An Ontology Based Semantic Data Model Supporting A Maas Digital Platform

Giuseppe Landolfi, Andrea Barni, Gabriele Izzo, Elias Montini, Andrea Bettoni
Department of Innovative Technologies SUPSI
Manno, Switerland
giuseppe.landolfi@supsi.ch

Marko Vujasinovic, Alessio Gugliotta
Innova S.r.l.
Rome, Italy
m.vujasinovic@innova-eu.net

António Lucas Soares, Henrique Diogo Silva
INESCTEC, University of Porto-Faculty of Engineering
Porto, Portugal
asoares@inesctec.pt

***Abstract*—The integration of IoT infrastructures across production systems, together with the extensive digitalisation of industrial processes, are drastically impacting manufacturing value chains and the business models built on the top of them. By exploiting these capabilities companies are evolving the nature of their businesses shifting value proposition towards models relying on product servitization and share, instead of ownership. In this paper, we describe the semantic data-model developed to support a digital platform fostering the reintroduction in the loop and optimization of unused industrial capacity. Such data-model aims to establish the main propositions of the semantic representation that constitutes the essential nature of the ecosystem to depict their interactions, the flow of resources and exchange of production services. The inference reasoning on the semantic representation of the ecosystem allows to make emerge non-trivial and previously unknown opportunities. This will apply not only to the matching of demand and supply of manufacturing services, but to possible and unpredictable relations. For instance, a particular kind of waste being produced at an ecosystem node can be linked to the requirements for an input material needed in a new product being developed on the platform, or new technologies can be suggested to enhance processes under improvement. The overall architecture and individual ontologies are presented and their usefulness is motivated via the application to use cases.**

Keywords—servitization; manufacturing ontologies; semantic data-model; knowledge discovery

I. Introduction

Servitization is nowadays a common trend adopted by companies active in a wide range of industrial sectors that shifted income generation from the sale of the physical products to charging customers for the availability of a service or functionality that the product bring forth [1]. This paradigm applies both to consumer as well as to capital goods, as companies are more and more willing to sell (and buy) manufacturing capacity instead of new equipment, making the Manufacturing as a Service (MaaS) paradigm arise. This paradigm can be particularly relevant to either small or new businesses in high-tech sectors seeking for the right supplier or competence who may see their effort hindered by lack of visibility, leading to incapacity to find the right partners [2]. It appears evident that, to fully achieve the MaaS paradigm, an aggregation and information sharing point is required to give visibility to the manufacturing capacity that any company may want to make available at any location and field [3].

Digital platforms have been showing the potential of creating the digital ecosystem able to matchmake needs of customers and suppliers globally and across multiple sectors [4], [5], [6]. The adoption of these instruments is able to provide a consistent added value by increasing consumer convenience (time saving, accessible at any hour of the day, personalization, etc), reduced information asymmetry (rating systems, comparison tools), improved awareness (more accessible product information), greater choice (diversity of products and sellers), monetary benefits (offering packages of goods and services, promoting deals) and additional sources of income (consumers can sell their products on marketplaces, offer services through sharing economy platforms) [7], [8], [9]. Moreover, it has been extensively agreed ([10], [3], [11], [12]) that the adoption of digital platforms is able to foster the shift of company business models towards a service oriented approach.

Bettoni et al. [13] proposed a digital platform in the MaaS domain that goes far beyond the matchmaking of manufacturing resources, by not limiting its capabilities to the sole manufacturing equipment sharing, but extending the sharing potential to the whole manufacturing ecosystem value network. The matchmaking of resources in a cross-sectorial, multi-resources environment requires a proper representation of the semantics behind the description of the ecosystem itself. Ecosystem actors with their production capacities, technologies and available resources need to be formally depicted, characterizing the interactions, the flow of resources and exchange of production services existent or that could arise.

This paper aims at describing the core abstract concepts and consequently at establishing the main propositions of the semantic representation that constitutes the backbone structure of the data model behind the platform described above. The paper is therefore structured in a first section where, upon an introductive description of the ecosystems and functionalities the platform is expected to support, an analysis of efforts spent for the representation of knowledge in the manufacturing domain is presented. The core section of the paper thus describes the main elements characterising the data model representing ecosystem knowledge and the core ontology implementation of such model. The coming steps of validation of the proposed model are eventually highlighted.

