Abstract

The aim of this research is to establish clear concepts for working with standards (e.g. ISO) in the context of connected factory. When machines or other output devices need to communicate with each other, e.g., to autonomously perform processes in manufacturing processes, all the necessary information must be accessible to them. Today, however, the required information is documented in standards that are only available in PDF or even paper form. Thus, a prerequisite for using information from standards is that these are automatically provided in output devices. A deeper look leads to three levels of maturity for using standard information in output devices: machine-readable, machine-actionable and machine-interpretable. Starting from a clear definition of these terms an approach is developed to implement machine-actionable standards. It is assumed that not all standards are suitable for machine-actionability and therefore the first step is to classify the standards in order to identify the appropriate standards. The second step describes how information from standards can be modelled to be available in a machine-actionable form. The last step clarifies how the machine-actionable content must be provided afterwards so that they can be used in all output devices with less effort. This research work closes with the validation of the developed approach.

Keywords: Standards, Industry 4.0, Machine-readable, Machine-actionable, Machine-interpretable, Knowledge, Webservice

1. Introduction

A survey among the management boards of German industrial companies showed that the following concepts are interesting until 2022 in regarding to Smart Factories: Data-enabled resource optimization (up to 77%), connected factory (up to 60%), transfer of production parameters (up to 32%) (PwC, 2017). To implement these concepts, it is necessary that the systems and machines have access to digital data and information. Standards are an essential knowledge carrier for information in the product development process (Feldhusen et al., 2013). The question arises how standards can be made available to industrial companies? They are usually available in PDF format, which means that relevant information in the standards cannot be accessed automatically by an output device like a machine (Rieger et al., 2014). For this purpose, this information must be prepared manually by the user and subsequently made available to the required output device. This manual step must be carried out automatically with regard to the automatic linking of objects, e.g. within the framework of the "Smart Factory". In 2016, DIN and VDE launched the "Standardization 2020" program with the aim of offering comprehensive services and digitized products "around standards". An important application is the direct integration of information into the control of products or production plants (DKE, 2016).

The aim of this contribution is therefore to determine the prerequisites for a machine-actionable standard and to develop an approach for automatically making this information available for implementation in all output devices.
2. Definitions

Different contributions on the subject of providing information and data in combination with Industry 4.0 (I4.0) address terms such as machine-interpretable (Sinoara et al., 2018) (Fleischmann et al., 2016), machine-actionable (Nüttgens et al., 2009) or machine-readable (Studer et al., 1997) (Neururer et al., 2015). However, there are no clear definitions of these terms in practical or scientific articles. Thus, a publication of the German federal parliament on Big Data under the term machine-readable defined that programs are needed that make evaluative statements about products, brands, etc. or emotions recognizable (Deutscher Bundestag, 2013). In contrast, DIN-TERM-Online defines machine-readable as the coding of information so that it can be read by a computer or machine and interpreted by hardware and software. Machine-readable technology includes optical character recognition (OCR) and barcodes (DIN-TERM, 2019).

A search on the definition of machine-actionable shows that processes are described by semiformal semantics and thus become machines-executable. (Houy et al., 2009).

The term machine-interpretable is defined by the Fraunhofer Institute as an explicit formalization of concepts, relationships and limitations with the help of ontologies. Hereby, an automatic conclusion is possible based on the description log. This enables, e.g., the automatic identification of restriction violations and the extraction of unspecified but implicit information in technical systems (Büscher et al., 2015).

As the various definitions of the individual terms above show, it is not possible to make a clear distinction based on the literature review. In part, different definitions and levels of maturity of the available data are understood under the same term. Therefore, the first step is to classify and define the terms as they are used in this paper.

All three terms are to be used in the context of I4.0. They describe different powerfulness of representation forms and need to be arranged hierarchically. Powerfulness means that, depending on the degree of maturity of information, the machine gains more possibilities for processing the information, like it is shown in Fig. 1. This increase in power is explained in the following by defining the terms machine-readable, machine-actionable and machine-interpretable.

