IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 24, 2020, accepted December 10, 2020, date of publication December 16, 2020,
date of current version December 28, 2020.

*Digital Object Identifier 10.1109/ACCESS.2020.3045111*

# A DIN Spec 91345 RAMI 4.0 Compliant Data Pipelining Model: An Approach to Support Data Understanding and Data Acquisition in Smart Manufacturing Environments

**KEVIN NAGORNY[1], (Associate Member, IEEE), SEBASTIAN SCHOLZE[1], (Member, IEEE), ARMANDO WALTER COLOMBO[2], (Fellow, IEEE), AND JOSÉ BARATA OLIVEIRA[3], (Member, IEEE)**

[1]Institute for Applied Systems Technology Bremen GmbH, 28359 Bremen, Germany
[2]Department of Electrotechnical and Industrial Informatics, University of Applied Sciences Emden / Leer, 26723 Emden, Germany
[3]Department of Electrical Engineering (DEE), Universidade NOVA de Lisboa, 2829-516 Caparica, Portugal

Corresponding author: Kevin Nagorny (nagorny@atb-bremen.de)

This work was supported by EU H2020 BOOST 4.0 Project under Grant 780732.

**ABSTRACT** Today, data scientists in the manufacturing domain are confronted with various communication standards, protocols and technologies to save and transfer various kinds of data. These circumstances makes it hard to understand, find, access and extract data needed for use case depended applications. One solution could be a data pipelining approach enforced by a semantic model which describes smart manufacturing assets itself and the access to their data along their life-cycle. Many research contributions in smart manufacturing already came out with with reference architectures like the RAMI 4.0 or standards for meta data description or asset classification. Our research builds upon these outcomes and introduces a semantic model based DIN Spec 91345 (RAMI 4.0) compliant data pipelining approach with the smart manufacturing domain as exemplary use case. This paper has a focus on the developed semantic model used to enable an easy data exploration, finding, access and extraction of data, compatible with various used communication standards, protocols and technologies used to save and transfer data.

**INDEX TERMS** Data pipeline, industry 4.0, RAMI 4.0, smart manufacturing, semantic model.

## I. INTRODUCTION

One of the most time consuming processes in data analytics is the data understanding and data preparation phase according to the Cross Industry Standard Process for Data Mining (CRISP-DM) [1], [2], which can take more than 60 % of the time in an entire data analysis process [3]. The data understanding phase includes an initial data collection, their description and their exploration, while the data preparation phase includes - among others - the selection and extraction of data. This is also valid for a data analysis process in the (smart) manufacturing domain. By investigating the causes for this high time consumption, three main causes for data scientist in the data understanding and data preparation (selection and extraction) phases in the (smart) manufacturing domain were identified.

The associate editor coordinating the review of this manuscript and approving it for publication was Ting Wang.

**Cause 1 - Data Chaos:** Even in modern digitalized manufacturing environments, different kinds of data are stored unstructured in several kinds of data sources. Some data are single files while others are bundled into data buckets, and others are only available as a data stream or in a request-response mechanism using different kinds of communication protocols.

**Cause 2 - Inefficient data understanding and selection:** Efficient approaches to explore, search, filter, identify, understand and select required data are missing and available data is often proprietary without applied classification and standardisation approaches.

**Cause 3 - Time and resource consuming data extraction:** Even if the data - required for a data analysis process - is known, extracting heterogeneous data for a later data analysis is still a challenge because multiple communication protocols and technologies need to be considered. In practice this often means that data has to be collected from different

data sources (message brokers, databases, services, etc.) and that the correct location/reference within a data source has to be identified (for instance a column in a specific database table). This is often a time and resource intensive process in which several IT experts with domain knowledge have to be involved.

These three causes lead to three research challenges addressed in this work:

**Challenge 1 - Find a methodology to structure the data chaos:** The growing volume, variety, velocity and the growing complexity of data in the smart manufacturing domain requires new approaches to manage the resulting data chaos. It has to be answered how data can be classified and how established and newer standards - emerging from the Industry 4.0 - can support this process. Very heterogeneous digital brownfields as well as modern green field ecosystems where any kind of data can be part of an asset life-cycle have to be considered for a comprehensive solution.

**Challenge 2 - Find data exploration approaches for heterogeneous (big) data environments:** Find approaches that makes the exploration, search, filtering, identification, understanding and selection of required data in heterogeneous data environments easier and faster.

**Challenge 3 - Find time-efficient data extraction approaches for heterogeneous (big) data environments:** Find approaches for the data extraction in a heterogeneous data environment that consider the path dependency in the industrial domain to assure an applicability in brownfields as well as in modern green fields. The approach needs to extract data easily should be independent from used communication technologies, data storage technologies, data formats or data schemas.

To tackle these challenges was built a semantic model based DIN Spec 91345 compliant data pipelining approach which combines SotA data classification and description standards in Industry 4.0 related smart manufacturing systems with data pipelining technologies to enable an technology independent direct data access and extraction for a subsequent data forwarding and/or processing (Extract Load Transform (ELT) or Extract Transform Load (ETL)). The results of the overall research are divided into ($1^{st}$) a novel methodology for semantic model-based DIN Spec 91345 compliant data pipelining, ($2^{nd}$) a semantic data model, ($3^{rd}$) a software architecture and ($4^{th}$) an implemented prototype validated in an industrial environment. This paper briefly introduces the elaborated methodology for semantic model-based DIN Spec 91345 compliant data pipelining to give an overview and focuses then on the developed DIN Spec 91345 RAMI 4.0 compliant semantic data pipelining model which is used to describe a smart manufacturing environment and to initiate the developed prototype.

The paper is organized as follows: Section I introduced the semantic model based DIN Spec 91345 compliant data pipelining approach; Section II presents an introduction in the approach which shall support the understanding of this paper; Section III presents a state of the art analysis in smart manufacturing, semantics and data pipelining approaches; in addition Section IV describes some fundamental knowledge; section V describes the semantic model; Section VI presents a use case of the approach. To round down the described approach, a discussion of the presented approach and results achieved so far is given in section VII; finally, the paper ends with a summary and outlook in section VIII.

## II. A BRIEF INTRODUCTION IN THE METHODOLOGY FOR SEMANTIC MODEL-BASED DIN-SPEC-91345 COMPLIANT DATA PIPELINING

This section provides an overviews on the methodology for semantic model-based DIN-Spec-91345 compliant data pipelining. FIGURE 1 shows a conceptual high-level view on the self-descriptive semantical smart manufacturing infrastructure. The first part (see number (1)), visualises I40 components (see robot symbols). These components provide their descriptive standardized meta data and raw data, including the reference to the data coming from/located in a data source, along their life cycle through a self-descriptive semantic model. Following the RAMI 4.0 specification, an I40 component provides its meta-data via an Asset Administration Shell. The part of the figure labelled with the number (2) shows the linkage of compatible I40 component semantical models (see smaller semantic model symbols) with a smart manufacturing environment semantic model (see middle semantic model symbol). Both linked semantic models build a semantical network of a smart manufacturing environment which is manged by a Semantic Manager module. This environment describes hierarchy levels and main OT/IT [4] components (for example central databases or message brokers). In the infrastructure follows a data selector module labelled with the number (3). It is responsible to explore, search, filter and select data based on the linked semantic models. Based on selected data (binary object, buckets, streams, etc.), the data selector module generates a pre-configuration for data source connectors. These connectors are then deployed in the data provider module labelled with the number (4) in FIGURE 1. The data provider deploys the list of connectors/ data source processors using parameters described in the pre-configuration file issued by the data selector in number (3). One processor could be for instance a MySQL database connector. The last gets pre-configured with access information (IP, Port), authorisation information (user, password), as well as a pre-configured query to extract the selected data from a specific table column.

As soon as the pre-configured data source connectors have been deployed, they can be executed (see number (5)). The execution of the connectors allows to extract the selected data from semantic model referenced data located in the data sources addressed in number (2) (for example monitoring data, meta data, context information or historical data along the life-cycle of I40 components). The data provided in number (5) will have to be prepared for being later used by the data analytics module represented by number (7) in FIGURE 1.

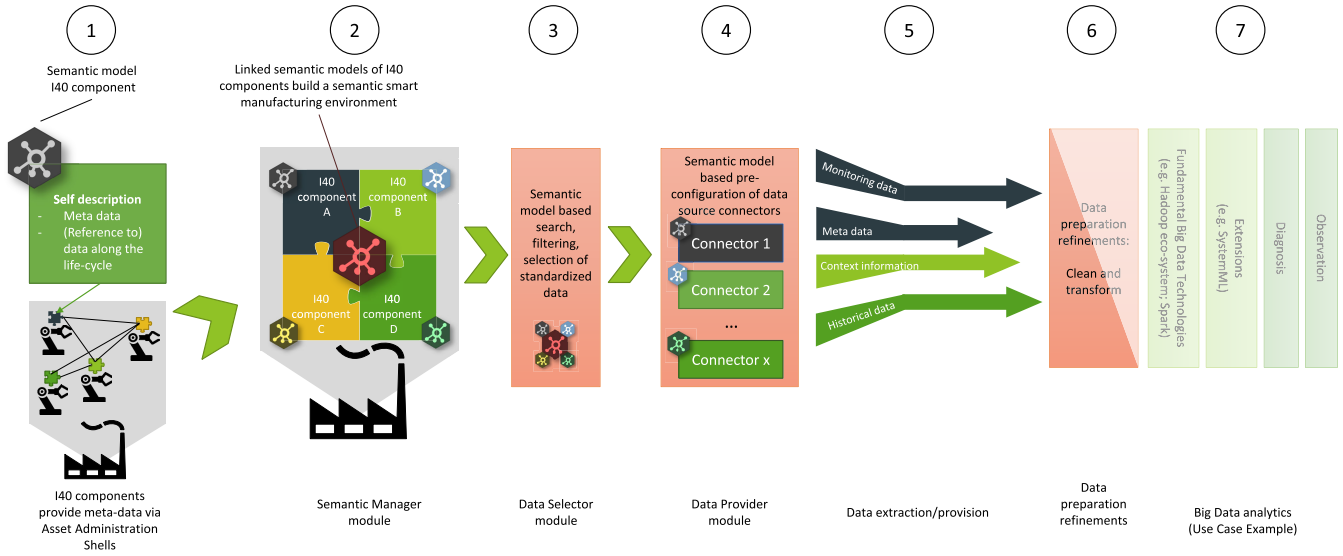



**FIGURE 1.** Self-descriptive semantical smart manufacturing infrastructure.

This functionality is performed by software components of the infrastructure and identified with the number (6).