978-1-5386-7097-2/18/$31.00 ©2018 IEEE

II. BACKGROUND

A. Manufacturing as a service platform to support sharing of unused resources

Starting from the idea of dematerialising production resources and of embracing servitization, the MANU-SQUARE platform creates an ecosystem that acts as a virtual marketplace bringing un-exploited production capacity, as well as other virtual and physical assets, closer to the production demand, to obtain their optimal matching. This has two main advantages: (i) the rapid and efficient creation of local distributed value networks for innovative providers of product services; (ii) the reintroduction and optimization in the loop of unused capacity and potential that would otherwise be lost. In doing so, MANU-SQUARE establishes an ecosystem that is organized to match the needs of buyers with the availability of sellers in terms of know-how, technology, manufacturing capacity and waste. The user needs matching is performed on the top of two flagship services: namely Resources finding & sharing, and Open Innovation Support.

Resource finding & sharing is the platform service which allows users to share resources available in their manufacturing environment. Users can share manufacturing capacities, capabilities, know-how, technology and by-products. To facilitate service access, from both users and platform side, this service is then further split in three sub-categories:

- Supply and demand of manufacturing capacity;
- Supply and demand of knowledge;
- Supply and demand of by-products.

The role of the MANU-SQUARE platform is not limited to enable manufacturing resources sharing, but also to provide support to innovation by embracing the open innovation paradigm. Following the motto "Not all the smart people work for us"[14], the platform aims to act as a facilitator for unleashing the full innovation potential residing in traditional companies, SMEs and start-ups, bringing new ideas or new technologies, competences and processes into play. The Open Innovation Support service is therefore divided in two sub-categories, which differ for the type of stakeholders these involves:

- Open Innovation Management (Innovation Facilitators)
- Consumers Involvement (Consumers)

In order to enable the delivery of the aforementioned services, the platform relies on a a set of functionalities running in the platform back-end (Table 1)

TABLE I. PLATFORM FUNCTIONALITIES DESCRIPTION

Functionality	Description
Production capacity matching	Matchmaking among Suppliers of available manufacturing capacity and Customers that aims to exploit that capacity. The platform proposes potential compliant Suppliers, filtering them according with user selected KPIs
Know-how capabilities matching	Matchmaking among Suppliers of available knowledge and Customers that require support in the related field of expertise.
By-product matching	The platform enables the matchmaking among manufacturers whose manufacturing system generates one or more by-products, and customers that can exploit the by-product as input resource.
Sustainability Assessment	The platform is able to support sustainability assessment of shared capacities, capabilities and by-products.
Ecosystem optimization	The platform is able to support the ecosystem optimization, adopting a specific objective function to rank suppliers and suggest most sustainable matchings.
User Profile management	The platform supports each user in the development of its profile reducing user efforts for data entering while optimizing the matching process.
Reputation management	The platform allows the users that are involved in a transaction to evaluate the involved parties.
Suppliers assessment	Ranking of results according to customer selected filtering parameters (cost, sustainability, , etc.).
Certifications management	The platform allows Auditors and Regulators to certify players through a verified and secure certifications management system.
Trust management	The functionality supports the management of information across the platform giving users the right to define the level of accessibility to provide to their information.
Communication support	The platform supports communication among platform users, streamlining connections and mediating the interactions among parties.
Innovation management	Starting from a user introduced idea, different users can provide tracked and structured contributions. The platform administrates the flow of contributions.
RFQ management	It provides the infrastructure to enable the definition and management of quotations, managing the level of visibility of the quotations and of the partners exchanging requests and transactions.
Transactions management	The functionality supports the creation of traceable transactions across the platform value network.
Reputation management	The platform allows the users that are involved in a transaction to evaluate the involved parties.

On the top of these functionalities, the MANU-SQUARE platform is able to support manufacturers in their experience with MaaS. A manufacturer may have a role of a supplier (seller) or a customer (buyer). In the case of the latter, a manufacturing company uses the platform each time they require to engage with the MANU-SQUARE ecosystem to fulfil a need, such as additional production capabilities. The platform performs the search for the optimal matching on a wide number of possible candidates from the MANU-SQUARE ecosystem, using a criterion that ensures high level of quality, reliability and reduction of costs. The ecosystem composition allows optimal matches also for non-trivial offerings of resources other than production hours or tangible assets, with the aim to identify and exploit unexpected synergies between participants and to promote the mutual interaction of diverse industries for beneficial reuse of competences and flows. In the following picture, the ecosystem, its needs and how the platform can address the is graphically depicted (Figure 1).

