AST 2019
Int. Workshop on Aircraft System Technologies

NETWORK DESIGN CRITERIA TO INTRODUCE DATA ANALYTICS WITHIN THE AIRCRAFT CABIN

Fabian Giertzsch[1], Hartmut Hintze[1], Burkhard Heinke[2], Ralf God[1]

[1]Institute of Aircraft Cabin Systems, Hamburg University of Technology
Hein-Saß-Weg 22, 21129 Hamburg, Germany

[2]KID Systeme GmbH
Lüneburger Schanze 30, 21614 Buxtehude, Germany

fabian.maximilian.giertzsch@tuhh.de

Abstract

The functions within the aircraft cabin are provided by various intercommunicating systems. Enhancing the operational reliability of cabin functions therefore depends on improving the availability of the underlying systems. As a prerequisite to improve the availability of a system a deep understanding of its behaviour while interacting with its environment is needed. A typical approach to gain this knowledge is the application of data analytics: this requires the collection of data describing the past behaviour of the system (*i.e.* data at rest) as well as an online evaluation of data streams (*i.e.* data in motion) to enable, for example, predictive maintenance, which traces the remaining useful life (RUL) of the system. If the RUL is known, a system or its components can be replaced before the system is no longer able to provide its intended functionality. Moreover, if a system is replaced on a regular basis but the RUL indicates that the system can be safely used until the next aircraft check, the maintenance interval of this specific system can be extended whilst still assuring its availability.

In order to efficiently access and process system data the data analytics must be an integral part of the cabin network. This introduces additional requirements for the network functions which might have an impact on the resulting network architecture. Thus, the requirements need to be translated into network design criteria in order to select an appropriate architecture for the network as depicted in Figure 1.

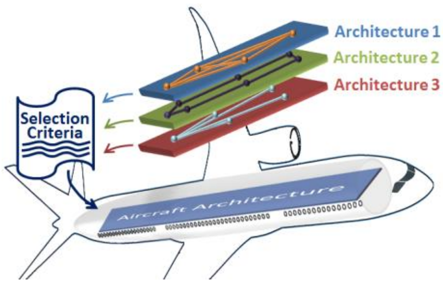

Figure 1. Network Design Process for introducing Data Analytics.

This paper provides a set of possible design criteria and discusses the derivation process of a suitable network architecture. The criteria serve as a starting point for the design of a future cabin network with native data analytics support.

1 INTRODUCTION

Today's passenger air travel is characterized by two main streams: the low fare airlines and the full service carriers. The operation of the aircraft of low fare airlines is driven by minimal turnaround times and a maximized load factor of the available seats whereas the main characteristic of full service carrier is the maximized passenger comfort. Both properties are highly correlated with the reliability and therefore also the availability of the cabin functions, which are provided by multiple interconnected systems. A failure of a system on the Minimum Equipment List (MEL) might, for example, imply that some seats cannot be occupied anymore or might even delay the departure. This type of failure influences both types of airlines, low fare and full service. In addition, full service carriers are even more prone to service restrictions as the corresponding aircraft cabins are equipped with more fragile components and systems. For this, the seat actuators of business and first class seats can be mentioned, as they can fail more often than pure mechanic seat adjustment based functions. In order to increase the availability of a function, an often discussed approach is predictive maintenance [1], which allows getting an indication before one of the underlying systems runs into a faulty condition. Thus, the system can be replaced before the provided functionality is restricted or even not available anymore. The indication is typically based on sensors monitoring the corresponding system as well as additional context data, e.g. the bank angle of aircraft. The data is then evaluated using analytical models, data-driven models or a combination of both in order to get an indication of the remaining useful life (RUL) of the system. As data-driven models have the advantage that no in-depth understanding of the internal structure of the monitored system is needed and due to today's affordable data processing technologies, data-driven models experience an increasing recognition. The evaluation of such models requires the existence of the data-driven model itself as well as the availability of the data, which are both handled by the data analytics process that will be introduced in Section 2 of this paper. Furthermore, this example also illustrates the intention of this paper that is depicted in Figure 2: The *use case* of providing an indication on the RUL of a system imposes a *communication need*, i.e. the data-driven model requires sensor data and context data in order to be evaluated. Furthermore, it has to be decided, which system will take the responsibility of evaluating the model or if new systems, e.g. a central server in the avionics bay, will be introduced, i.e. for a given set of *physical architectures,* the suitable one has to be selected. As part of this paper, the generation of the set of possible physical architectures incorporating data analytics functions will be described. Then, criteria will be proposed that allow a profound decision for a physical architecture (*resulting physical architecture*) based on the inputs (blue arrows in Figure 2) described before by performing the *selection process*.

