Margin requirements for first page
Paper size this page A4

99 pt
1.375 in
34.9 mm

# Modelling Artifact Interdependencies in Technical Documentation as a base for Maintenance Assistance Systems ⋆

**Melanie Stolze * Mike Barth ** Christian Diedrich *****

* *Institute for Automation and Communication, Magdeburg, Germany (e-mail: melanie.stolze@ifak.eu).*
** *Karlsruhe Institute of Technology, Karlsruhe, Germany (e-mail: mike.barth@kit.edu)*
*** *Otto von Guericke University Magdeburg, Magdeburg, Germany (e-mail: christian.diedrich@ovgu.de)*

**Abstract:** To increase the effectiveness of maintenance of production machines, assistance systems are being developed to guide personnel more quickly and safely. The knowledge partly originates from technical documentation (TD), which today is mostly document-based. With the Model Based Systems Engineering (MBSE) approach, a shift towards a model-based TD emerges. This paper analyses how an MBSE model can be used for assistance systems and what is necessary to facilitate it. A theoretical structure model for TD is described which shows the interdependencies between maintenance tasks and artifacts of a TD. The modelling is done by using Systems Modelling Language and the Asset Administration Shell.

*Keywords:* Intelligent Maintenance Systems, Model Based Systems Engineering, Systems Modelling Language, Asset Administration Shell, Intelligent Information Request and Delivery Standard

43 pt
0.597 in
15.2 mm

## 1. INTRODUCTION

Due to Industry 4.0, Afolalu et al. (2024) states that production machinery becomes more complex, and at the same time, automation and digitalization of machines and production processes increase. This requires higher-level human expertise for machine operation and maintenance. To support people in handling machines, the development of assistance systems (AS) arises. Examples for such AS can be found in Koppaetzky and Nicklas (2013), Schlagowski et al. (2017), and Dhiman and Röcker (2019). Maintenance Assistance Systems (MAS),a special kind of AS, are made up of dynamic and static data. Afolalu et al. (2024) depicts the usage of data-driven methods and recorded live data of machinery that help to diagnose or predict a failure and in the best case allows to locate the cause of the failure. After the prediction the cause of a possible fault needs to be rectified. To this purpose, information and equipment must be organized and the tasks to be carried out defined, taking into account the safety of people. The information to specify the tasks is provided by the technical documentation (TD) of a machine as well as by the experience of the employees. Nowadays, Robers and Fritz (2022) points out that this information is extracted manually but with a look into the future there must be an intelligent information base which shows user-defined and context-related information regarding a specific request.
Due to the current increasing shortage of skilled employees, the knowledge based on experiences decreases. To capture expert knowledge, the focus is on the model-based provision of TD that can be flexibly adapted during the machine life cycle. The Model Based Systems Engineering (MBSE) approach, which relies on the model-based development of systems, provides assistance here. Furthermore, artifacts of the TD which evolve mostly from the multidisciplinary engineering phase needs to be more interrelated with the tasks in the operational and maintenance phase. This paper describes an approach on how to relate multiple artifacts of TD with the help of Systems Modelling Language (SysML), Asset Administration Shell (AAS) and Intelligent Information Request and Delivery Standard (iiRDS) to extend the scope of MBSE to the operational phase, supporting the vision of INCOSE (2021) and to facilitate the development of MAS.
Section 2 presents challenges and requirements in conjunction with the TD available today. Considering the mentioned requirements, Sec. 3 describes the current state of the art to shift TD from document-based to model-based. Furthermore, multiple methods of machinery development are described and combined into a structure model of TD, designed by the authors. With the help of a use case, Sec. 4 explains the structure model and shows its usage based on SysML, AAS and iiRDS. Finally, the paper concludes with a summary and an outlook on future work.

43 pt
0.597 in
15.2 mm

## 2. CHALLENGES AND REQUIREMENTS

The transition to a more model-based TD presents several challenges and requirements (Req.). The following challenges and their listed requirements are taken from

⋆ Funded by Federal Ministry of Economic Affairs and Climate Action, Germany

71 pt
0.986 in
25 mm

the detailed analysis of Stolze et al. (2024), as they are considered the basis for the research presented.

