*Article Identifier*
*DOI: 10.22032/dbt.58900*
*URN: urn:nbn:de:gbv:ilm1-2023isc-059:7*

*60th ILMENAU SCIENTIFIC COLLOQUIUM*
*Technische Universität Ilmenau, 4 – 8 September 2023*
*URN: urn:nbn:de:gbv:ilm1-2023isc:1*

# DIGITAL TWIN IN INDUSTRIAL APPLICATIONS – HOW MODEL-BASED SYSTEMS ENGINEERING (MBSE) AND ASSET ADMINISTRATION SHELL (AAS) COMPLEMENT EACH OTHER

*Fabian Geibel*

Bosch Rexroth AG, Lohr am Main, Germany

## ABSTRACT

In the development, production and usage of cyber-physical systems, the number of stakeholders, involved interfaces and volatile environmental conditions is constantly rising. In addition, use cases require more consideration of the entire system life cycle. This significantly increases the administration effort and forms a barrier for the digital transformation of industrial companies. While model-based systems engineering (MBSE) addresses internal challenges within the product development and Asset Administration Shell (AAS) addresses vendor independent information exchange and interoperability, both approaches need to be coupled to address today's challenges. In this publication typical tasks within product development are discussed: "search the right information", "integrate the right information" and "provide the right information". It is shown how they were approached today, without the alignment of MBSE and AAS, what technological concepts exists to address the challenges and how the tasks are realized by an alignment of MBSE and AAS.

***Index Terms –*** Digital Twin, Asset Administration Shell (AAS), Model-Based Systems Engineering (MBSE)

## 1. INTRODUCTION

In this publication the research question is approached by a concrete industrial use case: a motor-pump group is the system under development using MBSE. The logical architecture consisting of an electric motor (sub-system 1) that drives a hydraulic pump (sub-system 2) is shown in Figure 1.

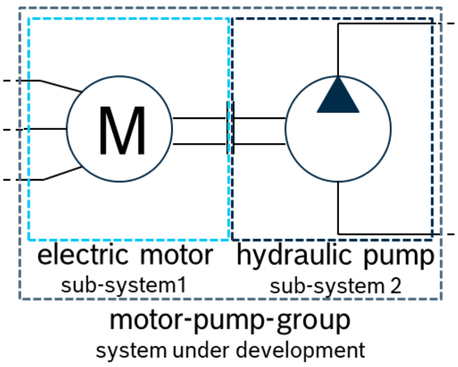


*Figure 1: Architecture of system under development – motor-pump group*

An electric motor must be selected that suits the requirements shown in Table 1 and integrated into the system model.

*Table 1: Required properties for the electric motor*

| | |
|---|---|
| Nominal speed | > 5000 min$^{-1}$ |
| Maximum rotation speed | > 7000 min$^{-1}$ |
| Maximum torque | > 3 Nm |
| Maximum current | < 10 A |

Also, the provision of the product information from the point of view of a vendor is considered.

In summary there are three main tasks:

- Search the right information
- Integrate the right information
- Provide the right information

In the following sections it is shown how the engineering process of searching, integrating and providing is done today without the connection of MBSE and AAS and what are the challenges.

**Search the right information**

The scope of the task "search the right information" is the selection of a suitable electric motor from a broad portfolio of existing variants. The requirements for the electric motor are derived from the system under development context (Table 1). Note that the solution space is even reduced to an electric motor and not a motor in general. It is assumed that the engineer starts the search from scratch.

Common sources for product information are online catalogues with data sheets, marketplaces or configurators (Figure 2). Even internal and experienced sales associates will eventually make use of these three sources.

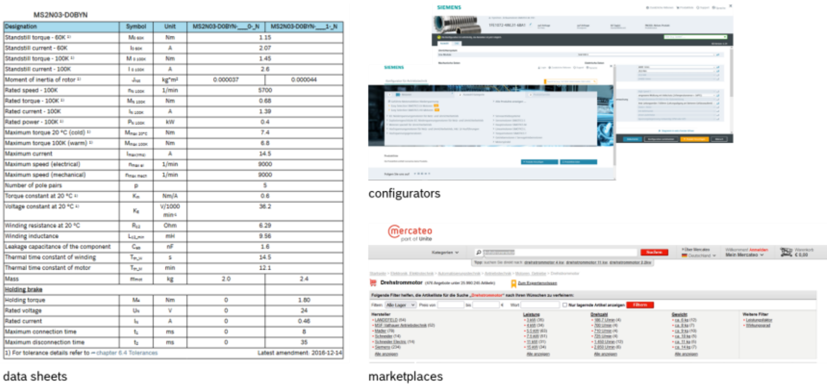


*Figure 2: Online sources – data sheets [2], configurators [3], marketplaces [4]*

Most vendors use their own naming for properties. Through lack of conventions multiple definitions for the same information often cause misinterpretations [1]. The effort for searching, comparing and integrating product information rises.

- Challenge 1: Heterogeneous semantics.
  This challenge is amplified by challenge 3.

The degree of supported search can be divided into three classes:

 

1) Full-text search
Search support: Highly manual work and searching with different terms for the same meaning
Solution space: Dependent on the search terms chosen by the user and provider

2) Structured search
Search support: Allows search operations and easier comparison within a solution space
Solution space: More narrowed due to search properties that are only unique in a local context

3) Configurators
Search support: Allows input of requirements and optimization features
Solution space: Tailored to a specific solution space

- Challenge 2: Higher degree of search support leads to a smaller solution space.
  This challenge is amplified by challenge 1 and 3.