Although the descriptions given above might indicate that data analytics is specifically tailored for predictive maintenance, data analytics can be generally applied to a large variety of use cases in the aircraft cabin. It can, for example, increase the passenger's comfort, e.g. a meal served according to the passenger's preferences, as well the cabin operation, e.g. a meaningful indication on the boarding process [2].

The descriptions above show that the aircraft is not an independent entity, but is integrated in a wider scope of processes, e.g. the catering process as well as the maintenance process. A report that a seat actuator will fail soon, for example, will be forwarded and processed by a large number of other entities that are participating in the overall maintenance process in order to ensure that actuator will be repaired or replaced on time. Still, within this paper, the aircraft is considered as an isolated entity and its internal network architecture is investigated. This assumption does not preclude a data exchange with the outside, however, the corresponding air-to-ground interfaces are not considered in detail in this paper.

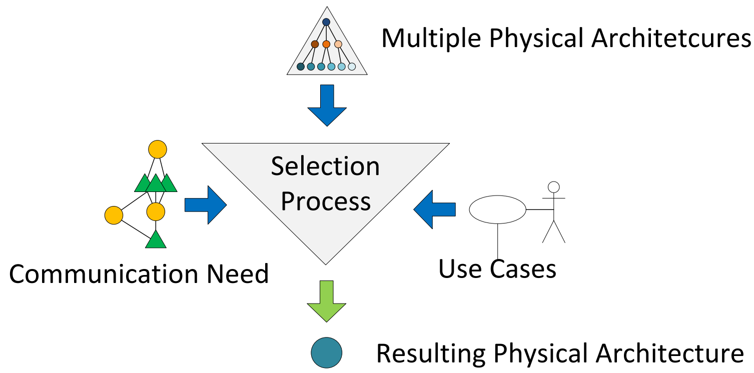

Figure 2. Overview of the Overall Network Architecture Derivation Process.

2 THE DATA ANALYTICS PROCESS AND THE CORRESPONDING NETWORK FUNCTIONS

Data analytics, as broached in the introduction, does not depict a single activity, but is composed of several activities, as shown in Figure 3 [3-5], referred to as the data analytics process. It allows gaining knowledge from pure data.

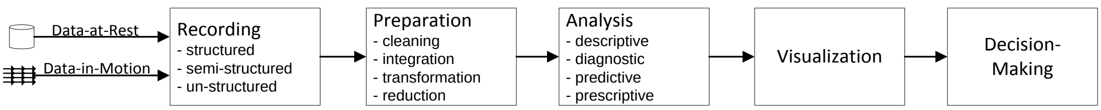

Figure 3. The Data Analytics Process.