**Challenge 1** INCOSE (2021) and Kunnen et al. (2019) highlight that in most industry domains engineering processes are still document-based. Artifacts of a TD can originate from different data sources such as paper based or PDF documents, XLSX files, or relational databases (Stolze et al. (2024)). Not all of them are machine readable and hence not automatically exploitable. Furthermore, Barthelmey et al. (2014) points out that after the commissioning of a machine, the document-based approach hinders the synchronisation of the TD with the as-build status of a machine.

- Req. 1.1: Every artifact of the TD should be model-based.
- Req. 1.2: The modelled artifacts should contain, allow or link semantic definitions, respectively.
- Req. 1.3: The model-based TD must be able to be flexibly adapted over the course of the product life cycle.

**Challenge 2** Stolze et al. (2024) and Robers and Fritz (2022) mention that during planning or execution of a maintenance task, relationships between artifacts of a TD are manually identified because they are separated from each other.

- Req. 2.1: Relationships between information of artifacts must be formalised.
- Req. 2.2: Relationships must be semantically interpretable.
- Req. 2.3: The information relevant to a maintenance activity must be extracted automatically.



**Challenge 3** INCOSE (2021) states that there are multiple modelling languages and domain specific engineering tools to create a model-based TD.

- Req. 3.1: The heterogeneity of existing modelling languages must be taken into account.
- Req. 3.2: The technology-independent interpretation of the information must be guaranteed as a result of the heterogeneity.

The aforementioned requirements can be considered when using the approaches described in Sec. 3.2. However, most MBSE methods focus on the engineering phase of a system. To reuse the engineering system model in operation and especially in maintenance the following research questions evolve:

What is necessary to reuse a system model, preserved via MBSE, to use it for MAS? Especially, how can the maintenance information be modelled as artifacts and what types of relationship exist between maintenance artifacts and the rest of the TD? How can the relationships be modelled and annotated with semantic to enable the extraction through a MAS in order to execute a special maintenance task?

## 3. STATE OF THE ART

With the emergence of digital twins, the modelling of data becomes more and more important. MBSE and long-existing modelling languages such as SysML (OMG (2024)) as well as the AAS (Fuchs et al. (2019)), which has only been around for a few years, promote the way to a fully model-based TD. Furthermore, the usage of ontologies facilitates semantic interpretation and thus the filtering of relevant data. In particular, the iiRDS ontology (iiRDS Consortium (2025)) can be used to semantically enhance a TD. The following sections show the state of the art of the methods and technologies mentioned above.

### *3.1 Methods in machinery development*

The development of machinery is subject to a product life cycle,as defined in ISO (1999). In each life cycle phase, both document-based and model-based artifacts are created and shape the TD of a machine.
To organize and structure the development of machinery, multiple methods evolved over time. A well known method is the general V-Model, which can be applied to the development of mechatronic and cyber-physical systems (VDI (2021)). The levels of the V-model begin with the requirement engineering, followed by the system architecture design and the detailed definitions of the mechatronic domains. After this, the implementation and integration follow with the handover of the developed product itself and its TD as the last step. A similar method is described in the VDI (2019) guideline. Other approaches for product development summarized by Ponn and Lindemann (2011), are the pyramid model of product specification, the construction model space and the Munich Product Specification Model (MPSM).
All approaches mentioned above share the same idea. They structure a system or product model into discrete levels or spaces. However, all of the methods mentioned before and MBSE are applicable particularly in the early phases up to commissioning. It needs to be considered that after the commissioning the TD of a machine is used, updated or enlarged in its content.



