations shall be used to model a concept system: hierarchical relations and associative relations”. In detail, this implies generic and partitive relations (hierarchical relations), as well as associative relations (non-hierarchical) (ISO 2009).

In practice, these three kinds of relations combined amplify the applicability of a reference designation system (RDS) by enabling explicit references between multiple system models. Hereby, RDS defines the nomenclature containing the principles of unambiguous reference, so called reference designation, for the systems, their elements, and their three relation kinds. These principles are defined in the ISO/IEC 81346 covering various engineering disciplines and particularly addressing the industries of construction works and power systems (Balslev 2015; ISO 2015, 2018, 2022b, 2022a). The standard is in the process of being constantly expanded by including additional core industries such as aircraft systems, oil & gas systems, and manufacturing systems (Balslev 2017).

Despite existing standards, it can be observed that the identification of systems experiences ambiguous references across all engineering disciplines and their different system models. In small teams, this ambiguity can be counteracted by human-to-human communication as part of a community. For instance, in a company certain esoteric terms derived from daily life. In a digital space, however, models and information are stored, managed, and exchanged by computer systems (machine-to-machine) involving some form of machine-to-human interface. Adding the challenge of large-scale engineering projects, multi-cultural workforce, and geographical locations, the degree of difficulty and risk of miscommunication increase drastically causing potential overbudget and failed projects (Hicks et al. 2020).

Consequently, this paper addresses the need for an unambiguous common language to enable accurate, faultless, and understandable human-to-human and human-to-machine communication within and across systems. This common language is established by introducing a system reference model based on RDS in accordance with ISO/IEC 81346, which is understood by humans as well as digital solutions. The aim is to promote the practical application of systems thinking and modelling of systems. Using AIRBUS as a use case, the AIRBUS Common Language is presented by using fundamental systems engineering breakdown of systems into system elements, designated unambiguously by RDS, and finally represented in a system reference model.

Reference Designation System

Difficulties in Relating Today's Models

By default, different engineering disciplines think in specific views when creating models of their respective systems engineering. The difference becomes clear when one must work interdisciplinary and connect these different models from different domains, i.e., when the models need to refer to each other. Figure 1 shows an example of different models and documents representing system elements of a thrust system in an aircraft from a functional and physical perspective.

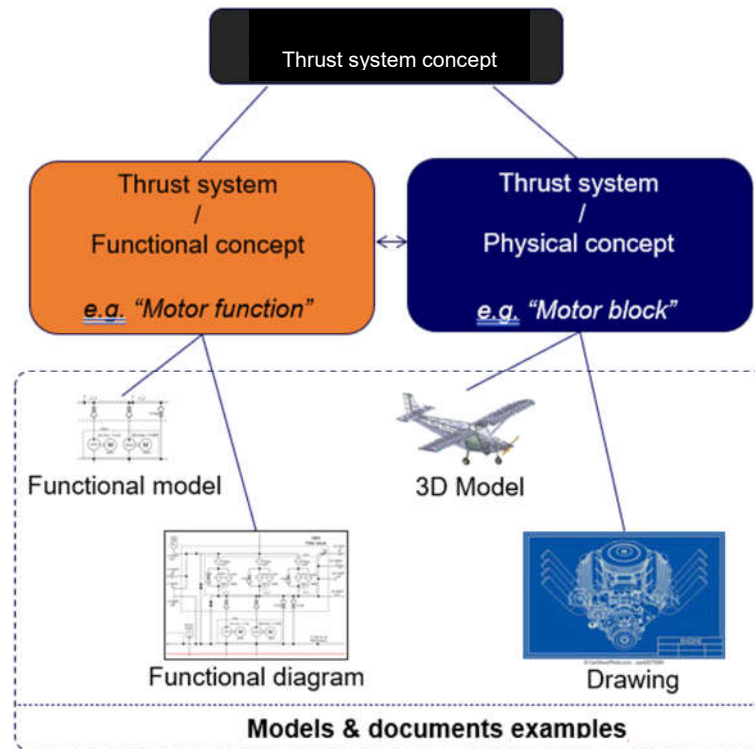


Figure 1. Models and documents including a thrust system from different disciplines.

Electrical and process engineers typically design functional models depicted in diagrams, whereas mechanical engineers are orientated towards physical constructions with a representation in, for example, 3D models or drawings. At the end, not one single model or view of one discipline is superior since every model solely serves the specific need of a given discipline.

Figure 2 provides an example of two different models and their individual strengths and weaknesses: The functional model (left) can be created in a fast fashion and uses intuitive symbols to delineate functionality while being limited in detail and physical location. In contrast, a physical 3D model provides a detailed, realistic, and geometrical representation of the system but it excludes functionality aspects.

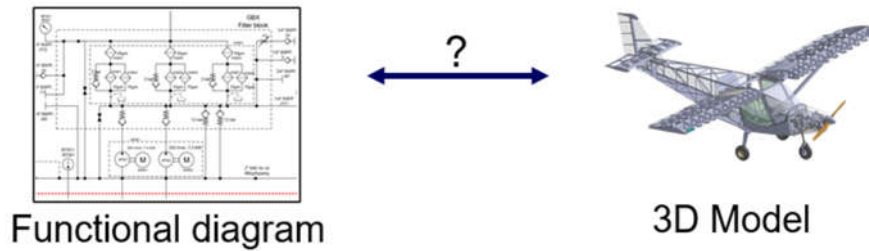


Figure 2. Two models from two engineering disciplines representing the same aircraft system: Functional view of thrust system (left) and physical view (right) of entire aircraft system.

Relating the functional model with the physical model is not only challenging but also misleading because the models represent different information, which are based on a distinct view on the system. This nature represents a hurdle in linking these sets of information together especially when designations and relations must be comprehensive by humans and computers. Even if the models are referring to the same database, which in principle should be a neutral point, reference across and between models require a tailored and complex digital environment.

As a result, a common language is required across all models and disciplines which is understood by humans and IT systems. Such a common language is defined by RDS. It offers different views for different needs, supports by default systems thinking (systems and their elements) and finally defines a syntax for reference purposes called reference designations, which are used to refer to the reference model defined by RDS.

Figure 3 shows how the RDS Reference Model can play the role of the neutral points (i.e., non-discipline dependent) of synchronization for all models. Hereby, the thrust system concept represents the system of interest, which encompasses the functional, physical, assembly process, and maintenance process concept. The indicated relations between these concepts serve as a simplified illustration of relations between certain elements within different concepts. Under section 2.7, a concrete example of relations between elements is depicted in Figure 8.