The data pipeline approach is basically composed by the sequence 2-5. Since this work does not deal with comprehensive data transformation tasks nor with data analysis and related use cases, step 6 and 7 in FIGURE 1 are partly shadowed.

In summary, this approach aims to contribute to the addressed challenges in section I trough the:

- description of heterogenous data environments in smart manufacturing based on a semantic model according to DIN Spec 91345 which describes the Reference Architecture Model Industrie 4.0 (RAMI4.0)
- classification of data based on Industry 4.0 related standards like eCl@ss
- exploitation of a semantic model-based structured data environment for comprehensive data exploration, search, filtering, identification, understanding and selection.
- integration of modern data pipelining approaches to extract selected data by using a semantic model-based data source connector pre-configuration.

This paper has a focus on the developed semantic data model which was elaborated to describe DIN Spec 91345 (RAMI 4.0) compliant I40 components, their data, the access to these data and their integration in a smart manufacturing environment. A semantic data model was chosen to follow and enforce/strengthen the overall trend which infiltrates also the manufacturing domain caused by various benefits related to integration, adaptation, standardisation and exploitation.

For the elaboration of a model for Industry 4.0 based smart manufacturing systems, it is necessary to consider the state of the art in smart manufacturing, the semantical movements in this domain, as well as the state of the art in data pipelining. Further some fundamental knowledge about reference architectures and standards for the manufacturing domain is needed. These aspects are described in the next two sections.

This paper is also supplemented by a complementary video, which provides further information on the methodology and helps to understand our approach.

## III. STATE OF THE ART

### A. SMART MANUFACTURING

The term "'smart manufacturing"' defines a wide field around ongoing digitalisation activities in the manufacturing domain. In the past, manufacturing systems were built based on simple controllable units that were hierarchically structured and separated by multiple communication technologies. The ongoing digital evolution leads to an integration of communication networks on levels 5-7 according to the Open Systems Interconnection (OSI) model. This allows a cross-level and cross-domain data exchange between connected units. To exploit the evolved new potentials, such units start to get their digital representation which is in literate - based on the context - often called virtual ghost, digital twin or cyber shadow. This digital representation enables units

- to cooperate and collaborate with units in their environment,
- to learn to behave well in their environment e.g. by using artificial intelligence approaches (for instance KPI based effectiveness and efficiency improvements),
- to describe themselves, their skills and their current state (for instance in form of realistic simulated models synchronized in real-time with the physical unit), or
- to share their data, information and gained knowledge.

Units with such features are - again based on the context - often called Cyber-physical Systems (CPS), Things (in relation to the Internet of Things) or I40 components (in relation to Industry 4.0). Next to many other emerged topics [5], [6], these developments generate also an awareness for the value in big data where [7]. Forecasting that

technology will enable an accessibility of all produced data leads to the thought ''how value can be extracted out of these data?'' [8]. This field is part of the data science discipline which evolved based on several parent disciplines [9]. This work aims to contribute into this field with the goal to exploit the ability of self-descriptive units to easily channelize needed data (streams) by using data pipelining approaches for a sub-sequent data analysis process. Therefore, two main topics need to be introduced in the following sub-sections. (1st) Semantic in smart manufacturing that enable the self-description of digitalised units, and (2nd) Extract, Transform, Load (ETL) [10], Extract, Load, Transform (ELT) [11] and Data Pipelining techniques [12] that are used to transfer/pipeline data including optional usable data transformation/pre-processing steps.

### B. SEMANTIC IN SMART MANUFACTURING SYSTEMS

Building a unified semantic for smart manufacturing systems plays an essential role for the evolution in this area. The harmonization of semantics is an increasingly important topic in this domain and is one of the key prerequisite to guarantee system interoperability [13], [14]. Many networks and organisations like the Semanz40 [15], eCl@ss e.V. (see www.eclass.eu), ProSTEP e.V. [16], AutomationML e.V. [17], OPC Foundation [18], PLC Open e.V. [19], International Electrotechnical Commission (IEC) [20], International Organization for Standardization (ISO) [21] or the Institute of Electrical and Electronics Engineers (IEEE) [22] are currently working on this issue and several already existing standards that aim to standardise and unify the used semantic in industry, are basis for their work. Many results show that semantics in smart manufacturing systems will be based on semantic models. Those results are for instance reported in VoCol [23]–[25], a collaborative space to achieve unified semantics. There is also available an initial semantic model related to the DIN Spec 91345 (RAMI 4.0), which was an initial inspiration for the semantic model developed in this work [26].

### C. DATA PIPELINING

Data pipelining [27] evolved basically from ETL approaches. ETL is a term, emerged in the data warehousing domain [28] and is used when data have to be moved or copied from a source system to a target system through an extraction (source), transformation (data transformation compliant to the target system(s)) and loading (move/copy transformed data to target system(s)) process. With the need of data pipelining a sub-topic of ETL evolved: ELT. ELT approaches are used for data pipelining to build e.g. data lakes. Such approaches are needed when it comes to transfer data from one point to another while the data transformation happens in the target system. ETL and ELT related technologies provide different I/O interfaces and processors for data cleaning or transformation. Many ETL/ELT/Data Pipelining approaches and technologies are already available - for instance:

- **Apache NiFi [29]** - A distributed dataflow engine by the Apache Software Foundation written in Java. It is a flow-based programming approach, is very flexible and include several processors to handle different kinds of data source interfaces or to clean and transform data. It also allows to write custom data flow processors [30].
- **StreamSets Data Collector [31]** - A distributed dataflow engine by the StreamSets Inc. written in Java. Provides an attractive user interface and provides typical features for data pipelining.
- **Hevo Data [32]** - A data pipeline tool by Hevo Data Inc. to clean, enrich, and transform data on the fly. It offers several standard data source processors for several kinds of data source interfaces and offers also an automatic data type and schema mapping for selected source and target system technologies.
- **Apache Airflow [33]** - A data pipeline tool by the Apache Software Foundation written in Python. Airflow is a platform to programmatically author, schedule, manage, and monitor workflows which represents these as - so called - directed acyclic graphs (DAGs) of tasks.
- **Talend Open Studio [34]** - Is an ETL software by Talend written in Java. Talend Open Studio is the open source version which contains selected core modules of their Data Management Platform and is therefore limited in its functionality.

Synthesizable is that the most available technologies provide processors to access several kinds of data sources, to transform extracted data, to map data types between selected data sources, and to access the application by an API. What is not available yet is a data pipelining approach that uses semantic models that describe the access to data located/accessible in various kinds of data sources to support the user in the creation of a data pipeline. An semantic model that describes a smart manufacturing environment, its units, the data along their life-cycles, and the access to this data, can enable a semantic model-based pre-configuration of available data source processors/connectors in ETL and ELT related technologies to support the creation of a data pipeline as presented in this work. While the most available semantical standards for a self-description of units only provide a simple pointer/reference to related data (e.g. in form of an URL), this approach goes one step beyond and adds an approach that directly extracts and channelize/pipeline data independent from used data source technologies, data types or data formats to generate a central data access point. This work shows (1st) how standardized self-descriptive semantic models of units and their environment (based on RAMI 4.0) can be exploited to easily find needed data, and (2nd), how this semantic model can be exploited for data extraction in established data pipelining approaches. The overall goal of our work is to introduce semantic-model based data pipelining to the manufacturing domain.

## IV. FUNDAMENTALS

To support the reader in understanding the presented approach and to provide some background information, some basic knowledge is described in this chapter. To have a

common ground for discussions in the community, smart manufacturing systems need reference architectures to provide a big picture, a framework, main elements, their relations and a vocabulary. Therefore, the next section introduces the main influencing reference architectures related to smart manufacturing systems.

### A. REFERENCE ARCHITECTURES RELATED TO SMART MANUFACTURING SYSTEMS

This work has a strong focus on the RAMI 4.0 as this reference architecture addresses directly the manufacturing domain, considers several manufacturing standards and is basis for the Industry 4.0. It is represented as a three-dimensional model which describes layers, the life-cycle and hierarchy levels of an Industry 4.0 compliant system. The RAMI 4.0 suggests also a range of standards to be used to establish a semantical interoperability [35]–[37]. But the RAMI 4.0 is not the only reference architecture that has a relation to smart manufacturing systems as some example described below show.

- Smart Grid Architecture Model (SGAM) [38]: The Smart Grid Architecture Model describes an architecture for smart-grids. It is a three-dimensional model which describes interoperability layers, domains and zones. Main ideas of this architecture were derived for the RAMI 4.0 which is described in DIN Spec 91345.
- Industrial Internet Reference Architecture (IIRA) [39]: After the release of the RAMI 4.0, the Industrial Internet Consortium released the Industrial Internet Reference Architecture, which is covering cross-application-domains while the RAMI 4.0 has a focus only on production. It is divided into three parts: (1.) key system characteristics and assurance, (2.) viewpoints and (3.) key system concerns. It is possible to map the most elements of the IIRA into the RAMI 4.0, as explained in [40].
- Big Data Value (BDV) Reference Model [41]: Was developed by the Big Data Value Association (BDVA) and is a reference framework to locate big data solutions on the overall IT stack. It addresses the main horizontal and vertical concerns and aspects to be considered in big data systems. Because Big Data becomes also a factor in smart manufacturing, this reference architecture should be considered when big data related solutions are developed in this domain.

When it comes to the question how smart manufacturing systems and data should be unified to achieve interoperability then this has to be done in standardisation activities. Some important standards used in this work are introduced in the next section.

### B. STANDARDS RELATED TO SMART MANUFACTURING SYSTEMS

For this work some relevant standards related to smart manufacturing systems were considered. For the architectural aspects of smart manufacturing systems, among others,

- IEC 61512-1 [42]: Is a standard for batch control and defines design recommendations for software, hardware and process flows. It is described in DIN Spec 91345.
- IEC 62264-1 [43]: Is based on IEC 61512 and is part of a series of standards for the integration of IT and control systems in industry. Especially IEC 62264-1 describes models, terminology and hierarchical levels used and extended in DIN Spec 91345.
- IEC 62890 [44]: Describes requirements for life-cycle management of systems and products used in industrial-process measurement, control and automation. It distinguishes between ''types'' (components/assets in the design phase) and ''instances'' (instantiated types).

Concerning the semantic model aspects related to the classification and description of data, among others the standard eCl@ss in combination with IRDI as global identifiers were considered:

- eCl@ss [45]: Describes an ISO/IEC-compliant data standard for the classification and description of products and services. This standard is frequently updated and candidate for one of the languages of Industry 4.0 compliant eco-systems. The data model is based on IEC 61360; especially the IEC property classification.
- IRDI [46]: The ''International Registration Data Identifier'' is based on ISO/IEC 11179-6, ISO 29002 and ISO 6523 and is candidate for the globally unique identifier in Industry 4.0 compliant eco-systems.