### *3.2 Systems Modelling Language*

As stated in OMG (2024), SysML was developed to facilitate the modelling of suitable system models for MBSE. Its specification allows the extension of SysML base elements to enable a broader semantic definition of model elements. Multiple research groups contributed use case specific extensions of SysML. These include, for example, the SysML4Mechatronics library (Kernschmidt (2019)), to model mechatronic systems with their operating principles between software, electrical and mechanical engineering. Furthermore, Kunnen et al. (2019) presents a SysML extension to model failures evolved by risk analyses. Another extension developed by Bremer (2020) shifts the focus from product-oriented to production-oriented MBSE by modelling production systems in conjunction with the product and process.
The possibilities for extending SysML satisfy the requirements (Req.) 1.1, 1.2 and 2.1 of Sec. 2 and its broad application characterizes the language to use it in the concept explained in Sec. 4.

### *3.3 Asset Administration Shell*

Today, most modelling languages are used independently of each other because every language has its advantages

in a special life cycle phase of a machine. Whereas SysML and Automation Markup Language (AML) (Drath (2021)) are predominantly used in the engineering phase of a machine, OPC Unified Architecture (OPC UA) (Mahnke et al. (2009)) was initially developed for the machine's operational phase. To overcome the heterogeneity of all modelling technologies the AAS was developed (Fuchs et al. (2019)).
However, the AAS is an additional member in the group of modelling languages and this makes it difficult to differentiate what language can be used for which use case. Due to this in Drath et al. (2023) the heads of AML, OPC UA and AAS discussed possibilities to make the three modelling approaches interoperable. This means that requirement 3.1 from Sec. 2 can theoretically be fulfilled with the exception of SysML. A closer look at the gaps and their solution can be found in Sec. 4.

### *3.4 Intelligent Information Request and Delivery Standard*

Robers and Fritz (2022) point out that the potential of information as it is created and used today has great opportunity for improvement. Due to that, the iiRDS standard evolved to make the TD of products such as machinery, machine-interpretable. The standard can be applied to brownfield solutions. Therefore documents can be annotated with intelligent information semantics and then be stored in a particular iiRDS package. (Robers and Fritz (2022))
The possibility of iiRDS to add role metadata to individual information enables the creation of different views on information. Furthermore, it enables the provision of information in an "[..] individual, user-oriented, and context-related way [..]" to facilitate "[..] interactive and adaptive human-machine or machine-machine communication [..]". (Robers and Fritz (2022))
Together with one of the latest AAS submodels named "Intelligent Information for Use" (see IDTA (2025)), a way to harmonize SysML, AAS and iiRDS is given. The combined use of all three technologies can fulfil Req. 1.3 of Sec. 2 to enable flexible model-based TD over the life cycle of a machine. In addition, iiRDS provides the specification of semantically enriched relationship definitions that enable a refinement of the general SysML relations and thus the fulfilment of Req. 2.2.

## 4. CONCEPTUAL WORK

### *4.1 Structure Model for Technical Documentation*

Comparing the multiple models of Sec. 3.1 shows many similarities. All of them categorize the information into four main levels or spaces in a rough way, but the development of a system involves many artifacts from different engineering disciplines. Furthermore, in the operational phase multiple artifacts are generated, e.g. work instructions. Usually, executing work instructions requires additional information in the form of engineering artifacts. For this reason, the authors suggest that artifacts from the operational phase must be linked to the artifacts from the engineering phase. Therefore, the authors designed an extended structure model to group and link artifacts of a TD up to the operational phase.