Besides that, the user most likely rely on his or her experience in considering the sources for the product search and the solution. Just to start with the requirements from the system model is not possible at all. Even within one vendor the kind of semantics for data sheets, configurators and search possibilities may vary.

**Integrate the right information**
Information is often provided within documents (e.g., pdf- or spreadsheet-files) and must be integrated manually into the target systems. Technical solutions and complete toolchains using e.g., BMEcat are not yet state of the art. Even if BMEcat is used, there must be a harmonized semantics and data structures, otherwise a manual mapping is necessary.

- Challenge 3: No common standard for information exchange.
  This challenge is amplified by challenge 1 and 4.

**Provide the right information**
The amount of system knowledge, which is stored in separate, non-linked silos is rising [5]. Bringing this data to the customer often leads to a copy of the data into a new silo, without a continuous data flow. Relevant information across the products lifecycle is often not accessible for the end user without manual effort and risk of redundancy.

- Challenge 4: Rising amount of non-linked system knowledge.
  This challenge is amplified by challenge 1 and 3.

**Summary of the Challenges**
- Challenge 1: Heterogeneous semantics.
- Challenge 2: Higher degree of search support leads to smaller solution space.
- Challenge 3: No common standard for information exchange.
- Challenge 4: Rising amount of non-linked system knowledge.

This leads to gaps within IT toolchains and loss of data integrity. Furthermore, processes within the value streams frequently can neither be standardized nor automized. Existing solutions are mostly strictly tailored to a fixed use case and not robust enough against the increasingly volatile requirements and constraints. As a consequence, redundant manual work and business inefficiencies have negative impact on the companies' competitiveness.

## 2. STATE OF THE ART

This publication will show in chapter 4, chapter 5 and chapter 6 how the challenges stated in the introduction (chapter 1) will be addressed by utilizing state of the art technologies and concepts as:

- "Standardized Semantic Dictionaries" (chapter 2.1)
- "Asset Administration Shell (AAS), the Standardized Digital Twin" (chapter 2.2)
- "Repositories and Marketplaces for standardized Digital Twins" (chapter 2.3)
- "Model-Based Systems Engineering (MBSE)" (chapter 2.4)

### 2.1 Standardized Semantic Dictionaries

Dictionaries like IEC CDD [6] or ECLASS [7] provide standardized Concept Descriptions for industrial product classifications and product properties. The Concept Description is referenced by a unique semantic ID, e.g., by an IRDI (International Registration Data Identifier) [8].

ECLASS provides an entry for some product classes with a language dependent "Preferred name" and "Definition" as well as an "IRDI". Figure 3 shows an ECLASS entry for the product classification "27-02 Electrical drive".

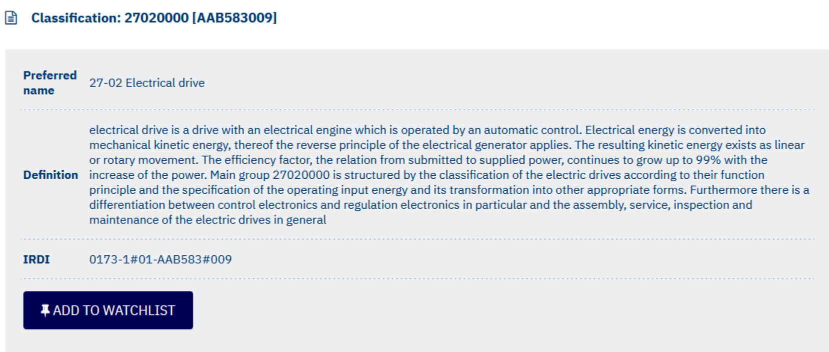


*Figure 3: Product classification – ECLASS entry "27-02 Electrical drive" [7]*

For products on the lowest hierarchical level ECLASS provides a set of properties. Figure 4 shows an ECLASS property list for "27-02-26-02 Servo synchronous motor".


*Figure 4: Product properties – ECLASS entry "27-02-26-02 Servo synchronous motor" [7]*

For each property ECLASS provides an entry with a language dependent "Preferred name" and "Definition", a data "Type" and for some properties a "Unit", as well as an "IRDI" (Figure 5).

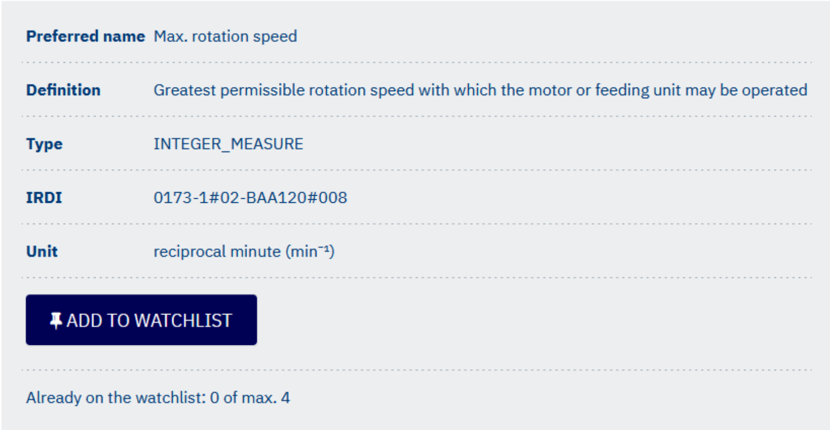


*Figure 5: Property classification – ECLASS entry "Max. rotation speed" [7]*

For this publication ECLASS is used as the exemplary dictionary.