After introducing the state of the art and needed fundamentals, the next chapter presents the semantic model used for DIN Spec 91345 compliant data pipelining.

## V. A SEMANTIC MODEL FOR SEMANTIC MODEL-BASED DIN-SPEC-91345 COMPLIANT DATA PIPELINING

The self-descriptive semantical smart manufacturing infrastructure introduced in section II, which is a semantic model derived from standards addressed in section IV, is described in detail in this section.

### A. STRUCTURE OF THE SEMANTIC MODEL

To build the structure of the semantic model, following core concepts from the following set of standards and technologies were derived:

- Industry 4.0 component from the RAMI 4.0: An Industry 4.0 component provides information about an asset and provides access to its functionalities. It includes an Asset Administration Shell (AAS) [47], [48] which is divided into a header with a manifest and a body which includes several sub-models according to IEC/TS 62832, to describe the component itself.
- RAMI 4.0 dimensions: Hierarchical Levels based on IEC 61512-1 and IEC 62264-1; Life-cycle and value stream dimension according to the IEC 62890; and digitalisation dimension containing six Layers.
- Data Access Views from the RAMI 4.0: Concept of data access views where specific data properties/sub-models are linked to specific user (groups).

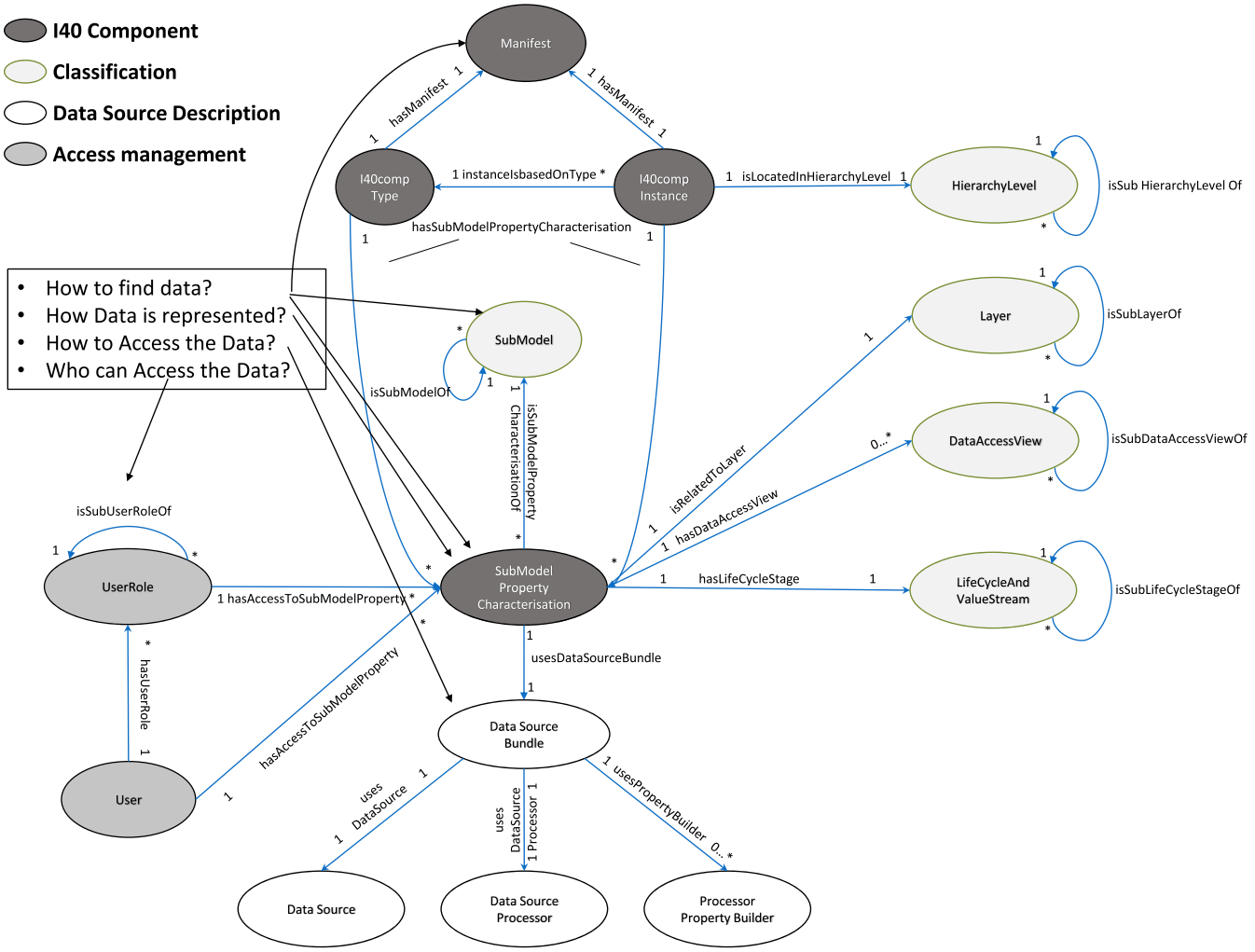



**FIGURE 2.** Semantical representation of an I40 component embedded in a smart manufacturing environment.

- Classification of products and services from eCl@ss: the separation between classification classes and product description classes. This concept is mapped to the sub-model concept described in DIN Spec 91345.
- Identifying globally and unique I40 components and sub-models through their IRDI identifier.
- Property models from eCl@ss: Derived will be the property model released in eCl@ss 10.0.1 which is derived from several other standards [49].
- FlowFile Processors from Apache NiFi: parametrization through different attributes to act as pre-configured data source processor/connector (e.g. a MySQL processor with pre-configured access details and a query to read specific data).

As result, FIGURE 2 shows the main classes of the elaborated semantic model for an I40 component embedded in a RAMI 4.0 oriented smart environment. I40 component related classes follow the I40 component (I40comp) concept described in DIN Spec 91345. An I40 component, according to DIN Spec 91345, is divided into an "I40comp Type" that represents the design/specification phase of an asset and an "I40comp Instance", which represents an instantiation of a specific I40comp type. Types and Instances have a "Manifest", which includes meta data of an "I40comp type" or an "I40comp instance". An "I40comp instance" can be associated to a specific location in a smart manufacturing environment through the "HierarchyLevel" class. The "HierarchyLevel" class represents the structure of a smart manufacturing environment (for instance a hierarchical structure of an enterprise in a tree or graph representation). Instantiations of hierarchy levels can be used to nest/link I40 component instances and their relations into bundles. Any data linked to an I40comp type or I40comp instance is described in a "SubModel Property Characterisation" (SMPC). An SMPC is according to DIN Spec 91345 linked to

- a "Layer" (SMPCs are linked to the "information" layer) to describe to which RAMI 4.0 layer the data belongs,
- a "DataAccessView" to group/categorize different SMPCs (for instance Energy Efficiency Data),

- a ''LifeCycleAndValueStream'' to show the life-cycle stage to which the linked data belongs (for instance a CAD file in the design life-cycle phase of an I40 component) (Note: According to the IEC 65/617/CDV:2016, DIN EN 62890:2016 the linked data can belong to one or more of life-cycles to manage systems and products used in industrial-process measurements, control and automation as e.g. product life-cycle, order life-cycle, factory life-cycle and technology life-cycle.),
- ''UserRole''s and ''User''s to configure users enable authentication base on single users or user groups.

An SMPC describes accessible data of an I40 component (for instance an attribute, a CAD file, a data stream, a specific information in a database, etc.). To enable an easy finding of these data, a standardized classification is needed. Such data classifiers are called ''Sub Models''. Sub Models are - similar to the eCl@ss approach - standardized data classifiers and identifiable by an IRDI [50], [51]. Note: If the reader is not familiar with the classification classes of eCl@ss, it is recommendable to take a look at their wiki (http://wiki.eclass.eu/wiki/Classification_Class). Sub Models can be based on eCl@ss but also other classification standards are integrable.

In case that the data cannot be stored as a simple attribute/data property in a SMPC (as basically done in eCl@ss), because the data is a binary file/object, a document, a data stream, etc., then a ''Data Source Bundle'' has to be defined. The Data Source Bundle includes detailed information that point/refer to the data in a data source (for instance: a specific MQTT stream that can be subscribed to by an MQTT broker or a query that gathers specific data from a specific database). The ''Data Source Bundle'' is divided into three classes:

- a ''Data Source'' class which includes access information to a data source (for instance a NoSQL database (user, password, IP, etc.));
- a ''Data Source Processor'' class which represents a parametrizable data source connector. For example, in case Apache NiFi is the chosen data pipelining technology used as basis to extract data, then the list of all Apache NiFi Flow processors that enables access to a data source would be imported to the semantic model, including all required parameters of each processor. Apache NiFi was chosen because it is a scalable and customizable solution which allows data routing, transformation and system mediation also for big amounts of data. The Apache NiFi Flow processors in this work are called ''Data Source Processors''. An individual of a data source processor, linked to a data source bundle, initializes the needed parameters for the chosen data flow processor. This approach is described in detail in the next section V-B).
- a ''Property Builder'' class which defines a list of properties needed to generate dynamically a parameter for the data source processor, if needed. Such a property builder could be for instance a MySQL SQL query builder that requires information typically for an SQL query as a table name and a column. The Property Builder individual would contain the values for the table and the column of the data addressed by the data source bundle. An algorithm (in this example an SQL query builder; to be developed) uses these values to build the query which is then used as parameter to initialize the data source processor. A property builder can also be used for dynamic data filtering or aggregations. This enables e.g. the collection of data with spatial and temporal characteristics (different in space and time like GPS or weather data). An example could be traffic data (live and historical) for a city. Available meta-data like timestamps and the GPS can be used to filter the traffic data for a specific time frame in a specific area. Again, this approach is described in detail in section V-B).

The data source bundle, used to parametrise a data source processor, is explained in detail in the next section.

### B. THE DATA SOURCE BUNDLE

FIGURE 3 shows how the data source bundle is built in detail. Number (1) in FIGURE 3 shows the Data Source Bundle. An individual of a data source bundle is linked to an SMPC (not visible in the figure; therefore, compare FIGURE 2). It is possible to add data properties to describe the data source bundle if needed. One data source bundle individual uses one data source processor individual (see number (6) in FIGURE 3). Following the infrastructure described in FIGURE 1, an essential step is the selection of a technology for implementing the pipelining approach. As a consequence, the data source processor defines a processors/connector to be used based on that selected data pipelining technology (see section III-C). For example: If Apache NiFi is used as data pipelining technology, a processor class could be ''org.apache.nifi.processors.mqtt.ConsumeMQTT'' with all data properties needed for the configuration (see FIGURE 4). An instantiated individual has this class as type and would initialise related data properties.