The input for the data analytics process is data which can be available in different shapes, at-rest or in-motion. Data-at-rest describes datasets with a defined beginning and end, stored in files or databases. The underlying activity from which the data has been gathered, has already been taken place, thus, it represents historical data. In contrast, data-in-motion describes continuous data streams gathered from currently ongoing activities. This type of data is usually available via a network, e.g. as a TCP stream. The *recording* step is responsible for gathering both types of data and therefore has to tackle the challenges introduced by the respective data shapes. Examples for these challenges are, high data rates especially for data-in-motion and large datasets for data-at-rest. In both cases, it has to be considered that in real-world scenarios, the data is often unstructured, e.g. texts in natural language. In contrast, structured data refers to data which follows a pre-defined structure, e.g. measurements of a temperature sensor with a timestamp in a JSON format. After the data is available it has to be *prepared* before it can be analysed, this includes, for example, the standardization of data formats and rules for missing values. The third step describes the actual *analysis* of the data and can be categorized according to the following questions:

- Descriptive: What happened?
- Diagnostic: Why did it happen?
- Prognostic: What will happen?
- Prescriptive: How can we make it (not) happen?

The first two questions deal with history of the underlying activities whereas the latter questions predict the future progress of the activities, e.g. the RUL evaluation of a system. The predictions are nowadays often based on the data-driven models of the system which can be, for example, designed with machine learning methods, e.g. artificial neural networks. As these methods require historical data available as data-at-rest, the data analytics process can be further detailed as shown in Figure 4 [6].

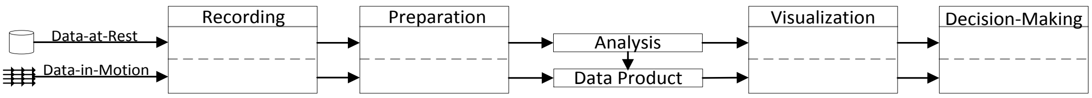

Figure 4. Evolved Data Analytics Process for Predictions.

The historical data is analysed with the goal of generating a data product (e.g. a trained artificial neural network) which encodes the gained knowledge from the analysis step and is used to evaluate data-in-motion, e.g. sensor values. The output of such a data product can, for example, be the RUL of a system. Within the *visualization* step, the analysis results are processed in order to allow a well-founded *decision-making*. If the decision is made by a human, the visualization step actually refers to the process of generating histograms, scatterplots or reports. However, it can be interpreted in a more general way if the embedding into a machine-to-machine (M2M) network is considered. In this case the analysis results can be translated into a command for a connected actuator, i.e. a decision is defined based on the analysis results.

As shown in Figure 5, the data analytics process can be translated into a generic functional flow a network has to implement if it incorporates data analytics. Data is generated by one or more *sources*, which is then *processed (F1)*. A *compute (F2)* function, which can for example depict the evaluation of a data product, collects the processed data of several sources and generates an analysis result which in turn is *processed for data sinks (F3)* that are interested in this result. The specializations of, for example, the generic function F2, i.e. the concrete functions that are realized by a data analytics network, are correspondingly denoted as F2(1) to F2(n).

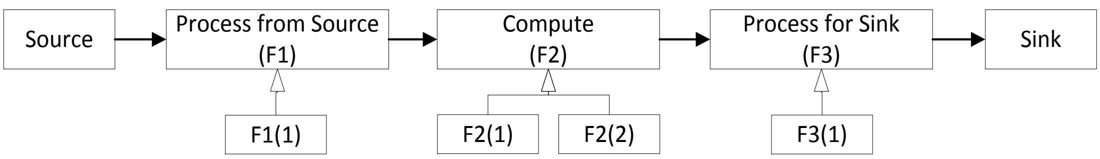

Figure 5. Generic Functional Flow for Data Analytics.

3 IDENTIFICATION OF COMMUNICATION NEED

The generic functional flow derived in the previous section introduces the notion of data sources and sinks, which will be described in more detail within the first part of this section. Furthermore, in the second part of this section, the logical linking of the sources and sinks, which results in a communication need, will be discussed. The goal of this section thus is the description of the input *communication need* to the architecture *selection process*.