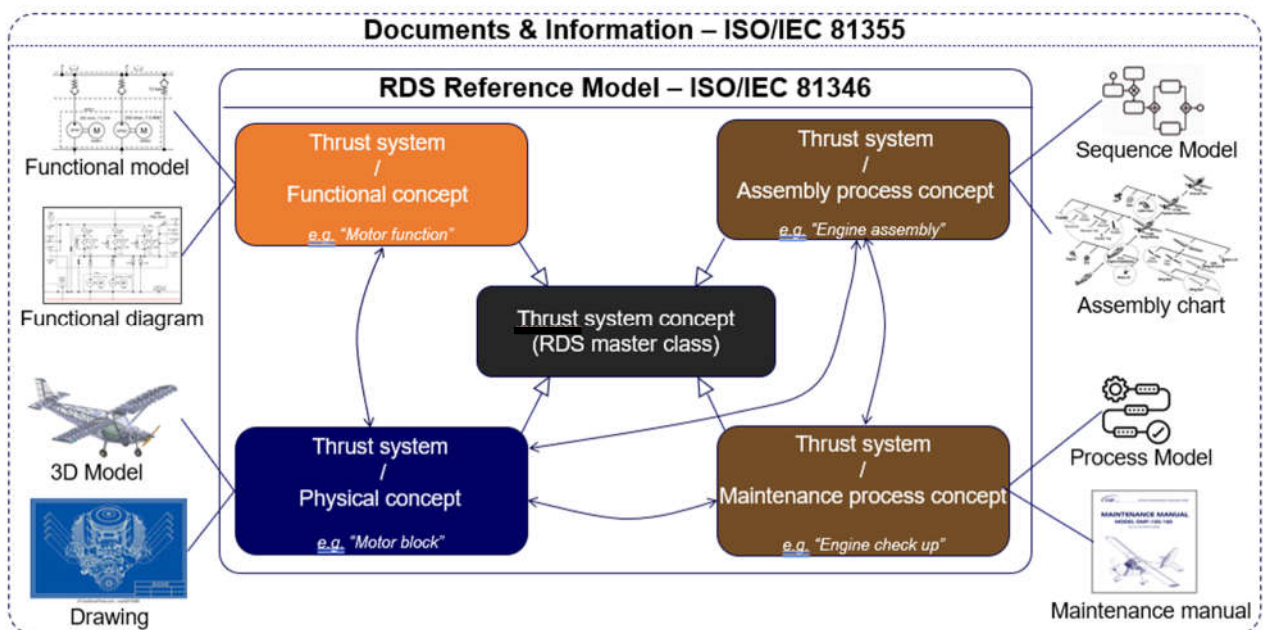


Figure 3. Different models and documentation from different disciplines, related to the RDS Reference Model (center)

The RDS Reference Model

The reference model is a discipline neutral model, i.e., it only represents systems, their elements, and the relation among them, which are not specified to any engineering discipline. The reference model offers different views, called aspects, to view the system and its elements. Within the reference model, RDS is used to apply designations to systems, their elements, and relations. Mastering the different kinds of relations is the key to “speak the language” of systems and designate them with RDS (RDS 81346 Technique ApS 2021).

The most useful relations are the hierarchical relations (part-of and type-of) and the associative (or pragmatic) relations. Hierarchical relations are by nature very organized, as in contrast to the associative relations, which tend to grow wild but can be managed with classification. The part-of relation is a 100% match to support any system composition (i.e., breakdown) in a model, which makes the model easy to establish and to modify.

RDS and the reference model can be applied in line with other popular models such as the Systems Modeling Language (SysML). SysML is based on the Unified Modeling Language (UML) and represents a graphical modeling language to design complex systems along the entire systems engineering lifecycle. In contrast, RDS is a system for referencing components within those systems. Consequently, SysML and RDS can benefit from each other by (i) including RDS information into SysML models, (ii) identifying components in SysML models using RDS, (iii) constantly developing and updating SysML using information from RDS, and (iv) vice versa.

As proof of concept for the reference model, several prominent digitalization programs have seen the potential in using RDS to create their reference model. Among these are the Airbus DDMS program (Digital Design, Manufacturing & Services) and the READI ((Requirement Asset Digital life cycle Information) project in the Norwegian Oil & Gas sector (Airbus 2021; READI 2020).

How the RDS 81346 Technique works

The introduction of ISO/IEC 81346-1:2022 (general rules) defines the following (ISO 2022, p. 11) :

“By applying the structuring principles, even very large sets of information in a complex installation can be handled efficiently. [...]

The structuring principles and the rules for reference designations:

- *provide a system that is easy to navigate within and easy to maintain;*
- *provide an excellent overview on a technical system¹ since composite structures are simple to establish and understand; [...]*
- *allow, by accepting more than one aspect, that more than one coding principle can be applied;*
- *allow ‘old structures’² to be handled together with ‘new structures’ by using multiple unambiguous identifiers;*
- *support individual management for the establishment of reference designations and enable subsequent integration of modules into larger constructs;*
- *support the establishment of reusable modules, either as functional specifications or as physical deliverables;”*

¹ Systems and systems thinking is an integral part of the standard.

² Meaning, existing structures.

The scope of RDS is to establish a reference designation, which shall be assigned to objects within models, which refer to the reference model. So, a RDS reference model hosts reference designations and elements inside models. Hereby, the model has one or more reference designations as well, so other models can refer to the reference model.

The rule is that at least one reference designation shall identify the system or its element unambiguously. Subsequently more than one reference designation can be allocated to an element, depending on the complexity of the model(s).

Syntax of the Reference Designation

The syntax of a reference designation is representative for the common language among models and follows the subsequent principle: If the system and its elements are represented in more than one model, for example an electrical, process, and instrumentation diagram, the same reference designation will be used in multiple models across disciplines.

Same reference designation means same system element, no matter how it is represented or even called in daily life. The RDS syntax is robust and immune to misspelling of system naming and preferred discipline languages as it includes letter codes representing classes of systems with precise definitions of each class.

RDS is based on three simple mechanisms which put together provides the reference designation, i.e., the common language used for reference purpose:

1. **Libraries classify** the system and its elements into descriptive classes. Classes are recognized by one, two, or three letter codes representing different abstraction levels of systems. The RDS ISO/IEC 81346 classification is robust as each class is distinctly defined according to the principles in ISO 704 and ISO 22274.
2. **Aspects define** the views on the system and uses a prefix () to designate this. The aspect prefix shall be a part of the reference designation. The aspects are Function (=), Product (-), Location (+), Type (%) and “open to user” (#).
3. Within each aspect, **systems are organized** with their system elements in part-of relations. One refers to a given system or its element in their respective structures by combining reference designation from each level in the structure to be an address pointing at the system element.

The syntax of the reference designation is shown in Figure 4 and its application is elaborated in an example adjacent to Figure 8.