Some data properties get not initialised in the data source processor individual directly because the values are already initialised somewhere else. This is happening if data source access details are needed (see number (2) in FIGURE 3). A data source class defines access parameter to a data source. A data source could be a message broker, a database or another central node for data access. A data source access individual for a specific database will be just instantiated once in a smart manufacturing environment semantic model. A data source bundle, which references data in this data source, will be linked by an ''usesDataSource'' relation (/an usesDataSource object property). The mapping in the data source processor individual follows following syntax:

*<data property>=DS(<data property>)*

**An example (compare FIGURE 3 number (6)):**

*access=,,DS(IP):DS(PORT)''*

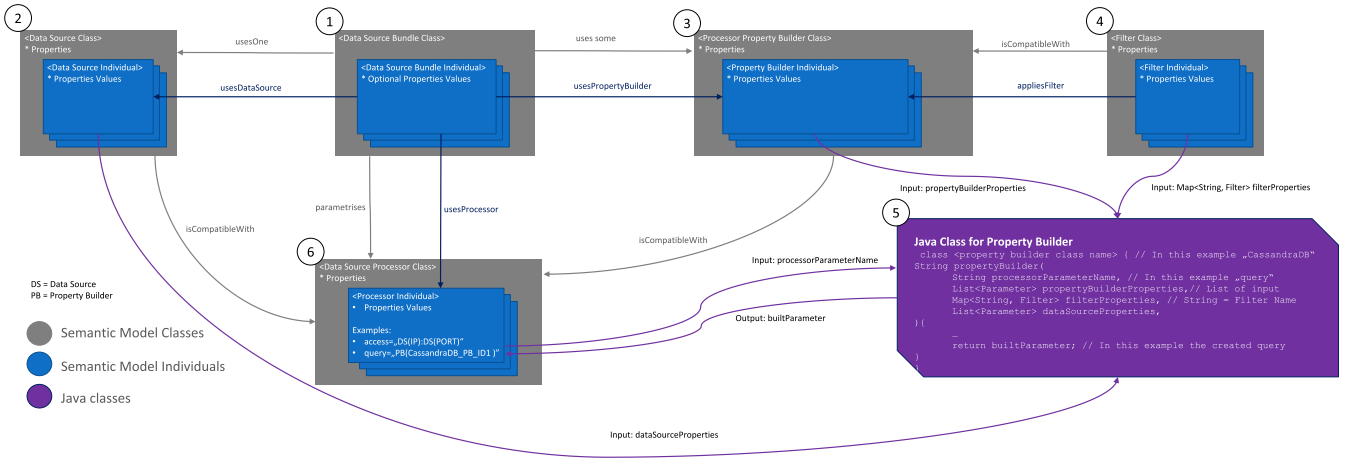



**FIGURE 3.** Data Source Bundle.

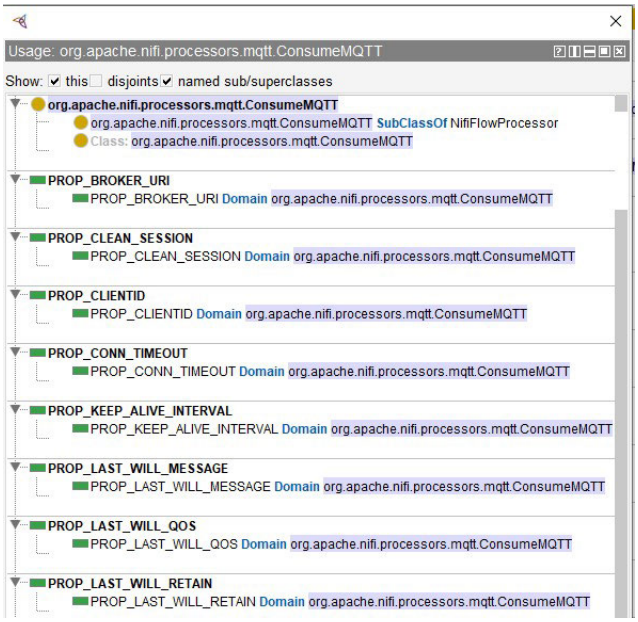



**FIGURE 4.** Data Source Processor Class exemplary based on Apache NiFi.

This example shows the simplest/static property mapping. Properties defined in a data source (DS) are referenced using the function DS(). A string builder which uses the semantic model as basis, has to be implemented to build the string by using the linked properties. The generated result in this example would be for instance "127.0.0.1:8080", which would be mapped to the parameter/data property "access" of the data access processor individual. Some other data properties get not initialised in the data source processor individual directly because the value, which has to be generated, is not static and needs some logic for its generation. This happens for instance if an SQL database query has to be generated based on a table name, optional filters or aggregation rules. For those properties are used processor property builders (PB) (see number (3) in FIGURE 3). Such property builder individuals contain input parameters for different builder classes that have to be implemented (see number (5) in FIGURE 3). The Processor Property Builder concept is extended by optional filters (see number (4) in FIGURE 3). In case that a property builder individual applies a compatible filter or filters (plural), then these filter properties are also considered by the linked property builder algorithm shown in number (5) of FIGURE 3. Note: This concept shall be extended with other features like aggregation rules. The use of a property builder in a data source processor individual follows following syntax:

*<data property>=PB(<property builder individual name>)*

**An Example (compare FIGURE 3 number (6)):**

*query="PB(CassandraDB_PB_ID1)"*

This example shows the dynamic property building. The function name PB() stands for Property Builder. CassandraDB_PB_ID1 in the example is the individual name of a specific PB class. The class brings a list of required properties to be initialised in the individual (transfer the concept shown in FIGURE 4). For example: following Cassandra Query Language (CQL) [52] query would address the needed data.

*SELECT columnName FROM keyspaceName.tableName*

This would mean that the related CassandraDB PB class would require the properties columnName, keyspaceName and tableName to build a query. These data properties are initialised in the related individual CassandraDB_PB_ID1. The implemented PB (see number (5)) uses these parameters to build the CQL query. In addition, FIGURE 3 shows also how a PB can be extended by filters. As example will be extended the CQL query by a WHERE clause:

*SELECT columnName FROM keyspaceName.tableName*
***WHERE date > '2020-01-01'***

A filter class with the name beforeDate is compatible with the PB class CassandraDB and would require the property date. In case that the PB CassandraDB should apply the filter beforeDate, then an individual of this filter has to

be instantiated which includes the initialised property date. The PB would then build the CQL query including the WHERE clause. How an PB algorithm can be implemented, is introduced number (5) in FIGURE 3. PB algorithms are part of the Data Selector module introduced in section II, FIGURE 1, number (3). Although the Data Selector is part of the implemented prototype, which will be explained in detail in an upcoming publication, the Property Builder has a direct connection to the Data Selector and therefore a small anticipation is needed. The Data Selector module uses a SPARQL engine to navigate through the semantic model. In case a data source processor has to be configured, the data selector module searches in the data source processor individual properties for syntaxes that requires a static mapping (for instance DS(IP)) or a dynamic data property creation (for instance PB(CassandraDB_PB_ID1)). Steps in case of a static Data Source mapping:

- Step 1: Data Selector identifies the DS() function name
- Step 2: Data Selector collects needed data properties from the linked data source individual (for instance IP and Port).
- Step 3: Data Selector replaces the static mapping syntax with the property values and uses the result for the data source processor pre-configuration.

Steps in case of a dynamic property building:

- Step 1: Data Selector identifies the function name PB()
- Step 2: Data Selector reads the addressed individual (for instance CassandraDB_PB_ID1) and identifies the related class name based on its type (for instance CassandraDB).
- Step 3: Data Selector searches for the related Java class which has the same name as the PB class in the semantic model (compare number (5)). If successful:
- Step 4: Data Selector collects data property values from the linked individuals as shown as inputs in FIGURE 3.
- Step 5: Data Selector builds based on all inputs the property value (for instance a query).

### C. VOCABULARY

Instantiated classes/individual are described by a range of data properties. In this work will be used a property definition partly based on the IEC 61360 [53] to standardise these properties. According to DIN Spec 91345, mandatory (standardized), optional (standardized) and free (not standardized) properties can be defined to describe an I40 component. This concept will be derived for specific classes of the semantic model. FIGURE 5 shows a few derived mandatory vocabularies used for the instantiation of the described semantic model classes. Some classes as ''User'' get other properties. TABLE 1 sumarises mandatory properties addressed in FIGURE 5.

*Remark:* The reader interested in more information concerning ongoing property standardisation activities in the area of smart manufacturing can find it for instance in [54].

**TABLE 1. Property description.**

| Property | Description |
|---|---|
| IRDI | International Registration Data Identifier (IRDI). |
| preferredName | Name of a property |
| definition | Definition of a property |
| versionNumber | Number of the version |
| versionDate | Date of the version publication |
| note | Note to a property. |
| labels | Labels are keywords related to a property. Format is $[label1, label2, \ldots]$ |
| username | The username of a user |
| password | The hashed password of a user. |
| userGroupName | The name of a user group. |

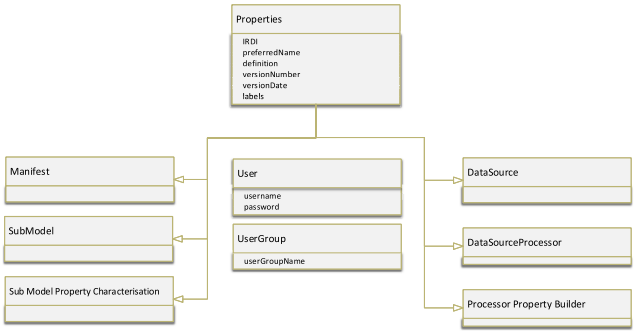



**FIGURE 5. Mandatory properties partly based on IEC 61360.**

### D. IMPLEMENTATION

The semantic model implementation is divided in different namespaces (/Internationalized Resource Identifiers (IRI)), as shown in FIGURE 6. One major requirement for this implementation is to enable an easier integration of new vocabularies, I40 components, smart manufacturing environments and sub models, but also to achieve a better overview, simplifying the engineering and the maintaining of the semantic model. FIGURE 6 shows the different namespaces represented as bubbles. While the schema and linkage namespaces are given by our implementation, the instantiations use the schema templates to represent the individual customised contents using also individual namespaces. The namespaces for schemas and linkages are beginning with ''http://www.i40semanticmanager.de'' and have the following endings:

Schemas:

- Vocabulary Schema: /schema/vocabulary
- Smart Manufacturing Environment Schema: /schema/smartmanufacturingenvironment
- Sub Model Schema: /schema/submodel
- I40 component Schema: /schema/i40component
- Date Selector Schema: /schema/dataselector

Linkage:

- Vocabulary Linkage: /linkage/vocabulary
- I40 Components, Smart Manufacturing Environment, Data Selector Linkage: /linkage/smartmanufacturingenvironment
- Sub Model Linkage: /linkage/submodel

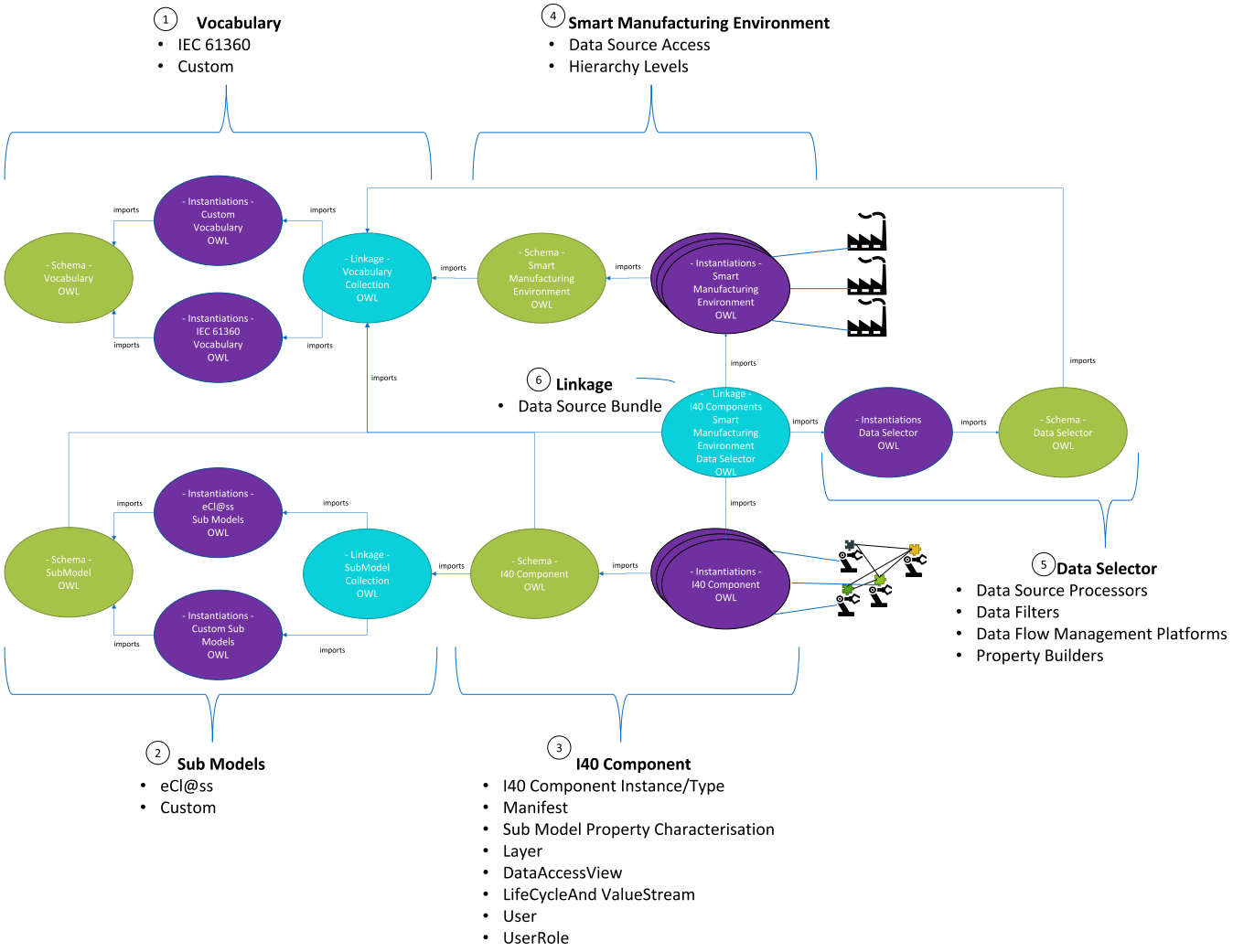



**FIGURE 6.** **Semantic model division into different namespaces/IRIs.**

The schema bubbles include classes, object properties, data properties, some generic individuals and their relations to describe main elements of the semantic model. The instantiations bubbles show the instantiated individuals that uses the related schemas. One instantiation could be for instance an I40 component that uses the I40 component schema.

The linkage bubbles show linkages and collection of/ between main elements. They act as a collection of individuals to generate one central point of access and to simplify the engineering process. For instance, the sub model collection includes all eCl@ss based sub models and all custom sub models. The main elements of the semantic model are:

- **Number 1:** Vocabulary – which describes the mandatory and optional vocabularies in form of data properties partly based on standards like the IEC 61360. This bundle is divided in a schema namespace, instantiation namespaces (IEC 61360; Custom) and a collection namespace that bundles all instantiations.
- **Number 2:** Sub models – which describes the structure of classification classes partly based on standards like eCl@ss. This bundle is divided in a schema namespace, instantiation namespaces (eCl@ss; Custom) and a collection namespace that bundles all instantiations.
- **Number 3:** I40 component – which describes classes and object properties that are directly linked to an I40 component. An I40 component includes classes to describe the I40 component category (is it an instance or a type), the manifest, layers, sub model property characterisations, data access views, life-cycles and value streams and user access. The bundle is divided in a schema namespace and in instantiation namespaces that initialise I40 components.
- **Number 4:** Smart manufacturing environment – which defines classes and object properties to describe a smart manufacturing environment. A smart manufacturing environment includes hierarchy levels and data sources. The bundle is divided in a schema namespace and in instantiation namespaces that initialise smart manufacturing environments.
- **Number 5:** Data selector – the data selector part includes for instance data source processors, data filters, data flow management platforms and property builders.

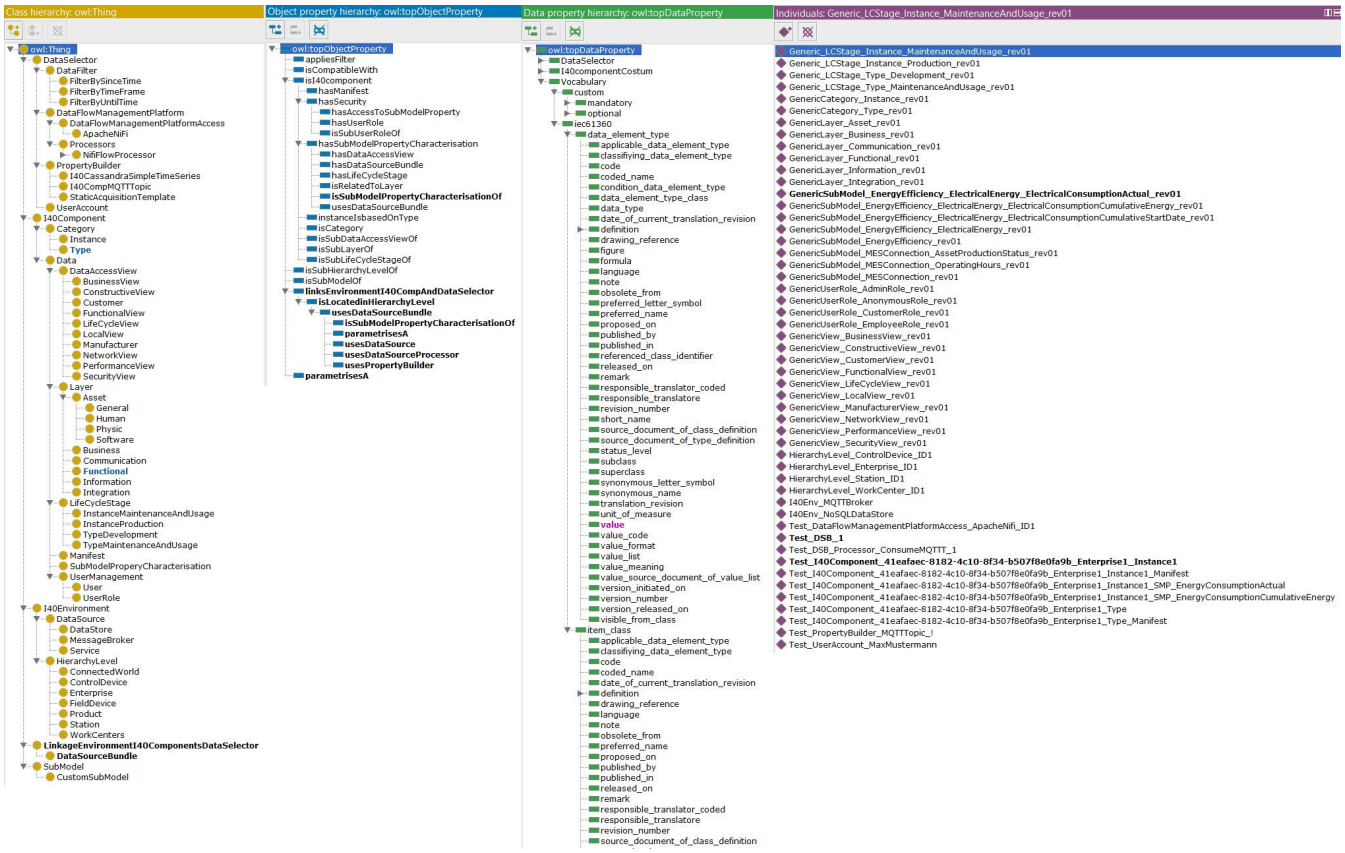



**FIGURE 7.** Semantic meta data model including some example individuals.

- **Number 6:** Linkage – the central point where all individuals of smart manufacturing environments, I40 components and the data selector get joined. In this namespace are also defined the data source bundles.

FIGURE 7 shows the complete semantic meta data model including some example individuals. It includes all classes, object properties, data properties, some testing individuals, and their relationships caused by the imported individuals and all related indirect imports (compare to FIGURE 6).

At this stage the major results of applying the pipelining approach is a prototype implementation of the semantic manager, the data selector and the data provider modules (presented in section II). It is important to note that the semantic meta data model is instantiated for the prototype and will be handled by the semantic manager as well as exploited by the data selector (data understanding, exploration, filtering, etc.). It follows the work of the data provider, which is able to extract the data described in the semantic model based on information provided by the Data Source Bundle. Although the implemented Data Selector prototype is a web application offering comprehensive search functionalities and the Data Provider is based on Apache NiFi, they can also be integrated e.g. in KNIME or RapidMiner to extract fast needed data for data analytics; could become an extension of Node-RED or could be integrated in further existing data pipelining technologies.

