34th Annual INCOSE international symposium
hybrid event
Dublin, Ireland
July 2 - 6, 2024

An ontology example in Configuration Management at Airbus

Balachandar Ramachandiya Amarnath
Airbus
Toulouse, France
+33 628113085
balachandar.amarnath@airbus.com

Adriana D'Souza
Airbus
Bristol, UK
+44 7793905547
adriana.dsouza@airbus.com

Bernd Podey
Airbus
Hamburg, Germany
+49 1607167661
bernd.podey@airbus.com

Javier Reines
Airbus
Madrid, Spain
+34 652992423
francisco.reines@airbus.com

Moises Ramon Martinez
Airbus
Toulouse, France
+33633714690
moises.martinez@airbus.com

Abstract. Today data is increasingly available in our engineering domains but without an ontology to structure it, one might get drowned in the data lake. Configuration Management data being central to maintaining the configuration of the aircraft throughout its life cycle, need to be well described, understood and interpreted appropriately. This paper presents the effort performed at Airbus in the configuration management domain to structure the data through abstract ontology models of processes, tools and workflows. The ontology objects are then implemented and exposed as a data product to be used in digital transformation initiatives. This ontology based approach has encouraged harmonization of digital initiatives across different aircraft programs, and the associated graph style implementation ensures a more efficient exploration of Configuration Management data, leading to shorter lead time for development of data driven initiatives and knowledge ramp up.

Introduction

Configuration Management (CM) is a key discipline to manage complexity and changes for a product development from specification, design and all the way to in-service and disposal.

Configuration management is there to make sure we know what we are doing at each step of the way and to be able to reproduce our products.

Configuration management has its origin in a legend where an industry prototyped a missile and ran a customer demonstration where the missile hit the target. The customer was pleased and wanted to order 100 more. The industry had the following dilemma:

- the prototype was expended
- the technical publication didn't reflect the build
- the documentation didn't keep up with the changes

…a second success could not be guaranteed.

CM at Airbus has a rich history dating from the 1980s, and has been implemented in heterogeneous manner across Product Lifecycle Management (PLM) systems. This implementation was adapted to different regulatory, engineering, production or maintenance needs. CM is everywhere, and needs simplification. It requires flexibility and modularity to adapt to business needs, while ensuring language, methods and rules are consistent across all programs and disciplines.

CM fosters the business knowledge capitalization and provides powerful simulation, prediction, prescription and exploration tools to assist the users in decision making and help them focus on added value work. However, due to the history of various product data management systems and CM paradigms used across different programs, the CM data are often not understandable, or misinterpreted due to the lack of a deep knowledge on the history of its creation.

At Airbus, there are lots of great initiatives for data consolidation and in making data widely available (Skywise 2023). However, very minimal attention has been given in these initiatives towards ensuring understandability of data and providing a unified view of data, independent of the tools and processes, and across multiple programs. As a consequence, there is an immense amount of data available, but it is not exploited in an efficient way so as to extract maximum business value out of it.

The major limitations that warrants attentions are:

- Too much data - *Exposure of technical raw tables for a Single Aisle program which is about 40 years old (close to 4000) across many different data platforms.*
- Same information in different formats - *As process matures or evolves, the underlying schemas of the tables are often modified, that in turn leads to different formats over the years*.
- Raw data not understandable by business users - *Risk of wrong interpretation of the data due to lack of knowledge about the process and data description. This risk increases because of multiple usage and sources.*
- As multiple sources evolve, establishing a single source of truth and ensuring the veracity of the data is always a challenge.

Hence, it is imperative to ensure the processes and data involved in every step of the configuration management is well described and understood, not only by the experts but also by the information Management community that strives to derive business value from it.

Airbus CM Community has invested significant efforts in building ontology for various CM Processes, describing business objects involved in them. In addition, to ensure practical usage, datasets conformed to the description of the objects are being built and exposed as a data product across Airbus data platform. This approach opens up plenty of opportunities for technologies to exploit them in a much (c)leaner way.

In the rest of the paper, the key concepts behind CM ontology and data products are explained. To demonstrate further the concepts, two key CM processes have been considered, and the CM ontology model built for these processes are explained briefly. Neo4j, a graph database management system, has been proposed to explore the resultant ontology model, and potential benefits are discussed. In addition, the paradigm shifts towards data product based development of tools and analytics dashboards are highlighted. In the end, the importance of harmonization across different programs are also presented, which in turn would help the company to ramp-up production and management of our aircrafts.

Background

In this section, key concepts that have inspired the CM ontology initiative in Airbus are presented. First and foremost is Ontology and its role in giving sense to the data.

Several processes are defined and implemented in Configuration Management for Airbus. These processes involve complex workflows that often flow back and forth. There are many tools involved that are being used by a diverse population within the company to operate the process, and thus generating very diverse data at a very rapid pace in multiple platforms. A data driven tool built with wrongly interpreted data can lead to wrong decisions and hence cannot be trusted. Hence, it is very important to understand and describe the processes, their workflows and associated data before developing data driven tools.

Ontology (L.Ding, 2007 and Sharmanet R, 2007) is a formal explicit description of concepts in a domain. It essentially defines the structure of knowledge for various domains. In the business domain context, one way to see it is that the nouns represent classes of objects and the verbs represent relation between objects. Web Ontology Language (OWL) (Horrocks I, 2003) is introduced as a semantic web language to represent rich and complex knowledge about things, groups of things, and relations between things. OWL is part of the W3C's semantic web technology stack (W3C, 2004).

For the ontology implementation, Protege software (Musen M.A, 2015) is the ontology editor and framework that was used. Protege software provides a graphical user interface to define the ontology schema and its populations called individuals. It also includes deductive classifiers to validate that working ontologies are consistent. It possibly infers new information based on the analysis of the existing ontology.

In this literature (Amarnath B, 2009), it was demonstrated that Ontology can not only be used in health care and travel industries, but also be exploited very well to describe information technology infrastructure domain. After all, Ontology is about expressing the domain in a meaningful way. Similarly, ontology was well used for describing configuration management in the computing domain.

The concept of ”data product” is the idea of treating data like a product (DJ Patil, 2012). It was rightly defined as the application of a unique blend of skills from analytics, engineering & communication aiming at generating value from the data itself to provide benefit to another entity.

Unlike in all this literature, the challenge that we need to address is two fold, that is, giving sense to the Configuration Management Data and ensuring its consistent interpretation and usage in applications. Hence, we proposed to deal with Ontology and Data Products together such that ontology is used to define business objects in an abstract manner and relations between them, whereas data products were built to the schemas defined in the ontology objects. This approach allowed the authors to explain the CM process and workflow through interconnected business objects while data products allowed the entire community to promote consistent usage of CM data in their applications.