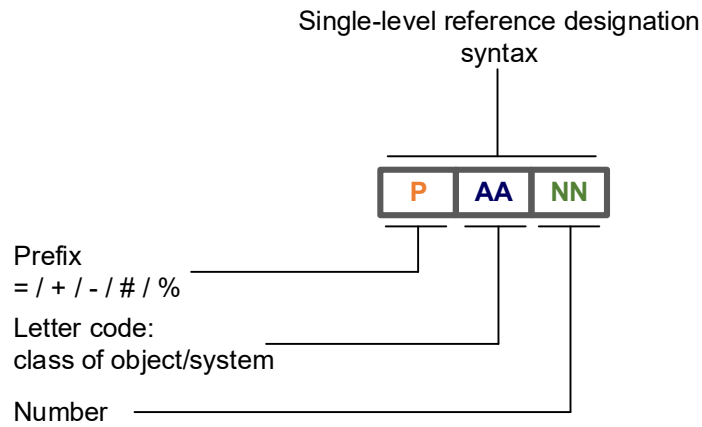


Figure 4. Reference designation syntax used to designate a system element in a reference model.

Relations in RDS

In RDS, the relations between systems and their elements are fundamental. Relations in a reference model are part-of relations, type-of relations, and associative relations used in combination as required by ISO 704. In RDS, type-of and associative relations are classified, which allows to properly relate the systems together in the reference model and to navigate within this graph.

To refer to structures in different aspects, the single level reference designations are concatenated to make an unambiguous reference designation for systems and their elements. By assembling the individual reference designations, the system breakdown in a part-of relation can be addressed as shown in Figure 5:

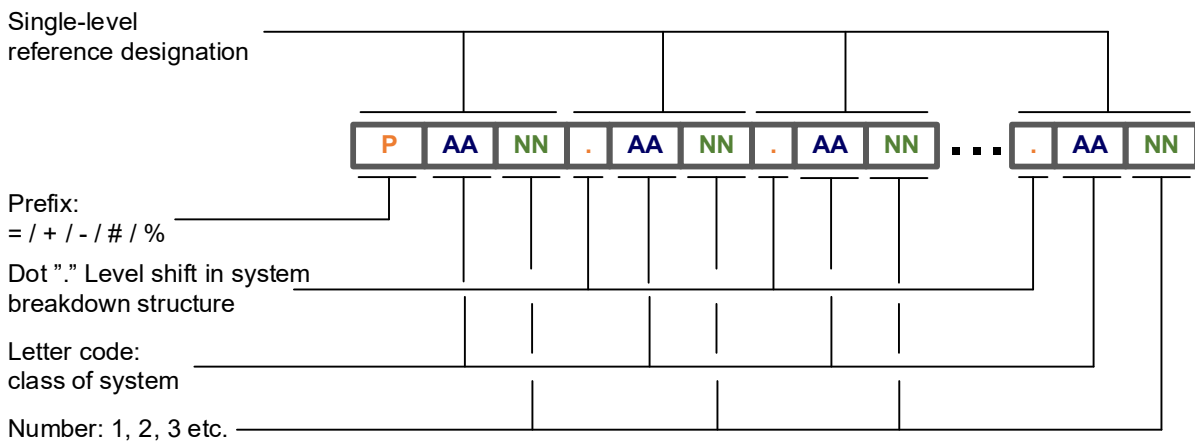


Figure 5. Concatenation of reference designations to address systems and their elements in a system breakdown structure.

Type-of relations are used to create classification schemes, which are the libraries used for the reference model. Part-of relations are used for decompositions, i.e., manage systems and their system elements. Associative relations are used to relate systems across aspects or model System of Systems.

The reference designation is an attribute (metadata) to any system or system element, which identifies the system unambiguously. Therefore, the reference designation shall be shown next to (or nearby / as a part of) the system or the element in the system documentation or system model representation. The reference designation acts as a TAG for systems and creates the common understanding of

different views of systems and their elements along with a common understanding of which system (or element of the system) are viewed.

Classification of Systems

One important feature of RDS is the system library with definitions of systems and their system elements. The libraries are made robust by means of classification hierarchies including definitions for each class provided. Thereby the RDS system libraries:

1. Can host any term used for any system yet making the recognition of it unambiguous by referring to the letter codes used to represent the class.
2. Become human and machine language independent because all libraries can be translated to any language (Spanish, Danish, German, etc.), keeping the class letter codes unaffected and thereby internationally recognized:

When classifying any system or system element, according to the rules in ISO/IEC 81346, one, two or three letter codes are used as a part of the reference designation as follows:

- ONE letter code for designation of high-level systems, i.e., high abstraction level
- TWO letter codes for designation of mid-level systems, i.e., mid abstraction level
- THREE letter codes for designation of component system, i.e., elementary abstraction level

The classes of systems (one-two-three letter codes) are independent from each other, so each class can be included in a part-of breakdown to represent systems and its elements without limit.

Figure 6 shows how this hierarchy and abstraction levels are defined in RDS:

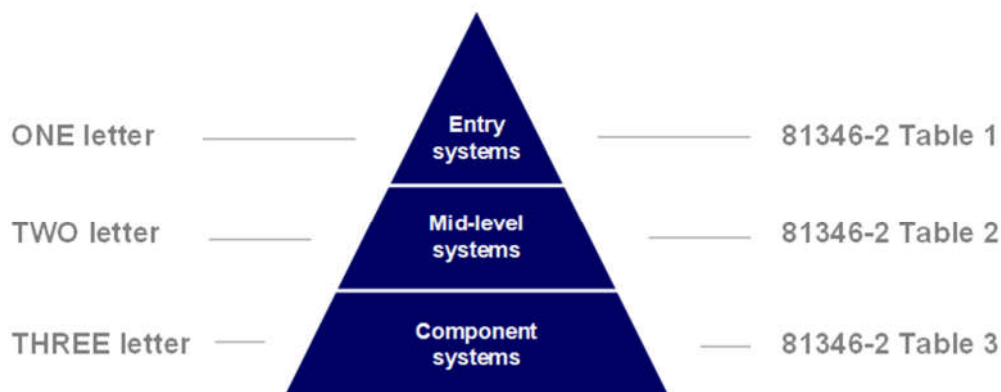


Figure 6. RDS classification library principle.

Even though there are three levels of abstraction, i.e., ONE-TWO-THREE letter codes to choose from, this does not mean that one always must use all of them and in that 1-2-3 sequence. It works more like LEGO® bricks: RDS has classified terms in these three boxes, and one are free to build any reference designation one may like out of it, for example in a ONE-TWO-TWO-THREE combination or ONE-THREE-THREE combination etc.