![Hierarchical classification of the terms machine-readable, machine-actionable and machine-interpretable](image)

Definition machine-readable:
Physical information is represented (e.g., visualized) in an output device (e.g., machine).

In the classification of powerfulness, machine-readable represents the weakest form of processing information. Physical information is represented by the machine as individual, stand-alone representable signs. This form can be used to represent the information for the user. There are no semantic, logic or meaningful connections between these signs. Forms of representation are e.g. a PDF document and possible output devices can be mobile devices, stationary computers, etc.
Definition machine-actionable:
The operations in an output device are performed automatically based on models derived from the physical information.
From the semantics of physical information, a given operation is automatically forced in an output device. The specified operations are automatically derived from the various information objects, such as values, tables and formulas.

Definition machine-interpretable:
The decision to perform the operations in an output device is ideally made autonomously derived from the physical information.
An output device, for example in the form of a machine operating software, is able to determine the optimal machining operations and carry them out independently. Thus, machine-interpretable represents the most powerful representation form of information. The information is represented in such a way that terms and formulas are related by ontologies. Consequently, a machine can automatically put the existing data into context, execute instructions and recognize restrictions according to the content - automatic conclusions are therefore possible. Quality characteristics, tolerances and test protocols can be automatically generated, accepted and evaluated.

3. State of the art

In Figure 2 a chronological classification is shown in which forms of representation and visualization standards exist and should exist in the future, it is assigned for whom the information is intended. There is also a mark that indicates at which point we are today. If standards are used today, they are available in paper form or as PDF documents (Beuth, 2019). The user has to search through the documents to find the solutions for his problems. This means that if he has found the information relevant to him, for example a calculation formula, he has to manually transfer the dimensioning variables for standard parts, tolerances or safety into his processes. The information is then processed, either by the user's knowledge or by external programs that can handle the individual pieces of information (Manoharan, 2018). In summary, this means that standards are available in such a way that only manual operations are possible in which information can be reproduced realistically to the original source and independently of the original operating system or hardware platform, but further processing is not automatically possible. Standards have recently been created by DIN in XML format with the aim of central storage and further use in various forms of recycling and products (Wischhöfer, 2016), such as search functions in documents. This suggests that, nowadays, the search for alternative and neutral data formats to provide information in neutral modelling notations is already ongoing. However, XML is suitable as modelling notation, but in the current used form of XML the contents are not available in machine-actionable form.
Knowledge/information that must be managed in support systems was already divided by O. Kuhn and A. Abdecker into the following classes: formal, semi-formal and informal knowledge (Kuhn et al., 1997). Thus, the knowledge to make individual knowledge available in a company and usable for solution finding and decision support is classified as follows:

Definition formal knowledge:
Formal knowledge is knowledge characterized by a complex formal representation which can be processed by complex conclusions (e.g. deduction, induction, abduction, etc.) and not only by retrieval mechanisms. Formal knowledge must be acquired by AI techniques. This can be done by dealing with formal knowledge content, e.g. business rules or design guidelines, which typically represent "compressed" knowledge. However, a predominant part of the individual knowledge will not be available in this form. (Kuhn et al., 1997)

Definition semi-formal knowledge:
Semi-formal knowledge refers to explanations and rule interpretations that are appended to formal knowledge units as text characters. This approach can be described as a combination of a formal knowledge base with hypertext. The data is not automatically processed, but only made available. (Kuhn et al., 1997)

Definition informal knowledge:
Informal knowledge representation is defined as lessons learned archive. The design rules contain the essence of the company's design competence. They indicate which criteria (attribute-value combinations) should be fulfilled in good designs regarding technology, cost efficiency and manufacturing aspects. Each design rule can be accompanied by an informal explanation. The definitions and explanations of terms represent a kind of ontology of all terms and structures that may occur in the types of knowledge mentioned above. In addition to a formal definition of concepts and relationships, it also includes informal explanations and information on how the various objects are to be presented to users. (Kuhn et al., 1997)

From the different definitions of terms and today's way of working with standards it can be concluded that it is not possible to integrate them into an I 4.0 environment without a great deal of additional effort by third parties.

