

IMPLEMENTING A SYSTEM ARCHITECTURE MODEL FOR AUTOMATED AIRCRAFT CABIN ASSEMBLY PROCESSES

Markusheska, Nastasija^{1,2} • Srinivasan, Venkatachalam² • Walther, Jan-Niclas²  • Gindorf, Alex² • Biedermann, Jörn² • Meller, Frank² • Nagel, Björn² 

Abstract

Aircraft manufacturers provide their customers with a number of options for aircraft customization, including a wide variety of pre-qualified optional equipment from which they can select different components according to their requirements. The numerous options cover a variety of engines, navigation systems, and interior cabin designs. This flexibility gives the possibility to the airlines to differentiate their brands. Moreover, a unique cabin interior design leaves an enduring impression on their customers and constitutes their expectations for the upcoming flight. On the other hand, many choices result in many different specifications, long delivery periods, complicated installation procedures such as stopping the running assembly of the cabin, disassembling already installed cabin components, and assembling new cabin models. Therefore, aircraft customization increases the cost and the lead time of the aircraft manufacturing processes and thus decreases the production rates. It is in the best interest of aircraft manufacturers and airlines to automate and optimize the customization processes to make them more time and cost-efficient. This paper presents a method for establishing reconfigurable and optimized scheduling for aircraft cabin assembly. The data necessary for calculating the optimal schedule is retrieved from cabin system architecture that is built with semantic web language. The focus is on building a system architecture ontology model for automated scheduling of assembly processes of an aircraft cabin, which opens up the possibility for cabin customization at every assembly step. Moreover, the cabin ontology can be used as a foundation for co-design where each expert of their branch can further upgrade the model. With the algorithm presented in this paper, the ontology can be upgraded with new data, which will automatically correlate with the existing data in the cabin ontology. The knowledge-based ontology model provides a view of the whole chain from design to realization and feedback links to all included parties. Moreover, it gives the possibility for agile changes in the assembly sequence in response to the updated demands of the clients.

Keywords: Aircraft cabin assembly • Robotics and Automation • Scheduling • Optimization • Semantic Web Language

1. INTRODUCTION

The future factories need to operate in highly dynamic environments with constantly evolving requirements dictated by the market [1]. To meet these demands, the manufacturers are in constant search of improvements. The new requirements could manifest in varying production rates, customized orders, reorganization of available resources, etc. An essential step towards improvement is the digitalization of the system architecture model. It enables a fast reorganization of resources, reconfiguration of production or assembly tasks, agile response to last-minute reconfiguration, late delivery of resources, or technological upgrades. Digitalization of the system architecture model is a significant milestone for achieving a digital twin, a virtual representation of the factories for optimizing the designing, building, and manufacturing processes, as the most promising research technology to handle multi-discipline, multi-level, and multi-site processes for aircraft production [2].

When a cabin is designed from scratch or changes in the processes are introduced, the feasibility of the manufacturing processes of the cabin parts and their installation in the fuselage is not always guaranteed. Instead, the analysis of the first manufacturing samples and their installation in the fuselage is done manually in a subsequent step. If problems appear, either the product needs to be altered or the installation processes. Due to the

lack of automation, altering the product or the installation step is usually very time-consuming [3]. Therefore, a digital model and method are needed that enable a complete overview from the design stage to manufacturing. Furthermore, the feedback data from the manufacturing stage is made available digitally to all the stakeholders, which increases the knowledge upstream and thus accelerates the development cycles.

This paper presents a methodology for developing a cabin ontology as a digital knowledge-based format for storing and organizing technical information about the aircraft cabin. Furthermore, it is presented the application of the ontology in an algorithm for optimal and automatic scheduling and rescheduling of cabin assembly processes of an aircraft cabin. Section 2 gives an overview of the research made in the area of ontology design for assembly processes. Section 3 presents a method for implementing an aircraft cabin assembly ontology into an algorithm for optimal scheduling the assembly and disassembly and reconfiguration of the aircraft cabin. Section 4 presents more details about parsing the data from the ontology model for automatic scheduling the assembly processes. The concluding remarks of the presented research and future perspectives are highlighted in Section 5.

¹ To whom correspondence should be addressed: E-mail nastasija.markusheska@dlr.de; VoIP: +49 40 248 9641 384

² Deutsches Zentrum für Luft- und Raumfahrt Forschung e. V. (DLR), Institute for System Architecture in Aeronautics, Hein-Saß-Weg 22, 21129 Hamburg, Germany

2. RELATED WORK ON ONTOLOGY DESIGN

By definition, an ontology is a data set containing structured knowledge about a particular subject domain [4]. Moreover, it uses common, agreed-upon vocabulary with defined rules and constraints to explicitly represent processes and automatic reasoning to infer knowledge from the entered data and detect inconsistencies. Ontologies as a knowledge-based dataset have become an important segment of so-called "intelligent" processes, which offer more flexibility, efficiency, and cost-effectiveness due to the possibility for inter-dependency between experts, machines, and applications [5]. Based on the specified vocabulary, an algorithm is developed in this research paper that interprets the rules and constraints to automatically and optimally plan the designated processes.

In literature are introduced several approaches for modelling an ontology. In [6], a skill description model is presented, which describes the programming of assembly systems by specifying the input and output variables. The model is based on five essential skills: change, compare, connect, move and store. The skills are further sub-classified into attributes, ID, inputs, interfaces, name, and output. However, this method is not addressing the reconfiguration of the resources.

The paper [7] presents an ontology where the data is divided into three different classes: product, processes, and resources. A list of resources executed in a process defines that process, and a list of required processes executed for manufacturing a product defines the products. The ontology is built to match the product requirements and resource skills. The method is based on three different classes: product, processes, and resources.

In [8] is described a research framework for supporting modular architecture designs for aerospace engineering ontologies. Their framework is divided into three main parts. The first part is the ontology design based on the studies presented in [9] and [10], the second part is ontology validation, and the last one is structuring the ontology knowledge and implementing it in a product-design information platform. Their ontology modelling is based on the identification and representation of low-level relationships between instances and concepts instead of high-level relationships between the different concepts.

In [11] is presented a new ontology-oriented programming module for Python. Moreover, it is given a comparison between previously available ontology programming interfaces. Among them is the *SPARQL* [12], a query language and the *OWLAPI* [13], which both are not as easy to apply as an object-oriented programming language. Therefore, they have developed the Python package *Owlready*, which has simpler syntax, but it is significantly compacter. Today there is an available newer version of the module, namely *Owlready2* [14], which is also used for developing the methodology presented in this paper. The decision was made based on the analysis presented in the previous paper [11].