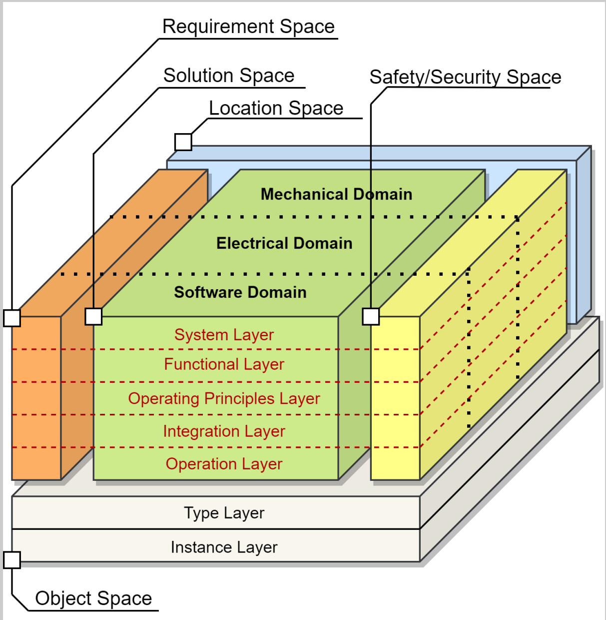


Fig. 1. Structure Model for Technical Documentation

The extended Structure Model for Technical Documentation (SMTD), named by the authors, is shown in Fig. 1. It consists of 5 main spaces which are *Requirement, Location, Solution, Safety/Security* and *Object Space*. All spaces are defined by the authors, except the *Requirement* and *Solution Space*, which are derived by combining the methods of Sec. 3.1.

At the beginning of system development, evolved requirements will be documented through suitable artifacts and grouped in the *Requirement Space*. In parallel, the structure of the location where the system will ultimately be situated is also specified. In complex systems, the location can be split into sub locations to pinpoint subsystems. This space is called *Location Space* and has interdependencies to the:

- *Requirements Space*: to link location specific requirements
- *Solution Space*: where the developed system with its sub-systems are linked to its later location
- *Safety/Security Space*: for the necessity to link risks to a location
- *Object Space*: in order to link single objects to locations.

The *Safety/Security Space* contains all artifacts derived from risk analysis methods, which can be linked to requirements and solution models as well as to single objects from the *Object Space*. All three, the *Requirement, Solution and Safety/Security Space* are divided into layers called *System, Functional, Operating Principles, Integration,* and *Operation Layer*. The first four layers are known from the models and methods of Sec. 3.1. The authors added the fifth to highlight the design of operation models, like workflows in maintenance, and their dependency to the artifacts of the upper layers and surrounding spaces. For example, when exchanging a drive, the electrical circuit plan from the *Integration Layer* in the *Electrical Domain* as well as a mechanical explosion drawing from the same layer but of the *Mechanical Domain* are necessary. The