## 2.2 Asset Administration Shell (AAS), the Standardized Digital Twin

IEC 63278 describes the meta model of the standardized Digital Twin, the Asset Administration Shell (AAS). The AAS contains specified data structures, so called Submodels (SM). A Submodel contains Submodel Elements (SME) like files and properties, or groups of Submodel Elements, so called Submodel Element Collections (SMC) [9]. SM and SME are referenced to Concept Descriptions (CD) from standardized Semantic Dictionaries (chapter 2.1) like ECLASS by a unique semantic ID. This allows the SME "property" to have – besides a "Name" and a "Value" – a "Type", "Unit" and "Definition" from the referenced CD. Figure 6 shows the AASx Package Explorer [10] with the AAS of a synchronous servomotor. On the left side the data structure with SMs and SMEs is shown and on the right side a viewer for the SM Technical Data. Besides the classification of the properties, the SM "Technical Data" has a specified SMC that contains properties to classify the product itself. In this case a classification in accordance with ECLASS 13.0 27-02-26-02 "Servo synchronous motor".

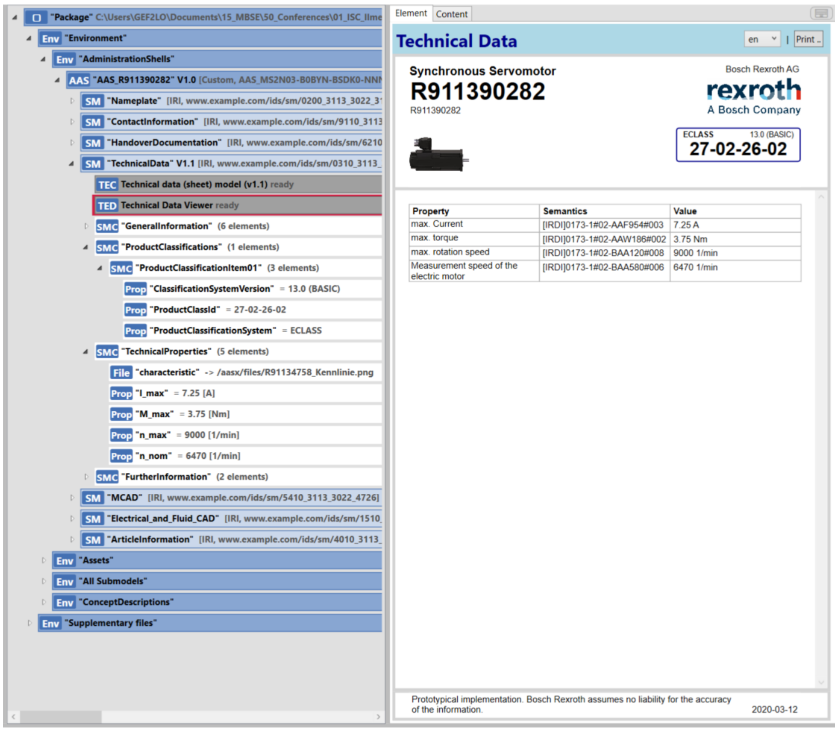

*Figure 6: AASx Package Explorer [10] with SMs and SMEs (left) and Technical Data Viewer (right) for Servo synchronous motor*

The data structure within the AASx-file is described by XML. Using the corresponding OMG standard XMI [11], the AAS can be exported from an AAS modeling environment like the AASx Package Explorer and imported in other systems compatible with XMI. For this publication the AASx Package Explorer with the AAS Meta Model Specification V2 is used.

## 2.3 Repositories and Marketplaces for standardized Digital Twins

Independent online marketplaces like the “asset administration shell marktplatz” of META-LEVEL Software AG [12] or repository viewer like the “AAS-Portal” of Fraunhofer IOSB-INA start growing. They adapt the IEC 63278 standard for the meta model of the standardized digital twin, the Asset Administration Shell (AAS). As most common marketplaces users can search with human readable full-text search as shown in Figure 7 in the “asset administration shell marktplatz”.

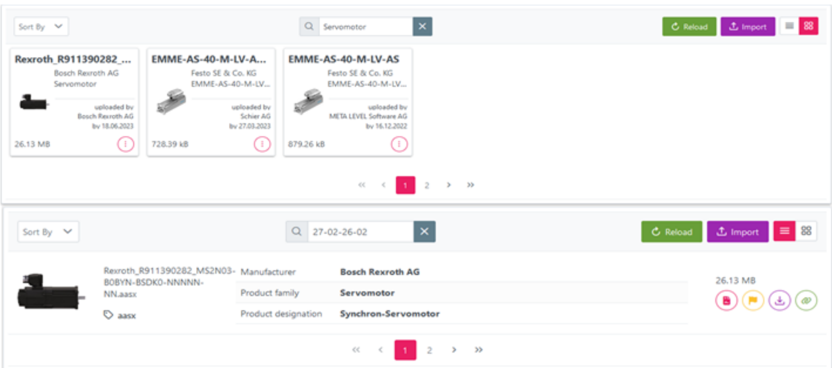

*Figure 7: META-Level Software AG “asset administration shell marktplatz”– full-text search [12]*

Besides that, solutions like the “AAS-Portal” [13] (Figure 8) have implemented a structured search. This allows e.g. a filter operation with a unique semantic ID from dictionaries like

ECLASS (chapter 2.1). The AAS-Portal was not released during the creation of this publication.