In [15] is presented a novel method for semantic modelling of architectural design space called Architecture Design Space Graph (ADSG). In the system architecture are implemented three types of architectural decisions, and those are function-component mapping, component characterization and component connection. The components which fulfil the functions are the ones included in the architecture. If there are more components that could

fulfil the same function, then a design choice has to be made. In the next step of the design process, the components are characterized by their number of instances and attributes. In the final step, the components are interconnected with formal relationships, modelled using ports. Moreover, the presented method is demonstrated by finding architectural decisions for the Apollo mission [16]. The European Commission founded this innovation project.

The following section presents trivial Cabin Assembly Ontology (CAO) for evaluating the outcome of using ontologies for optimal planning and reconfiguration of aircraft cabin assembly processes. This research is part of the leading research topic, Model-Based System Engineering (MBSE) at the Institute of System Architectures in Aeronautics at the German Aerospace Center (DLR-SL). The aim is to create a digital twin of the complete processes for aircraft development from the conceptual stage to operation.

3. CABIN ASSEMBLY ONTOLOGY MODEL

A Digital Twin of an aircraft represents an exact digital representation of the physical system with its processes and services. It will save valuable time and resources for developing new concepts, and a priori can identify possible errors or drawbacks [17]. The digital thread enables the flow of information between all the sectors of the digital twin [18]. The beginning of the digitalization on any engineering project is to gather technical information about the components. Under technical information is considered all the data about their mechanical electrical properties and the interfaces the components need or provide. Instead of using the standard PDF (Portable Document Format) for storing and sharing data, ontology format is preferred as a machine-processable form [19]. First of all, it will eliminate the possibility of misinterpretation of the data in the PDFs. Second, the data will be available in a format that many computer tools use and can be shared and stored.

3.1. Principles for creating system architecture for automated cabin assembly processes

In the future, aircraft manufacturers are looking to implement more digital, agile and thus more effective production methods, allowing them to plan the complete processes from the conceptual stage to manufacturing and assembly and feed information back to the conceptual stage [20].

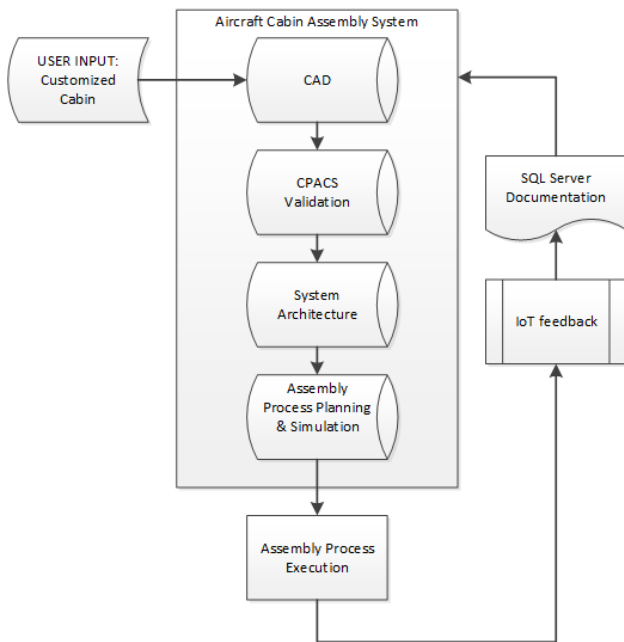


Figure 1 - Workflow of Aircraft Cabin Assembly System

In Figure 1, the schematic of such a structure is depicted, which is described further in this section. The Aircraft Cabin Assembly System (ACAS) is the leading segment of the overall process. It consists of several steps: Computer-aided Design (CAD) models, Common Parametric Aircraft Configuration Schema (CPACS) for design validation [21], System architecture of aircraft cabin and Assembly Process Planning.

Aircraft cabin design engineers develop the new cabin designs and store their work in CAD files. CPACS is a central data model which enables engineers an interdisciplinary collaboration between decentralized Multidisciplinary Design Optimization (MDO) of air transport systems [22]. Structured data sets written with the Extensible Markup Language (XML) [23] as open standard computer-processable language are used for developing aircraft cabin ontology. The modelling approach in CPACS from the XML data is called middle-out approach as a combination of top-down and bottom-up hierarchical structure [22]. Furthermore, a geometric library TiGL [24] is developed, that translates the parametric description of the aircraft into three-dimensional space (see Figure 2). Additionally, the parameters of the aircraft geometry could be modified or newly created [25].



Figure 2 – Cabin section of short distance passenger plane in TiGL viewer

The next block in the structure (Figure 1 - Workflow) is the System Architecture of an aircraft cabin. It holds information about the properties of the aircraft cabin and the cabin parts needed for scheduling the assembly processes. For instance, the size of a sidewall panel, its storage location, and assembly location are some of the properties stored in the aircraft cabin's system architecture. Semantic modelling is used for creating an interpretive and interactive cabin system architecture. For example, the instances in the model are classified according to their properties that match to specific class's properties. Therefore, the inheritance in the model is flexible and compact.

The Assembly Process Planning section collects and analyzes the information of the aircraft cabin system architecture to determine the optimal schedule for the assembly processes of the aircraft cabin. The global assembly sequence is stored in the model of the aircraft cabin architecture, and is modified to fit the requirements, avoid constraints and optimally use the available robot resources. Simulation is an essential part of this section because a simulation does feasibility analysis for the robot motion required for the assembly processes (Figure 3). The [26] presents a use-case for simulation of assembling an aircraft cabin and the collection of the meta-data for the feedback loop into the system architecture of cabin assembly.

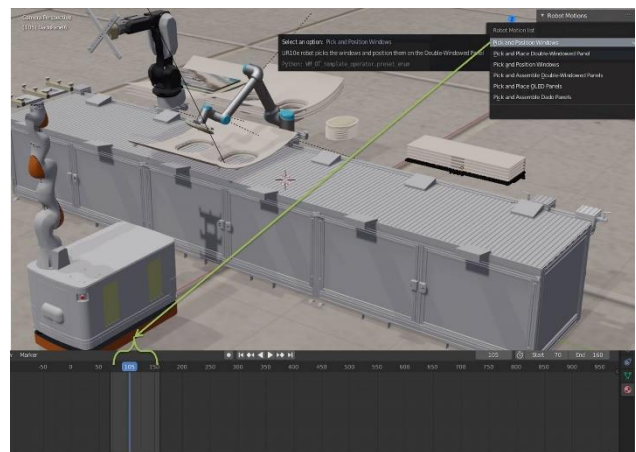


Figure 3 - Blender simulation of preassembling double window sidewall panels [26]

ACAS is a digital approach for designing an aircraft cabin, validating the model and scheduling the assembly

processes. The digital thread between the elements of the ACAS is an essential medium for exchanging, recording and storing data. Moreover, it provides substantial and time-effective access to data necessary for calculating the consequences of the changes from one component to any other component in the cabin. For example, changes in the dimensions of a cabin sidewall panel can be validated, and this change can be updated in the cabin architecture and automatically recalculated its assembly positions.