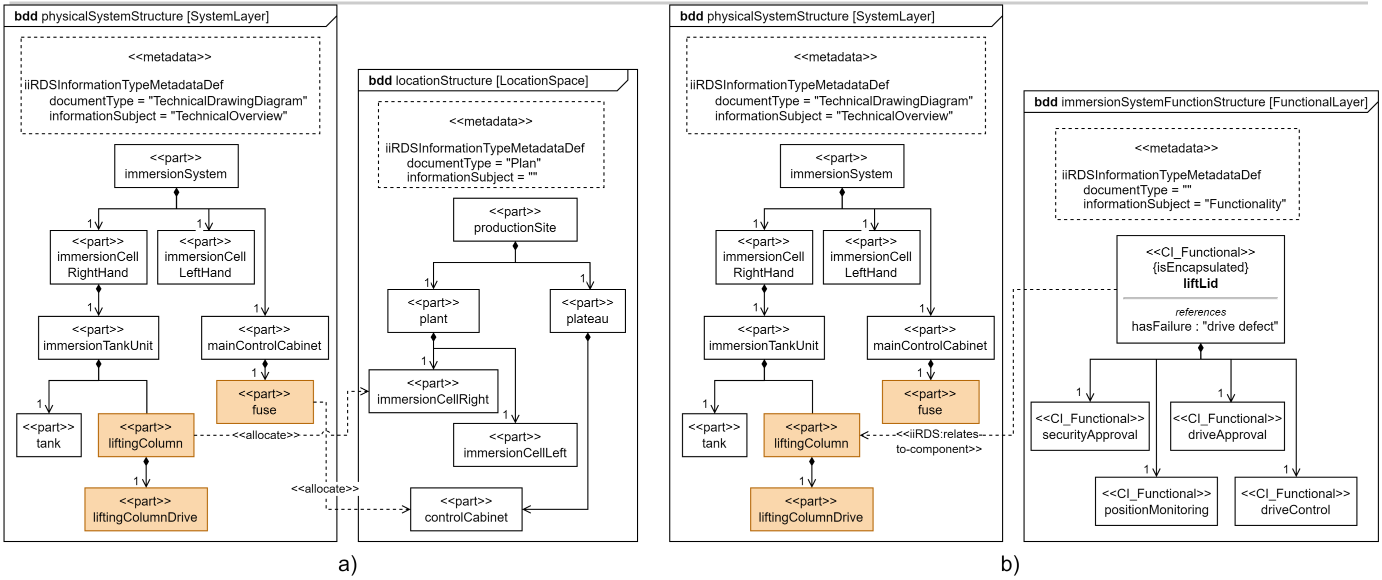


Fig. 2. a) Interdependencies between *System Layer* and *Location Space*, b) Interdependencies between *System* and *Functional Layer*

mentioned domains plus the *Software Domain* are introduced in order to subdivide the three mentioned spaces. An overlapping between both *Software* and *Electrical* as well as *Electrical* and *Mechanical Domain* is intended. At the bottom of the structure model an *Object Space* is added which is split into a *Type* and *Instance Space*. This is done in order to integrate the AAS concept, which facilitates an object-oriented modelling, but is not further explained in this paper. An object in the *Object Space* can be linked to all surrounded spaces.



The structuring of the model that has been developed is intended to help the artifact designers to assign developed artifacts to packages and structure the information for the later development of an MAS.

### *4.2 Integration of system models interdependencies in the SMTD*

The use case used to explain the application of the concept developed originates from an immersion system. Therein a tank unit with a lifting column is installed. The lid of the tank is lifted by a drive mounted on the lifting column. The power supply to the drive is protected by a fuse, which is located in a separate control cabinet. In the production process, a defect occurs in the drive, and the maintainer now has to search and resolve the issue.

One main function of an MAS should be the provision of necessary information to resolve the issue. Due to this, the focus in the concept lies on modelling maintenance workflows so that the artifacts which are modelled in the engineering phase are seen as a prerequisite for this use case. In reference to this, these models are not explained in detail. Furthermore, only parts of the overall model-based TD can be shown to explain the principles of the concept. To see in which space or layer of the SMTD the model occurs, all shown SysML diagram titles contain the name of the space or layer in rectangular brackets.

The concept defines SysML as a central point in modelling the TD because it fulfils most of the mentioned requirements of Sec. 2. The AAS is used to store additional data that can not be modelled within SysML. Furthermore, iiRDS is used to allow a more specific semantic definition of SysML elements. With the possibility to extend SysML, it can be jointly used with AAS and iiRDS.

Fig. 2 shows the system structure related to the use case and its interdependencies with the *Location Space* (Fig. 2a) and *Functional Layer* (Fig. 2b). The first defined SysML extension affects the semantic annotation of SysML diagrams itself. E.g., every block definition diagram (bdd) in Fig. 2 contains metadata which is semantically enriched by the iiRDS classes Document Type and Information Subject. For example, the Document Type of the system structure is semantically defined as *TechnicalDrawingDiagram* in iiRDS whereas the Information Subject is defined as *TechnicalOverview*. If there are no suitable values for one iiRDS attribute, they need to be extended or left empty, as is shown in the location structure of Fig. 2a.

43 pt
0.597 in
15.2 mm

Additionally, Fig. 2a depicts the interdependency between the *System Layer* and *Location Space*, e.g. the fuse in the system structure is allocated to the control cabinet in the location structure. The important components for the use case are highlighted.

The same applies to the interdependencies between the *System* and *Functional Layer* where the function *liftLid* is allocated to the lifting column (Fig. 2b). Due to the broad application of SysML the general relations like *allocate* are too abstract to get dedicated information for a MAS. For this reason, the concept also specifies new SysML relationships based on the iiRDS specification. In the example mentioned before the term *allocate* is replaced through *iiRDS:relates-to-component* (Fig. 2b). In the concept this relationship is used for every SysML element that contains information belonging to a component, see also Fig. 4.