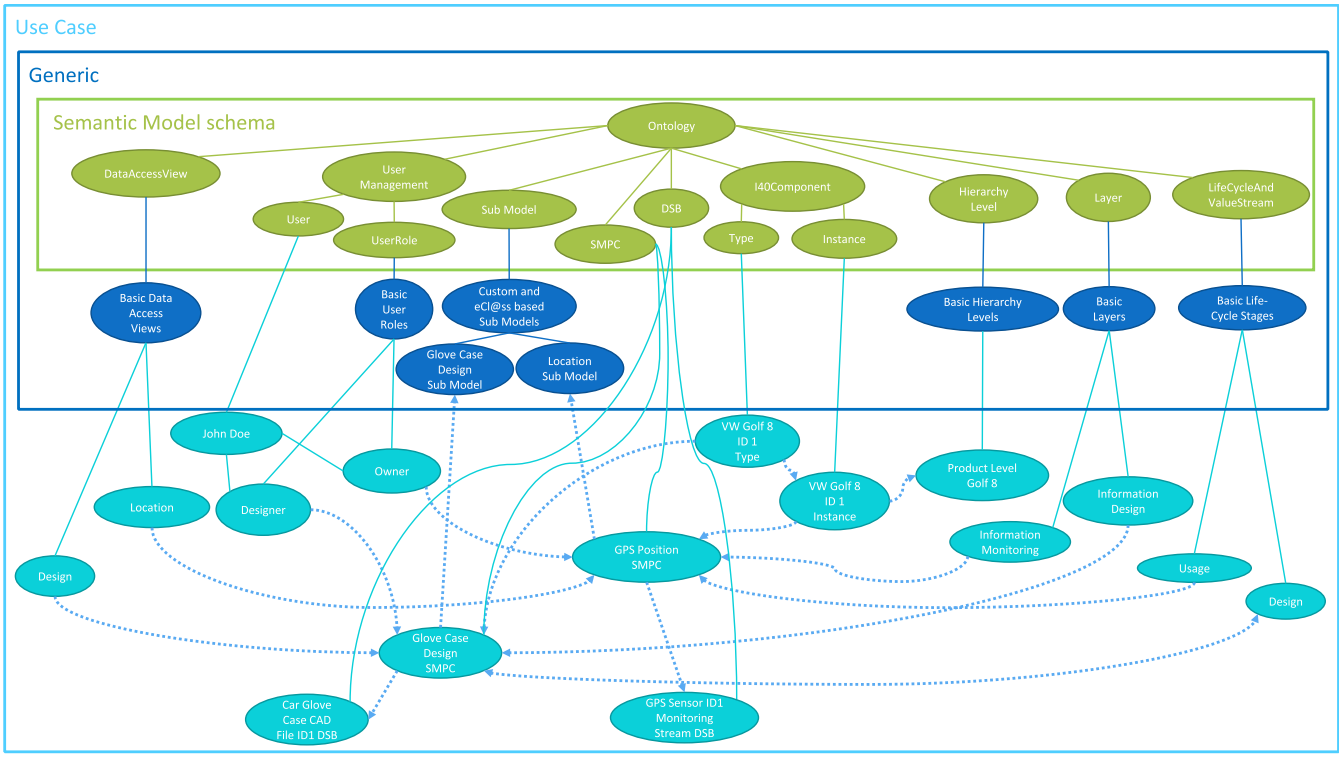
In order to show that the prototype results of the application of the pipelining approach can be classified at the level 6 of the international TRL scale [55], i.e. technological maturity with a prototype demonstration in an operational environment, the next section elaborates a real application.

## VI. USE CASE

In order to validate the usability of the semantic model defined, specified and described in the sections before, a real industrial use case for data pipelining in a context of Industry 4.0-compliant manufacturing will be detailed in this section.

The use case presents the application of the data pipelining approach over the life-cycle for the manufacturing of a car VW Golf 8 [56]. More specifically, this section describes the instantiation of the semantic model related to a glove case design (CAD file) of the VW Golf 8 type.

FIGURE 8 shows a simplified instantiation of a VW Golf 8 car specified as an I40 component with the instantiated semantic data model. The semantic data model schema area describes the basic taxonomy of the semantic model (compare FIGURE 2 to get more details). The generic area contents few generic basic classes and sub-models (e.g. sub models based on eCl@ss (see section VI)). The individual instantiation block presents the product, i.e. the car, as an I40 component. The reader can derive the meaning of the




SMPC = SubModel Property Characterisation; DSB = Data Source Bundle

**FIGURE 8. Simplified Instantiation of a VW Golf 8 car.**

relations between individuals and classes by comparing with FIGURE 2. In order to show few main parts of the semantic model, FIGURE 9 presents with more details the ''Glove Case Design Sub Model'', ''Glove Case Design SMPC'' and ''Car Glove Case CAD File ID1 DSB'' (Data Source Bundle)'' of the Glove Case Design example of FIGURE 8. More specifically, FIGURE 9 shows in detail an instantiation of Sub Models - exemplary based on eCl@ss classification classes -, a Sub model property characterization and a Data Source Bundle without the need of a property builder. The sub-model taxonomy classifies properties of a glove case in a vehicle. Next to the data properties standardized by eCl@ss, this example adds one property classification in form of a sub model. The added property classification gets the IRDI ''0174-nagorny-1#02-123456#001'' and classifies a CAD file for an object using the data source bundle approach. Remark: In this use case, the Volkswagen AG adds to the I40 component ''VW Golf 8'' a glove case CAD file with the IRDI ''0175-Volkswagen-1#02-D5GN2F#001''. The CAD file is located in the VW FTP server storage which has the IRDI ''0174Volkswagen-1#02-AFY951404#001''. Therefore, the data source bundle is linked to this data source access individual. For this real Use Case, the Volkswagen AG decided to use Apache NiFi as data pipelining technology. Apache NiFi provides the data source processor ''GetFTP'' to extract data from a FTP server. Therefore, a data source processor individual is created with details to reach the CAD file. The data source processor has the IRDI ''0176-ApacheNifi-1#02-1LSU4F#001'' and saves the reference to the CAD file (path and file name). Based on the given information, it is possible to generate a configuration for the ''GetFTP'' Apache NiFi Flow processor, as described in section V-B). A software which uses this information is able to deploy the pre-configured Apache NiFi Flow processor in an Apache NiFi instance using the Apache NiFi API. After this step, the NiFi Flow processor is directly ready to extract the needed data.

The supplementary video of this paper also contains a prototype demonstration related to this use case.

## VII. DISCUSSION

The overall approach as introduced in section II aims to improve data understanding and preparation. The questions to answer were: (i) how the time intensive data acquisition, data preparation and data understanding processes could be significantly and efficiently improved? (ii) Since the progressive semantic unification, driven by reference architectures like RAMI 4.0 or standards like eCl@ss in Smart Manufacturing, evolves self-describing cyber-physical systems, how this evolution could be exploited for data scientists in the data understanding and data preparation? The approach described in this paper is an attempt to start giving concrete and feasible answers to such questions, combining latest research and innovation outcomes in fields like the Industry 4.0 and data

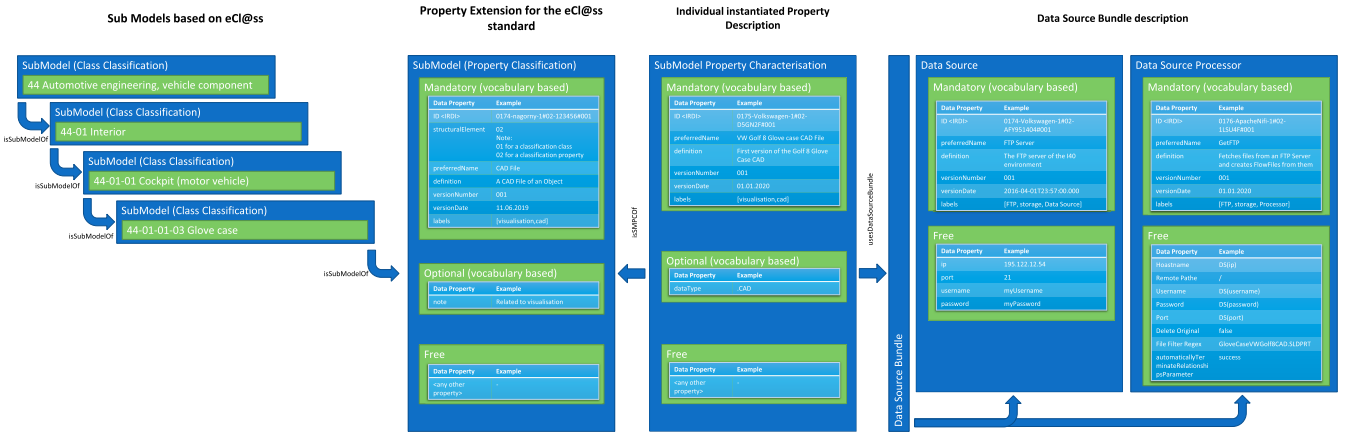



**FIGURE 9.** Exemplary Sub Model, SMPC and Data Source Bundle instantiation for a VW Golf 8 Glove Case CAD File.

analytics. Available data pipelining approaches still require high efforts and domain knowledge for their configuration, although many features, as automatically data type mapping, data source technology support and user-friendly interfaces, are provided. The semantical unification could support those kinds of approaches in the future, by using self-descriptive information coming from the cyber-physical systems in a smart manufacturing environment suitable to easily build a data pipeline of selected data. In the same context, there is another circumstance that will probably exist also for the next decades and needs to be carefully considered. Based on the historical progression can be derived that legacy systems in production will be used also in future for partly more than 20 years. The building of an Industry 4.0 compliant smart manufacturing system is therefore in many cases a brownfield transformation and does not emerge out of a greenfield. Also, the creation of a unified semantic is still a very volatile field where hundreds of standards are existing and many implementations are proprietary. All this concludes that also in the next decades we will struggle with heterogeneous data environments and that a central point of data access where all data producers provide their data in a unified and standardised way will not be available too fast as required. Therefore, the presented approach uses a basic semantic model schema based on the Industry 4.0 reference architecture RAMI 4.0, which can be built on top (smart) manufacturing systems, which is open for the integration of further data classification standards and which enables data extraction from each connected kind of a data source. And this by improving the use of established data pipelining approaches suitable for green-fields as well as on top of existing manufacturing brownfields without harming them. The approach could support several structured and unstructured problems [57]. A typical structured problem could be e.g. a failure in a manufacturing system. For such problems our approach would provide the possibility to build groups for data e.g. for diagnostic. Such groups can also be further refined/divided with sub-groups for more specific diagnostic problems. A typical unstructured problem could address quality deviations in a process which are often more complex. Such complex problems require a human expert who uses the available meta-data to search and find relevant data for an analysis. Such functionalities are provided by our first prototype as presented in the demo of the complementary video. However, to address a potential research in this area: The human expert could also be replaced by a self-learning AI which uses available meta data, provided by the Semantic Manager approach, to search, find and access needed data for a sub-sequent analysis.