- Examples of **entry systems** of an aircraft are Navigation system, Communication system, Lift system, Interior system, and Decision system (i.e., a human, for example a pilot).
- Examples of **mid-level systems** (called “technical systems”) of an aircraft are Body structure system, Joint system, Pitch control system, Roll control system, Yaw control system, Electrical power supply system, and Fuel distribution system.
- Examples of **component systems** (shared with all other industries) are Flow transmitter (e.g., pitot tube), positive displacement pump (e.g., fuel pump) and a Door (e.g., PAX door).

Entry systems (ONE letter) and mid-level systems (TWO letters) typically become industry dependent and vary between construction works, power systems, manufacturing systems, aircraft systems, and oil & gas systems, whereas the component systems (THREE letters) are a shared library for all industries.

Figure 7 shows an example of class definition of a component system class GPA. The definition of main class G is inherited to sub class GP, which is inherited to sub class GPA. Thereby, the class definition for each class becomes distinct and terms can be classified to its appropriate class by complying with the definition. See column “Examples of terms” in Figure 7. Mobile applications for RDS 81346 simplify the designation of elements (RDS 81346 Technique ApS 2016).

Class code	Class definition	Class name (Preferred term)	Examples of terms	Criteria for definition of subclasses
G_	<i>object for providing a controllable flow</i>	generating object		Kind of object flowing and kind of source
GP_	<i>generating object of liquid</i>	liquid flow generating object		Kind of method applied
GPA	<i>liquid flow generating object by forcing trapped liquid to the discharge side</i>	positive displacement pump	peristaltic pump, piston pump, rotary gear pump, screw pump, vacuum pump, pump	

Figure 7. Example of definition of class GPA – Pump derived from (ISO 2019, Tab. 3)

Example of RDS Reference Model

With the RDS 81346 Technique in hand, a non-discipline dependent reference model can be established. The reference model will by nature represent systems with their elements and their relations. RDS provides a reference designation as the unambiguous reference to the model from any other model as shown in Figure 8 for the fuel pump from different aspects.

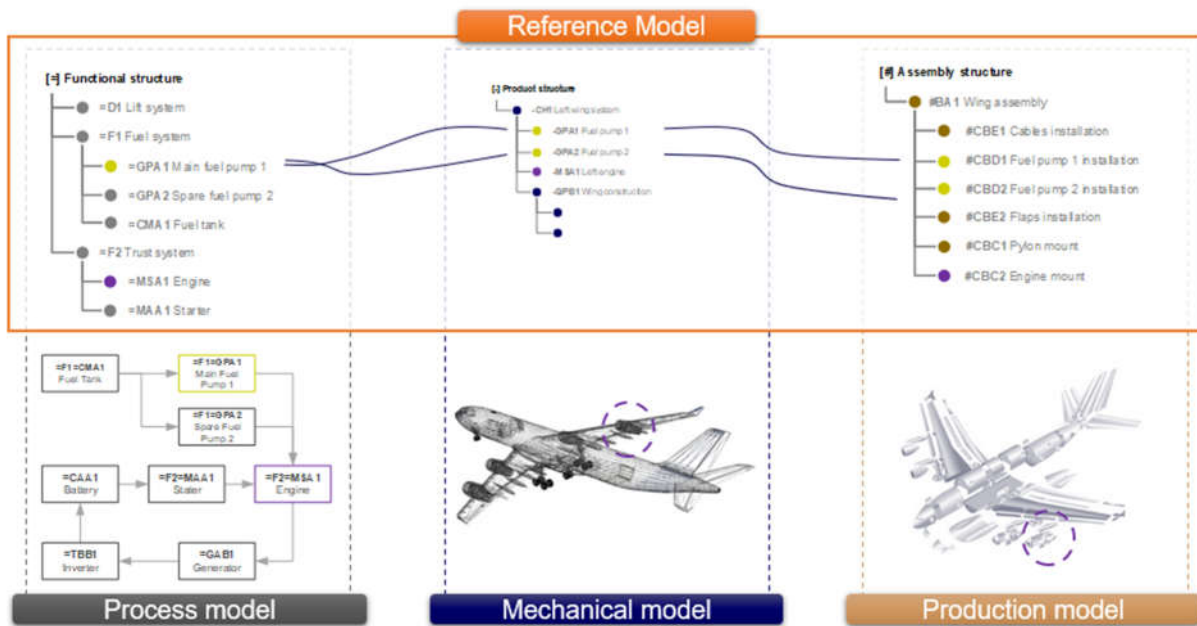


Figure 8. Different models referenced to the RDS reference model.

To refer to the reference model, the reference designations are used as shown exemplary in Figure 7:

=F1.GPA1: Main fuel pump 1 (class GPA), part of Fuel system 1 (class F)

In this case, the functional “main fuel pump 1” is in fact realized by two fuel pumps in the left wing, designated **-CH1.GPA1** and **-CH1.GPA2**. The relation between **=F1.GPA1** and (**-CH1.GPA1** and **-CH1.GPA2**) is an associative relation, which can be written:

=F1.PA1 | realized by³ **-CH1.GPA1** and **=F1.GPA1** | realized by **-CH1.GPA2**

The fuel tank in the aircraft is designated to:

=F1.CMA1: Fuel tank 1 (class CMA), part of Fuel system (class F)

The letter codes applied in the reference designation (above) are defined in the RDS System Libraries. In Table 1, examples from the RDS-A/C for aircrafts are shown:

Table 1. Examples of classes with definitions

Class	Definition
F	aircraft system generating speed
CH	shape system directing flow to create lift
CMA	enclosed stationary storing object for fluids
GPA	liquid flow generating object by forcing trapped liquid to the discharge side

In contrast to the fuel pump, there is no fuel tank part assembled into the aircraft as such. The fuel tank function is achieved by sealing wing structural parts already achieving other functions such as load transmission or lift creation. The fuel tank is therefore a sealed area in a wing (a “wet wing”). Hence, it refers to the fuel tank by means of the functional aspect (prefix ‘=’). This is a useful mechanism in the RDS 81346 Technique because if the system element can be designated to at least

³ Classes needed to control associative relations are defined, but not in scope for this paper.

one aspect, it also can refer to any model. In this case, the system element is justified to be represented in the reference model.

Likewise, products do not need to have a specific function, i.e., not all system elements need references across structures if it doesn't add value to the reference model. One entry in the model is enough but often more references are useful.

Applying RDS at Scale in a Digitalization Program

This part of the paper introduces the method of implementing a common language at AIRBUS, which is understood by humans as well as IT systems. The AIRBUS Common Language (ACL) is achieved by using the fundamental systems engineering breakdown of systems into system elements, subsequently represented in the reference model, and designated unambiguously by RDS.