4. Goal of the research work

In order to reduce the deficits from the current state of the art, the aim of this work is to examine the following research question: How can standard contents be formalized and made available for a machine-actionable form?

To achieve this goal, the following three subgoals were defined within the framework of this work:

Subgoal 1:
Standards are classified according to the possibility of machine convertibility (formal to abstract and focus on formal standards): It is assumed that not all existing standards are suitable for machine convertibility. Therefore, a method should first be developed to identify suitable standards by classification.

Subgoal 2:
A suitable format specification is identified: To formalize the information in the standards, a selection procedure shall be carried out to identify the appropriate format specification.

Subgoal 3:
A procedure for the implementation of the machine-actionable standards is available.

5. Method for digitization of standard contents

The following model can be derived from the objective of the contribution as shown in Fig. 3. It starts with the evaluation of a standard regarding the machine-actionability, the second step is to model the knowledge into machine-actionability and the last step is to provide the content.

![Fig. 3 Method to digitize standard content](image-url)
In the following this procedure model will be explained in more detail. First, it is necessary to classify the standard, then to investigate how the contents can be formalized and finally how these contents can be made available across systems.

5.1 Classification of standards

The effort required to digitize information depends on the form in which the information is available, as mentioned in chapter 3. So far, the information or the existing knowledge is divided into formal knowledge, semi-formal knowledge and informal knowledge. This classification already represents a very good approach, but is insufficient for the application of standards, since at the time the classification had been made (1997) IT development had not yet progressed so far. Therefore, a proposal for a new, contemporary classification of information needs to be made within the framework of this work. With the help of this classification it should be possible to not only classify information contained in documents but also to estimate how much effort is required to make the information available in a machine-actionable form.

Therefore, the first step is to analyze which objects are used in standards to represent the information. Following objects can be identified: texts, formulas, tables and figures. These objects may have different properties, hence further subdivisions are necessary for some objects to describe their properties as well.

Texts:
Texts can have following properties, which enables the transformation into a machine-actionable form in different ways: They can be descriptive, can specify properties, can serve as explanation/explanation for illustrations or contents and can be formulated as specific instructions.

Formulas:
Formulas are given in standards and are used to calculate values.

Tables:
Tables can be filled with numbers or values, but there are also tables that are used to assign different classes that are described by properties.

Illustrations:
There are different figures used in standards. Graphs or diagrams can be used to describe calculation values or dimensioning variables. In the same way, however, technical drawings can also be shown or, for processes, the corresponding workflow charts.

Table 1 Assignment of the clarity to the objects identified from standards: text, formula, table, figure

| Object  | property | clarity | transformability |
|---------|----------|---------|------------------|
| text    | Descriptive | Low    |                  |
|         | Property   | High   |                  |
|         | Explanation | Low    |                  |
|         | Instruction | Medium |                  |
| Formula | High      |        |                  |
| Table   | Number     | High   |                  |
|         | Classification | Medium |                  |
| figure  | Graph      | Medium | Table            |
|         | Diagram    | Medium | Table            |
|         | Drawing    | Medium |                  |
|         | Workflow   | Medium |                  |

For a machine-actionability a clarity is assigned to these objects. This clarity is divided into high, medium and low. The higher the clarity, objects are more suitable for machine-actionability. To increase the clarity of some objects, it is
possible to transform them into other objects. For example, diagrams can be transformed into tables.

As shown in Fig. 4, to classify the objects into three categories, the following rules are used. If there is a large number of objects that have a high proportion of clear definable objects, they are assigned to formal; if they have a medium proportion, they are assigned to semi-formal, and if they have a low proportion, they are assigned to abstract.