The next sections explain the concept of Ontology based data products for CM, and illustration of its benefits through few use cases.

Configuration Management Ontology

Configuration Management Ontology (CM Ontology) in Airbus is based on an Ontology defined for this domain. Significant efforts went in developing CM ontology wherein the entities involved in CM have been described using common language. This allows anyone who does not possess any knowledge of the CM process to understand CM data without any ambiguity. The CM Ontology proposed three high level notions, and suggested every other CM entity fall within these three notions.

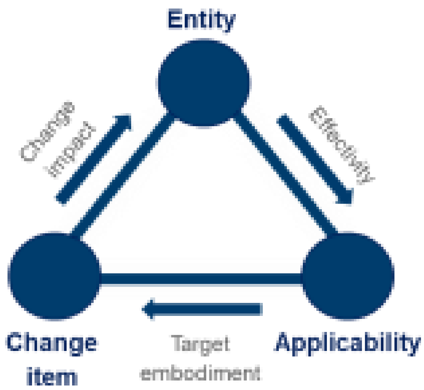

Figure 1. High level CM Ontology

CM Ontology was the first step towards describing CM domain knowledge in a meaningful and understandable way. With just three classes, it was too abstract and difficult to connect with the underlying real data. Hence, there was always a need to cascade this abstractness down to the real data. One way to achieve this is to cascade down the abstractness through intermediate core ontology comprising high level business objects.

Business Objects are not new to the industries as there are several business objects already built. They provide some level of abstraction but their primary purpose is to expose the data aggregated across several tables from the underlying data source. These objects are predominantly used for providing a unified view of a particular “topic” (e.g.: Modification, MOD in short) quickly. They are closer to the data as their descriptions follow the naming and description convention determined by the process and tools handling the data. As a consequence, they are essentially *data driven objects* and need to be rebuilt if the process and tool changes.

Imagine, if these business objects are abstracted to a higher level that are independent with processes and tools, following business friendly naming conventions, and that encapsulates the scope of the underlying business objects. It would be then possible to realize a layer of high level business objects that are easily understandable to the transversal communities, and the one that connects the data driven objects with the CM Ontology. This thought process has led to the proposal of Configuration Management Data Product. (CM Data Product)

Configuration Management Data Product

CM Data Product can be formally defined as a set of derived datasets built out of abstract modeling of processes with high level business objects, either standalone or linked with another, that satisfy certain needs of the business. Building a data product in CM starts with defining respective ontology object(s) and ends with publishing corresponding dataset(s) for the object(s) in multiple Airbus data platforms. Defining a data product is not simple as it could lead to several misunderstandings. This is because of the fact that there is already a notion of business objects in the industry. These business objects are essentially data driven, that is, they were built from the data and NOT from the business perspective. One can find these business objects in various Airbus data platforms.

The CM data Product, on the other hand, proposes to create business objects from the business perspective. For example, a change can be triggered either through product improvement or customisation. In this example illustrated in Figure 2, these two possibilities were abstracted into a high level object called "Change Trigger", and two sub-objects "Product Change" and "Product Customisation" were created. The "Change Trigger" can have common attributes of its sub-objects. The "Product Change" shall be built using the existing data driven objects such as Change Request (CR), whereas the "Product Customisation" object shall be built using Request for Customisation (RFC). If such data driven business objects are not existing already, one can always build the abstract objects directly from the raw tables.

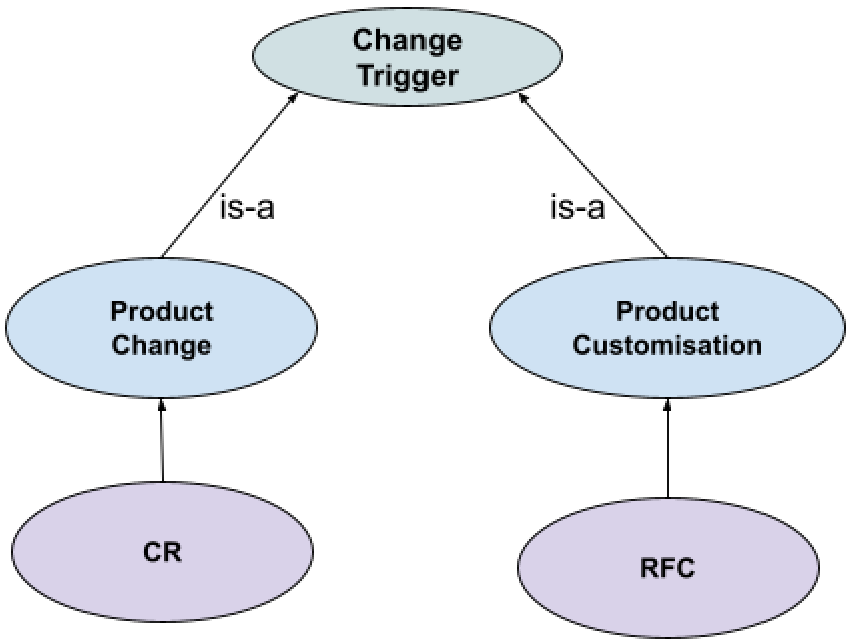

Figure 2. Example business object derivation

Architecture

Elaborating the definition of the CM Data Product, a multi layered architecture has been proposed for its implementation as shown in Figure 3. The purpose of the architecture is to clearly identify the different levels of abstraction starting from the CM Ontology down to the real data. This will give an idea on where different types of objects are positioned and how they are linked to other objects in the same layer, and also with other layers.

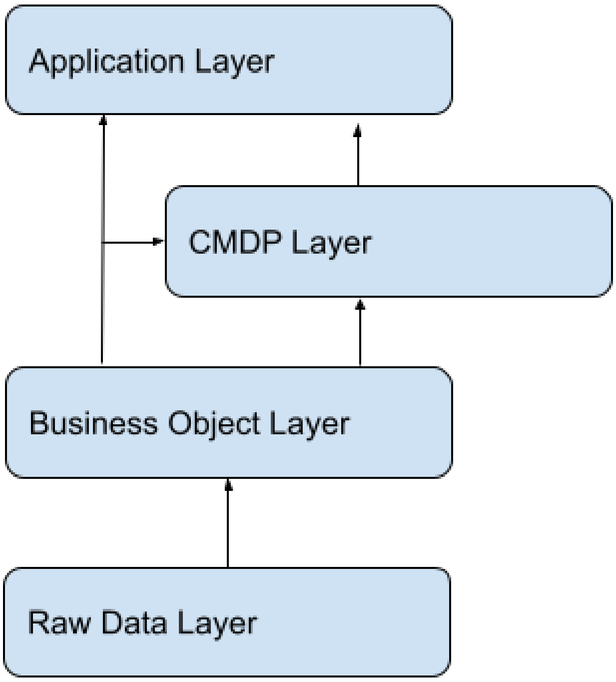

Figure 3. CM Ontology layers

Raw Data Layer

This layer refers to the raw data sources from where the data is consumed by the upper objects. Ideally, the data should be extracted from the source (e.g. Legacy tools or Authoring Tools). However, if it is not possible, then the next most reliable source should be selected (e.g. Company data warehouse). The data values shall be consumed and/or up-cascaded to the upper layers as is. In other cases, the values are suitably transformed based on the requirements of the objects in the upper layer.

