TOWARD A DIGITAL PLATFORM FOR SPACECRAFT MANUFACTURING

Philipp Matthias Schäfer(1), Philipp Martin Fischer(2), Nico Brehm(3), Christian Erfurth(3), Andreas Gerndt(2), Kobkaew Opasjumruskit(1), and Diana Peters(1)

(1)*DLR Institute of Data Science*
Mälzerstraße 3, 07745 Jena, Germany
Email: p.schaefer@dlr.de
Email: kobkaew.opasjumruskit@dlr.de
Email: diana.peters@dlr.de

(2)*DLR Simulation and Software Technology*
Lilienthalplatz 7, 38108 Braunschweig, Germany
Email: philipp.fischer@dlr.de
Email: andreas.gerndt@dlr.de

(3)*EAH Jena*
Carl-Zeiss-Promenade 2, 07745 Jena, Germany
Email: nico.brehm@eah-jena.de
Email: christian.erfurth@eah-jena.de

ABSTRACT

Professionals of many disciplines are involved in a spacecraft mission. They all use different software tools that are tailored to their tasks and they share data in various ways among themselves. These data sharing activities form a network, which, given modern software engineering practices, offers a lot of opportunities for improvement: simplify data source discoverability, automate previously manual data sharing activities, and better make use available data sources. To simplify data source discoverability, we propose a digital platform with a service-oriented architecture. Such an architecture also helps to better make use of available data sources. Additionally, we present our projects that automate previously manual data sharing activities and that make better use of available data sources. With the development of the digital platform we aim at providing a significant reduction in resource expenditure, especially time expenditure, for spacecraft missions.

1. INTRODUCTION

Many disciplines are involved in a typical spacecraft mission. One can get an idea of the breadth of these disciplines by looking at the ECSS Document Tree [3]. Professionals of these disciplines use software tools that are tailored their tasks and share data in various ways among themselves, for example: Suppliers provide data sheets to their customers, telemetry and telecommand system engineers provide specifications to ground segment operators, etc.

The data sharing activities form an interdisciplinary data sharing network. Given modern software engineering practices, the existing networks in and between various organizations offer a lot of opportunities for improvement. Ways to improve the networks that are relevant to our work are

- **I1** to simplify data source discovery,
- **I2** to better make use of available data sources, and
- **I3** to automate manual data sharing activities.

Such improvements are a major part of the current drive toward further digitalization of economic and industrial processes.

To simplify data source discovery (**I1**), we propose a digital platform with a service-oriented architecture. The platform consists of framework software applications and discipline specific software applications (participating applications), where both sets of software applications provide services as a means for sharing data. The framework has several tasks: It manages discovery and access to services provided by the platform's software applications and it provides identity and rights management to the platform's software applications.

Currently, various cloud computing platforms, such as Amazon Web Services [1], Azure [4], and SAP Cloud Platform [6], offer such functionality. Due to data security concerns, some organizations, in particular in the space industry, discourage or even forbid their use. Therefore, an on-premises solution is required.

We design the digital platform so that multiple instances, each operated by a separate organization, can be connected to form a federated network of digital platforms. Such cooperation enables the use of data sources from any participating organization by any of the other organizations, provided that access rights are granted, and it allows us to better make use of available data sources (**I2**). An important aspect of making the platform instances interoperable is to establish conventions, for example by creating or using appropriate standards.

To automate manual data sharing activities (**I3**), we need to find automatable activities in existing processes; to better make use of available data sources (**I2**), we need to find unused data sources that can be used to improve existing processes.

Once we have identified projects suitable from a technical point of view, non-technical aspects have to be considered. Regulations, for example ITAR, and policy, for example safeguarding trade secrets, may pose problems, particularly, when data is supposed to be shared between organizations. Regulations or policy may preclude the use of data sources, require human compliance checks that prevent full automation, or require additional technical, mainly security, measures.