The *Safety/Security Space* stores relevant models for risk management, like failure descriptions shown in Fig. 3. Related to the use case, only safety aspects are analysed but the same applies to security risks in maintenance tasks. Fig. 3 shows an example of a failure for the lifting

71 pt
0.986 in
25 mm

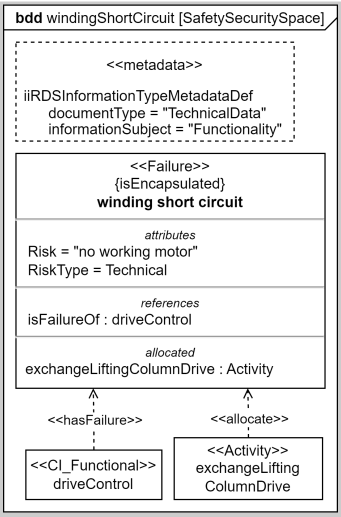


Fig. 3. Interdependencies between *Functional*, *Operation Layer* and *Safety/Security Space*

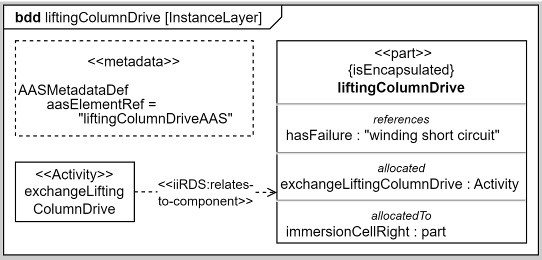


Fig. 4. Interdependencies between *Operational, Functional Layer*, and *Object Space* as well as to non-SysML artifacts

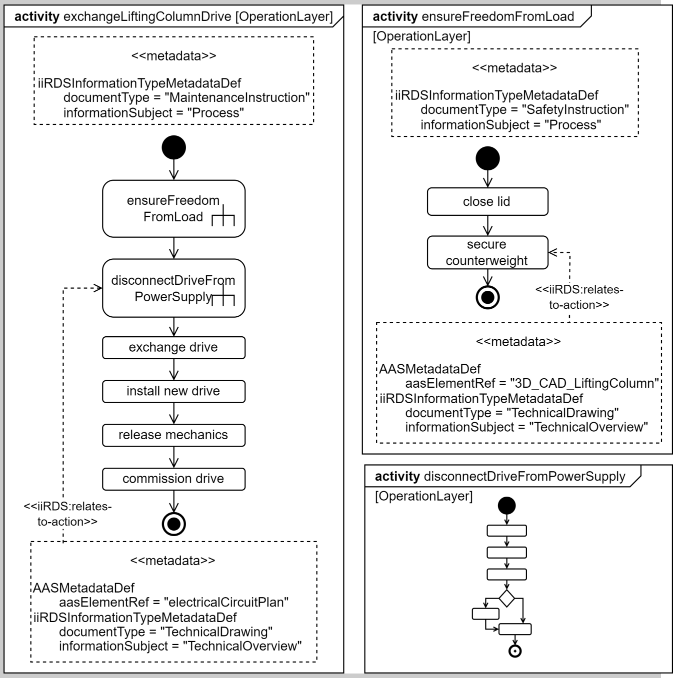


Fig. 5. Interdependencies between *Operation Layer* and non-SysML artifacts

column drive that leads to the drive defect. The specified metadata categorize this SysML model as technical data which covers a functionality. The interdependencies show that this failure is related to the drive control function and that the activity *exchangeLiftingColumnDrive* is allocated to it. In this case, no iiRDS specialized relationship is used. Further interdependencies of the use case occur between the *Object Space* and its surrounding spaces. To get more details about one object the concept uses encapsulated bdd's wherein all interdependencies to one component are displayed at a central point. For example, Fig. 4 shows the definition of the column drive in the *Instance Layer*. The part definition contains all referenced elements. Additionally, the diagram integrates a further SysML metadata extension, which is the integration of a link to an AAS or AAS element through a globally unique identifier. By following this link, additional information on the lifting column drive can be received, such as an OPC UA nodeset, AML file (see Gudder et al. (2024)) or CAD model, which can not be modelled in SysML. The authors suggest using SysML in the planning and concept phase of a machine, while AML shows its advantages in the integration phase thanks to its detailed information models (see Drath (2021)). In this way, both modelling languages complement each other.