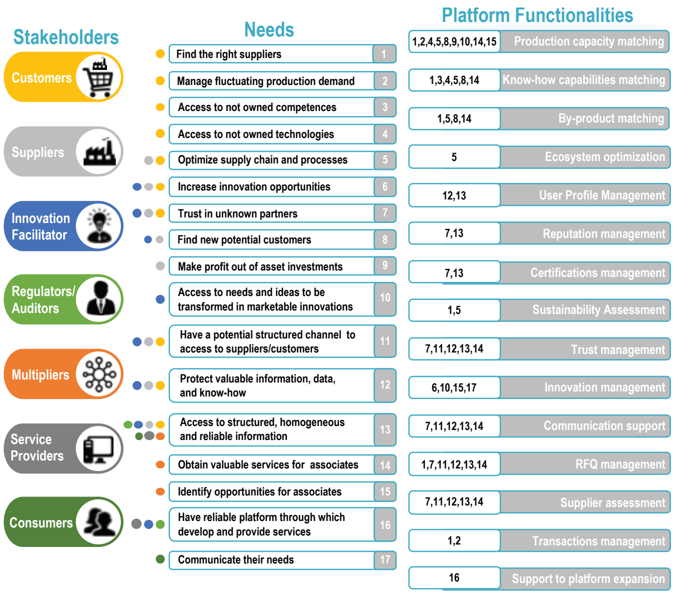

Fig. 1. The MANU-SQUARE ecosystem: icorrelation among Stakeholders, Needs and platform Functionalities

B. Knowledge representation in the manufacturing sector

The problem of developing a comprehensive data model for the manufacturing domain has already been addressed in the literature and the main contributions can be traced back to two main approaches: standard data models and reference frameworks for information representation based on ontologies. Data modelling is a complex process, consisting of defining and analysing of data requirements needed to support the business processes of an organisation. Data models formally define data objects and relationships between data objects [15]. Typical applications of data models are the development of databases and the enabling of the data exchange for a particular area of interest [16].

1) Standard knowledge representation

Entity Relationship Modelling – ERM: it is an abstract and conceptual representation of data. It deals with static structural properties such as entities, attributes and relationships. It clearly shows all types of concept abstractions, various relationships, mapping constraints and cardinalities. An entity may be defined as an Item which is recognised as being capable to exist independently and which can be uniquely identified [17]. Each entity has certain characteristics, known as attributes. A relationship is an association among entities and is represented by diamond-shaped symbols.

Unified Modelling Language – UML: Unified Modelling Language (UML) is a standardised general-purpose modelling language for object-oriented requirements analysis, modelling, specification and documentation of data and programs. Compared to other modelling languages the description of static aspects as well as dynamic aspects of a system is enabled by UML. UML defines nine different types of diagrams to support modelling [18]: use case diagrams for describing different application scenarios, component diagram and class diagram are used for the illustration of a structural and static system architecture, UML provides several types of diagrams for describing behaviour and dynamic aspects.

EXPRESS: it is a standardised modelling language for product data and is formalised in the ISO Standard for the Exchange of Product model STEP (ISO 10303) [19]. It is useful for displaying entity and type definitions, relationships and cardinality.

Integration DEFinition – IDEF: IDEF (Integration DEFinition) is a family of modelling languages in the field of software engineering. They cover a wide range of applications, from functional modelling to data modelling, simulation, object-oriented analysis and design as well as knowledge acquisition. The most-widely recognised and used IDEF methods are IDEF0, a functional modelling language, and IDEF1X, which addresses information models and database design issues [20].

Object Role Modelling – ORM: it is a fact-oriented method for creating a conceptual model. In Europe, the method is called Natural Language Information Analysis Method (NIAM), as well. The focus of ORM is on data modelling, although several ORM extensions have been intended for process or event modelling [21].

2) Ontology based knowledge representation

Ontologies represent a possible way to generate a more flexible data model integrating different knowledge domains and fragmented data models into a unique model without losing the notation and style of the individual ones [22]. Traditional data models describe the data schemas and do not formally describe the meaning and semantics of the data. When someone builds a system using a traditional data model approach, usually there is a Closed World Assumption (CWA). With the data model) implementation under CWA: (i) you can only enter data that you know to be valid/true; (ii) you are "encouraged" to enter complete information; (iii) there are no other data (what is not known to be true must be false); (iv) the data model implementation reveals no semantics about the data, but rather provide a data entry templates; (v) because of the lack of formalized semantics of concepts and their instances (data), the knowledge and data sharing, integration and reasoning are limited. Conversely, an ontology assumes the Open World Assumption (OWA). Hence, with an ontology and ontology-based knowledge bases: (i) you can enter what you know to be true; (ii) you can enter incomplete information/knowledge; (iii) you (and the ontology applications) can understand the meaning of data and the applications can infer additional information/knowledge based on the ontology entailment and reasoning/inference rules; (iv) ontology classes are simply sets of things, with precise semantics and do not act as a data entry templates in the system; (v) ontology boost knowledge or data sharing, integration and reasoning. Ontologies, as a formal and explicit description of concepts in a domain, are implemented using ontology languages. There are different ontology languages available today, with a higher or lower level of adoption from the ontology development community, with different levels of technology maturity and toolset support. The most recent and relevant applications of ontologies are in the field of the so-called Semantic Web [23].