The rest of this paper has the following structure: In Section 2, we describe the architecture of our proposed digital platform. In Section 2.1, we present our ideas regarding the federated use of digital platforms. In Section 3, we present our projects to improve existing processes by automation of manual tasks and by further make use of available data sources. And finally, in Section 4, we provide a summary of the other sections.

2. DIGITAL PLATFORM

As stated above, a lot of software applications are used during a spacecraft's life-cycle, from its design to its disposal. Data needs to be shared between many of them. To enable software application to software application communication the involved software applications need to offer services, ranging from pure data sources—for example, an interface to a product catalog—to pure control interfaces—for example, an interface for the emergency shutdown of a machine. To manage the services from a central point (per organization), we propose a digital platform.

The digital platform we propose consists of framework software applications and participating software applications. The framework software applications manage discovery of and access information about services provided by any of the platform's software applications, that is, they provide information on where services with a given interface are located and how given services authenticate their users. The framework software applications also provide identity and right management for all of the platform's software applications. The platform's participating applications are all those management and engineering software applications that offer or use a service through the platform. Since all applications forming the digital platform communicate via services, the platform has a service-oriented architecture ([15], page 8).

To enable the improvements discussed above, the framework software applications of the digital platform should fulfill the following requirements:

- **R1** Enable the management of authentication and authorization for services
- **R2** Enable service discovery for participating software applications
- **R3** Enable humans to browse registered services
- **R4** Enable the exchange of information about services with other platforms

We need three abstract framework applications to satisfy these requirements: an identity and rights manager (**R1**), a service registry (**R2** and **R4**), and a human interface (**R3**). How the framework applications, as presented here, are implemented, whether, for example, as three separate applications or as part of a larger system with more functionality, is not relevant to our considerations.

Fig. 1 gives an overview of the proposed architecture, with the framework and its three software applications on the left and the participating software applications indicated on the right.

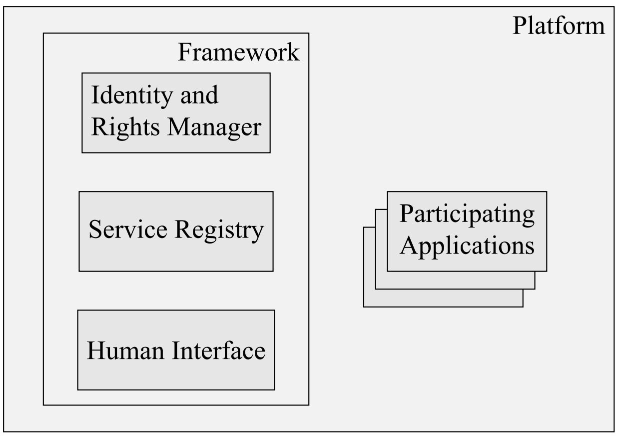

Figure 1: Architecture of Digital Platform

The identity and rights manager manages user identities and access rights for other software applications. Identities and rights can be registered and deregistered and rights can be granted to identities and revoked from identities. Software applications can then verify their users' identities and their users' access rights using a service provided by the identity and rights manager. It thus enables the platform to satisfy requirement **R1**.

The service registry stores information about services. Other applications can register their services and query for information about other services, that is, where to find and how to authenticate with them. Access control to service information is managed by the identity and rights manager. The service registry thus enables the platform to satisfy requirements **R2** and **R4**.

The human interface provides humans with read and write access to the other two framework software applications. It, therefore, enables the platform to satisfy requirement **R3**.

We already mentioned in the introduction that there are cloud platforms that provide all that functionality, but that we do not want to use them for policy reasons. The alternative is an on-premises solution. There are standalone software applications, even free and open source software applications, that offer the described functionality and can be set up on-premises. Such software applications and own developments will be the basis of our digital platform. We start by implementing our own instances of the framework software applications in order to allow for easy experimentation. However, we follow industry standards (REST APIs) for the interfaces of participating services that we develop as part of the use cases described in Section 3, so that adaptation to alternative framework software applications should require minimal effort.