**Definition abstract standard:** Standards which are assigned to the class abstract consist mostly of objects which have a low uniqueness, e.g. texts. These are not further considered for machine-actionability. An example of this category is ISO 9001.

**Definition semi-formal standard:** Semi-formal standards are standards that contain information in various forms of representation. They include not only textual components that describe states, output classifications or test procedures, but also information in formal description like tables with values or mathematical formulas. Or however descriptive/explanatory/definitive texts can be converted easily into if/then instructions. Then it is possible to transform semi-formal norms into formal norms.

**Definition formal standard:** Formal standards can be classified as standards whose information has a high degree of clarity, such as formulas or tables with values. As an example, a standard for the description of component sizes (e.g. for flanges) can be mentioned. Formulas can also be used to determine dimensioning variables.

### 5.2 Analysis of suitable format specifications

After classifying standards, the next step is to examine which format specifications are currently available and which database types are suitable for storing information as well as for classifying and labelling it in a suitable form. These format specifications and databases are used to make the information available in a machine-actionable form, so the content can then be made available in respective application areas.

After researching the available format specifications, JSON, XML, HTML and YAML were found as common formats for structuring data.

XML is the abbreviation for "Extensible Markup Language" and represents hierarchically structured data in a human-readable ASCII encoding, so it is comparable to a text file. However, the elements are organized in a tree structure, which makes efficient reading and further processing possible. An error-free syntax is required in order to make it possible to read the information/data.

JSON (JavaScript Object Notation) is a lightweight data exchange format that is easy for humans to read and write and easy for machines to parse (analyze data structures) and generate. It is based on a subset of the JavaScript programming language. It is a universal data structure that is supported by almost all modern programming languages. Therefore, it makes sense that a data format that can be exchanged between programming languages also builds on these structures. JSON does not support cyclic graphs and is not intended to exchange binary data. (ecma international, 2017)

JSON and XML are comparable, but unlike XML, JSON is not a markup language. The advantages compared to XML, however, are the smaller size, since fewer characters are required to subdivide data sets and a better handling of flexible interfaces is given. (Köhler et al., 2018a)
HTML (Hypertext Markup Language) is a markup language that describes components such as references, tables and graphics, and the structure of a document. It is the most commonly used language in the world wide web. HTML serves as a markup language to semantically structure texts, but not to format them. (Schaedler, 2004)

YAML is also a markup language that gives data a structure. Like JSON, YAML stores the data in a human readable form, with the focus even more on it. YAML serves as a format for the serialization of data and offers the possibility of text marking. Since version 1.2, YAML is a superset of JSON, which means that YAML can replace JSON in all applications. YAML natively supports scalar data types, lists and associative arrays. YAML is stored in a text file. (dev-insider, 2019)

The following requirements were defined to find a machine-actionable format: universal applicability, practical relevance, complexity of the language, support through Open Source, sustainability, human readability, automated further processing, possibility of revision and the required resources of the computer.

| Requirement                  | XML  | JSON | HTML | YAML |
|------------------------------|------|------|------|------|
| Universal applicability      | High | Medium | Low  | Medium |
| Practical relevance          | Medium | High | Medium | Medium |
| Complexity of the language   | High  | Medium | High  | High |
| Open Source                  | High  | High | High  | High |
| Future viability             | High  | High | High  | High |
| Human-readable               | Medium | High | Low  | High |
| Automated further processing | Medium | High | Low  | Medium |
| Possibility of revision      | Medium | High | Low  | High |
| Resourcing                   | Medium | Low  | Medium | Medium |

The universal applicability of JSON can be classified as medium, since no complex awards are possible, and XML can be used more flexible. In terms of practical relevance, JSON can be classified as high because it is a simplified markup language and provides a standardized structure that clearly defines how the data is represented. Regarding the complexity of the language XML is to be classified higher since it offers more functions than markup language and thus complex markups are possible. Related to the use of Open Source XML and JSON do not differ and are to be classified as high. The same can be concluded for the future ability, JSON is still a relatively young format specification and even if XML is used longer, it is still used today constantly for new approaches and efforts are done to extend it. In terms of human readability, JSON can be classified as high since the structuring of the data appears more natural for humans. In reference to automated further processing, JSON can be classified as high due to its direct feasibility in JavaScript. Also the revision is clearly easier to classify in comparison to XML and thus as high since the files are easier for humans to read and additions are possible. The resource utilization requirement is rated low by JSON, as the files are smaller than XML files and allow faster access.