To make the reference model useful and practical for the AIRBUS DDMS project, further development in interlinking mechanisms is added and defined resulting in the ACL. The ACL covers systems, their relations, and properties as well as the manufacturing systems (enabling systems) of the product. Today, the ACL based on RDS ISO/IEC 81346 is considered as a very promising enabler for use of model-based engineering at AIRBUS in the DDMS program.

Developing the Common Language for AIRBUS

Applying a digital thread at the scale of AIRBUS means to manipulate hundreds of models, which derive from different sources and disciplines, and manage the resulting model, which contains billions of elements and connections.

When operating at such a large scale, the task of federating various models, identifying systems and their components, and threading through billions of raw identifiers presents a significant challenge, especially across domains, IT platforms and enterprises.

The solution encompasses adding semantic tags as identifiers of the information elements. These semantic tags are in fact reference designations, used to refer from any given model to the reference model. The ACL is the language providing the syntax of these tags. It is based on the ISO/IEC 81346 standard series. It provides structured knowledge to identify and relate the different pieces of information.

The overall goal encompasses:

- the ambition of the ACL is to create a universal data identification strategy to clarify and relate information at scale.
- the vision is to apply this data identification strategy to support codesign and reuse of data at scale.
- It is based on:
 - a common language available for anyone exposing new or legacy data across actors, disciplines, enterprises, platforms, programs, contexts etc. readable by humans and machines
 - associated services and governance enabling the use of this common language to designate & retrieve any new or legacy data persisted at AIRBUS and its extended enterprise
 - incremental deployment of these common language applications inside AIRBUS and with extended enterprise early adopters

- incremental expansion and optimization of this common language to cover all the aerospace sector data exposition needs.

Historically, AIRBUS progressed to its current state through a process that involved testing the fundamental principles of the RDS 81346 Technique, which was introduced at a previous INCOSE symposium in 2014. In 2017, AIRBUS took the first steps towards determining whether the technique could serve as an enabler for digital continuity. At this time, however, the aerospace sector was not specifically covered by RDS. AIRBUS took the decision to create an extension of RDS dedicated to civil aircraft, which became the starting point of the ACL project. Some important features of RDS were instantly recognized to be useful:

- Basics are defined in an international standard which ensures that this language is not stakeholder specific.
- It is open source, and there was no need to spend years to develop it.
- It is ready to use. The coordination effort in definition of tags to be used is reduced to the minimum even if we would decide to link different system domains such as aerospace and automotive industries.
- The RDS language is easy to understand and implement in daily life. There is no need for certification nor long training courses as only a few main classes need to be known to start naming systems on a given domain.
- Creating reference designations (tags) goes fast. During real application cases we have created millions of tags at the rate of 10.000.000 tags per full time equivalent per year.
- When using RDS for designation of systems and their elements, the related data become crystal clear so retrieving data is easy, even for non-domain specialists.
- The language offers different views of the same system. This common language can be used on new and legacy data and remain stable even for long lifecycles such as the typical 40 years of aircraft programs.
- Scale up is possible as this language can be used in a distributed mode and in extended enterprise. Testing the RDS at real scale leads to the conclusion that it works as intended, in fact even better, with a lot of unforeseen benefits.
- This Business language is readable and thereby understood by humans and machines.
- It is only labels (i.e., tags) so it minimizes the footprint on IT while being agnostic to the IT tool and languages used.
- It eases a lot the models federation as most of the parameter's exchanges between models are properties, which are classified by this language.

AIRBUS not only designs but also produces aircrafts. So, the next step was to create another extension of RDS dedicated to the manufacturing domain.

Both these new domains are based on the RDS standard foundations: They use the same aspects and the same elementary systems libraries. The point was to create the abstraction levels dedicated to these new system domains. With these relations in hand, AIRBUS was able to link all the aspects and domains together in a system reference model.

Furthermore, AIRBUS has created a classification of the properties related to the systems represented in the reference model. Properties are characteristics of a system. For example, the length, width, speed, etc. are properties of this aircraft.

Application of RDS at AIRBUS

During the process of defining classes of systems to accurately designate an aircraft, AIRBUS also developed manufacturing procedures, established relationships and properties, and created tests and applications. The designation and manufacturing procedures were the subject to the two main applications of ACL: Co-design and design-to-cost.

ACL contains 100 classes of relations, 2000 classes per domain (aircraft, manufacturing, construction work, process etc.), which allow straightforward navigation. ACL allows to describe detailed systems and their relations thanks to a panel of about 2.000 classes per domain, which are easy to retrieve as they are organized in libraries structured per abstraction level.

ACL is used to tag the pieces of information, which are filtered on their nature, aspects, and classes. Hereby, valuable answers can be provided to answer practical questions such as:

- On the Co-design side, it was demonstrated that by a simple navigation across the aspects of the aircraft and manufacturing domain, ACL provides the list of aircraft functions that can be tested when only a section of the aircraft is manufactured. This allows to reduce the number of issues discovered when the whole aircraft is powered up.

On the Design-to-cost side, the same filtering principle has been applied to retrieve the tools & consumables used to realize a given work order. The costs of tools and consumables are properties attached to them, so non-recurring cost and recurring cost can be provided.

The maturity of this project is such that it is used for real applications, e.g., the torquing tools catalogue has been digitalized using the ACL. It allows the users to retrieve the torquing tools that fit their own application considering the fact that AIRBUS has 200.000 different types of bolts and the catalogue contains 500.000 different types of torquing tools.

The match between bolt type and tool type depends on size, shape, worker position, energy available, environmental constraints etc.

Without the ACL, nobody managed to create this catalogue. The ACL team has created about 60 million of tags to properly designate every bolt and torquing tools on various aspects including their properties and relations. This has been achieved in about 6 weeks by 3 people at a rate of 100.000.000 tags per full time equivalent per year. A human-to-machine interface has been created to filter these tags and retrieve the needed tools type.

The conclusion after this practical test has demonstrated the following:

When the AIRBUS ACL is used to index the available data, a crystal-clear data lake is created.

As all the pieces of information are linked together and classified this is clearly an enabler of Model Based Systems Engineering at the scale of AIRBUS.

Of course, this language eases co-design, Design to Cost, models federation in bottom-up of information or top-down cascade of structures, trade-off at system-of-systems level and comparison of systems decompositions.

To succeed with this application case, AIRBUS developed several services. The first exposes the language as a common meaning reference.

In order to distribute the AIRBUS ACL through the highest number of disciplines and IT platforms in the company, AIRBUS decided to adopt the World Wide Web Consortium standard model called Resource Description Framework (RDF).