## VIII. SUMMARY AND OUTLOOK

### A. SUMMARY

The paper introduced a semantic model-based DIN Spec 91345 compliant big data pipelining approach and presented the major specifications of the used semantic model using a real industrial product, a VW Golf 8 car, as result of a first exemplary TRL-6 implementation of the approach to support data analytics in a smart car manufacturing environment. The paper was also linked to supplementary video which provides further information on the overall approach and improved its understanding. The application of the pipelining approach shows how the data, stored in various data silos or distributed over various communication technologies in current (smart) manufacturing environments can be accessed and extracted. The last function, i.e. the extraction of data, is performed by the definition of a data source bundle usable to parametrise data source processors, which can then be deployed in a data pipeline solution. As explained in section VI, for the use case of VW described in the paper, Apache NiFi is the scalable data pipelining technology selected to generate a central access point to data. Moreover, with their results the authors were able to show how through an association of this data source bundle to a RAMI 4.0 compliant semantic model, integrated with data classification standards (as exemplary shown with the standard eCl@ss in section VI), the approach is suitable to explore, search, filter, identify, understand and

select adequate data using various SPARQL queries. It is important to reinforce that this work is based on the use of standards (as explained in section IV and V) that are currently applied in the area of smart manufacturing and digital factory [47]. The provision of a DIN Spec 91345 compliant semantic model, considering standardised data classifications classes, creates a highly flexible data knowledge graph that can also be extended by or integrated with other existing ontologies. As major features of the approach, it is possible to conclude that the presented semantic model:

- provides unified and standardized templates to classify and describe data.
- enables an easy integration of classification standards as exemplarily presented with eCl@ss.
- enables the management of data along the whole life-cycle of an asset.
- remains generic by separating data classification, I40 component description, data pipelining technology and data source technologies, which enables customization and an easy integration even into existing/legacy environments.

Based on the specified semantic model, an adequate SW-architecture was developed and prototype implemented. The first results of the implementation are enable to define a smart manufacturing environment including RAMI4.0-compliant I40 components. The prototype provides a user-friendly web-based interface where various SPARQL queries are used in the background to search for needed data based on IRDIs, Hierarchy Levels, Life-cycle stages, Subs Models, Data Properties, etc. Related/linked data source bundles are used to deploy pre-configured data source processors in a running Apache NiFi instance.

### B. OUTLOOK

The prototype implementations are currently in a testing phase and get frequently further refinements. Although key performance indicators (KPI) measurements are still missing, the first results so far are promising and show that data understanding and data access can be significantly improved by our approach. Further innovations are required to increase the potential technologically and functional impact of the application of the approach, as e.g.

- Refining the logical and physical architecture of the prototype which exploits the presented semantic model and derived data pipelining approach.
- Providing a guideline with a generic workflow for using the approach based on the lessons learned from the implemented prototype and use case.
- Extending the set of practical use cases for proving the applicability of the approach.
- Discussing, interpreting and particularly assessing the impact of implementing the approach in other data ecosystems contexts (e.g., pipelining data from a smart manufacturing shop floor providing data harvesting within the CROSS-CPP [58]).
- Support the evolution towards blockchain-enabled manufacturing systems: The Semantic Manager could use blockchain technology to manage the semantic model. This would enable the tracing of smart manufacturing environment changes in an unmanipulable way.
- Step 6 in FIGURE 1 opens new research opportunities to address data preparation challenges. Following questions could be addressed: how could the approach be extended, (1st) to fill data into standardized data models addressing data transformation processes, (2nd) to bundle and link datasets for specific data analysis challenges, and (3rd) to clean/classify/balance data(-sets) for a sub-sequent data analysis.
- The Data Selector (as shown in the Demo addressed in section VI) is extendible by many features for visual filtering like in Google Maps for traffic or Windy for weather.

However, while this paper had a focus on the specification of the semantic model to initialize the data pipelining approach, some aspects have not been addressed in detail as the e.g. the integrity of data, interoperability and compatibility with SotA ontologies as OntoCAPE [59] or efficient high-performance data extraction from large semantic models. Such aspects have to be focused and will be published in the next reports.

## REFERENCES


[1] C. Shearer, "The CRISP-DM model: The new blueprint for data mining," *J. Data Warehousing*, vol. 5, no. 4, pp. 13–22, 2000.

[2] M. Jones. (2011). *Crisp-Dm Model, Cross-Industry Standard Process for Data Mining*. [Online]. Available: http://geuder.tumblr.com/post/3028113424/crisp-dm-model-cross-industry-%standard-process

[3] K. J. Cios, W. Pedrycz, R. W. Swiniarski, and L. A. Kurgan, *Data Mining: A Knowledge Discovery Approach*. New York, NY, USA: Springer, 2007. [Online]. Available: https://books.google.de/books?id=YvTxwaLJJ2kC

[4] O. Givehchi, K. Landsdorf, P. Simoens, and A. W. Colombo, "Interoperability for industrial cyber-physical systems: An approach for legacy systems," *IEEE Trans. Ind. Informat.*, vol. 13, no. 6, pp. 3370–3378, Dec. 2017.

[5] Y. Qu, X. Ming, Z. Liu, X. Zhang, and Z. Hou, "Smart manufacturing systems: State of the art and future trends," *Int. J. Adv. Manuf. Technol.*, vol. 103, pp. 3751–3768, Aug. 2019.

[6] S. S. Shipp, N. Gupta, B. Lal, J. A. Scott, C. L. Weber, M. S. Finnin, M. Blake, S. Newsome, and S. Thomas, "Emerging global trends in advanced manufacturing," Institute For Defense Analyses Alexandria, Alexandria, VA, USA, Tech. Rep. 0704-0188, 2012.

[7] D. Laney. (2018). *Explanation of 3v's Model of Big Data*. [Online]. Available: https://www.cosoit.com/explanation-of-3v-model-of-big-data

[8] J. Qiu, Q. Wu, G. Ding, Y. Xu, and S. Feng, "A survey of machine learning for big data processing," *EURASIP J. Adv. Signal Process.*, vol. 2016, no. 1, p. 67, 2016.

[9] Wikibooks. (2018). *Data Science: An Introduction/a Mash-up of Disciplines*. [Online]. Available: https://en.wikibooks.org/w/index.php?title=Data_Science:_An_Introduction/A_Mash-up_of_Disciplines&oldid=3484786

[10] P. Vassiliadis, "A survey of extract–transform–load technology," *Int. J. Data Warehousing Mining*, vol. 5, no. 3, pp. 1–27, 2009.

[11] V. Ranjan, "A comparative study between ETL (extract, transform, load) and ELT (extract, load and transform) approach for loading data into data warehouse," 2009. [Online]. Available: https://www.semanticscholar.org/paper/A-Comparative-Study-between-ETL-(-)-and-ELT-(-)-for-Ranjan/b6aa6cd1aec2c36c8d7e573b109a8d1d2e87b593

[12] N. Miloslavskaya and A. Tolstoy, "Big data, fast data and data lake concepts," *Procedia Comput. Sci.*, vol. 88, pp. 300–305, Jan. 2016.