As a result of the selection process JSON was selected as the format specification. The characteristics of low complexity by the specification, low resource requirements on the system and that it is easy for humans to read offer a very good possibility for further automatic processing and have led to this decision. Figure 5 shows an example, that presents the structure of a JSON file, which demonstrates how a standard can be modeled in JSON. The standard is saved with the following metadata: The ID, document number, description and release date.
5.3 Analysis of suitable Databases

To save the data, a suitable database will be searched. As database types relational and non-relational databases are available. For a relational database, an SQL database is a typical representative. For a non-relational database, graph databases (e.g. Neo4j) and document-oriented databases (e.g. MongoDB) are representatives.

Relational databases are particularly suitable for large databases if the data has the same structure, because they can then be summarized in tables. The structure of the respective tables can be defined via the schema, whereby the data types of the respective fields should already be defined at the beginning. If the structure must be changed afterwards, it is only possible with a lot of effort. The relationship between different tables can be defined by using keys. Relational databases have a rigid and predefined behavior. The data can be retrieved from other systems via SQL queries. The accessing system decides how the data is linked and evaluated. Changing the database structure means that all clients must be adapted. (Köhler et al., 2018b)

The world consists of a network of data and information. This network is divided into nodes, their properties and their relation with each other. Simple examples are mind maps or flowcharts. Graph databases try to display and store these relations. As a representative for a graph database Neo4j will be considered in this article. Neo4j is an open source graph database which uses the already mentioned techniques to store single graphs. Thus, Neo4j offers every user the possibility to save data into the database and to retrieve them again. For this purpose, Neo4j uses a special developed query language called Cypher. Cypher offers the possibility to query certain constellations of the graph. Since it concerns above all visual connections within the graphs, an equally visual query language offers itself. For this reason Cypher uses ASCII-Art. ASCII-Art is an art form in which images are created using the ASCII character code. (Hunger, 2014) Figure 6 shows two nodes connected to each other. As an example, a simple connection was chosen for illustration. For this reason, the nodes and the relation itself have no properties. Below the visual representation of the connection the identical one in Cypher is shown. It is clear to see why ASCII type is used, because it makes it possible to represent the visible connection in simple form by text. With the graphical user interface of Neo4j the stored results can be displayed visually.

The other subcategory of non-relational databases are document-oriented databases. Document-oriented databases store data in form of individual documents, which can be addressed with a unique identifier. The advantage of this form of storage is that data storage can contain completely different variables. In comparison to the classical relational databases, additional information can easily be added to a document, since none of them follow any precast file types. In a relational database, the storage follows a fixed pattern. Therefore, document-oriented databases have an advantage if no common type can be defined for the data to be stored and the information differs in many data sets. However, this also entails a disadvantage - the references. References can be represented by embedded documents. A document is

Fig. 5 Structure of a JSON file representing standard content modeled

```json
{
    "Description": {
        "ID": "1",
        "DocumentNumber": "DIN 28883",
        "Description": "Support brackets with reinforcing plate",
        "release": "2012-05"
    }
}
```