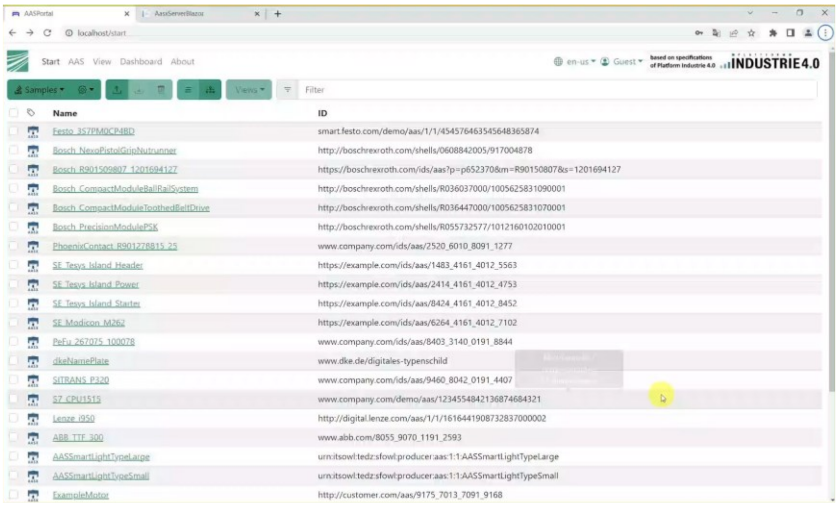


*Figure 8: Fraunhofer IOSB-INA "AAS-Portal" [13]*

## 2.4 Model-Based Systems Engineering (MBSE)

In Model-Based Systems Engineering (MBSE) structures, requirements and behavior of systems are modeled by using the systems modeling language (SysML). Block-Definition Diagrams (bdd) are used to describe the structure of a system and relationships between subsystems. This can be e.g. in terms of a physical bill of material using composition (Figure 9-left: an electric motor contains a shaft assembly, a housing assembly and a control assembly), or in terms of classification using inheritance (Figure 9-top: a synchronous electric motor is a specialization of an electric motor). Every block contains value properties with a "name", a "value type" and a "value". The "value type" can also be classified with further customizable tags like "unit", "name", or "semantic ID" [14].

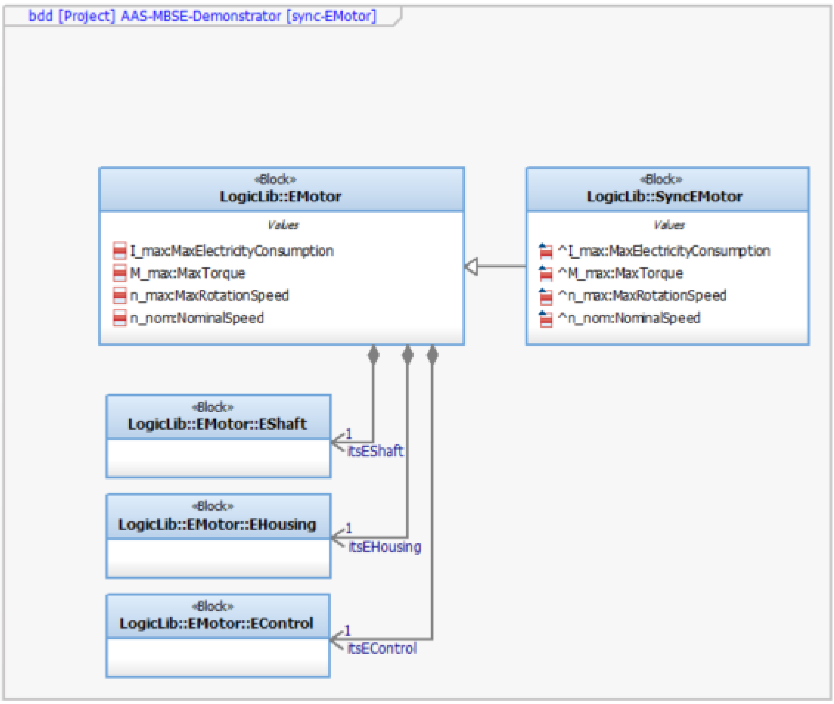


*Figure 9: Block-Definition Diagram (bdd) of a E-Motor – showing inheritance, composition, and values*

Tools like IBM Rhapsody are able to import and to export XML, using the corresponding OMG standard XMI [11]. For this publication IBM Engineering Systems Design Rhapsody is used.

## 3. RESEARCH QUESTION

Four challenges were described in the introduction (chapter 1):

- Challenge 1: Heterogeneous semantics.
- Challenge 2: Higher degree of search support leads to smaller solution space.
- Challenge 3: No common standard for information exchange.
- Challenge 4: Rising amount of non-linked system knowledge.

These challenges will be addressed by utilizing the state of the art technologies (chapter 2. This publication is based on the hypothesis that Model-Based Systems Engineering (MBSE) and Asset Administration Shell (AAS) complement each other in order to solve the underlying research question.
The solution and implementation suggestion are split into three parts. Part 2 and part 3 are based on part 1:

- Part 1 (chapter 4): Align MBSE system model with a semantic dictionary
- Part 2 (chapter 5): Find AAS and integrate it into MBSE system model
- Part 3 (chapter 6): Provide AAS based on MBSE system model

## 4. ALIGN MBSE SYSTEM MODEL WITH A SEMANTIC DICTIONARY

This chapter shows the part 1 of the solution, the process of how to align the MBSE system model with a semantic dictionary to proceed with part 2 of the solution "Find AAS and integrate it into MBSE system model" (chapter 5), and part 3 of the solution "Provide AAS based on MBSE system model" (chapter 6). From the state of the art technologies (chapter 2) "Standardized Semantic Dictionaries" (chapter 2.1), as well as "Model-Based Systems Engineering (MBSE)" (chapter 2.4) are used.