The main goal of ACAS is to assist the aircraft manufacturers in optimizing the utilization of their assembly plants by monitoring every step and all the assets in the processes [27]. The presented approach in this paper focuses on proving the ontology model, which is a practical solution for creating a cabin system architecture. It can be implemented into an algorithm for optimization of the schedule for aircraft cabin assembly processes.

The following block is the execution block which represents the physical execution of the modelled cabin assembly processes. For the validation of the proposed method, a scaled-down demo will be performed in the near future. There is a preassembly line with three robots at the DLR Institute of System Architectures in Aeronautics. Two robots, an Universal Robot Ur10e and a Bosch APAS are mounted on a linear axis, and one mobile robot Kuka KMR liwa is also available for transferring components from the preassembly line to the cabin mockup (Figure 4).



Figure 4 - Preassembly station at DLR Hamburg for validating the methodology

The feedback connection from the production plant to ACAS is another crucial element of the procedure. This block provides information from all sensors deployed in the assembly station, such as cameras, laser sensors and the robot location and orientation sensors. Furthermore, information about all assembled aircraft components, the assembly time and possible occurrences of error signals during run time is fed back. The data will be applied in future to assess the quality of the assembly processes and compare simulated assembly with actual execution of the assembly processes. Moreover, the feedback data is necessary for automatic assembly process reconfiguration. Sometimes, due to “last-minute” customization demands on the cabin design, rapid reconfiguration of the assembly processes has to be performed. Necessary changes to the processes might include a reallocation of resources, removal or replacement of cabin components or changes in production rate. Modifying the ontology with the feedback data from the Internet of Things (IoT) [28] devices can quickly overcomplicate the model. Therefore, introducing a

SQL [29] database server into the method could be a practical solution for storing feedback data like sensor’s reading, camera output, and instances of every preassembled or installed part in the cabin with more detailed data. With the SQL server on the feedback link, the ontology would not be overpopulated with data, ensuring a transparent system architecture for cabin assembly processes.

With this approach, the information and constraints for the assembly processes are easily obtainable from the aircraft cabin system architecture. Moreover, all cabin assembly processes are designed to be flexible and efficiently reconfigurable, as presented in Section 4. Additionally, this algorithm is designed to allow the customers to reconfigure the cabin according to their wishes even during later stages of the aircraft production (see Section 4.2.1).

The following section focuses on the system architecture and assembly process planning blocks and how they interlink and contribute to the complete intelligent aircraft cabin assembly process planning.

3.2. System architecture for cabin assembly

To abstract and interpret the system architecture of cabin assembly stations, an ontology model is created with one of the standards of Semantic Web Technology, namely Web Ontology Language (OWL) [30]. The ontology is a knowledge-based model in a disclosed domain, where the classified instances are described with properties and axioms. Moreover, OWL offers the possibility for automatic reasoning of the acquired data based on the defined axioms and properties for generating a logical interface between the data in the model.

3.2.1 Class definition

When modelling ontology, there are no strict rules of what should be defined as a subclass of a class and what should be defined as an instance of a class. However, a good practice is to define classes that describe the concept of the domain [31]. Therefore, the central system classes defined in the proposed model are (see Figure 5 –):

- Resources
- Assembly Stations and,
- Robots.

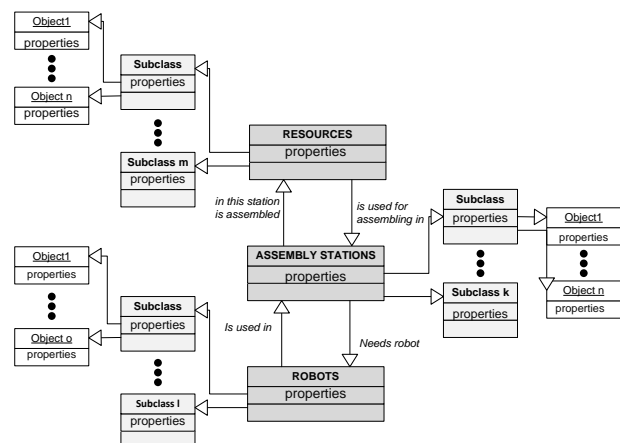


Figure 5 – Sample for domain declaration of aircraft cabin assembly sequence

3.2.2 Defining instance properties

Once they are defined, the model is refined in a taxonomic hierarchy (classes – subclasses - instances). In the next step, semantic-annotated properties are specified for defining different class axioms. The primary relations between the system classes are created with the following instance properties:

- "is used for assembling in"
- "needs robot"
- "is used in"
- "in this station is assembled"

For example, the following axiom defines that some robots are needed in the assembly station:

- *Assembly Station* → *needs robots* → *some Robots*

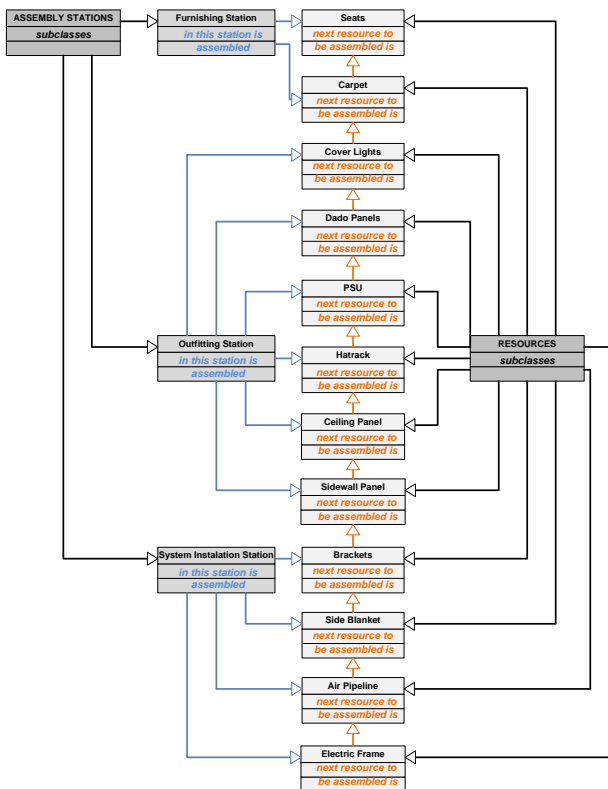


Figure 6 – Segment of a cabin assembly sequence

Moreover, the axioms represent the objective and the constraints of each cabin assembly process.

In the Web Ontology Language (OWL) [30], here are two kinds of properties: instance properties and data properties. To illustrate, blue arcs, which represent an instance property "in this station is assembled," are depicted in Figure 6. The defined axiom for the class "Furnishing Station" is:

- *Furnishing station* → *in this station is assembled* → *some Seats and some Carpet*

This axiom describes the objective of the furnishing station from the cabin assembly processes.