This standard enables description of ontologies via triples, containing a subject, predicate and object, gathered in a graph. Resources in the graph can then be structured to reflect for instance that a given class with its definition inherits from another one, and they both belong to a given RDS library associated to an aspect. For instance, Figure 9 illustrates the Door physical component (class QQC), which inherits from Space access object QQ_, which inherits from Controlling object Q__ in the Aircraft Physical library.

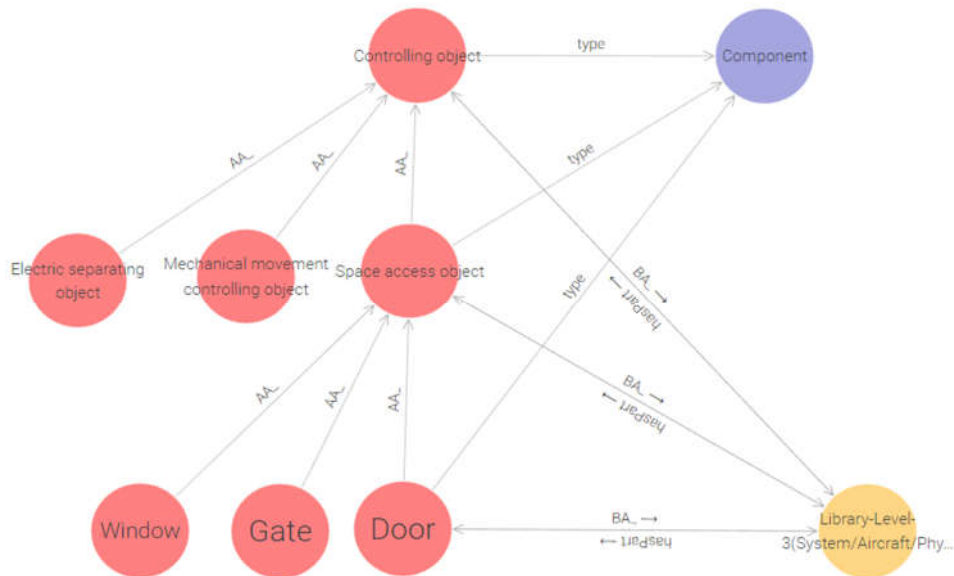


Figure 9 – Illustration of RDS classes structure principle in AIRBUS ACL RDF graph

The AIRBUS ACL knowledge is serialized into a file that can be exposed in any graph database. Concepts can then be queried, and anyone can refer to them. This way, RDS principles can be deployed.

Once a domain knowledge is captured, the tagging process comes into play. Each concept must be linked to a component of the ACL reference framework. In order to identify which one is the most suitable the ACL team has developed a Machine Learning based application that suggests definitions. Algorithms have been trained against both already tagged data and examples from the ISO/IEC 81346 standards. This service eases navigation within libraries but requires a mandatory verification of the suggestion, as shown in Figure 10.

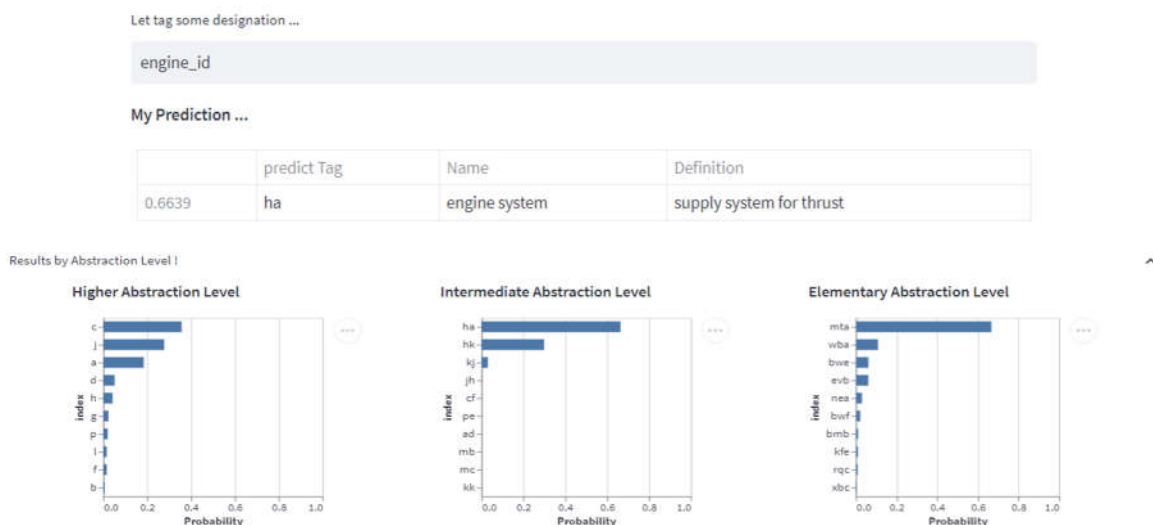


Figure 10. Class suggestion using Machine Learning algorithm, for an engine.

The materialized link between the ACL component and the concept can be stored and uniquely identified thanks to indexing functionalities of graph databases.

Any entity willing to designate its data with RDS needs to build a knowledge graph. It contains concepts (in a language that can be specific), linked together to express the entity’s daily business. It is all about federating information coming from various data sources. Two solutions have been implemented to perform this job in practice.

The first solution considers tabular data that could come from relational databases. Each column of the table is linked to an ACL component and is linked with other columns. When generating the graph, both the concepts and all row instances are explicitly designated. The designation process is displayed in Figure 11.

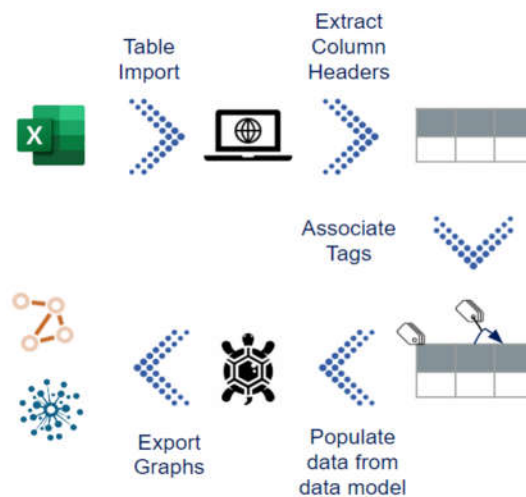


Figure 11. Digitalization process of a table, using ACL

A second way makes it possible to create concepts from scratch and link them together, also generating a graph. A screenshot illustrating real use is provided in Figure 12.