The process inside the system model is structured as follows:
1) Define block "EMotor" and assign ECLASS value types (from 2) to the values
2) Define ECLASS value types
3) Derive technical requirements as an instance of the Block "EMotor"

The definition of the values of the block and the definition of the value types might be an iterative process that considers the boundary conditions and requirements of the block and the available property classifications in ECLASS. Table 2 shows the properties relevant for this publication including ECLASS IRDI, unit and the required value for these properties.

*Table 2: Properties of the E-Motor in accordance with ECLASS and corresponding requirement constraints*

| [idShort] Preferred name | ECLASS IRDI | Definition | Unit | Required Value |
|---|---|---|---|---|
| [n_nom] Nominal speed | 0173-1#02-BAA580#006 | Speed corresponding to the motor used according to the name plate ratings | $min^{-1}$ | >5000 |
| [n_max] Max. rotation speed | 0173-1#02-BAA120#008 | Greatest permissible rotation speed with which the motor or feeding unit may be operated | $min^{-1}$ | >7000 |
| [M_max] Max. torque | 0173-1#02-AAW186#002 | Maximum mechanical torque which the motor can emit at the output shaft | Nm | >3 |
| [I_max] Max. electricity consumption | 0173-1#02-AAF954#003 | Maximum of energy which can be drawn | A | <10 |

**1) Define block "EMotor" and assign ECLASS value types (from 2) to the values**
The block EMotor must be created. Here it is recommended to model this block as neutral as possible to search later a broad range of possible solutions. As stated in the introduction (chapter 1), it should be an electric motor and not a motor in general. Once created, this block can be re-used in other projects as well. The corresponding ECLASS value types created in process step 2 are assigned. Figure 10 shows the block "EMotor" with the ECLASS conform value types.

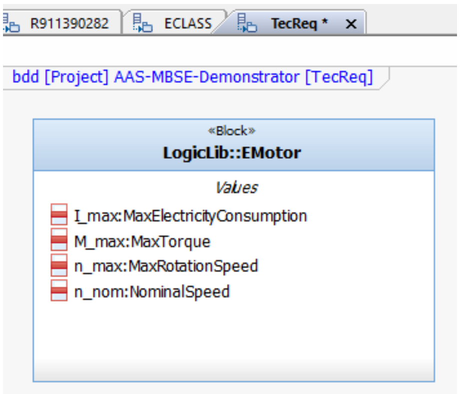


*Figure 10: Step 1): Define block "EMotor" and assign ECLASS value types (from 2) to the values*

**2) Define ECLASS value types**
The value types in accordance with ECLASS (chapter 2.1) properties are created inside an own package. Once created and released, this package can be re-used in other projects as well. Here it is recommended to select just a certain amount of properties and not all available properties from ECLASS in order to reduce the modeling complexity. Once a value type is created it should contain tags in accordance with the respective ECLASS entry. Figure 11 shows the structure of the value type "MaxRotationSpeed".

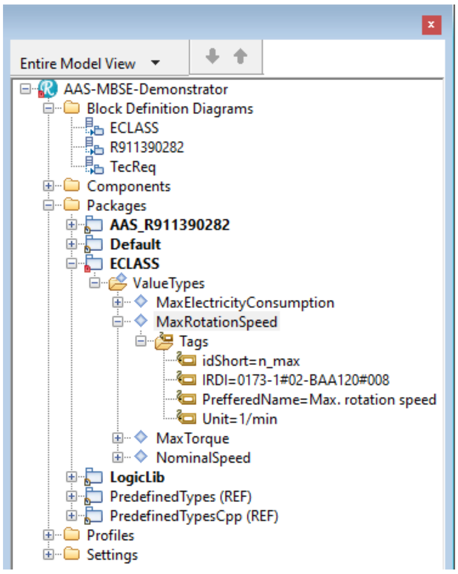


*Figure 11: Step 2): Define ECLASS value types*

**3) Derive technical requirements as an instance of the Block "EMotor"**
The technical requirements (TecReqEMotor) are derived in the system model as an instance of the Block "EMotor" with the corresponding constraints for each value. Figure 12 shows the technical requirements, derived from the EMotor block.

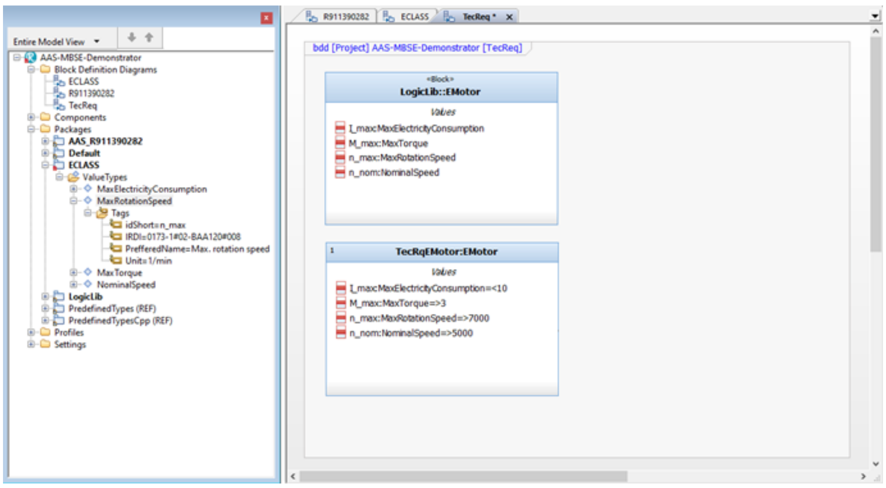


*Figure 12: Step 3): Derive technical requirements as an instance of the Block "EMotor"*

This is the starting point for the next parts of the solution part 2 "Find AAS and integrate it into MBSE system model" (chapter 4) and part 3 "Provide AAS based on MBSE System Model" (chapter 5).