The last and most relevant example for the use case is the *Operation Layer*. Therein all activities to maintain a system with its possible hazards for personnel are modelled with the help of SysML activity diagrams. In Fig. 5 the main activity to exchange the column drive with its sub-activities (ensure the freedom from load and disconnect the drive from the power supply) are modelled.

For single activities and actions, the maintainer needs additional information. For example, the electrical connection and the location of the fuse is important, to disconnect the drive from power supply. For this reason, an annotated metadata element in SysML is defined which defines the location in the AAS, where the electrical circuit plan for the activity *disconnectDriveFromPowerSupply* is stored as well as the iiRDS semantic definition of the artifact like in the examples before. Furthermore, an annotated metadata element is specified and linked to the action *secure counterweight*. This metadata element references a CAD model which can help to find the element to secure the counterweight of the lifting column. The activity diagrams itself are semantically enriched with iiRDS Document Types like *MaintenanceInstruction* for the drive exchange and *SafetyInstruction* which needs to be considered during the exchange. The link between the metadata element and the activities or actions is specialized through the *iiRDS:relates-to-action* relationship.

Following the diagrams, starting at the entry point that the drive control has a failure, an MAS can display all related failure possibilities to the maintainer. In this use case the failure occurred due to a winding short circuit of the drive. With the failure description in the *Safety/Security Space* the related maintenance task can be found. In parallel, the *Object Space* shows the physical location of the drive which is the immersion cell on the right hand of the immersion system. The activity to exchange the drive shows the necessary steps for it. For every step the MAS can show the additional artifacts linked to an action or activity to help the maintainer in efficiently as well as safely execute the task.

The MAS has to automatically derive the information described from the model-based TD. Furthermore, it has to interpret the different modelling languages (SysML, AAS and iiRDS). For this, the authors are investigating the implementation of the presented concept to validate it against still open requirement 2.3 and 3.2 of Sec. 2.

## 5. CONCLUSION AND OUTLOOK

There is a huge interest in shifting the product's technical documentation (TD) from document- to model-based to make it more modular and thus flexible as well as to enrich it with information over the product life cycle. This new TD can be used to improve the knowledge base of Maintenance Assistance Systems. The presented concept defines an extended theoretical model to structure a TD and to broaden its content with necessary information for the operational phase of machinery. The concept theoretically defines the model-based TD by combining and extending the respected standards SysML, AAS and iiRDS. With the annotation of additional metadata, user-defined and context-related information for use can be extracted. The authors are currently implementing the presented concept technically using SysML, AAS and iiRDS to enable the automatic extraction of the information required for a specific maintenance task. Furthermore, a methodology to create and link multiple artifacts of the TD especially for maintenance will be designed and steps in the methodology to be automated will be defined. In addition, a proposal is being developed that enables the consolidation of several information models of all stakeholders involved in the life cycle of a machine.



## ACKNOWLEDGEMENTS

This work was funded in part by the project TwinMaP under contract 13IK028G funded by the Federal Ministry of Economic Affairs and Climate Action, Germany.

## REFERENCES

Afolalu, S.A., Nwankwo, S.O., and Azeez, T.M. (2024). Evolution of Maintenance Processes in Industry 4.0- A Review. In *2024 IEEE SEB4SDG*. doi:10.1109/SEB4SDG60871.2024.10629886.

Barthelmey, A., Störkle, D., Kuhlenkötter, B., and Deuse, J. (2014). Cyber Physical Systems for Life Cycle Continuous Technical Documentation of Manufacturing Facilities. *Procedia CIRP*, 17. doi:10.1016/j.procir.2014.01.050.

Bremer, C. (2020). *Systematics for modeling flexible production systems in Model-Based Systems Engineering (German)*. Ph.D. thesis, Universität Paderborn.