Business Objects Layer

This layer refers to the already existing business objects or business objects to be created from the underlying tables. It requires significant effort to understand the description of the raw data and its location in the sources. In most of the cases, applications and/or the users need certain key attributes from different tables by establishing suitable links between those tables. Hence, business entities have already built ready-to-use objects that are essentially an abstraction of the raw data. The purpose of such objects is to simplify the access and exposure of the raw data.

The abstraction was modeled from the description of the data and hence it can be called data driven objects. When the process or tool changes, they need to be rebuilt from the scratch, as they are tightly linked with it in terms of attributes naming and means to access the data from the tool. This layer can be multi-layered in itself wherein a business object is created from more than one business objects, that are in turn created from the underlying raw data tables. As they are not insulated to the changes in the processes and tools, they cannot be considered to be the objects in the CM Data Product (CMDP) layer but nevertheless serve as a data product to the end users or applications.

CM Data Product layer

When the underlying business objects are not abstracted enough to be understandable to the business, and that the description is not independent to the language of the tools and processes, then, one has to create more abstract objects to represent the underlying business objects. The abstract business objects are positioned in the CMDP layer and are typically delivered as a data product to the end users and applications. The objects in this layer are built from objects from the underlying business object layer. When the tools and processes change, the objects in the CM Data Product layer will not be impacted, as they rely on the business object underneath and is not created from raw data directly. In addition, the schema definition of these objects are kept agnostic to various aircraft programs. This enables building cross program applications, eliminating redundant effort in maintaining a different data pipeline for each aircraft program.

Characteristics

In order to ensure consistency and understandability of the data, CM Data Products are defined with some key characteristics. They are:

Well documented. This is fundamental and most important. The root cause of ambiguity and mis interpretation of the data is the lack of documentation. Hence, high level business objects intended to be published in the CMDP layer should be well documented. It shall follow high level naming convention while modeling its attributes. These attributes should include documentation on its purpose, type, and examples on what kind of values they can contain, history of creation and its last update. The object should be documented with respect to how it is built. For instance, this part should contain information on transformation logic, and the underlying objects that form the input to create it. These documentation should be accessible from the environment in which the model is viewed.

Extendable and Re-usable. CM is transversal. This means that objects used in one CM process will be needed in the workflow of other processes. Hence, objects in the CMDP layer should be re-usable at best, and hence be able to form a link with other objects in other CM processes and outside of CM too. Having said this, it can be difficult to satisfy or anticipate the data needs of an object beforehand. As a consequence, the objects in the CMDP layer may not contain all the attributes required by a CM process or an application. Hence, there will be a requirement of new additional attributes in the object. In this case, existing objects should be extended to address new data needs rather than re-creating similar objects from scratch. This will avoid duplication of data and objects.

"Pull" or "Push" model. Modeling activity is an ever evolving task. To start the creation of CM Data Product objects, one can use the popular use cases or popular objects that are widely used by the business as a guidance. This is called the "Pull" model. There can be cases when CM process owners create objects anticipating certain needs of the business, or willing to drive a transformation. In this case, objects can be created by the process owners and promoted to the business. This is the "Push" model.

Good level of Abstraction. The objects in the CMDP layer shall exhibit a good level of abstraction with respect to the underlying processes and tools. One should be able to obtain at least the overview of an object throughout its evolution with the processes and tools, and across multiple programs.

Consistent with data sources. Currently, there are cases where the same data is available in multiple business objects created from multiple data sources. These objects were not modeled with the CM process owners in the loop. Hence, at times, it becomes very difficult to validate the data values seen in these objects. To address this problem, objects in the CMDP layer need to be created using the data available close to the source, and modeled in close consultation with the respective process owners. These objects shall then be deployed in Airbus wide infrastructure where there is a need or a need is foreseen.

Benefits

Some of the benefits seen through the implementation of CM Data Product are listed below:

- *Impact analysis*: Perform an analysis using the CM Data Product to ensure applications are not affected by the process change. It also supports cross program analysis, that is, to develop analytical solutions for multiple programs by using one abstracted model.
- *Consistency*: Data Product ensures applications and users always work with CM data that is consistent with the data sources.
- *Validation*: Modelling and Abstraction were performed in close collaboration with Process Owners to ensure validation of the model and the datasets associated with the objects.
- *Accessibility*: In the long term, it reduces effort spent in accessing CM data. Further, it has the potential to be synchronized with other similar initiatives to encapsulate objects of those initiatives in the CMDP layer.
- *Reporting*: Applications used for business reports can be modified to avoid working with the real data for every user request, but instead re-using the abstracted business objects in the data product layer for building the reports and KPIs. This will significantly reduce human efforts.

CM Ontology Modeling Process

Defining ontology for CM itself involves several challenges as the process has to be revisited once again, and its evolution in the future needs to be brainstormed. This will help to build abstract objects with more abstract naming. In the current situation, the data model has been derived from the existing data without much involvement from the process owner community. Hence, several derived datasets were built through linking several raw tables through primary and foreign keys. Such links are purely technical and follow a bottom up approach and more importantly, these links do not capture the semantic relationship between the objects involved in. Hence, it was decided to approach the modeling from the top down perspective, starting with the process flow. A group of respective process owners and experts were involved, brainstormed and thus supported the modeling of respective processes.