3.1 Source and Sinks Layout

The output of the architecture selection process is a physical architecture, i.e. the architecture describes actual physical devices with a location within the aircraft. Therefore, in order to optimize the physical architecture, the spatial distribution of the data sources, e.g. sensors, and the data sinks, e.g. actuators or displaying devices, in the aircraft needs to be known. Depending on the location of the data sources requested by sink, the location of the device which implements, for example, the compute function of the functional flow (Figure 5) can be different as discussed later in Section 4 on the selection process. In order to encode the knowledge on the sources and sinks, the sources and sinks layout as exemplarily shown in Figure 6 is proposed. For this, the sources and sinks are placed in the Layout of Passenger Accommodation (LoPA) which allows depicting the spatial distribution of the sources and sinks including their multiplicities. The layout is not limited to the design phase of the network, but can also be used to properly define the sink interests in data that is generated by the sources. A temperature sensor, for example, placed in the aircraft cabin will likely be installed multiple times at different locations. Thus, the layout can help to specifically select the intended temperature sensor. Furthermore, the identification of all sources and sinks reveals the variety of interfaces that need to be implemented (*process from source, process for sink*) that need to be implemented in order to communicate with the respective devise.

The example layout in Figure 6 shows the sources and sinks for a predictive maintenance scenario in the aircraft cabin, namely for the prediction of the RUL of seat actuators inside business and first class seats. There exist multiple sensors inside the seat observing the actuators (power consumption and speed) as well as the passenger sitting on the seat (weight). In addition, a source, providing information about the flight attitude, is located in the avionics bay. The entirety of the available data should be used to calculate the RUL of each seat actuator and to display appropriate information on the overall seat status to the crew on a mobile HMI (sink), possibly enriched with operational instructions. In order to decide on a physical architecture implementing this functionality, further details need to be added to the sources and sinks, e.g. information on the data generation process, i.e. how often the data is generated, together with respective data size. Finally, collecting existing sources and sinks is not only an input for design criteria for the network design, but can also be used for the communication inside the project. It allows every stakeholder of the system to get an overview of the already available data and might trigger new interests in sources. Furthermore, such a collection of sources and sinks is not limited to the current network design, but can be reused for further network designs.

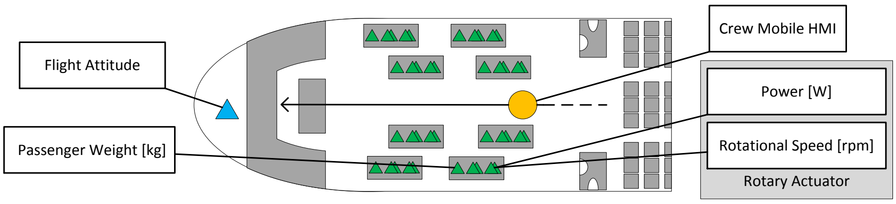

Figure 6. Example Source and Sink Layout for the Prediction of the RUL of a Seat Actuator.

3.2 Use Case driven Links of the Sources and Sinks

A network does not exist for an end to itself, but allows transmitting data and possibly implements additional networks functions. Thus, a list of source and sinks as introduced in the previous section is not sufficient to design a network, but the communication needs to be gathered. Sinks do not directly request data generated by a specific source, but sinks are interested in specific information independent of the source that provides this information, e.g. a temperature within a specific area of the aircraft cabin. Based on this requirement, a logical link between this sink and a possible source providing this information, e.g. a temperature sensor installed in this area, is established. This linking also shows if the available sources already cover all sink interests or if additional source need to be deployed. Furthermore, in order to be able to design an intelligent network being capable of dynamically adjusting the data streams, it has to be known how often a sink actually needs an update of the source. Depending on the use case, a sink might request data from a source with adjusted properties. An example for this property is the data generation process, i.e. a sink can request the data at a lower rate as it is generated at. In this case, network bandwidth could be saved by reducing the rate at which the source is transmitting the data, e.g. by dropping samples or taking their mean.

4 NETWORK ARCHITECTURE DERIVATION

In the previous sections two of the three inputs of the architecture selection process have been described, namely the use case and the communication need. However, in order to be able to select an architecture, a set of *physical architectures* needs to be supplied before. A guideline on how to generate this set is introduced in Section 4.1. Thereafter, in Section 4.2 the architecture selection process being based on multiple proposed network design criteria is discussed.