Fig. 6 Two nodes connected visual and with Cypher

(A) –[:referred_to]-> (B)}

Another subcategory of non-relational databases are document-oriented databases. Document-oriented databases store data in form of individual documents, which can be addressed with a unique identifier. The advantage of this form of storage is that data storage can contain completely different variables. In comparison to the classical relational databases, additional information can easily be added to a document, since none of them follow any precast file types. In a relational database, the storage follows a fixed pattern. Therefore, document-oriented databases have an advantage if no common type can be defined for the data to be stored and the information differs in many data sets. However, this also entails a disadvantage - the references. References can be represented by embedded documents. A document is
simply appended to the current one as a sub-item. Another possibility is to place the reference to an object ID. In this paper the representation of references by document ID is chosen. Document-oriented databases are particularly suitable for storing data that is not strongly networked and has different properties. With strongly networked data it is recommended to rely on other databases. (Klöckner, 2015)

For machine-actionability the following requirements are defined: the dependencies shall be visible, it shall be possible to integrate images, the databases shall be easily extendable and an efficient possibility for modeling the dependencies shall be available.

| Requirement                                  | Relational Database | Graph Database | Document-oriented Database |
|----------------------------------------------|---------------------|----------------|---------------------------|
| Dependencies can be displayed visually       | Low                 | High           | Low                       |
| Integrate figures                            | Low                 | Medium         | Low                       |
| Easily extendable                            | Low                 | High           | Medium                    |
| Efficient possibility for modeling the       | Low                 | Medium         | Medium                    |
| dependencies                                  |                      |                |                           |

After assessing the degree of fulfilment of the requirements, it is necessary to shift away from a relational database and consider at non-relational databases. Because only these make it possible to serve as a structured data memory and get along without fixed table schemata. There is also a big difference between this type of storage and relational databases. Instead of storing the data records in tables and linking them with each other, a different approach is chosen. Not the individual nodes, but the entire connection is stored as a data set in a database. This leads to the fact that the effort for saving the data is greater, but the call will be faster. So, finally, a graph database was chosen.

5.4 Provision of machine-actionable content

In the last step, the information that can now be transferred to a machine-actionable form must be made available again so that it can be used by software systems on machine tools or CAD-Systems. The requirements for such a provision are that the contents are available in a system neutral way and do not represent isolated solutions. In addition, the solution should provide as many interfaces as possible so that all systems have access to the content. One problem that results is that each system has its own interface. For example, each system must have an add-in according to the requirements of both systems for coupling to another system. If a software system must be exchanged, all add-ins that have a connection to this software have to be adapted.

The combination of web services and microservices is one possibility to solve this problem. Microservices make the content/information from standards, which are mapped in databases, available to other systems. The web service serves to link the different software systems. This procedure also enables no direct linking between systems and thus the content can be exchanged in the background without individual interfaces having to be adapted.

5.5 Procedure for converting a standard into a machine-actionable standard

Once a tool has been selected for each subgoal, the procedure model can be developed. The elaborated procedure model is presented in a tree structure (Fig. 7). It begins with the classification of the standard and the decision whether it can be classified formal or semi-formal. If the standard is classified as an abstract standard, the process is aborted and there is no transfer to machine-actionability.

 Afterwards, it is checked whether the standard can be classified as formal. If this is not the case, objects with low clarity are transformed into objects with a higher one if possible.

In the next step, the standard content are extracted and formalized with the help of JSON. The structure of the standard is then stored into a graph database and the content is made available via web services.
6. Validation

To validate the approach presented, the DIN 28083 standard is now considered, which describes claws with reinforced plates. The standard consists of eleven pages and is therefore not very large and consequently well suited for a limited example.

The first step is to classify the standard according to the procedure model. When considering the standard, the following objects can be identified:

- Illustrations with technical drawings whose clarity is classified as medium.
- Tables with dimensions, i.e. numbers, whose clarity is to be classified as high.
- Formulas whose clarity is also high.
- Illustrations with graphs whose clarity is to be classified as medium, but by means of a transformation into a table, the clarity is also high.
- Texts which describe the parameters properties and thus are to be classified as high.
- Texts which represent the properties of the designation, whose clarity is also to be classified as high.