The orange arcs represent another instance property "next resource to be assembled is". This property is used to

describe constraints. For example, the carpet has to be mounted before installing the seats in the aircraft cabin in the furnishing station. This is demonstrated with the following axiom (Figure 6):

- *Carpet* → "next resource to be assembled is" → *some Seats*

3.2.3 Defining instances and data properties

In an ontology model, the instances of a class could also be an instance of another class. The reasoner classifies the instances according to the defined properties and axioms [31]. The different models of aircraft cabin parts, such as "Classic sidewall panel", are defined as instances of the class Sidewall Panels in the ontology presented here. Because they are used in an assembly station, the reasoner will classify the instance "Classic sidewall panel" also as an instance of the specified class Assembly station. In Figure 7, the instances from the "Double Window Sidewall" class are depicted in purple. They all have the exact dimensions and assembly constraints but differ in decorative foils. This information is modelled with another type of property called data property.

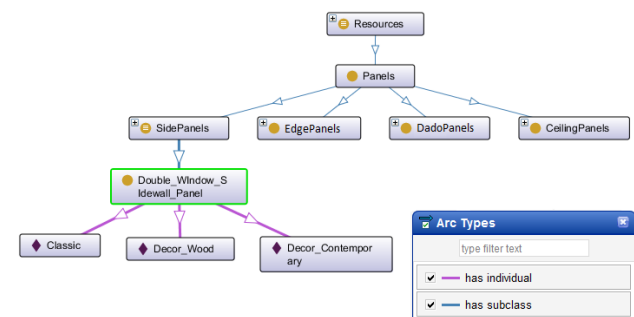


Figure 7 – Sample of class instance from aircraft cabin sidewall panel visualized in OntoGraf [32]

They could be of a type integer, float, bool, string, date or time. In the ontology presented here, the defined data properties are the following:

- *Size*
- *Initial Position with respect to global coordinate system*
- *Delta position for the next component installation*
- *Total time for installation in the aircraft cabin*
- *Conditions for installation in the aircraft cabin*
- *Number of available parts*
- *Number of installed parts*
- *Storage location*

These properties are necessary for calculating the time-optimal schedule of the assembly processes and for agile rescheduling of them. The fundamental properties are also stored in the standard file format CPACS.

3.2.4 Defining axioms

Axioms are expressions used for autogenerating relationships between different classes and instances [33]. In the ontology presented in this paper, axioms are used to classify the data. For example, when the ontology is populated with a new set of sidewall panels, instead of checking every class if the sidewall panel should be an instance of the specific class, axioms make the selection automatically. For writing the axioms was chosen the Semantic Web Rule Language (SWRL), which is a combination of OWL sublanguages and Rule Markup

Language (RuleML) [34]. Examples of the axioms defined in the ontology model are the following:

- $Sidewall_panel(?s) \wedge has_design(?s, Custom_design) \rightarrow Custom_design_class (?s)$
- $Sidewall_panel(?s) \wedge has_design(?s, Double_windows) \rightarrow Double_Windows_Sidewall_class (?s)$

With the help of these axioms, the sidewall panels that have Custom design and have two windows will be automatically sorted as part of both classes: Custom design and Double Windows Sidewall. This approach reduces the possibility of human error and time for arranging the data.

As already mentioned in this paper, there are no strict rules for designing an ontology, but it is designed regarding the demands of the application. Therefore, the presented ontology is designed to satisfy the needs for scheduling an automatic and agile aircraft cabin assembly. For example, for assembly sidewall panels, the model has information on when to start the process, the storage location, the installation location, how long it needs to be executed, and which robot will assemble the sidewall panel.

4. CABIN ASSEMBLY PROCESS PLANING

The proposed ontology model encompasses all necessary aspects for demonstrating the automatic scheduling of an aircraft cabin assembly processes. Over the years, the introduction of industrial robots into assembly processes often yielded improved results in terms of speed and quality [35]. Therefore, in this paper, the proposed method is validated with three uses cases where robots are deployed for the assembly tasks.

The two main challenges that are presented in the method are time reduction of the assembly processes (Section 4.1) and enabling agile customization processes (Section 4.2). In other words, this method enables agile reconfiguration of the cabin assembly processes at a later stage according to the customer requirements for new cabin design. It automatically optimizes the assembly processes' execution schedule to improve the utilization of the available resources.

4.1. Time-optimized scheduling of assembly processes

The proposed method for developing an ontology aims to optimize cabin assembly and reassembly schedule. Changes in the schedule are necessary due to new cabin design changes and resources or tool availability. For example, if a robot is programmed to service just one task, such as assembling sidewall panels, and if the robot needs to be serviced, it will result in delays. On the other hand, if the robots are programmed flexibly to service different tasks, the assembly could be rescheduled with the available robots. Programmed flexibly, it might imply that the robots should be programmed with all the possible scenarios that could happen in the assembly line, which is impractical and unrealistic. First of all, it is implausible that the engineers could foresee all the future changes in the complete assembly procedure. Second, programming manually all the scenarios with all the positions where the robots need to pick and assemble different cabin parts would cost a lot of time and effort. Moreover, the aircraft cabin design is

often upgraded and alternated according to the customers' requirements. Therefore, the purpose of this algorithm is to decrease the time and effort required to adjust the assembly station according to the available resources, requirements and changes in the design. The availability of the robots in the assembly station has a substantial impact on the lead time. Therefore, this paper presents an application case in which only three types of aircraft components are installed in the outfitting station (Figure 6). The algorithm plans the assembly sequence of the cabin components using two and three robots. For comparison, are given the time flow charts of the optimized and not optimized schedule for assembling processes.

The required cabin components (Figure 8) for the assembly processes in the application case are:

- 4 x sidewall panels
 - Conditions: finished system installation.
 - Total time for installation is 3-time units³.
- 3 x linings between sidewall panels.
 - Conditions: two sidewall panels have to be assembled so the robots will not collide. For example, the lining *b1* can be assembled after sidewall panels *a1* and *a2* are assembled.
 - Total time for installation is 2-time units³.
- 4 x ceiling panels
 - Conditions: one lining between panel have to be assembled so the robots will not collide. For example, the ceiling panel *c1* can be assembled only after the lining *b1* is assembled.
 - Total time for installation is 3-time units³.



Figure 8 – Cabin lining components for assembly with different number of robots

The conditions in the ontology are stored as instance properties. From Figure 6, it can be observed that the cabin parts have to be assembled in a specific order:

- *Sidewall Panel* → *Lining between Sidewall Panel* → *Ceiling Panel*

If only one robot is available for assembly, then all the components are going to be assembled sequentially. Figure 9 illustrates the time flow chart for this case. The figure shows that the execution time will not be reduced when three robots are engaged in the processes if they do not execute the assembly processes in parallel when possible.

³ The given information is trivial (it is not recorded from the physical system)

However, the cabin assembly could be accelerated by reprogramming the robots to execute different processes flexibly. In cases where one robot has finished assembling one type of component, it could be reassigned to assemble another. Instead of manually reprogramming the robots, it could be done through the algorithm, which takes all the required information about the assembly processes from the cabin ontology.