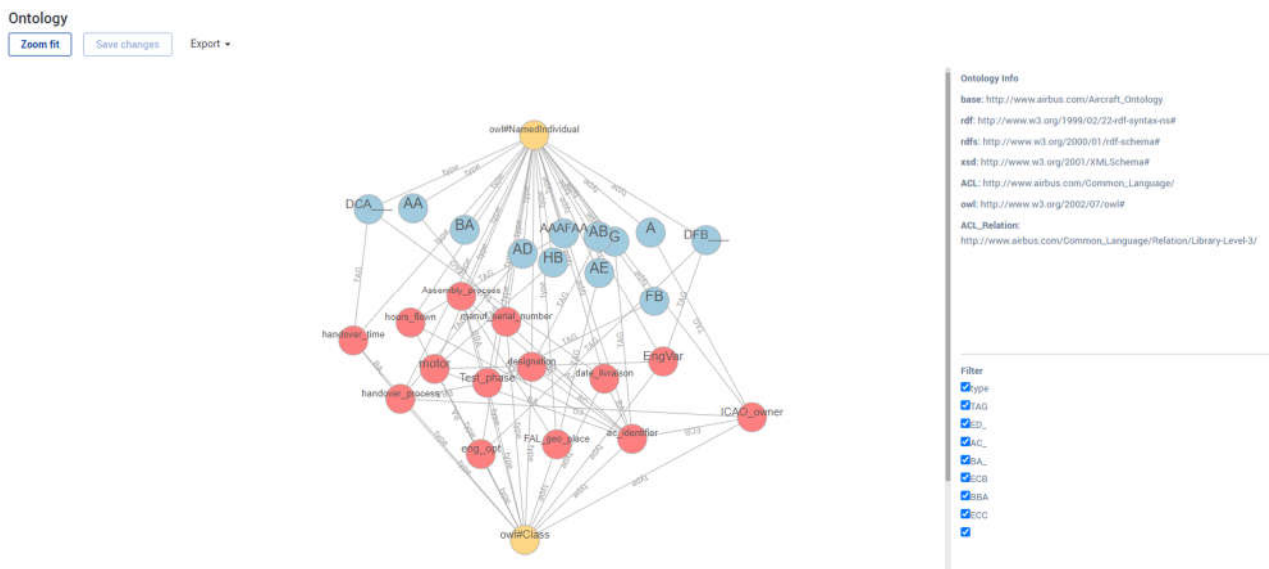


Figure 12. Visualization of an ontology in the ACL authoring application.

Conceptual layers are created to manage billions of data sharing certain communalities either regarding what they are or how they interact with their environment. The above-mentioned link to a foundation layer permits trans-domain, trans-application, and ultimately trans-company analogies.

Connecting ontologies at a large scale can be a time-consuming process. The RDS Technique contributes to a more efficient detection of such analogies.

Graph analysis methods can compute the distance between concepts of two graph ontologies via the meaning reference as shown in Figure 13. Thanks to its structure, the reference segregates components based on their semantics.

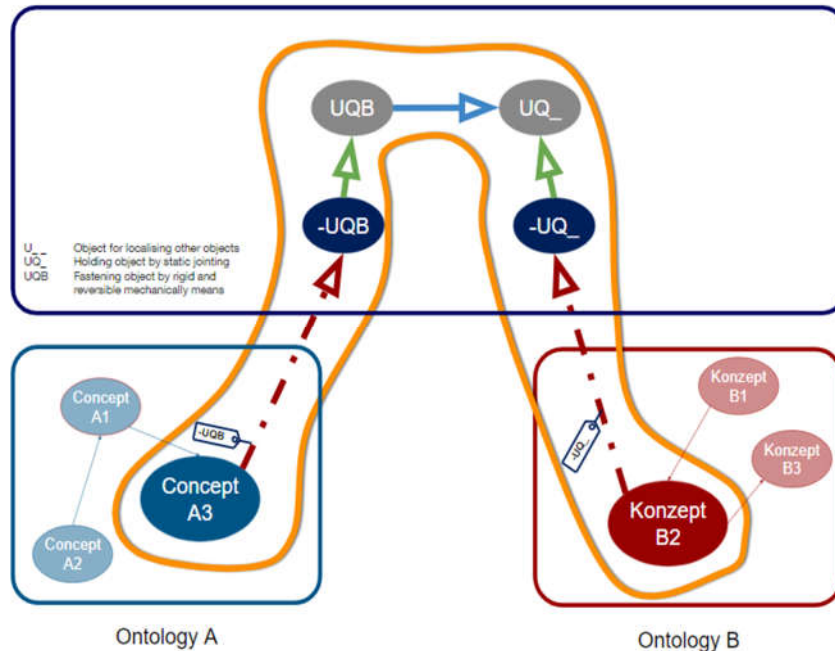


Figure 13. Illustration of Semantic distance between two ontologies: A and B

Semantic scores can be derived from semantic distance and displayed in a compact matrix as shown in Figure 14. Alignments can be quickly detected and agreed between stakeholders to drastically reduce concept duplication. This is powerful to federate models:

- Ontologies and the Common Reference are inserted in the tool
- A matrix provides potential similarities between concepts thanks to scores
- The highest scores are good candidates to be connected
- Ontology authors just confirm similarities to connect their ontologies.

It only takes seconds to connect ontologies even with no modeling expertise.

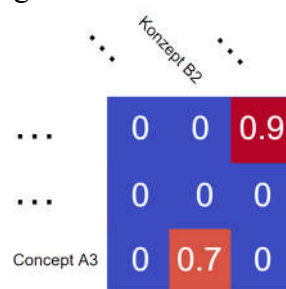


Figure 14. Example of a small semantic score matrix

Up to this point, structured, shared, and indexed meaning exposure can still be seen as a tedious and/or expensive task.

The goal is to query explicit data, wherever it is, based on the component it refers to. Therefore, answers to complex questions can be easily accessible, such as “*what are the masses of the wing*”

subassemblies of aircraft built in 2022 in the Toulouse assembly line?”. Each chunk of this sentence can be attached to common meaning components and relations to make the answer exhaustive, whatever the words used for communication, even with no query language expertise by just answering basic questions. This is performed via a user interface, illustrated in Figure 15.

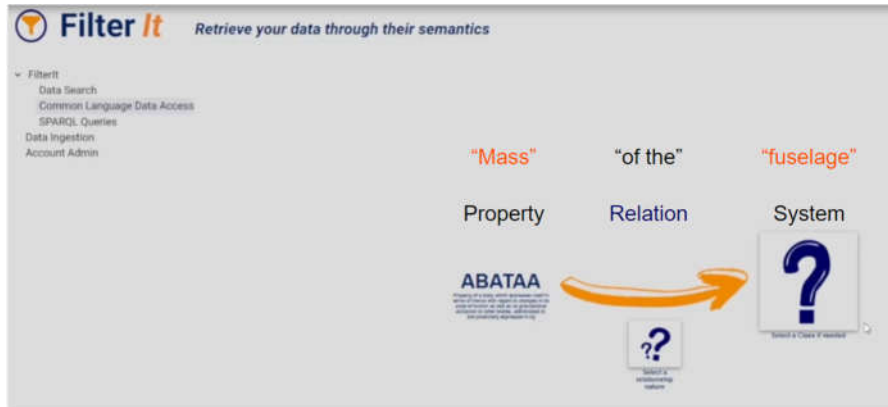


Figure 15. Building of an ACL-based query to gather results from a database.