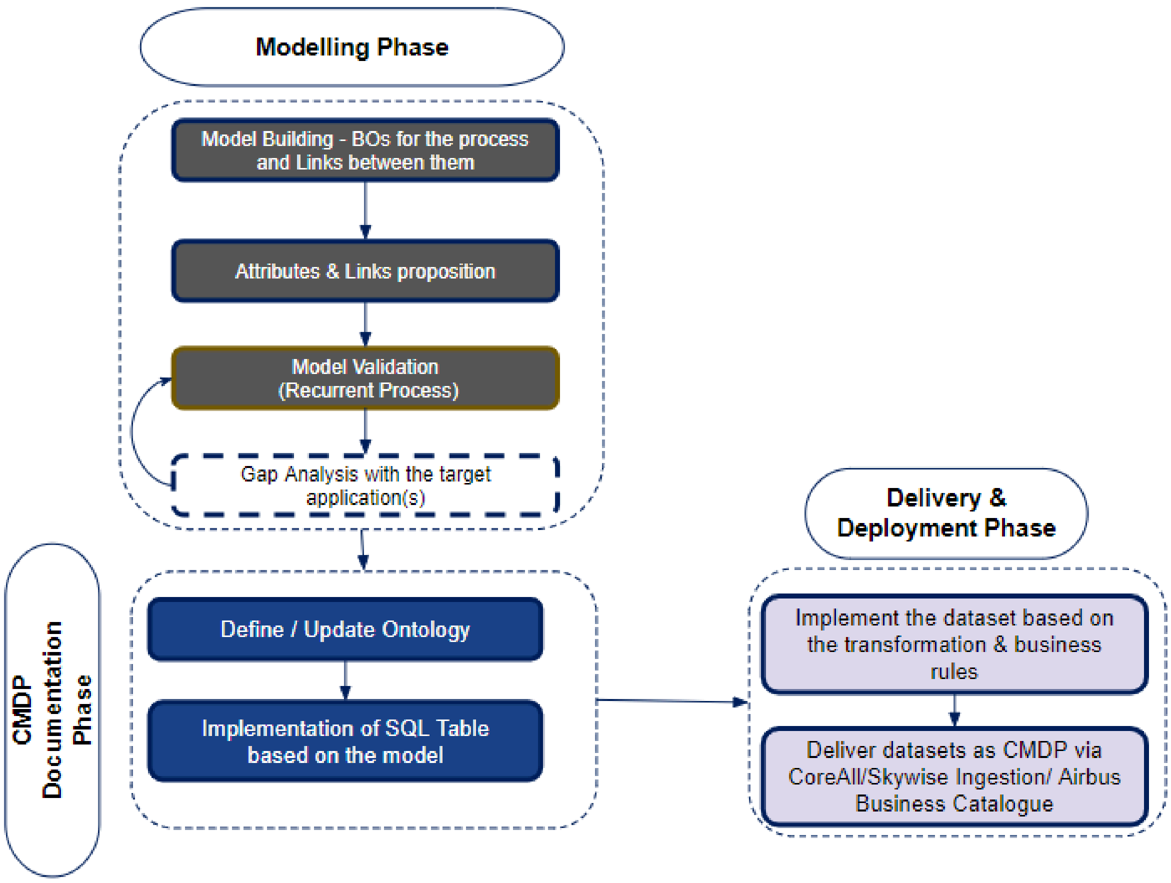

Figure 4. CM Data Product Modeling Process

Figure 4 above shows various steps involved in the CM Data Product modeling process. The first step is obviously the modeling that is performed with the support of process owners. A series of questions are asked to the process owners to trigger the thought process towards abstract modeling of the process. As it is based on a top-down approach, the key business objects are listed around. It shall be noted that the configuration management process is largely common across various aircraft programs, which allows harmonizing the naming of the business objects at this stage.

In some cases, an abstract but more human-friendly name is assigned to the business object to make the model more understandable to wider audiences. For instance, in the change process, two key objects namely "product change" and "customisation change" are loosely modeled as CR and RFC respectively in the BO layer. "Product change" usually refers to the changes initiated by the product evolution whereas "customisation change" refers to changes requested by the customers, especially towards customisation of certain elements. Both these are considered as a trigger to initiate the change process in Airbus. Hence, an abstract object called Change Trigger has been proposed that encapsulates the common properties of "product changes" and "customisation changes".

In modeling terminologies, both Product change and Product Customization are subclasses of Change Trigger, and they are siblings to each other.

Next step is to understand and model the links (a.k.a. relationships) between those objects. In any process, the workflow is initiated with an object, and in the subsequent steps these objects are enriched with additional information and several other objects are created. Links are essentially a part of the process workflow that connects objects in the process, and also connects objects existing in the other processes. For instance, the change requests triggered by "product change" or "customisation change" are studied in detail to understand the impacts and thus several analysis documents are created. The steering committee reviews the documents and makes decisions on implementing the changes in the aircraft. At this stage, a new object called Modification (MOD) is created and then tracked during the entire lifecycle through design, implementation in the aircraft and certification with the appropriate Airworthiness authorities.

Such kinds of descriptions are available only in the documentation, but they are not captured in the data model underneath. To address this limitation, the objects in CMDP are connected via semantic links proposed by Process Owners. These links are named ones and sometimes, two objects are connected through more than one link depending on the context. An example of linked objects within change process is shown in the figure below