4.1 Generate the Set of possible Physical Architectures

The generation of the set of physical architectures is based on a three step approach [7], namely:

1. A *functional architecture* describes the interconnection of functions that the network has to fulfil. As described above, data analytics networks are based on a *functional flow (F1-F3)* that represents the data analytics process. Therefore, the functional architecture is composed of the specialized functions (*F1(i)-F3(i)*) of the generic functional flow discussed in Section 2.

2. Based on the functional architecture, multiple *logical architectures* can be derived describing which functions are grouped in order to be realized on the same physical device.
3. In the last step, multiple *physical architectures* can be derived for each logical architecture describing interconnected physical devices. By this, a physical architecture defines where the devices are located and which functions they implement. Furthermore, it describes the lower layer characteristic of the communication, i.e. the lower layers of the OSI communication model and based on which topology the communication is realized, e.g. a ring, star or meshed topology. The application layer protocols, however, are not part of the physical architecture and a decision for a specific protocol is therefore out-of-scope of this paper.

In order to elaborate on the 1-to-n relation between the functional and the logical architecture as well as the logical and physical architecture, the former is used for an example. For this, in Figure 7 a subset of two possible logical architectures for the seat actuator RUL scenario is shown, namely for the instantiation of the computing function (F2) that returns the RUL prediction based on the inputs of the sources. The architectures depict the two extrema that exist for the grouping and instantiation of functions namely, that all functions are realized in one central logical block, often denoted as (private) cloud computing (Figure 7, left) or that the functions are instantiated at or near the source and sink devices, often denoted as edge computing (Figure 7, right). This example can be extended to several functions allowing to realize not only the two extrema for the logical architecture, but an arbitrary combination of the cloud and edge computing approach.

Figure 7. Extreme Cases for the Grouping of the Functional Blocks. Left: Cloud Computing Approach. Right: Edge Computing Approach.

The variety of physical architectures for one logical architecture can be discussed in a similar way. There exist, for example, multiple ways to realize the communication between the logical blocks using a wired or wireless link. This also includes the dimensioning of the communication link.

4.2 Selection Process for a suitable Choice of a Physical Architectures

In the previous section two possible logical architectures for the example of the RUL prediction of the seat actuators have already been shown. Within this section, a process of selecting an appropriate and optimal architecture for the given use cases and communication needs is proposed based on a set of selection criteria.

In order to properly prepare the selection process, the communication need has to be further elaborated. It has been introduced as pairs of sources and sinks, i.e. logical links. However, the data is not transmitted directly from the source to the sink, but multiple steps of processing can take place in between, i.e. as described by the functional flow. Therefore, the paths through the functional architecture have to be evaluated as depicted in Figure 8. The sources and sinks are fixed in their position as they represent existing providers and consumers of data in the aircraft cabin. The remaining blocks represent the specialisations of the functions of the generic functional flow which need to be placed in the aircraft cabin with a possible grouping into the same device. For the optimization of such placement and grouping it has to be considered that the computing functions (F2) might require the data from multiple sources and that sinks might request the same computation results.

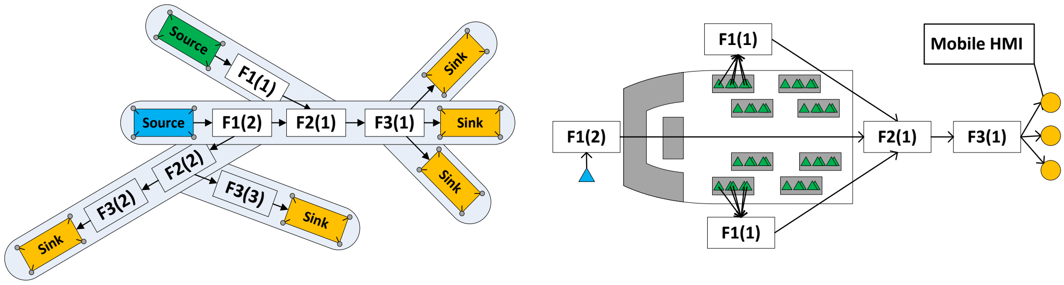

Figure 8. Visualization of the Data Analytics Functions that need to be allocated onto Physical Devices. Left:Generic Example. Right: Seat Actuator RUL Example.