## 5. FIND AAS AND INTEGRATE IT INTO MBSE SYSTEM MODEL

This chapter shows the part 2 of the solution, the process of how to find the right AAS based on the requirements of the systems model and how to integrate it into the system model. The starting point is the technical requirements block from chapter 4 in accordance with the semantic dictionary ECLASS. From the state of the art technologies (chapter 2) "Standardized Semantic Dictionaries" (chapter 2.1), as well as "Asset Administration Shell (AAS)" (chapter 2.2), "Repositories and Marketplaces for standardized Digital Twins" (chapter 2.3) and "Model-Based Systems Engineering (MBSE)" (chapter 2.4) are used.

Based on the technical requirements block from chapter 4, the process looks like this:
1) Perform semantic search
2) Transform AAS to XML
3) Integrate XML into system model

**1) Perform semantic search**
With tools like the AAS-Portal (chapter 2.3) from Fraunhofer IOSB-INA it is possible to have a structured search e.g., by ECLASS properties. The filter 0173-1#02-BAA120#008 > 7000 would show only products with a maximum rotation speed greater than 7000 min$^{-1}$. The AAS-Portal had not yet been released during the creation of this publication, therefore the semantic search was not tested in this particular case.

**2) Transform AAS to XML**
Once a suitable product is found, the corresponding AASx-file can be downloaded and opened in the AASx Package Explorer. For the MBSE system model considered in this publication only the Submodel (SM) "Technical Data" is relevant. This Submodel can be exported to an XMI conform XML-file (Figure 13). This XML-file has the same structure, naming and values as the source AASx-file. The Concept Description (CD) is not part of the XML-file. Line 115 – 120 in Figure 13 shows exemplarily the property "n_max".

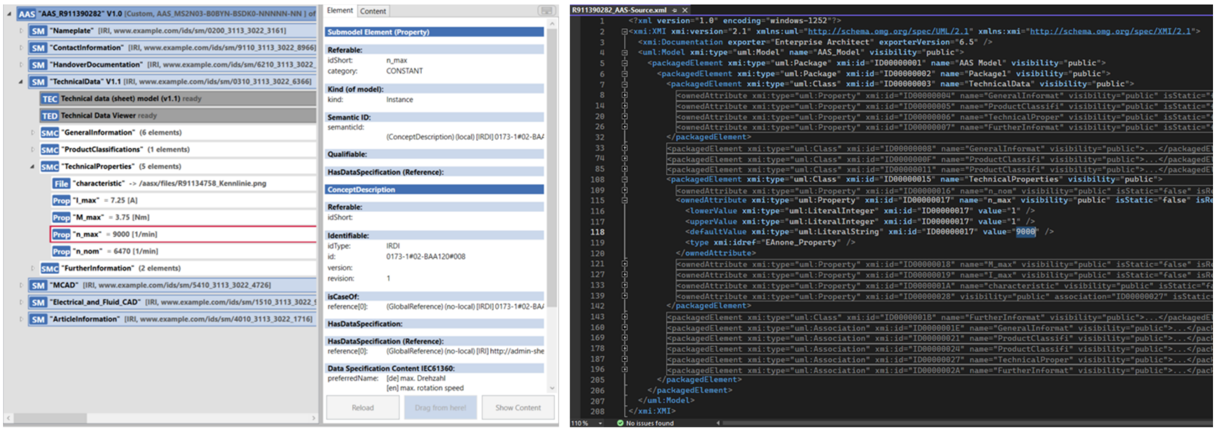

*Figure 13: Transform AAS to XMI conform XML-file*

**3) Integrate XML into system model**
The XML-file can be integrated into the system model via the import functionality of IBM Rhapsody (Figure 14). The structure of the submodel, the properties and the property values are integrated. As stated before, the Concept Description (CD) and with that the correct data type is not part of the XML-file.

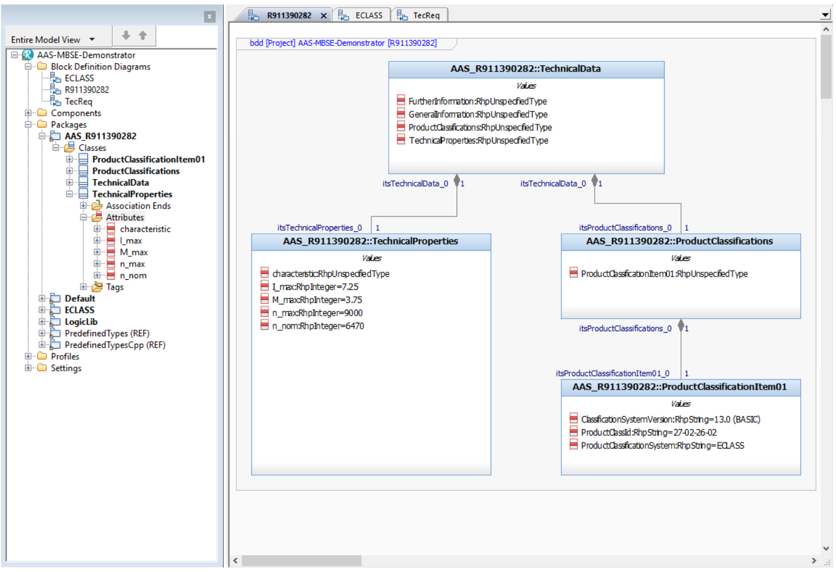


*Figure 14: Integration of XML-file*

Now the properties from the AAS are accessible inside the MBSE system model. Here it is recommended to assign the ECLASS "value types" to the properties (chapter 4) or to map the integrated properties to a block or part that will be used further (e.g. via Internal Block Diagram (ibd)). To enhance traceability, a satisfying relationship between the technical properties of the integrated AAS and the technical requirements (chapter 4) should be established (Figure 15).