a) Semantic Web Ontology Languages

Semantic Web technologies define and link data on the Web (or within an enterprise system) by providing the languages to express rich, self-describing interrelations of data in a machine-processable form. Thus, machines are not only able to process long strings of characters and index tons of data, but with the Semantic Web technologies they become able to store, manage and retrieve information based on the data meaning and logical relationships (semantics). So, the semantics adds another layer to the (Web) data and is able to show, interpret and infer related facts instead of just matching the string words. Semantic Web is based on a set of universal industrial and technology standards, as defined by the World Wide Web Consortium (W3C). The main standards that Semantic Web technology builds on are the following.

Resource Description Framework (RDF) is the format and a data model that a semantic technology uses to describe the resources and to store the resource descriptions on the Semantic Web or in a semantic graph database. RDF describes the resources in the format of triples: subject-predicate-object. In addition to the RDF language constructs, RDF Schema (RDFS) provides additional language constructs to define the classes and properties of the resources. These additional RDFS constructs allow for describing and embedding the semantics of user-defined vocabularies in RDF itself. Hence, by using RDFS, it is possible to structure RDF resources, by describing the classes (types) of the resources, properties of the resources, and by modelling the hierarchical subsumption of the classes and properties.

The Web Ontology Language (OWL) extends RDFS and allows for expressing further schema definitions in RDF. OWL is the computational logic-based language that is designed to capture the conceptual models and that may capture also a rich and complex knowledge about hierarchies of things (concepts, properties and individuals) and the relations between them. Most importantly, OWL overcomes some RDFS limitations in its ability to express rich semantic constructs. For example, RDFS does not allow expressions of property restrictions (value constraints and cardinality constraints) and it has very few constructs to make extensive inferences. There are three versions of OWL: OWL Lite, OWL DL, and OWL Full. The main differences are related to the different logics that can be expressed with the language. More complex logics (OWL Full) enable more comprehensive descriptions and automatic inferences, but introduce limitations from the computational point of view.

SPARQL is the semantic query language specifically designed to query RDF/RDFS data across various systems and databases, and to retrieve and process data stored in RDF format. Therefore, it is not an ontology language per se, but it can be used to exploit semantic representations in concrete applications, together with other languages (no standard yet) to express rules, such as Semantic Web Rule Language (SWRL).

b) Other Ontology Languages

Besides Semantic Web languages, there are other ontology languages. Some of them, such as DAML, OIL, DAML+OIL are predecessors of the OWL and were developed for the Semantic Web purpose. Other ontology languages, not strictly related to the Semantic Web initiative, include Knowledge Frame Language, CycL, FLogic, KIF, Ontolingua. They have been built to address specific purposes (e.g. CycL to create a rich foundational ontology), but no one of them is a standard and their diffusion is currently limited, compared Semantic Web technologies.

C. Ontology Representation Language Choices in manufacturing sector

This section reports a brief summary of an analysis of previous initiatives in a manufacturing/industrial domains ontology development. Most of the identified initiatives adopt OWL (in particular OWL-DL) as formalisation language. For example, the Design for Manufacturing (DFM) domain ontology focuses on capturing various manufacturing and assembly concepts and to share joining information along with the assembly geometry at the same time [24], as well as the Manufacturing ontology for Functionally Graded Materials [25]. More relevant to the present work, we can report further initiatives that aimed at formalising a core/upper ontology that could provide the primitive building blocks required for description of a wide spectrum of manufacturing services, such as MSDL-Manufacturing Service Description Language [26] or the MASON-Manufacturing's Semantics Ontology [27]. Moreover, Lin et al. [28] designed a Manufacturing System Engineering (MSE) ontology to provide a common understanding of manufacturing-related terms and to enhance the semantic and reuse of knowledge resources within global extended manufacturing teams, based on seven key classes: Project, Flow, Process, Enterprise, Extended_Enterprise, Resource and Strategy. Similarly, ADACOR (A Collaborative Production Automation and Control Architecture) could be classified as general-purpose manufacturing ontology [29]. Then, if we consider RDFS formalisations, we can report a recent work developed in the scope of the SatisFactory project [30] that developed an ontology to support automatic analysis and design of dynamically evolving shop floor operations.