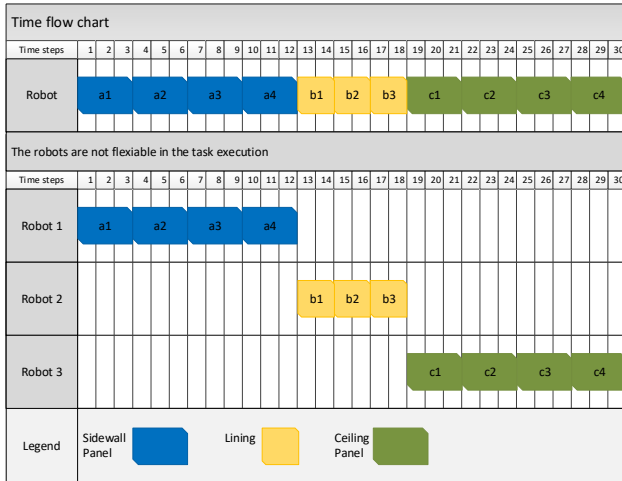


Figure 9 - Time flow chart of the cabin assembly processes before optimally scheduling

The primary constraint for parallel execution of the assembly processes is that two or more robots cannot assemble different components at the exact location. The location constraints are presented in Section 4.1 a), b) and c) for each cabin component expressed in time units.

The Owlready2 Python Package [4] is applied to parse the information from the ontology and make it available to the algorithm for optimal planning of cabin assembly processes. The algorithm presented below (see Figure 10) is an abbreviated version of the complete algorithm in order to present the procedure of assembly planning.

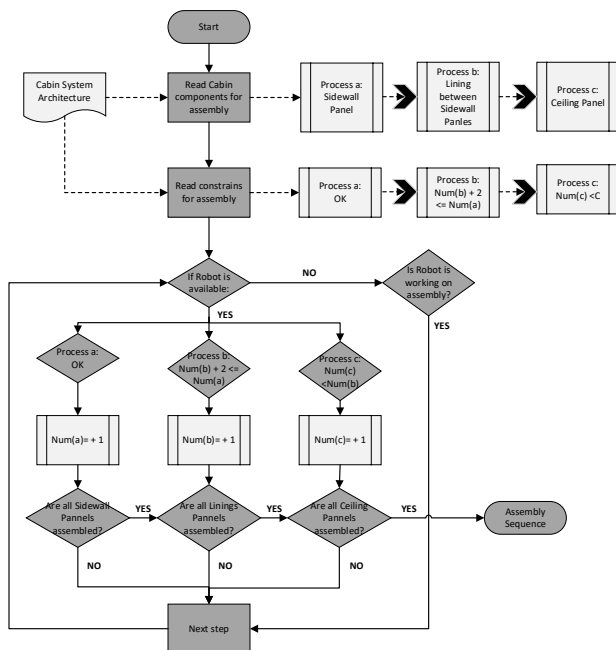


Figure 10 – Algorithm I: Optimization for scheduling cabin assembly sequence

The algorithm in Figure 10 starts with reading the aircraft cabin system architecture data. At first, it parses which cabin components need to be assembled and what is the initial sequence, later it parses the constraints for assembling the components. If the required cabin parts are available and the robot is available, the assembly of the first cabin part will start. Since only the condition for assembling sidewall panels is fulfilled, the algorithm will start with assembling the first sidewall panel *a1* (see Figure 8). When two sidewall panels, *a1* and *a2*, are already assembled in the cabin, the algorithm will allow assembling the first lining *b1* between the two assembled sidewall panels. After the lining *b1* between the sidewall panels *a1* and *a2* is assembled in the cabin, the first ceiling panel *c1* could be assembled. When two robots are available, sidewall panels *a1* and *a2* are going to be assembled simultaneously (see Figure 11 and Figure 12) and if three robots are available, sidewall panels *a1*, *a2* and *a3* are assembled simultaneously (see Figure 13). In Figure 11, after time slot 11, Robot 2 has finished assembling the sidewall panel *a4*, and it needs to wait for two-time units. It is not allowed a simultaneous assembly of the third lining between the sidewall panels *b3* and the third ceiling panel *c3* due to location constraints. In the presented use case, it is relatively manageable for a person to make the schedule for the assembly of the cabin parts considering the given constraints. However, when the use-case is extended for assembling the complete aircraft cabin, the task of creating the schedule becomes significantly more complex and challenging. Therefore, this algorithm is needed as part of the digitalization of assembling an aircraft cabin. This algorithm will automatically read the necessary data from the cabin system architecture and create the schedule for time-optimally assembly. In this way, manually scheduling the complete assembly with the method trial-and-error will be avoided.

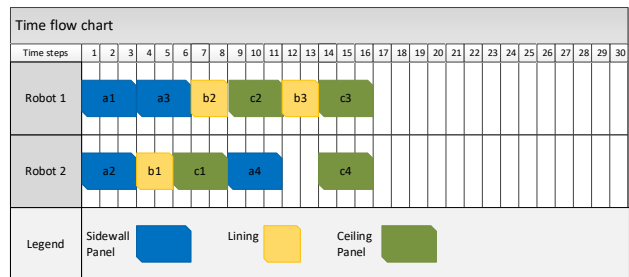


Figure 11 - Time flow chart of the cabin assembly processes after optimally scheduling with two robots (Scenario 1)

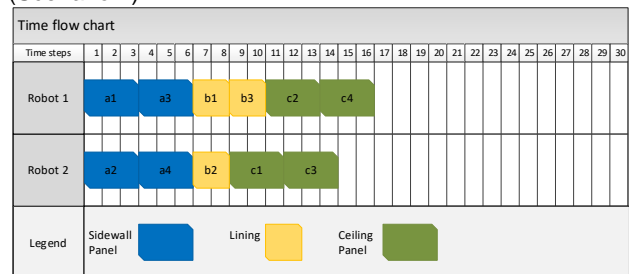


Figure 12 - Time flow chart of the cabin assembly processes after optimally scheduling with two robots (Scenario 2)

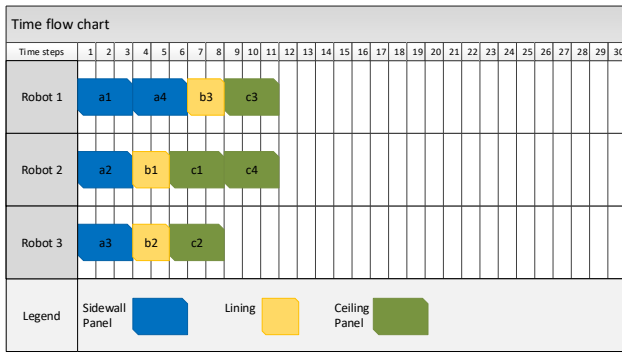


Figure 13 - Time flow chart of the cabin assembly processes after optimal scheduling with three robots

4.2. Reconfiguration of assembly and disassembly aircraft cabin processes

This section demonstrates the benefit of applying ontology modelling for dynamic and reconfigurable cabin assembly processes. Reconfiguring the cabin assembly processes can become necessary due to frequent customization demands, the introduction of new cabin products, disassembling some cabin parts or retrofitting. When the assembly processes are digitalized, the digital thread provides a constant flow of information between the actual assembly processes and their digital representation. This information is used in the algorithm for automatic reconfiguration of the cabin assembly. Reconfiguration of the processes could be requested at any time step, hence it could not be foreseen, and every time could be different. The first necessary information is to know which parts are assembled in the cabin and then analyze which parts need to be disassembled to assemble the new requested ones. A manual reconfiguration is a complex and time-consuming task. Alternatively, the reconfiguration algorithm can automatically recalculate the disassembly and reassembly of the aircraft cabin.