Thus, the standard can be classified as formal due to its high clarity and its content is suitable for extraction into the machine-actionable form according to the procedure model.

Figure 8 shows how to transform a graph into a table. In the table, the nominal sizes of the brackets are assigned to the coefficient C. Since the graph has a medium clarity, a transformation into a property of higher clarity is necessary - in this case the property table.
In the second step, the content is extracted into the JSON format. Figure 9 shows how the table created in Figure 8 is modeled in JSON format.

```json
"Table 1": {
  "Title": "Determination of the coefficient C on the basis of the nominal bracket sizes",
  "Table 1":
  {
    "nominal bracket size": [
      "1",
      "2",
      "3",
      
    ],
    "coefficient C": [
      "0,056",
      "0,055",
      "0,052",
      
    ]
  }
}
```

Fig. 9 Table modeled in JSON format

In the third step, the data is stored in the graph database. The structure of the standard shown in the graph database is shown in Fig. 10. The example shows how chapters are referenced to the standard. The chapter “material” refers to another standard, this reference (link) is also shown. The node of the chapter “labeling” has the syntax of the bracket labeling as a property.
In the fourth and final step, the content of the standards are made available. Here the example shall be shown that the user asks for the name of the claw. To do this, the user submits a request to a web service with the following URL:

https://standard28083.com/api/v1/standards/brackets/name?ng=4&material=S235JR

The query transfers the nominal size and the material. The web service then supplies the name of the claw as an answer. The answer is provided in HTML, text, JSON or XML format, depending on the query selection. In this way, as many interfaces as possible can be served. In JSON format, the answer looks as shown in Fig. 11.

```
{ 
  "response": "Pratze DIN 28083 - 4 - S235JR"
}
```

Fig. 10 Standard mapped in a graph database

Fig. 11 Answer of webservice in JSON-format

7. Discussion and outlook

The aim of this work is to develop the first fundamentals for machine-actionable standards, which play an important role in the context of smart factory solutions. It was shown how existing standards can be identified so that they can be prepared for machine-actionable at reasonable costs. For this, it was shown how the content can be formalized by the format specifications JSON and how the dependencies can be mapped in the graph database. The use of web services represents a suitable solution for making the content available across systems. The validation example demonstrates that the methods and solutions as well as the procedure are suitable to create machine-actionable standards.

The results from this work will be used to optimize the preprocessing process for the creation of standards. Thus, the standards are already to be created and made available according to the method presented. For this purpose, the Duisburg 3M model for the machine-actionable standards will be developed. The first M stands for modularization, the decomposition of standard contents into knowledge modules. The second M stands for modelling, the conversion of the knowledge modules into a machine-interpretable format, which is to be regarded as the highest and most powerful level. Lastly the third M, the management, serves for the provision and administration of the standard contents. Each individual M represents such a large subject area in itself that in the future a more detailed consideration of the individual modules will take place and specific methods will be developed or created for these.
References