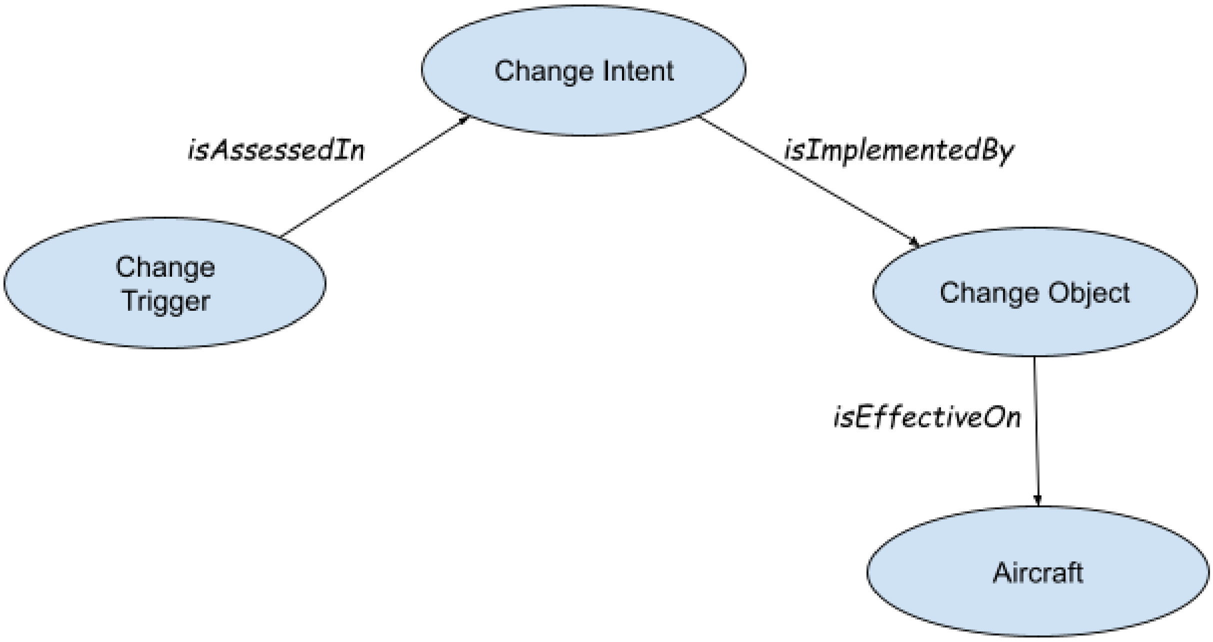

Figure 5. CMDP example of semantic links

Importance of Links and Linked Objects

When correctly designed, Links can connect literally all the processes, associated objects and underlying data, creating a connected graph of objects. When links are assigned a human friendly naming semantically, the graph of objects becomes a knowledge graph. More ambitiously, when all the data objects in Airbus are modeled using this approach, we will have an End-to-End connected data landscape which can enable building data driven tools that can understand the downstream impact when making a minor change in the upstream objects. With this intention, every object in the process is linked with other objects via one or multiple named links.

By reading the modeling diagram in the form of a graph, one can understand the workflow of the processes to a great extent, and thus be given an idea of how and when to use these objects. In addition, the link itself is implemented as a dataset that contains reference of source and target objects. Such link datasets will help to join multiple objects in the process to eventually build a flat table for exploring and solving certain business cases.

Documentation of the Ontology

Once the objects and associated links are proposed, modeling itself is performed in Protege ontology editor. Describing objects and links in Ontology enables the usage of semantic query languages such as SPARQL or SWRL, and inference engines such as Algernon to query the ontology model. This is ambitious at this stage due to scope limitations, but once extended for all the Airbus domains, this will be considered as a knowledge base to query key information about any aspects of documentation, data, processes, tools and aircrafts. This will significantly help Airbus to quickly ramp up newcomers in the company, and expose to them a huge repository of knowledge base built over the years. The figure below shows the ontology visual representation of Change Process in Protege Ontology Editor.

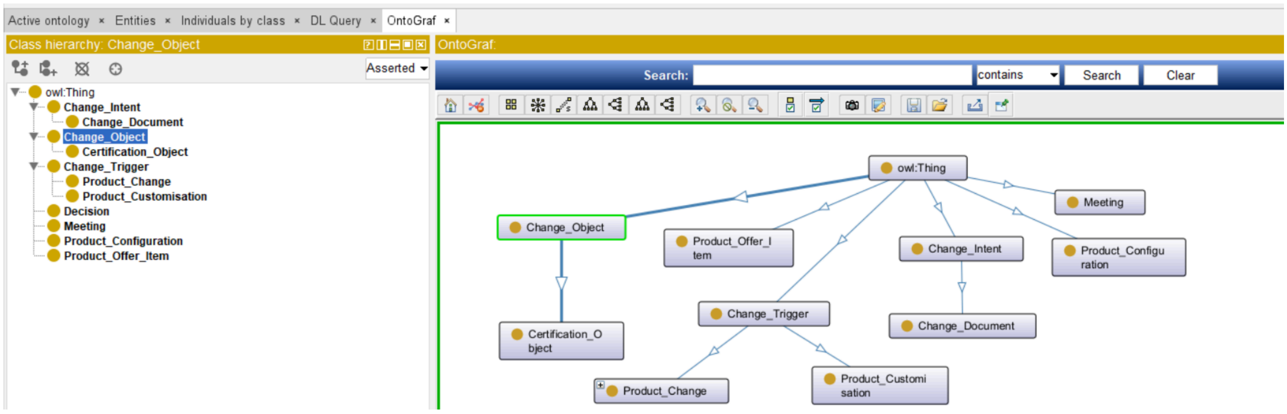

Figure 6. CM Ontology representation of Change Process in Protege Ontology Editor

Implementation

With the concepts and approach described in the earlier sections, the CM Data Product initiative has already built ontology based data products for various CM processes, namely Change Control, ConfigurationIdentification, Product Offer, and Status Accounting. In this section, ontology defined for two key processes are presented: Change Process and Product Structure Identification. Here, all the major business objects of these two processes, the links between them, most importantly, the object(s) from Change Process that interfaces with another object from Product Structure Identification are presented. This interface link is the key that establishes a connection at object level between two processes.

Such interface links across all the processes will ensure navigation of Ontologies of multiple processes seamlessly as a connected knowledge graph. This graph based representation also allows the use of graph based tools such as Neo4j (Neo4j, 2012) to explore and query various aspects of CM processes. For example, one can quickly extract various ways to connect a design solution and a change request, which in turn allows one to understand the impact on the product structure identification team, when implementing a new change request from the customer. Such detection of impacts at the early stage is a must for better planning and timely delivery of the aircraft. Hence, the interface links are key to achieve the intended objective.

Modeling Process

In the modeling phase, we capture key business objects in the change process and product structure identification processes based on a story-telling approach.

Changes can either be initiated by "*product change"* or from the "*product customization*" from the customer request. These two elements are considered as a trigger to the change process. A detailed assessment is then made to describe the intent of the requested change and its impacts. This assessed change is further processed through several meetings and a final decision is made on the implementation of the change. At each of these stages, the initial request is enriched with additional information and results in new objects in the process. When the change request is approved, a *modification* is created, for execution to the engineering and its subsequent implementation in the right aircraft. Once implemented, the modification goes through the *certification* process. The figure below shows the abstract business objects defined for the change process. The change trigger encapsulates the "product changes" and "customisation changes". The modification is represented into two objects namely Change Object which refers to the changes to be made in the aircraft, and Certification Object which refers to the details needed for the certification process.