Dhiman, H. and Röcker, C. (2019). Worker Assistance in Smart Production Environments Using Pervasive Technologies. In *2019 IEEE PerCom Workshops*. doi:10.1109/PERCOMW.2019.8730771.

Drath, R. (2021). *AutomationML: The Industrial Cookbook*. De Gruyter Oldenbourg, Berlin, Boston. doi:doi:10.1515/9783110745979.

Drath, R., Mosch, C., Hoppe, S., Faath, A., Barnstedt, E., Fiebiger, B., and Schlögl, W. (2023). Discussion paper - interoperability with the asset administration ahell, opc ua and automationml (german).

Fuchs, J., Schmidt, J., Franke, J., Rehman, K., Sauer, M., and Karnouskos, S. (2019). I4.0-compliant integration of assets utilizing the Asset Administration Shell. In *2019 24th IEEE ETFA*. doi:10.1109/ETFA.2019.8869255.

Gudder, R., Hoffmann, D., Hünecke, P., and Lüder, A. (2024). Integrated Engineering Data Transformation: An AutomationML-Based Approach for Efficient Data Exchange. In *2024 IEEE ETFA*. doi:10.1109/ETFA61755.2024.10710965.

IDTA (2025). IDTA 02063-1-0 - Intelligent Information for Use. URL `https://industrialdigitaltwin.org/wp-content/uploads/2025/03/IDTA-02063-1-0_Submodel__IntelligentInformationForUse.pdf`.

iiRDS Consortium (2025). tekom iiRDS Standard. URL `https://www.iirds.org/materials/version-12`.

INCOSE (2021). Systems Engineering Vision 2035 - Engineering Solutions for a Better World. URL `https://www.incose.org/docs/default-source/se-vision/incose-se-vision-2035.pdf?sfvrsn=e32063c7_10`.

ISO (1999). ISO 15226:2017-03 - Technical product documentation - Life cycle model and allocation of documents.

Kernschmidt, K.E.T. (2019). *Interdisciplinary structural modeling of mechatronic production systems using SysML4Mechatronics*. Ph.D. thesis, Technische Universität München.

Koppaetzky, N. and Nicklas, D. (2013). Towards a model-based approach for context-aware assistance systems in offshore operations. In *2013 IEEE PERCOM Workshops*. doi:10.1109/PerComW.2013.6529456.

Kunnen, S.G., Adamenko, D., Pluhnau, R., and Nagarajah, A. (2019). An Approach to Integrate Risk Management in Cross-structure SysML-models. *Proceedings of the Design Society: ICED*.

Mahnke, W., Leitner, S.H., and Damm, M. (2009). *OPC Unified Architecture*. Springer, Berlin, Heidelberg. doi:10.1007/978-3-540-68899-0. URL `http://link.springer.com/10.1007/978-3-540-68899-0`.

OMG (2024). OMG Systems Modeling Language™ (SysML ) - Part 1: Language Specification Version 2.0 Beta 2.

Ponn, J. and Lindemann, U. (2011). *Concept development and design of technical products (German)*. Springer Berlin Heidelberg.

Robers, R. and Fritz, M. (2022). From Manuals to "Information on Demand" with iiRDS. URL `https://www.iirds.org/fileadmin/downloads/documents/Whitepaper_2-2022_iiRDS.pdf`.

Schlagowski, R., Merkel, L., and Meitinger, C. (2017). Design of an assistant system for industrial maintenance tasks and implementation of a prototype using augmented reality. In *2017 IEEE IEEM*. doi:10.1109/IEEM.2017.8289899. ISSN: 2157-362X.

Stolze, M., Barth, M., and Diedrich, C. (2024). Semantic linking of information models for machines and systems - a system model for technical documentation (german). In *AUTOMATION*, volume VDI report 2437. Düsseldorf.

VDI (2019). VDI 2221 Part 1 - Design of technical products and systems - Model of product design.

VDI (2021). VDI/VDE 2206 - Development of mechatronic and cyber-physical systems.