In the seat actuator RUL prediction scenario, the generic functional architecture shown in Figure 8 (left) can be further elaborated as shown in Figure 8 (right) being a subset of the generic functional architecture as indicated by the colouring. One proposed selection criterion addresses the location of the sources and sinks together with the paths through the functional architecture as depicted in Figure 8. The prediction of the RUL of the seat actuator (F2(1)) requires the local sensor data of the actuator as well as the flight attitude. In this case, the sensors located inside the seat form a cluster in the aircraft layout. Thus, it is sensible to perform the computation for the RUL prediction within the seat (edge computing) instead of transmitting the data of each seat actuator to some other place, e.g. a central server in the avionics bay. In this case only the computation result, i.e. the RUL prediction, is shared within the network. This decision is strengthened by the fact that the sensors inside the actuators often sample at a far higher frequency than flight attitude data is generated. Thus, a logical architecture for the functional architecture shown in Figure 8 (right) satisfying the results of the discussion is shown in Figure 9.

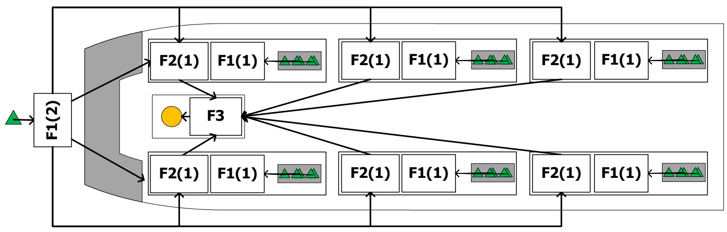

Figure 9. Selected Logical Architecture for the Seat Actuator RUL Scenario.

If, in contrast, only the seat occupation state of each seat is needed, e.g. for the prediction of the remaining boarding time, the data sources required for computation of the prediction are scattered over the layout. Therefore, no specific location can be assigned for the computation and the introduction of a centralized device (cloud computing) is sensible. However, this is not the only possibility: Instead of introducing a central instance, the computing function (F2) can also be instantiated in the sink, the mobile HMI in the example. For this decision, it has to be checked, if the computation result is required by several sinks or by just one (compare to Figure 8 (left) showing three sinks all requesting the computation result of F2(1)). In case of multiple sinks requesting the same data, a central solution is preferred as it allows reusing computation results instead of performing the same computation several times, i.e. in each sink.

In general, both approaches have their advantages and drawbacks: In the pure cloud computing approach, i.e. all functions (F1-F3) are instantiated in a central block, the interfaces of the sources are not harmonized near the respective devices. Thus, if a device has only a wired interface, a specialised wire needs to be installed from each device to the avionics bay, which is not necessary in edge computing approach. In this case, the process from source function (F1) can harmonize the interface which allows deploying a network being shared by all devices. However, the edge computing approach also increases the complexity of the edge devices.

The overall network design process can be summarized as depicted in Figure 10. The upper triangle describes how the set of all possible physical architectures is generated based on the functional flow for data analytics networks (Section 4.1). The lower triangle in turn describes how the optimal physical architecture is selected from the set of all physical architectures (blue circles) based on multiple design criteria (green arrows) as discussed in this section. The two proposed criteria, the location of the data sources and sinks and their path through the functional architecture as well as the data rate only represent example criteria as there exist far more. However, the completeness of the deduced criteria has to be evaluated and is therefore kept for future work.