4.2.1. Scheduling a disassembly processes for cabin reconfiguration

To demonstrate the flexibility of the designed ontology and the developed method is used the same scenario as in Section 4.1, where two industrial robots assemble the cabin components. This application case aims to demonstrate automatic reconfiguration of the assembly sequence when a reconfiguration of the cabin is demanded during run time. It is important to note that the disassembly sequence is stored in the system architecture of the aircraft cabin, but it is reversed from the assembly sequence from Section 4.1 (see Figure 6), and it is following:

- *Ceiling Panel* → *Lining between Sidewall Panel* → *Sidewall Panel*

The following properties are specified for the use-case which demonstrates the disassembly process plan of cabin parts:

- 4 x ceiling panels
 - Conditions: received command for disassembling
 - Total time for disassembling is 3-time units³.
- 3 x linings between sidewall panels.
 - Conditions: two ceiling panels have to be disassembled so the robots will not collide. For example, the lining *b1* can be

disassembled after ceiling panel *c1* and *c2* are disassembled.

- Total time for disassembling is 2-time units³.
- c) 4 x sidewall panels
- Conditions: the lining between the sidewall panel has to be disassembled. For example: to disassemble sidewall panel *a2*, the linings *b1* and *b2* have to be previously disassembled.
 - Total time for disassembling is 3-time units³.

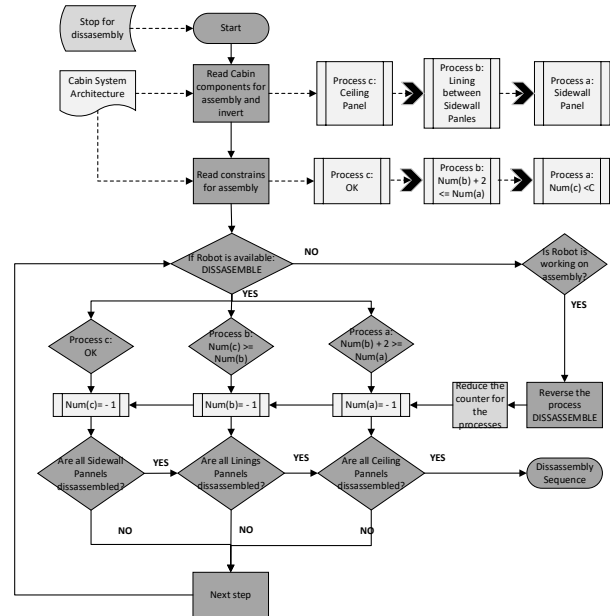


Figure 14 – Algorithm II: Optimization for scheduling cabin disassembly sequence

The algorithm for disassembling is depicted in Figure 14, and it is designed to interrupt the assembly processes and automatically reschedule the processes for disassembling the needed cabin components. For planning the disassembly processes, the initial sequence is automatically reversed from the assembly sequence, but the conditions are changed, and the task of the robot is to remove the cabin parts of the fuselage. Instead of reviewing and changing all the conditions manually, the algorithm does that task automatically and arranges a new schedule for disassembling.

An example of the disassembly with two robots is presented in Figure 15. A signal for disassembling the cabin components is received. The signal is received after the eighth time step is finished (red vertical line in Figure 15). Next, the execution sequence for assembling the three aircraft cabin components is reversed. The first disassembled component is the ceiling panel *c1*. Since two robots are deployed in this process, the robot assembling the sidewall panel *a4* currently disassembles it. Even if a third robot was available, it would need to wait because the condition for disassembling the lining between the panels *b1* is not fulfilled. The procedure finishes with disassembling all required cabin components. This state of the cabin is fed back to the aircraft cabin architecture, and it can be used for scheduling new assembly with other cabin components.

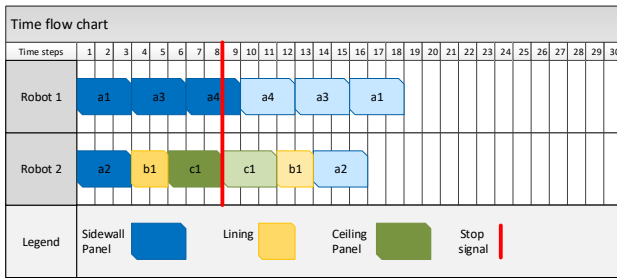


Figure 15 – Time flow chart of cabin disassembly processes with two robots

The proposed method for designing an ontology model in combination with the developed algorithm demonstrates reversibility with minimal human effort for scheduling time-optimal disassembly sequences.

4.2.2. Reconfiguration of cabin assembly processes

Reconfiguring the assembly processes to add new elements to the assembly sequence is an important attribute of the proposed methodology for agile rescheduling cabin assembly processes. When a new cabin component is added to the cabin design, the assembly sequence needs to be adjusted accordingly. This section has presented the algorithm's logic that automatically reconfigures the cabin assemble sequence. Necessary information for the new cabin components are the following:

- Which cabin components has to be assembled?
- Which cabin components must not be assembled?