Beuth, <https://www.beuth.de/de/erweiterte-suche/272754!search?alx.searchType=complex&searchAreaId=1&query=DIN+281&facets%5B%5D=hitsPerPage=10>, (accessed on 12 June, 2019).
Büscher, C., Jeschke, S. and Meisen, T., Ontology-based Information Management for Factory Planning, Digital engineering for planning, testing and operating technical systems, 18. IFF-Wissenschaftstage (2015), pp. 29-35 (in German).
Dev-insider, <https://www.dev-insider.de/was-ist-yaml-a-665391/>, (accessed on 17 June, 2019).
DKE, <https://www.dke.de/de/normen-standards/normung2020>, (accessed on 7 June, 2019).
Deutscher Bundestag, current term: Big Data, Wissenschaftliche Dienste, available from <https://www.bundestag.de/resource/blob/194790/c44371b1c740987a7f6fa74c06f518c8/big_data-data.pdf> (2013).
DIN-TERMinologieportal, <https://www.din.de/de/service-fuer-anwender/din-term/suche-nach-benennung/wdc-dinterm-beg:dn21:168267565?sourceLanguage=de&destinationLanguage=en>, (accessed on 7 June, 2019).
Ecma International, The JSON Data Interchange Syntax, available from <https://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf> (2017).
Feldhusen, J. and Grote, K.-H., Pahl/Beitz Design Theory (2013), pp. 152-199 (in German).
Fleischmann, H., Kohl, J. and Franke, J., A Reference Architecture for the Development of Socio-Cyber-Physical Condition Monitoring Systems, IEEE (2016), DOI: 10.1109/SYSOSE.2016.7542963.
Houy, C., Fettke, P. and Loos, P., Stylized Facts of the Event Driven Process Chain - Application of a Method for Theory Formation in Business Informatics, EPK 2009, 8. Workshop der Gesellschaft für Informatik e.V. (GI) und Treffen ihres Arbeitskreises „Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten (WI-EPK)“, pp. 22-41, (2009).
Hunger, M., Neo4j 2.0 - One graph database for all, entwickler.press (2014)(in German).
Klöckner, K., In Comparison: NoSQL vs. relational databases, Universität Siegen (online), available from <http://pi.informatik.uni-siegen.de/Mitarbeiter/mrindt/Lehre/Seminar/NoSQL/einreichungen/NOSQLTEC-2015_paper_9.pdf> (2015) (in German).
Köhler, P., Manoharan, T., Loibl, A., Bertram, M., Klemme, U., Sernetz, P., Hasanbegović, M., Vetterling, T., Keller, G. and Hoffmann, J., Knowledge-based CAx process chains for welded structures in power plant construction (WPSK): Final Report on the BMWi Project (2018a), p. 120 (in German).
Köhler, P., Manoharan, T., Loibl, A., Bertram, M., Klemme, U., Sernetz, P., Hasanbegović, M., Vetterling, T., Keller, G. and Hoffmann, J., Knowledge-based CAx process chains for welded structures in power plant construction (WPSK): Final Report on the BMWi Project (2018b), p. 121 (in German).
Kühn, O. and Abecker, A., Corporate Memories for Knowledge Management in Industrial Practice: Prospects and Challenges, J. UCS., Vol. 3. (1997), pp. 929-954, DOI: 10.1007/978-3-662-03723-2_9.
Manoharan, T. and Köhler, P., Knowledge Provision as a Service, 16. Gemeinsames Kolloquium Konstruktionstechnik 2018 (2018), pp. 224-235 (in German).
Neururer, S. B., Lasierra, N., Peiffer, K. P. and Fensel, D., Formalizing the Austrian Procedure Catalogue: A 4-step methodological analysis approach, Journal of Biomedical Informatics, Vol. 60 (2016), pp. 1-13.
PwC, Smart Factory - Use of business-relevant concepts in Germany 2017, PwC (2017), Digital Factory 2020 p.26 (in German).
Rieger, B. and Benjamins, A., Text Mining in scientific publications, Sammelband zum KI-Praktikum WS 2013/2014, Universität Osnabrück (2014), p. 8 (in German).
Schaedler, T., Use of Markup-Languages und Web-Technologies, Universität Augsburg (online), available from <http://wi.wu.ac.at:8002/rgf/diplomarbeiten/2004_Schaedler_A/Einsatz_von_Markup-Languages_und_Web-Technologien.pdf> (2004), p. 29 (in German).
Sinoara, R. A., Camacho-Collados, J., Rossi, R.G.,Navigli, R. and Rezende, S. O., Knowledge-enhanced document embeddings for text classification, Knowledge-Based Systems, Vol. 163 (2018).
Studer, R., Benjamins, V.R. and Fensel, D., Knowledge Engineering: Principles and methods, Data & Knowledge Engineering, Vol. 25 (1997), pp. 161-197.
Wischhoefer, C., Standardization of the ISO XML format for norms, DIN Mitteilungen (2016) (in German).