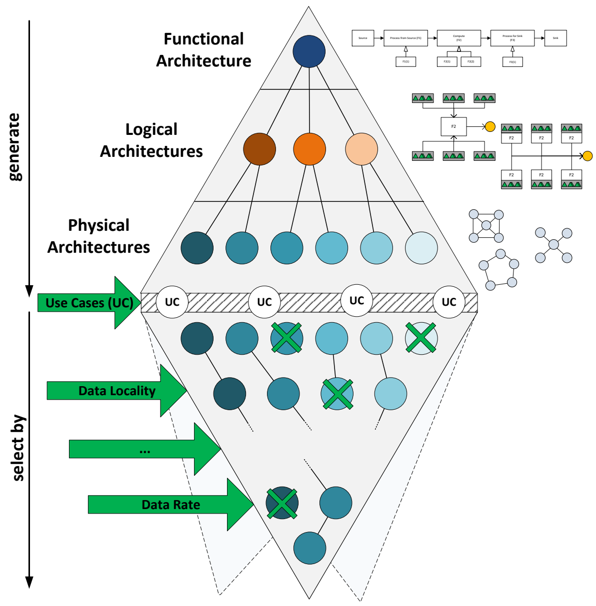

Figure 10. Overview of the Network Design Process.

5 CONCLUSION AND FUTURE WORK

Within this paper a network design process for data analytics networks being based on design criteria has been proposed. Prior to the elaboration of this approach no end-to-end process existed that allows tracing data analytics specific requirement through the complete design of the network. In order to establish this process, the data analytics process has been analysed and translated into a functional flow on which the functional architecture of each data analytics network is based on. In order to prepare a standardization of the development process, descriptions for inputs needed for the network design, namely the sources and sinks as well as the communication need between them, have been introduced. Based on these results, first a process has been described on how to generate a set of physical architectures that fulfil the requirements for the network. For this set of architecture, a criteria-based selection process has been proposed. In order to show the behaviour of the process, it has been evaluated using a predictive maintenance example in the aircraft cabin based on two criteria. The development of a complete set of criteria is left open for future work as well as the formalization of the design process allowing an integration into established systems engineering processes and, at the end, to automate the process. Furthermore, it has been mentioned in the introduction that the aircraft is integrated into a wider scope of processes, especially, when considering data analytics. Thus, in future work, criteria based on the requirements imposed by the embedding of the aircraft into a wider scope will be studied.

6 ACKNOWLEDGEMENT

This work was supported by the LuFo V-3 project “Die intelligente Kabine – vernetzte Informationen von hochentwickelten Kabinensystemen zu einem Gesamtzustandsbild für neue innovative digitale Fluggast-, Flugbegleiter- und Airline-Dienstleistungen (iCabin)” funded by the Federal Ministry for Economic Affairs and Energy based on the decision by the German Bundestag.

7 REFERENCES

[1] Mobley, R. Keith: “An introduction to predictive maintenance”. Elsevier, 2002.

[2] Michael Schultz: “The Seat Interference Potential as an Indicator for the Aircraft Boarding Progress”. In: SAE Technical Paper. SAE International, 2017.

[3] Jacques Roy: “Streaming Analytics with IBM Streams: Analyze More, Act Faster, and Get Continuous Insights”, Wiley, 2015.

[4] C. L. Philip Chen and Chun-Yang Zhang: “Data-intensive applications, challenges, techniques and technologies: A survey on Big Data”. In: Information Sciences 275, pp. 314–347, 2014.

[5] Hai Wang et al.:“Towards felicitous decision making: An overview on challenges and trends of Big Data”. In: Information Sciences 367, pp. 747–765, 2016.

[6] Cathy O’Neil and Rachel Schutt: “Doing Data Science: Straight Talk from the Frontline”. O’Reilly Media, 2013.

[7] Andreas Korff, Jesko G. Lamm and Tim Weilkiens: „Werkzeuge für den Schmied funktionaler Architekturen“. Tag des Systems Engineering, November 2011. English translation by J. Lamm.