III. A Data model to support ontological representation of a manufacturing ecosystem

A. Manu-square ecosystem data modelling approach

The UML language offers a convenient way to capture the classes (or concepts) of data in a domain and the graphical representation of the domain classes, accompanied with textual descriptions of the classes, provides an excellent basis for discussing, negotiating and finally formalizing the agreed meaning (semantics) of the domain classes. The conceptual models of MANU-SQUARE Ecosystem data will be first formalized as UML models (see paragraph C) their formal and machine-interpretable ontological representation (see paragraph D). The conceptual models of MANU-SQUARE Ecosystem data provided as UML models should not be mistakenly understood as any of traditional data models (e.g. E-R models, relational database schema). The semantic approach has been chosen as it offers the opportunity to escape rigid hierarchical formalisation of a complex domain such as that of industry. In fact, the MANU-SQUARE solution will benefit from ontologies in terms of expansibility to additional contexts when new sectors besides those of the project pilots are integrated or other applications being plugged on the platform infrastructure. Moreover, inference reasoning on the semantic representation of the ecosystem allows to make emerge non-trivial and previously unknown opportunities. This could regard not only the matching of demand and supply of manufacturing services but could point out possible and unpredictable relations: a particular kind of waste being produced at an ecosystem node could be linked to the requirements for a material needed in a new product being developed on the platform.

The data modelling approach aims therefore at: (1) designing the data-model containing the elementary propositions capable to generalise the key abstract concepts that act as backbone for the ecosystem model; (2) converting the knowledge beyond project consortium and stakeholders into ontologies; (3) adding the context-specific semantic models capable to support the specific functions of the service-providing applications running upon the platform infrastructure; (4) creating the inference rules working on the ontology and develop the reasoning system that allow to point out relationships between different aspects of the ecosystem; (5) designing and developing the technology infrastructure to populate (annotating), store and make use of (querying, inferring, matchmaking and recommending) the devised semantic representation of the MANU-SQUARE ecosystem.

B. Manu-square ecosystem data model representation

This section documents the data model developed to support the definition of a semantic infrastructure for the described platform. The General Model (see Figure 2) is composed by 5 sub-sections that correspond to the 5 main semantic areas of the data model.

other aspects have been designed, is the process that takes the cue from the IDEF0 [20] modelling standard. The choice of names has been kept as close as possible to the IDEF0 notation, in order to facilitate the communication between the various actors involved in the development of the platform. The Process class has input and output relationships with class Item which represents a specific level of a BOM. These relationships map the input and output defined by the standard. The "Control" part is instead modelled with the sub diagram represented by the main Allocation and CapacityAvailability classes. The last component of the standard is the "Resources".

<u>Stakeholder Data Model</u>: This part of the data model represents the stakeholders who are involved in using the platform. In addition to describing their specifications, the section related to the reputation mechanism has been also modelled, with the possibility of defining different assessment KPIs depending on the context in which the stakeholder operates.

<u>Innovation Ideas Data Model</u>: This part of the data model concerns the definition and collection of innovative ideas. The formalization of this part starts from the Idea class which is modelled to allow the association with multimedia contents and add some labels that can be useful during the research phase of the idea. Ideas are also linked with stakeholders or to whom conceived or interested in them by creating links. IdeaNetwork entity has been formalized in order to trace all types of links created along the life of the idea. For example, once an idea has been created, the link "ideation" will be added to the