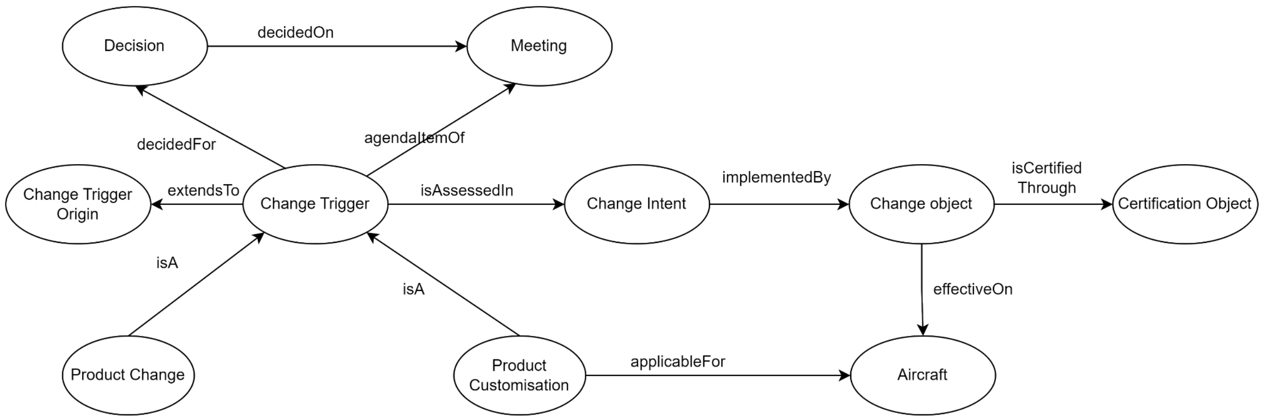

Figure 7. CM Ontology business objects for Change Process

The attributes for these objects are not shown due to confidentiality constraints. Further, it has been decided to create extended objects for the given business object when needed. For instance, the object *ChangeTrigger_Origin* has been proposed to define origin and its related information for the *change trigger* object. Such split and extension shall be discussed with the process owners and experts. In general, as the object grows bigger, it is advisable to split it appropriately.

Continuing the story-telling of the process, once modification is approved, it is then handled by the engineering and design teams. The change object specifies the modifications to be performed through the Configuration Items (CI). These are then *solved* by *design solution*s (DS) proposed by the design teams and these solutions are at the atomic level of the modification. For example, *installation of a light in the cabin* refers to a CI whereas this can be done through several implementations depending on the kind of light requested. Hence, a CI will have multiple DS-es to it. A CI is then linked to a design solution through Link Object (LO) which specifies the position and coordinates of the installation location of the DS for the given aircraft. This process workflow is captured in the image below

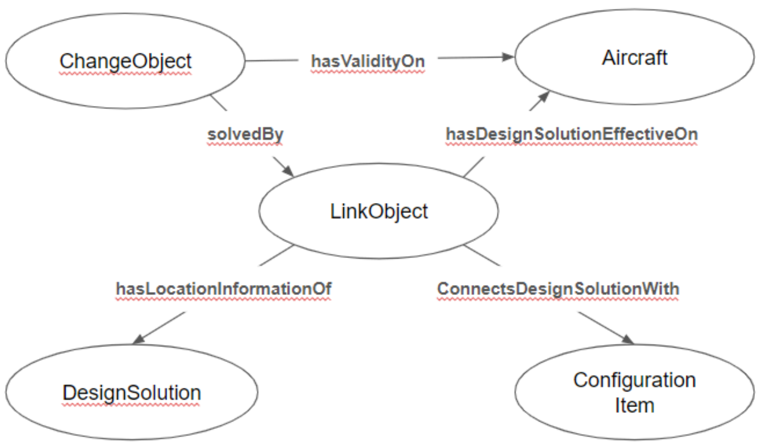

Figure 8. CM Ontology business layer

Graph based exploration

Configuration Management is a connected world. Understanding it requires processing rich sets of connections between the business objects involved in it. Often, the connections between objects are as important as the items themselves. Hence, a graph database system was chosen to explore the CM ontology. A graph database stores nodes and relationships instead of tables, or documents. Data is stored just like ideas sketched on a whiteboard. The data is stored without restricting it to a predefined model, allowing a very flexible way of thinking about and using it. Neo4j is one such graph database management system.

The data elements Neo4j stores are nodes, edges connecting them, and attributes of nodes and edges. It offers a query language with which one can navigate the graph and perform all kinds of graph exploration such as all paths between two objects, shortest path and much more. Here, the CM ontology objects are implemented as nodes and the links between them are implemented as edges. The following image shows the Neo4j query in action over the CM ontology that shows the paths between two objects namely *ChangeTrigger* and *Decision.* With this, one can easily compose the transformation to build the dataset that contains information about change requests and corresponding decisions taken on them.

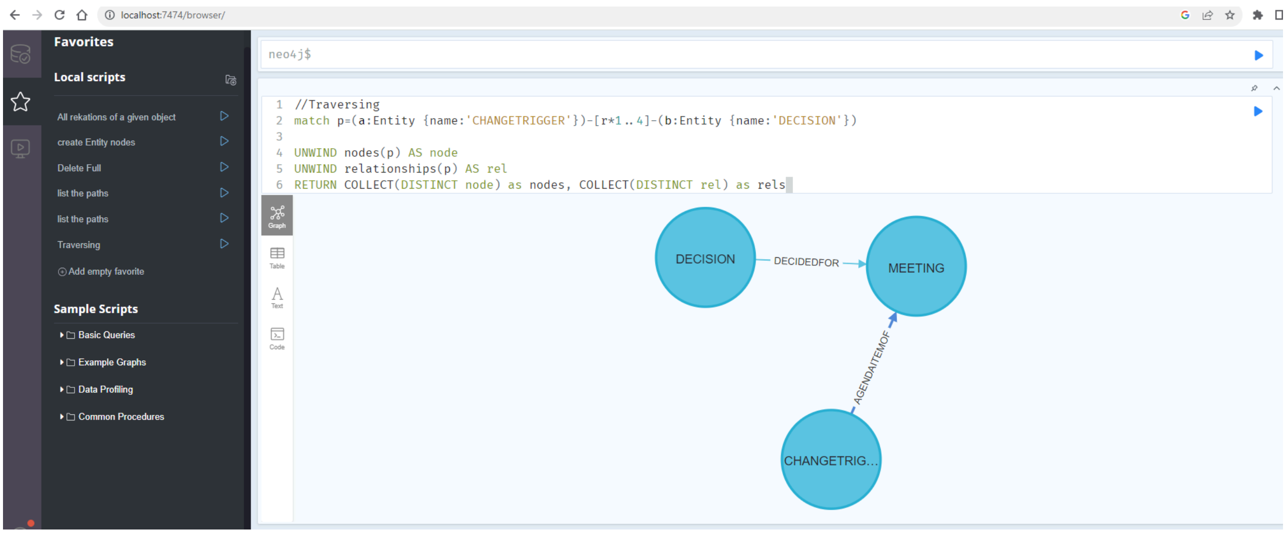

Figure 9. Neo4J Graph exploration

In the same spirit, a flat dataset can be constructed that contains the link between change request from change process and design solution linked with those change requests. Building such a linked dataset is otherwise quite complex looking into plenty of raw tables spending a significant amount of time. With ontology based datasets, such complex transformations are centralized and well documented enabling applications and dashboards to simply plug-and-play. Figure 10 below shows the portion of CM Ontology that contains objects from change process, offer and product structure identification. Exploring it in Noe4j allows us to quickly find various paths or ways to link objects from one domain with that of another. For example, in the image below, the link between Change Trigger and Design solution is highlighted.