2.1 Federation

Data sharing within organizations is common, though it is, as discussed above, not necessarily easy to keep track of all connections. To manage data sharing between two organizations poses further problems.

Looking at past and current developments of complex service oriented software architectures that involve different independent organizations, we observe that centralized, monolithic architectures are being replaced by decentralized, modular architectures, for example the microservice architecture. An example for a standard relevant to the former, centralized, approach is the now defunct UDDI standard [8], whereas we can find the modular approach thriving in all of today's major cloud computing platforms. However, these platforms are still centralized marketplaces for services.

Considering these developments in the field of large service-oriented software architectures, we intend to use the microservice architecture. This architectural paradigm allows more decentralized and independent development and operation of services. The reduced minimally required investments could lead to new developments in inter-organizational scenarios. Additionally, the microservice architecture better highlights the benefits of individual services, which should lead to more adoption.

Thinking about service-oriented software architecture as the foundation for an open marketplace where services are autonomously provided and consumed, the idea of federated digital platforms becomes obvious. Such a federated system consists of software artifacts that are created and provided by independent organizations [11]. As part of our future work, we will develop and propose an architecture for such federated digital platforms. We are planning to evaluate our ideas with various prototypes that are applied in inter-organizational spacecraft manufacturing environments. One important problem to solve in this context will be managing the service quality and data security requirements of the involved organizations.

3. USE CASES

To make use of our digital platform, we look at data sharing use cases that can be improved through digitalization (**I3**) and at new data sharing use cases (**I2**). Each use case involves developing services for existing or new software applications. By registering these services with the digital platform, the software applications become participating software applications.

The use cases we are currently working on are all related to the Model-Based Systems Engineering (MBSE) tool Virtual Satellite [13]. A modern web API for Virtual Satellite's database is a requirement for the realization of most use cases and is also worked on. Each of the following sections presents one of our use cases.

3.1 Reporting on MBSE projects

There are multiple standalone software applications that support MBSE for the design and development phases of a spacecraft mission, such as CDP [2], OCDT [5], VSD [10], RangeDB [12], and Virtual Satellite [13]. These applications all have in common—to the degree the authors could determine—that their features are aimed at engineers. However, the data they manage is relevant for other actors as well. Project managers need to keep track of a projects progress and some of that data is relevant for reports that have to be generated as part of the development and manufacturing process, for example the documents required by the process described in ECSS-M-ST-10C [7].

Functionality to create such reports would be misplaced, if implemented as part of the above mentioned standalone applications. Web-based tools, for example Valispace [9], better lend themselves to integrate reporting. However, tight integration of two applications targeting different user groups would go against the design principles of our digital platform.

For this reason, we develop a tool that creates reports and report snippets from MBSE projects. Virtual Satellite's web API will be the source for the necessary data. Given a standardized interface, the tool could eventually support arbitrary MBSE tools as sources, if they implement said interface.

3.2 Automatically Augmenting a Product Database with Supplier Data

Some companies keep databases with information about products of their suppliers. Product data is entered by hand, copied from data sheets, or imported automatically using electronic data sheets and web services. Companies obtain the data from suppliers or from specialized data brokers.

Those databases not only store information provided by the supplier, but also store information created by the company itself: values as measured (in contrast to values as stated in the specification), price history based on previous purchases, reliability information, etc.

We implemented a product database for spacecraft manufacturing. It is the basis for research into two methods of automatically augmenting the contained data:

- A standardized interface for sharing supplier product data within the satellite industry
- Computer aided extraction of product data from existing data sheets

3.3 Connecting MBSE Tools to a Product Database

In the previous section, we described product databases for parts sourced from suppliers. One application of such a product database is to give engineers direct access to reliable product information in an MBSE tool. This makes it unnecessary to copy information from data sheets and, thus, prevents errors and saves time.