When this information is available, the algorithm will automatically arrange and adjust the assembly sequence in the cabin system architecture. In Figure 16 the initial process chain is depicted. The yellow arcs represent the property "next for assembling is", and the orange ones represent the property "before is assembled" (see Figure 16). Another element A2.1 is introduced, and it needs to be assembled after element A2 but before element A3. For that purpose, its properties for assembling are defined as:

- A2.1 → "next for assembling is" → some A3
- A2.1 → "before is assembled" → some A2

The algorithm for reconfiguration the assembly processes inspects the properties of the classes that are part of the definition of the class A2.1. Moreover, the algorithm inspects, which of their properties need to be removed, for the new element to be placed in the correct segment of the assembly sequence. For this reason, the algorithm inspects if the classes A2 and A3 have common properties with the class A2.1. First, it is inspected if the class A3 is defined with the property "before is assembled". From Figure 16, the following definition can be observed:

- A3 → "before is assembled" → some A2

This definition is deleted and it is replaced with:

- A3 → "before is assembled" → some A2.1

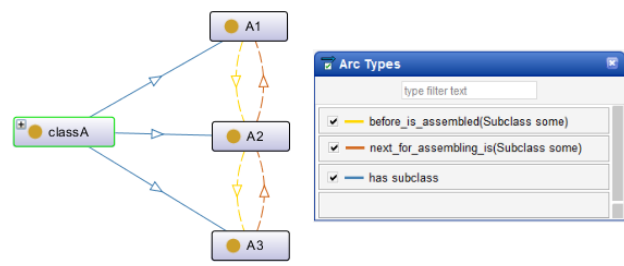


Figure 16 – Example of the initial cabin assembly sequence visualized in OntoGrav

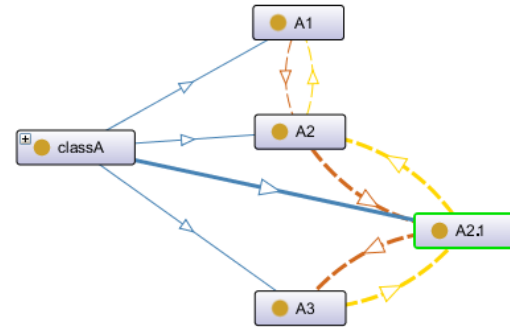


Figure 17 – Example for reconfiguration of the initial cabin assembly sequence visualized in OntoGrav

The second inspection is checking if this class A2 is defined with the property "next for assembling is". From Figure 16, the following definition is taken:

- A2 → "next for assembling is" → some A3

This definition is also deleted and replaced with:

- A2 → "next for assembling is" → some A2.1

As the result of the algorithm, the new class is appended in the correct position of the assembly sequence (Figure 17). This method offers simple reconfiguration of the ontology without analyzing and rearranging the complete cabin assembly architecture manually.

In conclusion, this methodology demonstrates to be suitable for time-optimal assembly reconfiguration without the need for a more in-depth and long examination of the overall procedure.

5. CONCLUSION

This research study proposes a method for ontology-based modelling of aeronautical system architectures (Section 3.2). The method demonstrated the two significant advantages of using ontologies for automatically optimizing the scheduling of cabin assembly. The first one is that semantic web language had proven to be an appropriate choice for creating an ontology for aircraft cabin assembly. Moreover, the ontology can be implemented in an optimal time scheduling algorithm that agile and automatically reschedule the cabin assembly processes and reduces unnecessary expert's effort. Customized aircraft cabins cannot be foreseen and pre-scheduled, therefore the aircraft manufacturers need to interrupt the serial production processes and reschedule it. Manually

rescheduling the assembly processes is a complex and time-consuming task. The second advantage is that ontologies connect various pieces of knowledge and perform logical reasoning for inferring new knowledge from the present data. This feature enables reconfiguration of the system architecture for an aircraft cabin assembly. At the beginning of this paper, an ontology model is introduced, which follows the general guidelines for designing ontologies. The model is designed to accurately abstract and model the system architecture for cabin assembly.

Moreover, this study presents an algorithm for parsing ontology data and using it for computing time-optimal cabin assembly sequences. The given application cases prove the concept for ontology-based scheduling of cabin assembly and disassembly processes. The time flow charts in Section 4.1 illustrate the potential of schedule optimization, a result of optimizing the schedule of an assembly process with an execution time of 30-time units (Figure 9) to 16-time units (Figure 11 and Figure 12) or 11-time units (Figure 13). This is a significant reduction of the time for assembling the cabin, due to the number of engaged robots and the algorithm for synchronizing the robots to assemble different cabin components in parallel. In practice, this method could be applied even at the conceptual stage to anticipate the challenges and lead times for assembling the novel designs.

Furthermore, in Section 4.2, the importance of ontology reconfiguration has been illustrated. It is a crucial capability for the assembly station because it offers the possibility to reconfigure the assembly sequence according to updated product requirements. The assembly process changes are often unforeseen and require much involvement in data analysis, detecting the constraints for rescheduling, and organizing the available resources. Nevertheless, manufacturers need to react quickly to avoid negative consequences such as cost increases or postponed delivery. Therefore, the digitalization of assembly processes and the development of algorithms for automatic reconfiguration are valid and necessary procedures. The reconfiguration capability of the developed algorithm has been demonstrated in two example cases in this paper, one for disassembling cabin parts (Section 4.2.1) and another for adding further elements to the assembly sequence (Section 4.2.2).

The methodology presented for optimized scheduling assembly processes and reconfiguration will be tested at the DLR facilities in Hamburg. Even though the research is in its early stages, it shows promising results. Also, it is important to mention the limitation of this research. First, the algorithm uses trivial execution time to assemble the cabin component instead of accurate data. Next, it is not considered the inverse kinematic for navigating the robots. Inverse kinematics, robot joint's speed and acceleration could contribute to the time optimization of the assembly schedule and will be considered in the following stages of this research. Further possible future work will involve expanding the ontology model with additional data for developing path planning algorithms, grasping points and additional connection rules and elements between the cabin components. With this information, the optimized schedule assembly plan computed by the process planning algorithm will be more precise, explicit and detailed.

Another challenge for future development is handling the feedback data from the IoT sensors. One option is to store the data directly in the ontology, and the other option is to

set up a complimentary SQL database. The methodology presented in this paper is focused on developing the two blocks from the complete process for intelligent, agile cabin production (see Figure 1), from CAD design to physical realization, including feedback of information to the design stage. Therefore, implementing an interface between the CPACS product data model and the cabin ontology is essential for further knowledge modelling for cabin assembly processes.

Ontology-based engineering is a promising approach for further developing knowledge-based engineering capabilities in the area of aeronautics. This paper presents a suitable, virtual, semantic modelling approach for time-optimized automatic scheduling of cabin assembling processes. Hence, the method presented in this paper significantly contributes to the digital thread for efficient and transparent transferring the knowledge from smaller-scale components into larger-scale subsystems of the digital twin.