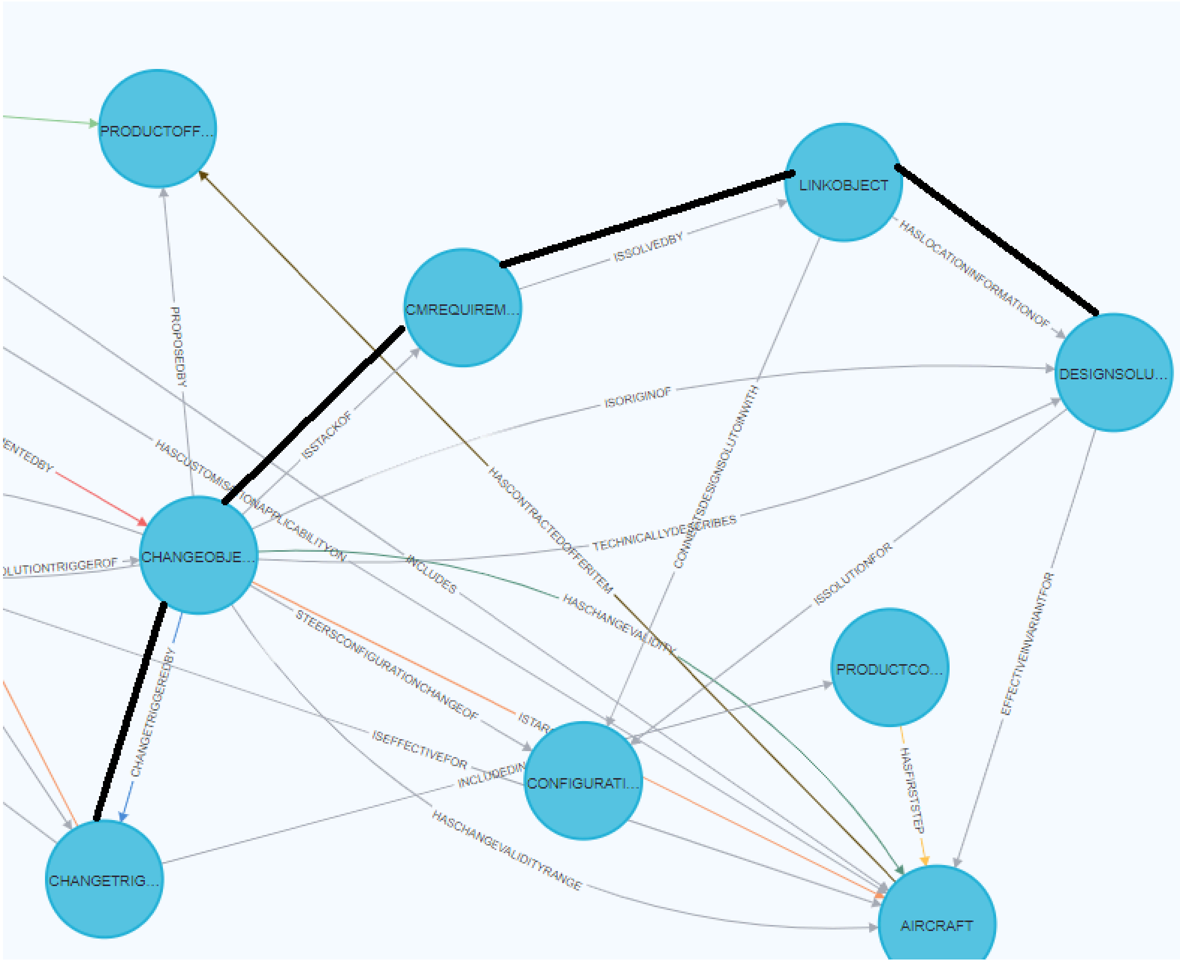

Figure 10. CM Ontology business layer extract

Such exploration allows quickly to understand the impact of downstream objects when making a change upstream. Alternatively, one can also find the root cause elements of a manufacturing issue encountered in the final assembly line.

Opportunities and Way Forward

Abstract ontology objects for Configuration Management has been an inspiration for similar initiatives in other domains in Airbus. With the aircraft programs dating back to the 1980s and an average lifecycle of an aircraft standing at 30 years, the configuration of the aircraft needs to be managed for the entire lifecycle of a program until the retirement of the last aircraft in service. Processes, technologies and tools have evolved so much since the beginning that the aggregation of configuration data of an entire period, that it poses huge challenges, consumes a significant amount of time and is susceptible to human error. In parallel, data is considered as an asset, and data driven insights creates new business opportunities in various streams, such as providing new and improved customer services, maintenance and repair overhaul, and improved time-to-market services. Many of these streams rely on heavy analysis of aircraft configuration data. The CM Ontology initiative not only proposes an abstract object modeling of the processes but also builds datasets encapsulating complex business rules and data transformations. These ontology based datasets are exposed as data products that enable the development of data driven initiatives in a rapid manner and ensure the information presented in those tools are consistent and accurate. Some of the key benefits of this approach are listed in the next sections.

Cross Program mindset

First and foremost of all the positive outcomes of CM Ontology and CM Data Product is the cross program mindset. Until recently, various aircraft programs initiated several digitalisation initiatives without synchronization due to their own priority and market. This has resulted in redundant development and maintenance efforts from the information management department. Another technical reason is that the legacy data is managed in different servers with different schemas per program.

The CM Ontology approach delivered a program agnostic model with common schemas across all programs. It has harmonized the variations with respect to tools, processes and raw data at the abstract object level. Building datasets by performing necessary transformations from the underlying data ensures the business objects are the same for all programs. Hence, it was possible to build an application that works for all programs as the program specific calculations are done at the data level. Looking at the real benefits of cost reduction, a cross-program digitalisation forum is being organized regularly wherein requirements are gathered, brainstormed for the calculation rules, and developed once for all programs.

As a byproduct of such a forum, the program community is now also sharing best practices with their counterparts of other programs. One such cross-program initiative is *End-to-End Modification control* which aims to centralize follow up activities of all objects such as Change Request, Modification and Request for Customisation under one major data driven tool with CM Ontology based data product as its backbone. About 57 business objects and 48 linked objects have been created and exposed as data products that support this initiative.