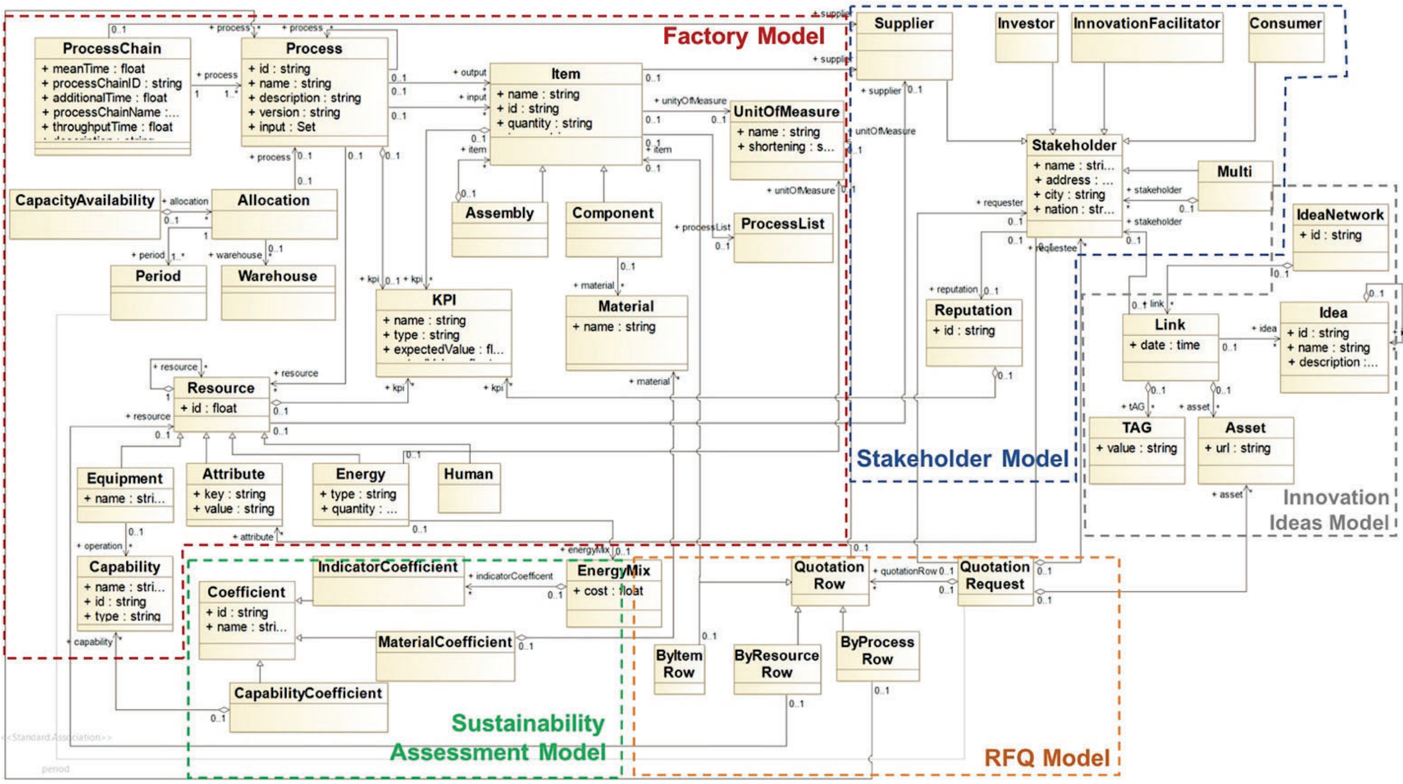

Fig.2 MANU-SQUARE Ecosystem Data Model Representation

<u>Factory Data Model</u>: it describes the most relevant aspects of a factory. The central aspect of this diagram, around which the

IdeaNetwork and as soon as it will be funded, the new link "funded" will be added.

RFQ Data Model: The MANU-SQUARE platform gives customers the possibility of searching suppliers meeting their needs in terms of technologies, products, input materials, etc. In order to manage the selection of the service the customer is interested in, the RFQ data model has been formalized in order to support the management of the whole "Request for Quotation" process. The model is structured in order to reflect a product/service quotation divided into rows. For each row, there is the option to request the cost to use a resource, to produce an Item or to perform a complete process. It is also possible to enrich the request by attaching additional information like CAD files, blueprints, descriptions etc. in order to provide to the supplier all the details needed for the quotation. The request for quotation is not limited only to a single supplier but can be addressed to many suppliers, allowing in this way the comparison between the different costs.

Sustainability Assessment Data Model: The result of an available capacity search could return a very long list of suppliers. So how to choose between the results? A criteria could be the evaluation of the sustainability impact which a Supplier's Resources, Processes or Items are associated to. This section of the Data Model is therefore dedicated to the integration of Sustainability Related analysis of the MANU-SQUARE ecosystem.

C. *Ontology Representation Language Requirements for MANU-SQUARE Core Ontology*

Before making a final step, which is the formalisation of the MANU-SQUARE core conceptual models with a specific ontology language, we came up with a precise set of requirements that drive our choice about the ontology language and about desired richness of the semantics in the MANU-SQUARE core ontology. Hereafter the identified requirements.

Standard language and interoperability: MANU-SQUARE will be dealing with an industry and one key element for industries to adopt a new technology is the use of standards. If the MANU-SQUARE ontology will follow a standard language representation, then it could be better accepted by the industry.

Reusability: As introduced above, other manufacturing and industrial ontologies are available, with manufacturing domain concepts (e.g. process, resource, items) and domain-independent concepts (e.g. quantities and unit of measures) applicable to MANU-SQUARE. Therefore, whenever is possible and system requirements clearly state the need, we will aim to reuse existing ontologies, trying to reduce as much as possible the (most likely necessary) adaptations/customisations. To be feasible to reuse existing ontologies or to align MANU-SQUARE ontologies with them, the MANU-SQUARE ontology has to be based on compatible ontology language.

Shareability and Extensibility: The project aims to develop a core set of conceptual models and a set of methodologies and tools to progressively extend and adapt such a core representation to multiple settings. In the project, we will experiment a few cases; but then third parties should be able to start from the project outputs and derive new representations. Therefore, the selected ontology language, as well as the associated tools and development methodologies, should be able to consistently describe and specify any element of a current or future system or the systems in its entirety including its concepts, relationships, processes, interactions and interoperation and must work across the different organisation levels. This must include new concepts or types of relationship.