Conclusion and Future Vision

ISO/IEC 81346 enables the unambiguous identification of components and functions within technical systems for various engineering use cases such as in the aircraft industry. Just one product from AIRBUS, for example an A350 aircraft, may include up to 4 million parts and their associated properties, functions, work orders, jig & tools, suppliers etc. which means about 100 million of tags, which could be created in a year by just a few people by means of ACL based on the RDS technique.

The AIRBUS case has demonstrated that the RDS 81346 technique is a power tool to create, manage and control such a huge and actionable data lake. The major value is associated with the explicit linkages of information between models using the reference model.

That is why the vision for the future is to generalize the use of ACL to clarify and relate the information used at AIRBUS including its extended enterprise applied in a systems of systems reference model. Integrating RDS can present some hurdles such as complexity for large technical systems and resistance to change. However, with proper planning, these hurdles can be overcome, resulting in improved communication within the technical system.

Like a wing without air cannot fly, our models of the aircraft cannot be useful on a large scale without RDS.

References

- Airbus 2021, *Digital Design, Manufacturing & Services*, viewed 3 April 2023, <<https://www.airbus.com/en/innovation/disruptive-concepts/digital-design-manufacturing-services>>.
- Balslev, H 2017, *A Guide to RDS - Reference Designation Systems.: TAG Numbers for Systems in Accordance with the ISO/IEC 81346 Standard Series*, 2nd edition., Danish Standards Foundation, Nordhavn.
- 2015, 'Implementing Model Semantics and a (MB)SE Ontology in Civil Engineering & Construction Sector', *INCOSE International Symposium*, vol. 25, no. 1, pp. 687–696.
- Hicks, B, Culley, S, Gopsill, J & Snider, C 2020, 'Managing complex engineering projects: What can we learn from the evolving digital footprint?', *International Journal of Information Management*, vol. 51, p. 102016.
- INCOSE 2015, *INCOSE Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*, 4th edn, John Wiley & Sons, viewed <<https://www.wiley.com/en-us/INCOSE+Systems+Engineering+Handbook%3A+A+Guide+for+System+Life+Cycle+Processes+and+Activities%2C+4th+Edition-p-9781118999417>>.
- International Organization for Standardization 2009, *ISO 704:2009 Terminology work — Principles and methods*, viewed 20 March 2023, <<https://www.iso.org/standard/38109.html>>.
- 2015, *ISO 12006-2:2015 Building construction — Organization of information about construction works — Part 2: Framework for classification*, viewed <<https://www.iso.org/standard/61753.html>>.
- 2022a, *ISO 81346-10:2022 Industrial systems, installations and equipment and industrial products — Structuring principles and reference designations — Part 10: Power supply systems*, viewed 3 April 2023, <<https://www.iso.org/standard/75471.html>>.
- 2018, *ISO 81346-12:2018 Industrial systems, installations and equipment and industrial products — Structuring principles and reference designations — Part 12: Construction works and building services*, viewed 3 April 2023, <<https://www.iso.org/standard/63886.html>>.
- 2022b, *ISO/IEC 81346-1:2022 Industrial systems, installations and equipment and industrial products — Structuring principles and reference designations — Part 1: Basic rules*, viewed 30 March 2023, <<https://www.iso.org/standard/82229.html>>.
- 2019, *ISO/IEC 81346-2:2019 Industrial systems, installations and equipment and industrial products — Structuring principles and reference designations — Part 2: Classification of objects and codes for classes*, viewed 3 April 2023, <<https://www.iso.org/standard/75265.html>>.
- RDS 81346 Technique ApS 2016, 'RDS 81346 App', *RDS 81346*, viewed 4 April 2023, <<https://www.81346.com/rds-81346-app>>.
- 2021, *RDS 81346 Technique*, viewed 3 April 2023, <<https://www.youtube.com/@RDSTechnique>>.
- READI 2020, *Reference Designation System for Oil and Gas*, viewed 3 April 2023, <<https://readi-jip.org/reference-designation-system-for-oil-and-gas/>>.



Biography – Henrik Balslev

Systems Engineering A/S
Born 1965.

1988 Bachelor of Science in Electrical Engineering.
Technical University of Denmark, Copenhagen

1989 - 2010 Consulting Engineer.
Head of Department, Business development,
Head of large & complex projects.

1998 - Present. International Standardization (ISO & IEC).
Co-author of several international standards with relations to Systems Engineering and documentation.

2013 - 2020 President INCOSE Denmark.
2020 – Present: member of Board of directors INCOSE Denmark.

2010 - Present.
Founder of Systems Engineering A/S
Specialist in Systems Engineering
INCOSE CSEP & ESEP certified.

With a strong technical profile and a personal dedication to create a common language among all technical disciplines, Henrik Balslev is a lead designer of implementing systems awareness in various industries like technical plants, machinery and building design. He is a specialist in system-of-systems, system integration and implementation of Systems Engineering for economic and technical benefits.

Henrik Balslev enables the bridge between practical engineering disciplines and the theory and rules anchored in International Standards (ISO & IEC), some of them of which he acts as the convenor. He is recognized as an international expert in Reference Designation Systems (RDS) in accordance with ISO/IEC 81346 standard series, which supports SE and design of system-of-systems.

As member of the boards of directors and co-founder of INCOSE Denmark, he has a personal dedication to promote systems engineering as an essential tool to Danish engineers and industries, and to promote the Danish profile in the international network of Systems Engineering.

See www.81346.com for more information and free downloads.

Biography – Thomas Barré

Airbus S.A.S.



Thomas Barré is graduated engineer with almost 20 years of experience at Airbus.

Thomas has participated as team leader to the successful design of a wide range of Aircraft types including long range (A350) & military (A400M).

He has created 10 patents which are mostly applied on aircrafts.

About 6 years ago he has joined the Airbus CTO team then the transformation program of Airbus called DDMS (Digital Design, Manufacturing and Service) to develop a universal data identification strategy to clarify and relate information at scale as a key enabler of digital continuity. Practically this strategy is a common language for the aerospace industry based on international standards such as ISO/IEC81346.

As a solution architect he is driving the development of the supporting software, governance and methods while testing and promoting this language inside and outside Airbus.

Looking for more information about this language:
Mail to: common.language@airbus.com