Alignment with Major Initiatives

The data product based application development approach has been accepted at the enterprise level and hence very well integrated with one of Airbus major initiatives, the *Digital Design Manufacturing and Services* (DDMS 2023*)*. It has been targeted to build 100 data products spanning across several processes and domains of Airbus to be deployed in the latest state-of-the-art PLM infrastructure. Two major value streams have been created with clear focus on building ontology based models and also data products. The CM Ontology approach is now leading in the front as it not only defines but also proposes an approach to connect the objects with the low level data and demonstrates the benefits through graph based exploration capabilities. In the coming years, Airbus will have deployed ontology models for many of its domains due to the strong buy-in from DDMS.

Conclusion

There is a huge emphasis on data and knowledge capitalisation across industries since the last decade. Airbus understands that and has been investing heavily in providing necessary training to all employees in data analysis and data science domain. This has encouraged many of our business experts to start analyzing their data, build a dashboard that improves their productivity in their daily operations. However, they lacked visibility of other domains, mainly due to the lack of understanding of other processes and associated data landscape. CM Ontology has addressed these limitations through establishing the links between various processes, and is well documented. Ontology alone cannot provide real benefits to the customers. Hence, datasets were built based on the abstract objects and exposed as official data products to be consumed by data driven solutions.

The CM Ontology approach has inspired several similar initiatives across various other processes within Airbus. These days, any major initiative starts with building ontology models and associated data products. Airbus big data platform is now equipped with advanced tools that automatically extract key KPIs, and insights straight out of the data products, showcasing potential business values. In some cases, errors in the data are also identified which allows them to fix it at a very early stage. Rapid prototyping before full blown application development ensures the development cycle is small and applications deliver intended business value right-at-first-time.

Ontology development and management is a continuous and iterative process. Airbus clearly sees that the knowledge capitalisation and data products are key to create new business opportunities, to provide state-of-the-art data services, anticipate disruptions and act proactively in this highly competitive and volatile market. With this in mind, Airbus continues its journey in digital transformation with incremental innovation every step of the way, and ontology is just one but a key enabler.

References

Amarnath, B. et al 2009, ’Ontology-based Grid resource management. Software Practice Experience’. 39. 1419-1438. 10.1002/spe.942.

DDMS 2023, Digital Design, Manufacturing & Services (DDMS) last accessed Sep 2023 <https://www.airbus.com/en/innovation/disruptive-concepts/digital-design-manufacturing-services>

Ding. L. et al , ‘Using ontologies in the semantic Web: A survey’. Part of the Integrated Series in Information Systems book series (ISIS,volume 14)

Gruber, T. 1993, ‘A translation approach to portable ontology specifications’ Knowledge Acquisition Volume 5, Issue 2, June 1993, Pages 199-220

Horrocks, I.et al. 2013, ‘From *SHIQ* and RDF to OWL: the making of a web ontology language’. J. Web Semant., 1(1):7–26, 2003

Musen, M.A. 2015, ‘The Protégé project: A look back and a look forward. AI Matters’. Association of Computing Machinery Specific Interest Group in Artificial Intelligence, 1(4), June 2015. DOI: 10.1145/2557001.25757003

Neo4j 2012, .Database graph social. Neo4j- The World’s Leading Graph Database. <http://neo4j.org>

Patil, DJ 2012. ‘Data Jujitsu: The Art of Turning Data Into Product’. O’Reilly Media, Incorporated, 2012

Sharmanet, R. et al. 2007, ‘*Ontologies: A Handbook of Principles, Concepts and Applications in Information Systems’*. New York: Springer, 2007, pp. 79–113. doi: 10.1007/978-0-387-37022-4_4

Skywise 2023, Last accessed Sep 2023 < https://aircraft.airbus.com/en/services/enhance/skywise>

W3C 2004, World Wide Web Consortium (W3C). Web Ontology Language (OWL). W3C Recommendation. Available at: http://www.w3.org/2004/OWL/.

Biography

Balachandar Ramachandiya Amarnath. has more than 13 years of experience in Digitalisation initiatives in Airbus. He has a strong working experience in the broad domain of High Performance and Distributed computing, and also in Big data domain. He received his PhD from Madras Institute of Technology (India) in the field of Ontology and Grid Computing. Having transitioned to Configuration Management six years ago, he is working on building a consistent data model for Configuration Management and data products to ensure the process knowledge is well captured and understood.

Adriana D'Souza (CSEP) is a Configuration Management Process Architect for Systems for Airbus, having previously worked as a Systems Architect and as a Design and Development Engineer on challenging and complex projects like large border security projects and air traffic management projects both at the subsystem, system and system of systems level in the Airbus Group. She was awarded an Honours M.Sc. in Computational Science and Engineering from the Technical University of Munich (Germany) and a B.Sc. in Mathematics and Computer Science from the Ovidius University of Constanta (Romania). Adriana is also a Certified Systems Engineering Professional with INCOSE.

Bernd Podey. has more than 20 years of experience in the Configuration Management Domain of Airbus - in Operations on A380 Program - as well as Process Architect for the Aircraft Change Process. He received the Dipl.-Ing. of the University of Rostock in Electrotechnics/ Communication Technology in 2000. Starting on A380 managing configuration of Water / Waste and Oxygen Systems he participate and lead on Transformation Projects in Airbus like the introduction of an **I**ntegrated **C**hange **C**ontrol application enabling a digital support to the decision process of Airbus since 2004 - the introduction of it's replacement - the new ***A***irbus **C**onfiguration Control **S**ervices Platform since 2020 - as well as the introduction of Advanced Visual Analytics towards operation since 2014. Currently he is acting as Data Custodian and Business Data Expert supporting the creation of the Configuration Management Data Product - and it's ontology.

Javier Reines. A trained Aeronautical Engineer, Javier has worked during the last 18 years in Configuration Management on the A380, A320, A400M, and A350 programs. Currently he is part of Configuration Management Innovation team. During his career, Javier has worked in Configuration Management processes definition and in daily operational support of several disciplines, such as product identification, change process, and configuration conformity. This includes relevant aircraft life cycle steps, such as the early stages of the A350 concept definition, the A380 design and production, and the A400M Retrofit in service campaigns. His configuration Management activities also covers collaboration with other Aircraft and Engine manufacturers and Aeronautical Engineering universities.

Moises Ramon Martinez has more than 20 years experience in the aerospace domain. First 10 years he gained experience working close to the product in different areas like production, aerodynamics and engineering. Last 10 years he took the lead in several initiatives on data and digitization services to address the needs of aerospace operations teams. Currently, he leads the Configuration Management Innovation team that contributes significantly to major digital transformation initiatives.