References

- [1] S. Vaidya, P. Ambad and S. Bhosle, "Industry 4.0 – A Glimpse," *Procedia Manufacturing*, vol. 20, pp. 233-238, Jan. 2018.
- [2] R. Henke, "An der Schwelle zu einem neuen Luftfahrt-Zeitalter," *DLR magazin, Deutsches Zentrum für Luft- und Raumfahrt e.V.*, no. 157, p. 4, April 2018.
- [3] N. Halfmann, D. Krause and S. Umlauf, "Assembly Concepts for Aircraft Cabin Installation," in *ASME 2010 10th Biennial Conference on Engineering Systems Design and Analysis*, Istanbul, Jan. 2010.
- [4] M. C. Keet, *An Introduction to Ontology Engineering*, University of Cape Town: Maria Keet, Minneapolis; Open Textbook Library, 7. November 2018.
- [5] J. Page Risueno and B. Nagel, "Development of a Knowledge-Based Engineering Framework for Modeling Aircraft Production," in *AIAA Aviation Forum*, Dallas, Texas, USA, 17-21 June 2019.
- [6] J. Backhaus and G. Reinhart, "Digital description of products, processes and resources for task-oriented programming of assembly systems," *Journal of Intelligent Manufacturing*, vol. 28, pp. 1787-1800, March 2015.
- [7] J. Pfrommer, D. Štogl, K. Aleksandrov, S. Navarro, B. Hein and J. Beyerer, "Plug & produce by modelling skills and service-oriented orchestration of reconfigurable manufacturing systems," *at - Automatisierungstechnik*, vol. 63, pp. 790-800, October 2015.
- [8] I. Sanya and E. Shehab, "A framework for developing engineering design ontologies within the aerospace industry," *International Journal of Production Research*, vol. 53, pp. 2383-2409, April 2015.
- [9] M. Gruninger, O. Bodenreider, F. Olken, L. Obrst and P. Yim, "Ontology Summit 2007–Ontology, taxonomy, folksonomy: Understanding the distinctions," *Applied Ontology*, vol. 3, pp. 191-200, Jan. 2008.
- [10] G. Brusa, M. Caliusco and O. Chiotti, "Towards ontological engineering: A process for building a domain ontology from scratch in public

- administration," *Expert Systems*, vol. 25, pp. 484-503, Nov. 2008.
- [11] J.-B. Lamy, "Owlready: Ontology-oriented programming in Python with automatic classification and high level constructs for biomedical ontologies," *Artificial Intelligence in Medicine*, vol. 80, August 2017.
- [12] E. Prud'hommeaux and A. Seaborne, "SPARQL Query Language for RDF," 01 2008. [Online]. Available: <http://www.w3.org/TR/rdf-sparql-query/>.
- [13] M. Horridge and S. Bechhofer, "The OWL API: A Java API for OWL ontologies," *Semantic Web*, pp. 11-21, 10.3233/sw-2011-0025 January 2011.
- [14] J.-B. LAMY, "Owlready2 Documentation," 23 Januar 2022. [Online]. Available: <https://owlready2.readthedocs.io/en/latest/>.
- [15] J. Bussemaker, P. D. Ciampa and B. Nagel, "System Architecture Design Space Exploration: An Approach to Modeling and Optimization," in *AIAA Aviation 2020 Forum*, 15-19 June, 2020.
- [16] W. Simmons, "A framework for decision support in system architectures," PhD thesis, Massachusetts Institute of Technology, Nov. 2008.
- [17] R. Hein and F. Heinecke, "Digitaler Zwilling - ein dynamisches Abbild und nicht nur eine digitale Kopie," DLR Deutsches Zentrum für Luft- und Raumfahrt e.V., Braunschweig, 2018.
- [18] L. Boggero and B. Nagel, "Towards the digitalization of the aircraft life cycle. Research activities at the DLR.," FERCHAU Aviation Partner Network Meeting 2020, Hamburg, Germany, 15 Jan 2020.
- [19] K. Opasjumruskit, D. Peters and S. Schindler, "DSAT: Ontology-based Information Extraction on Technical Data Sheets.," International Semantic Web Conference 2020, Nov. 2020.
- [20] P. D. Ciampa and B. Nagel, "AGILE Paradigm: The next generation collaborative MDO for the development of aeronautical systems," *Progress in Aerospace Sciences*, vol. 119, p. 100643, Nov. 2020.
- [21] "Common Parametric Aircraft Configuration Schema," DLR Institute for System Architectures in Aeronautics, [Online]. Available: <https://www.cpacs.de>. [Accessed August 2018].
- [22] M. Alder, E. Moerland, J. Jepsen and B. Nagel, "Recent Advances in Establishing a Common Language for Aircraft Design with CPACS," Aerospace Europe Conference, Bordeaux, France, 25.-28. Feb 2020.
- [23] C. Johnson, "The lexicon of XML," *InTech*, no. 50, May 2003.
- [24] M. Siggel, J. Kleinert, T. Stollenwerk and R. Maierl, "TiGL: An Open Source Computational Geometry Library for Parametric Aircraft Design," *Mathematics in Computer Science*, vol. 13, Sep. 2019.
- [25] J. C. Fuchte, "Enhancement of Aircraft Cabin Design Guidelines with Special Consideration of Aircraft Turnaround and Short Range Operations," PhD thesis, DLR Deutsches Zentrum für Luft- und Raumfahrt e.V. and Technische Universität Hamburg-Harburg, Hamburg, April 2014.
- [26] V. Srinivasan, N. Markusheska, J.-N. Walther, A. Gindorf, C. Hesse, J. Biedermann, F. Meller and B. Nagel, "Autonomous control of an industrial robot based on formalized process description for cabin assembly," in *Deutsche Gesellschaft für Luft- und Raumfahrt - Lilienthal-Oberth e.V.*, 01-03 Sep. 2020.
- [27] H. Meyer, J. Zimdahl, A. Kamtsiuris, R. Meissner, F. Raddatz, S. Haufe and M. Bäßler, "Development of a digital twin for aviation research," in *Deutsche Gesellschaft für Luft- und Raumfahrt - Lilienthal-Oberth e.V.*, 1-3 Sep. 2020.
- [28] A.-Q. Antar, E. Magesh and T. Srinivasulu, "The Internet of Things (IoT): An Overview," *Journal of Engineering Research and Applications*, no. 5, pp. 71-82, Dec. 2015.
- [29] B. Nevarez, High Performance SQL Server, Jan. 2016.
- [30] L. Jean-Baptiste, *Ontologies with Python*, Apress, Berkeley, CA, Jan. 2021.
- [31] N. F. Noy and D. L. McGuinness, "Ontology Development 101: A Guide to Creating Your First Ontology," *Knowledge Systems Laboratory*, vol. 32, Jan. 2001.
- [32] S. Falconer, "Stanford University; Stanford Center for Biomedical Informatics Research," April 2010. [Online]. Available: <https://protegewiki.stanford.edu/wiki/OntoGraf>.
- [33] S. Staab and A. Maedche, "Axioms are Objects, too — Ontology Engineering beyond the Modeling of Concepts and Relations," Institute AIFB, Karlsruhe University, Germany, March 2000.
- [34] I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosz and M. Dean, "SWRL: A Semantic Web Rule Language Combining OWL and RuleML," National Research Council of Canada, Network Inference, and Stanford University, <http://www.w3.org/Submission/SWRL/>, W3C Member Submission 21 May 2004.
- [35] T. Bock and T. Linner, *Robotic Industrialization: Automation and Robotic Technologies for Customized Component, Module, and Building Prefabrication*, Cambridge: Cambridge University Press, Januar 2015, pp. 1-238.