Open source and costs: linked to the previous point, but more focused on the tools (software) that will manage the MANU-SQUARE ontology, the project aims to deal with the open source and free of costs licences. This can be seen from these perspectives: The project aim to use existing open source solutions to develop our modules and services; in this way, we can also benefit of existing communities around of open source projects. The project aim to deliver open source tools to facilitate the future exploitation of developed tools. This approach is also linked to the idea of creating a specific community around the envisioned MANU-SQUARE platform. In addition, we should consider that one of the key stakeholder groups of MANU-SQUARE is SMEs, which have limited resources and thus cannot invest a lot on (not core) new technologies. The use of open source can indeed address the cost barriers for them.

Scalability: MANU-SQUARE aims to become a B2B cloud platform facing with many customers across Europe. In this context, the scalability aspect is a key element to take into consideration. This affect all architectural elements of the platform and, thus, also the semantic layers and thus ontology language/framework decision. And there is trade-off between highly expressive languages and computational speed. The actual need for complex inferences and reasoning (and thus highly expressive languages) in the project scenarios and platform services should be carefully taken into consideration.

D. *MANU-SQUARE Core Ontology Implementation*

Finally, the UML MANU-SQUARE conceptual models introduced in the previous sections are translated into their formal and machine-interpretable ontological representation. As a representation language for the MANU-SQUARE Core ontology, we have chosen RDFS and OWL Lite language from the Semantic Web technology stack.

There are other works related to RDFS or OWL creation from UML. Tong and Zhang [31] presented an approach for constructing RDFS from UML and implemented an experimental prototype. Belghiat and Bourahla [32] presented an UML to OWL transformation using eXtensible Stylesheet Language Transformation (XSLT). Viademonte and Zhan [33] demonstrated an OWL creation from UML models of core business capabilities. Gasevic et.al [34] presented a Model Driven Architecture (MDA)-based approach for automatic generation of OWL from a UML using XSLT. Our UML to OWL transformation work is mostly similar to the solution of Gasevic et.al. Both approaches are based on the MDA and apply the same core set of UML to OWL transformation rules, however, in our approach, *to ease the implementation*, we used Eclipse Foundation Modelling Framework as it provides: a standard implementation of UML data-model; the data-model creation from a XMI; and Java API to manipulate with the model. On RDFS and OWL side, we used an Apache Jena Framework [35], which provides RDFS and OWL metamodels in Java. Gasevic et.al used an OMG's Ontology Definition Metamodel, their own Ontology UML Profile (OUP), and two-way XSLT mappings between OUP and ODM, ODM and OWL. Our transformation is, therefore, procedural and based on existing, tool-supported metamodels, while in Gasevic et.al's work, it is a declarative and based on their own Ontology Profile for UML. In contrast to all other works, we wanted to get both RDFS and OWL ontology from a UML model, with a single transformation and in a single file, but this is more development than research issue. Our transformation sufficiently supported the current requirements of MANU-SQUARE Core Ontology representation. We may expand the tool to address the future requirements. For example, if needed, we may add in a transformation of UML cardinality constraints to OWL axioms, inverse and functional properties creation, but this may require an introduction of custom UML tags in the model, as discussed in [34].

The reason for using the RDFS/OWL languages is somehow obvious, but it is strongly based on our analysis of existing ontologies (section II.C) and MANU-SQUARE specific requirements (section III.C). Clearly, RDFS and OWL are standard languages, supported by the W3C community. There are open source, free of cost, and mature tools for RDF, RDFS and OWL ontologies and data management. Existing industrial ontologies are based on RDFS and OWL too. Even the scalability may be adequately addressed in the future, as there are commercial tools that, comparing to some free of cost tools, claim to provide better performance and scalability for the reasoning and querying over the RDF-based knowledge graph database. Hence, the ontology is implemented in both RDFS and OWL languages and published in a RDF/XML and Turtle syntax. We have adopted this idea of creating the ontology by combining RDFS and OWL from a well-known FOAF ontology [36]. FOAF is captured in both RDFS and OWL. It seems to us that the approach taken by the FOAF is very pragmatic also for the MANU-SQUARE project, as the project use-cases and inference/reasoning requirements are not very clear yet to have an ultimate decision about the RDFS versus OWL Lite versus OWL DL expressivity. This "dual" ontology representation provides a nice flexibility to proceed with the project and technology implementation, and if needed in a later stage of the project, to gradually increase semantics expressiveness of the MANU-SQUARE Core and domain/tool-specific ontologies.We call this ontology as a MANU-SQUARE Core Ontology, as it provides a set of generic concepts that are shared across multiple manufacturing domains and applications dealing with the concepts of (1) manufacturing capability and capacity, (2) manufacturing innovation ideas, (3) manufacturing sustainability assessment, (4) request for quotation or ordering, and (5) ecosystem stakeholders' reputation.