We implemented a plug-in for the MBSE tool Virtual Satellite that uses the product database's interface to provide Virtual Satellite's users read and write access to product information.

3.4 Connecting CAD Tools to a Product Database

Virtual Satellite offers functionality to create three dimensional sketches of a satellite during the early design phase [14]. Users can define shapes—cuboids, spheres, cones, cylinders, and geometries imported from STL files—for components and place them in three dimensional space as a way to test configurations during concurrent engineering sessions.

Basic shapes—cuboids, spheres, cones, and cylinders—can be too abstract for some satellite component geometries, yet, sufficiently complex geometries can be modeled on the fly in a CAD tool. With Virtual Satellite, there is an MBSE tool that allows the import, the export, or both of geometric shapes from CAD tools [16].

Having an interface between the CAD tool and the MBSE tool allows to work on geometries for components of a particular system. With bidirectional interface between the MBSE tool and the product database, mentioned in the previous section, an indirect path to work on geometries for products in the product database exists. As an alternative, independent of the MBSE tool, we want to build a direct bidirectional interface between the CAD tool and the product database.

4. SUMMARY

Our goal is to help carry the current drive for further digitalization industrial and economic processes into the space sector. To achieve this, we work on a digital platform that helps organize and manage a growing network of software to software interfaces, and we look for opportunities for improving processes through digitalization, either by automation or by finding new applications for available data sources.

We identified use cases for improvement through digitalization, we created the prototypes to test of some of these ideas, and we started to work on a framework for a digital platform.

REFERENCES

[1] Amazon Web Services (AWS) - Cloud Computing Services. `https://aws.amazon.com/`. Accessed: 2018-06-27.

[2] Designing the Future of Your Complex Engineering Projects. `https://www.rheagroup.com/cdp`. Accessed: 2018-05-29.

[3] ECSS Document Tree (ECSS Architecture). `http://ecss.nl/standards/ecss-document-tree-and-status/`. Accessed: 2018-04-17.

[4] Microsoft Azure Cloud Computing Platform & Services. `https://azure.microsoft.com/`. Accessed: 2018-06-27.

[5] OCDT Community Portal. `https://ocdt.esa.int`. Accessed: 2018-05-29.

[6] Overview | SAP Cloud Platform. https://cloudplatform.sap.com/. Accessed: 2018-06-27.

[7] Space project management - Project planning and implementation (ECSS-M-ST-10C).

[8] UDDI | Online community for the Universal Description, Discovery, and Integration. http://uddi.xml.org. Accessed: 2018-06-27.

[9] Valispace - Develop better products. Together. http://www.valispace.com/. Accessed: 2018-05-29.

[10] VSD-Project | Virtual Spacecraft Design. https://www.vsd-project.org/. Accessed: 2018-05-29.

[11] Nico Brehm and Jorge Marx Gomez. Federated ERP-Systems on the basis of Web Services and P2P networks. *International Journal of Information Technology and Management*, 9(1):75–89, 2009.

[12] Harald Eisenmann, Claude Cazenave, and Thierry Noblet. RangeDB the product to meet the challenges of nowadays System Database. In *9th ESA Workshop on Simulation for European Space Programmes*, 2015.

[13] Philipp Martin Fischer, Meenakshi Deshmukh, Volker Maiwald, Dominik Quantius, Antonio Martelo Gomez, and Andreas Gerndt. Conceptual data model: A foundation for successful concurrent engineering. *Concurrent Engineering*, 26(1):55–76, November 2017.

[14] Philipp Martin Fischer, Robin Wolff, and Andreas Gerndt. Collaborative Satellite Configuration Supported by Interactive Visualization. In *2012 IEEE Aerospace Conference*. IEEE, 2012.

[15] Sam Newmann. *Building Microservices*. O'Reilly UK Ltd., 2016.

[16] Katharina Roventa. Integration von mechanischen Arbeitsprozessen in den modellbasierten Systementwurf. Master's thesis, Universität Stuttgart, 2018.