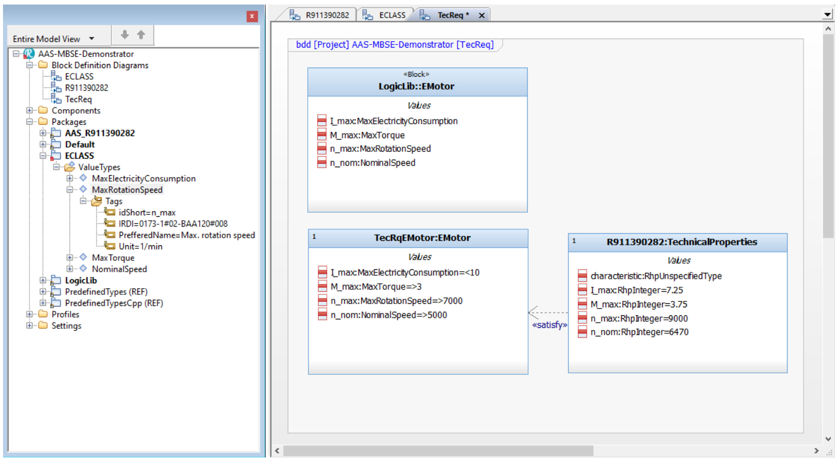


*Figure 15: Traceability within system model*

## 6. PROVIDE AAS BASED ON MBSE SYSTEM MODEL

This chapter shows the part 3 of the solution, the process of how to provide an AAS based on the MBSE system model. The starting point is the synchronous E-Motor block from chapter 4 in accordance with the semantic dictionary ECLASS. From the state of the art technologies (chapter 2) "Standardized Semantic Dictionaries" (chapter 2.1), as well as "Model-Based Systems Engineering (MBSE)" (chapter 2.4) are used. In this chapter it is assumed that the described solution is out of the viewpoint of an E-Motor supplier.

Based on the synchronous E-Motor block from chapter 4, the process is structured as follows:

1) Derive E-Motor type
2) Export block to XML
3) Import XML into AAS environment

**1) Derive E-Motor type**

One sellable product is the “oneType_EMotor” with a material number (Mat_No) R123456789. The block “oneType_EMotor” is derived from the block “SyncEMotor” and refines the property values of this Block as shown in Figure 16. The derived property value types are still the ECLASS types defined in chapter 4.

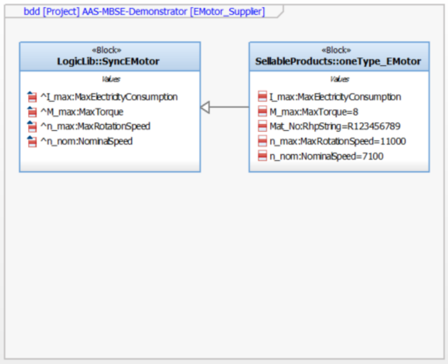


*Figure 16: Derivation of a specific E-Motor type*

**2) Export block to XML**

From IBM Rhapsody the Block with the corresponding properties is exported to a XMI conform XML-file. The structure is kept, as well as value and the name of the property (Figure 17). The value types are referenced to the locally created ECLASS elements in the system model and therefore not directly linked to a global identifier. Line 93 – 96 in Figure 17 shows exemplarily the property “n_max”.