[13] A. Fay, C. Diedrich, M. Dubovy, C. Eck, C. Hildebrandt, A. Scholz, T. Schröder, and R. Wiegand, "Vorhandene Standards als semantische Basis für die Anwendung von Industrie 4.0 (SemAnz40)," Universitätsbibliothek der Helmut-Schmidt-Universität, Hamburg, Germany, Tech. Rep., 2017. [Online]. Available: https://edoc.sub.uni-hamburg.de/hsu/volltexte/2018/3193/
[14] R. Jardim-Goncalves, J. Sarraipa, and A. Steiger-Garcao, "Semantic harmonization for seamless networked supply chain planning in the future of Internet," in *Enterprise Architecture, Integration and Interoperability*. Berlin, Germany: Springer-Verlag, 2010, pp. 78–89.
[15] A. Fay, C. Diedrich, T. Schröder, M. Dubovy, C. Eck, and R. Wiegand, "Semantik für industrie 4.0-systeme—Die basis für den informationsaustausch in industrie 4.0-anwendungsszenarien," Semantische Allianz Für Industrie 4.0, Institut Automatisierungstechnik—Helmut-Schmidt-Universität/Universität der Bundeswehr Hamburg, Hamburg, Germany, Tech. Rep. 1, 2017. [Online]. Available: https://www.hsu-hh.de/aut/wp-content/uploads/sites/670/2017/12/Brosch%C3%BCre_Semanz40.pdf
[16] U. Sendler, *ProSTEP iViP Verein*. Springer, 2009, pp. 377–382.
[17] A. Lüder and N. Schmidt, *AutomationML a Nutshell*. Berlin, Germany: Springer, 2017, pp. 213–258, doi: 10.1007/978-3-662-53248-5_61.
[18] O. Foundation. (2012). *What is the Opc Foundation?: Opc Foundation: The Interoperability Standard for Industrial Automation & Other Related Domains*. [Online]. Available: https://web.archive.org/web/20120220234456/http://www.opcfoundation.org/Default.aspx/01_about/01_history.asp?MID=AboutOPC
[19] PLCopen. (2019). *What is Plcopen*. [Online]. Available: https://plcopen.org/what-plcopen
[20] I. E. Commission. (2019). *About the IEC—What we do Facts & Figures*. [Online]. Available: https://www.iec.ch/about/activities/facts.htm
[21] (2019). *About Us: What we do, Structure, Members, News and Events*. [Online]. Available: https://www.iso.org/about-us.html
[22] (2019). *IEEE at a Glance*. [Online]. Available: https://www.ieee.org/about/today/at-a-glance.html
[23] L. Halilaj, I. Grangel-González, G. Coskun, S. Lohmann, and S. Auer, "Git4voc: Collaborative vocabulary development based on git," *Int. J. Semantic Comput.*, vol. 10, no. 2, pp. 167–191, 2016.
[24] L. Halilaj, N. Petersen, I. Grangel-González, C. Lange, S. Auer, G. Coskun, and S. Lohmann, "Vocol: An integrated environment to support version-controlled vocabulary development," in *Proc. Int. Conf. Knowl. Eng. Knowl. Manage. (EKAW)*. Bologna, Italy: Springer, 2016, pp. 303–319.
[25] I. Fraunhofer. (2018). *Vocol—An Integrated Environment for Collaborative Vocabulary Development*. [Online]. Available: https://vocol.iais.fraunhofer.de/
[26] I. Grangel-González. (2017). *Ramivocabulary, An Ontology to Represents the Reference Architecture Model for Industry 4.0 (RAMI), Including the Concept of the Administration Shell I4.0 Component.@en*. [Online]. Available: https://vocol.iais.fraunhofer.de/rami/
[27] G. Alley. (Apr. 26, 2019). *What is a Data Pipeline*. [Online]. Available: https://dzone.com/articles/what-is-a-data-pipeline
[28] M. J. Denney, D. M. Long, M. G. Armistead, J. L. Anderson, and B. N. Conway, "Validating the extract, transform, load process used to populate a large clinical research database," *Int. J. Med. Informat.*, vol. 94, pp. 271–274, Oct. 2016.
[29] T. A. S. Foundation. (2019). *An Easy to Use, Powerful, and Reliable System to Process and Distribute Data*. [Online]. Available: https://nifi.apache.org/
[30] B. Samal and M. Panda, "Real time product feedback review and analysis using apache technologies and nosql database," *Int. J. Eng. Comput. Sci.*, vol. 6, no. 10, pp. 22551–22558, 2017.
[31] StreamSets. (2019). *Streamsets Data Collector—Data Collection Challenges With Building Robust Dataflow Pipelines*. [Online]. Available: https://streamsets.com/products/sdc
[32] H. D. Inc. (2019). *Unified Data Platform for Customer Focused Companies—Break Data Silos and Put Your Data Into Action*. [Online]. Available: https://hevodata.com/
[33] A. S. Foundation. (2019). *Apache Airflow Documentation*. [Online]. Available: https://airflow.apache.org/
[34] J. Bowen, *Getting Started With Talend Open Studio for Data Integration*. Birmingham, U.K.: Packt Publishing Ltd, 2012.
[35] M. Hankel and B. Rexroth. (2015). *Das Referenzarchitekturmodell Industrie 4.0 (RAMI 4.0)*. Zentralverband Elektrotechnik-und Elektronikindustrie e. V. Accessed: Jan. 11, 2019. [Online]. Available: https://www.zvei.org/fileadmin/user_upload/Presse_und_Medien/Publikationen/2015/april/Das_Referenzarchitekturmodell_Industrie_4.0__RAMI_4.0_/Faktenblatt-Industrie4_0-RAMI-4_0.pdf
[36] *Smart Manufacturing-Reference Architecture Model Industry*, document PAS and IEC, 63088: 2017, 2017, vol. 4.
[37] *Reference Architecture Model Industrie 4.0 (rami4.0)*, document D. S. 91345:2016-04, 2016. [Online]. Available: http://www.beuth.de/de/technische-regel/din-spec-91345/250940128
[38] J. Trefke, S. Rohjans, M. Uslar, S. Lehnhoff, L. Nordström, and A. Saleem, "Smart grid architecture model use case management in a large European smart grid project," in *Proc. IEEE PES ISGT Eur.*, Oct. 2013, pp. 1–5.
[39] S.-W. Lin, B. Miller, J. Durand, R. Joshi, P. Didier, A. Chigani, R. Torenbeek, D. Duggal, R. Martin, and G. Bleakley, "Industrial Internet reference architecture," Industrial Internet Consortium (IIC), Needham, MS, USA, Tech. Rep. 304, 2015.
[40] M. Happacher. (2015). *Industrie 4.0 Architekturen—Rami Und Iira Im Vergleich*. [Online]. Available: https://www.elektroniknet.de/rami-und-iira-im-vergleich-121818-Seite-3.%html
[41] *Big Data Value Strategic Research and Innovation Agenda*, B. D. V. Association, Seattle, WA, USA, 2016.
[42] *Chargenorientierte Fahrweise, Teil 1, Modelle Und Terminologie*, document DIN, 61512-1, DIN Deutsches Institut Für Normung, Berlin, Germany, 2000.
[43] *Enterprise-Control System Integration—Part 1: Models and Terminology*, document IEC/FDIS, 1-62264, I. E. Commission, IEC, Geneva, Switzerland, 2003.
[44] *Life-Cycle Management for Systems and Products Used in Industrial-Process Measurement, Control and Automation*, document IEC 62890, I. E. Commission, 2016.
[45] A. Bondza, C. Eck, R. Heidel, M. Reigl, and D. S. Wenzel, "Mit daten und semantik auf dem weg zur industrie 4.0," eCl@ss e.V, Cologne, Germany, Tech. Rep. 1, 2018.
[46] *Information Technology—Metadata Registries (MDR)—Part 6: Registration*, Standard ISO/IEC, ISO/IEC 11179-6, 2015.
[47] *Structure of the Administration Shell—Continuation of the Development of the Reference Model for the Industrie 4.0 Component*, Federal Ministry for Economic Affairs and Energy (BMWi), ZVEI, Frankfurt, Germany, 2016.
[48] B. Boss, S. Malakuti, S.-W. Lin, T. Usländer, E. Clauer, M. Hoffmeister, L. Stojanovic, and B. Flubacher, "Digital twin and asset administration shell concepts and application in the industrial Internet and industrie 4.0. An industrial Internet consortium and plattform industrie 4.0 joint whitepaper," German Federal Ministry Econ. Affairs Energy, Berlin, Germany, Tech. Rep., Sep. 2020. [Online]. Available: https://www.plattform-i40.de/PI40/Redaktion/DE/Downloads/Publikation/Digital-Twin-and-Asset-Administration-Shell-Concepts.html
[49] E. V. eCl@ss. (Oct. 9, 2018). *ecl@ss—Property*. [Online]. Available: http://wiki.eclass.eu/w/index.php?title=Property&oldid=9069
[50] E. V. eCl@ss. (Jan. 31, 2019). *Irdi (International Registration Data Identifier)*. [Online]. Available: http://wiki.eclass.de/wiki/IRDI
[51] U. Döbrich, M. Hankel, R. Heidel, and M. Hoffmeister, *Basiswissen RAMI 4.0: Referenzarchitekturmodell und Industrie 4.0-Komponente Industrie 4.0*. Berlin, Germany: Beuth Verlag, 2017.
[52] T. A. S. Foundation. (2016). *The Cassandra Query Language (CQL)*. [Online]. Available: https://cassandra.apache.org/doc/latest/cql/
[53] I. Reihe, *Standard Data Elements Types With Associated Classification Scheme for Electric Items—Part*, vol. 1, Standard 61360 IEC 61360-1, Jul. 2009.
[54] T. Hadlich, "Verwendung von merkmalen im engineering von systemen," M.S. thesis, Dept. Elect. Eng. Inf. Technol., Otto-von-Guericke-Univ. Magdeburg, Magdeburg, Germany, 2015.
[55] (2015). *General Annexes—G. Technology readiness levels (TRL)*, E. Commission Horizon 2020 Work Programme 2014-2015. [Online]. Available: https://ec.europa.eu/research/participants/data/ref/h2020/wp/2014_2015/annexes/h2020-wp1415-annex-g-trl_en.pdf
[56] Wikipedia. (2019). *Vw Golf Viii*. [Online]. Available: https://de.wikipedia.org/w/index.php?title=VW_Golf_VIII&oldid=195024505
[57] C. G. Pont. (Nov. 2020). *Analysis of Business Problems*. [Online]. Available: https://www.coursera.org/lecture/analysis-business-problem-iese/3-struc%tured-vs-unstructured-problems-DWLQy
[58] C.-C. Consortium. (2017). *Ecosystem for Services Based on Integrated Cross-Sectorial Data Streams From Multiple Cyber Physical Products and Open Data Sources (Cross-CPP)*. [Online]. Available: https://cordis.europa.eu/project/rcn/214254/de
[59] J. Morbach, A. Wiesner, and W. Marquardt, "Ontocape—A (re) usable ontology for computer-aided process engineering," *Comput. Chem. Eng.*, vol. 33, no. 10, pp. 1546–1556, 2009.

**KEVIN NAGORNY** (Associate Member, IEEE) received the M.Eng. degree. He is currently pursuing the Ph.D. degree with the NOVA University of Lisbon. He is an Research Assistant with the Institute for Applied Systems Technology, Bremen. He studied electrical and automation technology, and industrial informatics at the University of Applied Sciences Emden-Leer, Germany, with a special focus on manufacturing systems. Since 2013, he worked in more than six international research projects as well as in industrial projects. As Ph.D. candidate at the NOVA University of Lisbon, he is researching in the topic of Big Data observation, analysis and diagnosis in the manufacturing domain. He has more than 18 publications on technical and research topics.

**SEBASTIAN SCHOLZE** (Member, IEEE) received the Dipl.-Inf. degree. He studied computer science at the University of Bremen. Since 2000, he is working as scientific staff member at ATB. He is involved in diverse CEC funded RTD projects since the 5th FP. He is active in researching on context aware approaches and systems, object-based software models, and methodologies for optimizing the software development process for distributed, SOA, agent-based and interoperable and context aware systems and web-based applications. He is working as project coordinator and local project manager in several EU and direct research projects. He has more than 50 publications on technical and research topics.

**ARMANDO WALTER COLOMBO** (Fellow, IEEE) received the Dr.-Ing. degree. He is currently a Professor for industrial informatics, automation and robotics with the University of Applied Sciences Emden/Leer, Germany. From 2001 to 2018, he worked as a Manager for collaborative innovation projects and also as Edison Level 2 Group Senior Expert at Schneider Electric. His research and innovation interests are in the fields of industrial digitalization, engineering of industry 4.0-compliant solutions, and system-of-cyber-physical systems. With his innovations, he has performed scientific and technical seminal contributions that are nowadays being used as one of the basis of what is recognized as ''The 4th Industrial Revolution'': Industrial Cyber-Physical Systems.

**JOSÉ BARATA OLIVEIRA** (Member, IEEE) received the Ph.D. degree in robotics and integrated manufacturing from the New University of Lisbon, in 2004. He is currently a Professor with the Department of Electrical Engineering, New University of Lisbon. He is a Senior Researcher with the UNINOVA Institute. He has participated in more than 15 International research projects involving different programs. His main research interests are in the area of intelligent manufacturing with particular focus on complex adaptive systems, and involving SOA based intelligent manufacturing devices. He has published over 60 original articles and is member of the several IEEE technical committees as for instance industrial agents or self-organization and cybernetics for informatics (SMC).