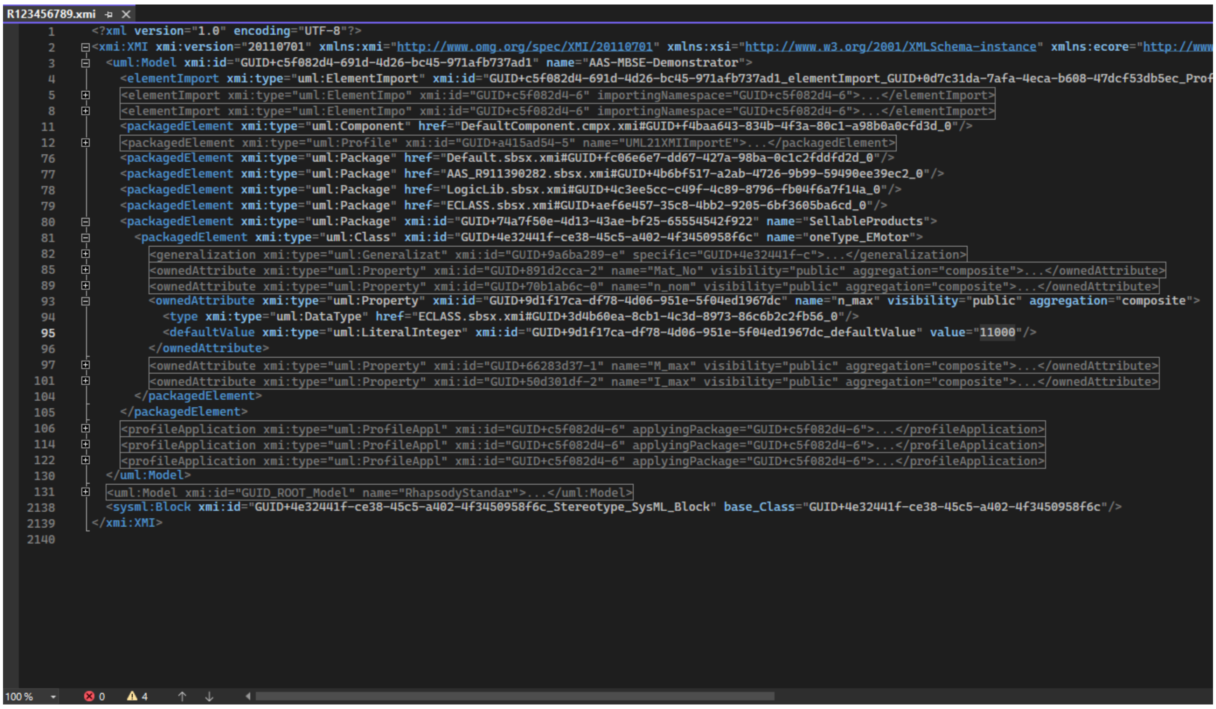

*Figure 17: Exported XML-file of the specific E-Motor type*

**3) Import XML into AAS environment**

For the considered tool chain, it is not possible to import the XML-file generated with IBM Rhapsody into the AASx Package Explorer.

## 7. DISCUSSION AND CONCLUSION

Existing technologies like MBSE and AAS as well as semantic dictionaries like ECLASS lead to a higher degree of automation and data integrity within IT toolchains and engineering processes. Harmonized semantics and standardized data structures as core concepts of Digital Twins [15] concretely enables:

- a requirement-based product search and selection
- a decrease in manual effort for integrating product information into system models
- a higher degree in information continuity across toolchains

To achieve these goals comprehensively, it is necessary to address processes, IT solutions and people across the whole product lifecycle. This means in particular:

- People must change their way of working from document-based to model-based approach. Also, externalizing their knowledge and referencing to common semantic dictionaries is required.
- IT solutions must rely on standards for data exchange and semantic descriptions.
- Processes must shift from a high focus on domain specific solutions to a more generic approach with a higher initial effort but a raised model reusability and data continuity.

This publication contributes in particular to bridging the gaps between requirements and the search for suitable products as well as the provision of product information from and the integration into a system model.

As shown in this publication smaller gaps still exists. To bridge them it is proposed to:

- extend the XML export from the AAS-file with a reference to a Concept Description
- adapt export from system modeler and import to AASx Package Explorer
- reference to a global Concept Description identifier IRDI
- integrating APIs between engineering tools and AAS providers

Furthermore, the results shown in this publication will provide a basis for:

- higher flexibility in terms of supply chains and customization
- strengthen collaborations to deploy the best domain experts worldwide
- increased transparency to establish more data driven business models
- more sustainability to meet regulations and responsibilities

## REFERENCES


[1] S. Reinhardt, "The GDPR Challenges Classic Data Management Approaches", Big Data Insider, 2018-12

[2] Bosch Rexroth AG – "R911347581_05 – Operating Manual". Available online: https://www.boschrexroth.com/en/de/media-details/770ca9e0-e16a-4789-82f5-3de546ed0038 (accessed on 2023-06-27)

[3] Siemens – Configurators, Available online: https://mall.industry.siemens.com/mall/en/en/Catalog/Configurators (accessed on 2023-06-27)

[4] Mercateo – B2B Marketplace. Available online: http://www.mercateo.com (accessed on 2023-06-27)

[5] T. Grosser, "Why Data and Knowledge Silos Prevent the Digital Transformation", eccena, 2021-05-20

[6] IEC CDD – Dictionary. Available online: https://cdd.iec.ch/cdd/iec61360/iec61360.nsf/Welcome?OpenPage (accessed on 2023-06-27)
[7] ECLASS – Dictionary. Available online: https://eclass.eu/en (accessed on 2023-06-27)
[8] E. Barnstedt, B. Boss, C. Ziesche et al. – "Specification – Details of the Asset Administration Shell, Part 1 – The exchange of information between partners in the value chain of Industrie 4.0 (Version 1.0)", 2018-11
[9] J. Neidig, A. Orzelski, S. Pollmeier, „Asset Administration Shell – Reading Guide", Plattform Industrie 4.0, 2022-01
[10] AASx Package Explorer. Available online: https://github.com/admin-shell-io/aasx-package-explorer (accessed on 2023-06-27)
[11] ISO, "Information technology - Object Management Group XML Metadata Interchange (XMI)". ISO/IEC 19509:2014(E). April 2014. Accessible online: https://www.omg.org/spec/XMI/ISO/19509/PDF
[12] META-LEVEL Software AG „AAS marktplatz". Available online: https://aas-markt.meta-level.de/ (accessed on 2023-06-27)
[13] J. Tikekar Fraunhofer IOSB-INA. „AAS-Portal". Presented at the IDTA "AAS Software & Research Landscape Event" in Garching, 2023-06-14). Information available online: https://www.linkedin.com/feed/update/urn:li:activity:7074679924686483458?updateEntityUrn=urn%3Ali%3Afs_feedUpdate%3A%28V2%2Curn%3Ali%3Aactivity%3A7074679924686483458%29
[14] S. Friedenthal, A. Moore, R. Steiner, "A Practical Guide to SysML: The Systems Modeling Language", Elsevier Science & Technology, 2014-10-24
[15] S. Malakuti, P. v. Schalkwyk, B. Boss, C. R. Sastry, V. Runkana, S.-W. Lin, S. Rix, G. Green, K. Baechle, S. V. Nath, "Digital Twins for Industrial Applications", Industrial Internet Consortium, 2020-02-18

## CONTACTS

F. Geibel
email: fabian.geibel@boschrexroth.de
ORCID: https://orcid.org/0000-0002-3360-6717