An argument to call this ontology as a core ontology, is found in the literature. According to [37] the core ontologies in manufacturing sector cover concepts such as process, product, and resource. In addition to providing formal specifications of the semantics of these generic concepts, core ontologies are also designed to maximize shareability and reusability, and hence do not make any ontological commitments that are not shared by all related domain ontologies. That is exactly what the scope and purpose of MANU-SQUARE Core Ontology. It describes the core concepts and their relations of the MANU-SQUARE platform; it will be the conceptual "glue" of the several area-specific models (ontologies). The actual translation of UML representation of MANU-SQUARE conceptual models to the formalized RDFS/OWL ontology was done by applying the following translation rules:

TABLE II. TRANSLATION RULES APPLIED TO TRANSFORM UML IN RDFS/OWL

UML Class	<rdfs:class>, <owl:class>
UML Class inheritance	<rdfs:subClassOf>
UML Primitive Data Type Attribute	<rdf:property> and <owl:DatatypeProperty> with a proper <rdfs:domain> and <rdfs:range>. For example, if the UML attribute data type is String, the property range is a plain literal <rdf:Literal>, or if the UML attribute data type is decimal, the property range is a typed literal <xsd:decimal>
UML Relationship	<rdf:property> and <owl:ObjectProperty> with a proper <rdfs:domain> and <rdfs:range> expression. E.g. <rdfs:class> of an UML Class that owns the relationship is a property domain (<rdfs:domain>) and another UML class that is at the end of relationship is the propery range (<rdfs:range>)
UML Instance	rdf resourse with appropriate <rdf:type>

To be able to perform the MANU-SQUARE UML to Ontology transformation more accurate (less error-prone) and to be able to be easily repeat the transformation process in a case of UML models change, we have implemented an automated UML-to-Ontology transformation tool. The tool takes an XMI as its input and provides RDFS/OWL representation of the MANU-SQUARE Core Ontology. XMI stands for an XML Metadata Interchange (XMI) and it is a standard interchange format for UML models. The transformation tool is supported by the UML2 Java library of Eclipse Foundation Modelling Framework [38] that implements the data-model of the Unified Modelling Language (UMLTM) 2.x OMG.

IV. CONCLUSIONS AND NEXT STEPS

In this paper, a technical description of the Ecosystem Data Model considering the core abstract concepts behind the MANU-SQUARE platform has been provided. The data model formalization has been based on a well-defined approach starting from a standard representation of the MANU-SQUARE Ecosystem Data Model, by means of UML technique, thus translating it in ontology representation, in order to exploit the advantages of the latter technology.

The design of the proposed data model poses the basis for the instantiation of the data model in the MANU-SQUARE reference industrial sectors where the platform service-providing applications will be contextualized and let available to the upper layers of the platform architecture. At this level, the validation of the designed ontologies will become therefore fundamental to verify and demonstrate that the developed infrastructure is actually able not only to represent the ecosystem, but also to answer non-trivial questions identifying unexpected interdependencies among ecosystem resources. To this aim, two main activities are expected to represent milestones in the success of the core ontology as well as of the platform effectiveness: (i) validation and ontology refinement by means of application of competency questions; (ii) integration of inference rules. One of the ways to determine the scope of the ontology is to sketch a list of questions that a knowledge base based on the ontology should be able to answer, competency questions [39]. Competency Questions(CQs) play an important role in the ontology development lifecycle, as they represent the ontology requirements. These questions will serve as the validation test of the implemented ontology: Does the ontology contain enough information to answer these types of questions?

The next extended versions of the data model, will be subjected to a set of inference rules containing the actual propositions required for supporting the functioning of the service-providing applications. Inference reasoning on the semantic representation of the ecosystem allows to discover previously unknown opportunities in order to exploit the full potential residing in the involved industries. Service-providing applications, like the matching of demand and supply of manufacturing services, could exploit these unpredictable relations.

ACKNOWLEDGMENT

The work presented here was part of the project "MANU-SQUARE - MANUfacturing ecoSystem of QUAlified Resources Exchange" and received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreements No 761145.

REFERENCES

[1] [1] S. L. Vargo, P. P. Maglio, and M. A. Akaka, “On value and value co-creation: A service systems and service logic perspective,” *Eur. Manag. J.*, vol. 26, no. 3, pp. 145–152, Jun. 2008.

[2] [2] D. Dutra, V. C. de Oliveira, and J. R. Silva, “Manufacturing as Service: the challenge of Intelligent Manufacturing,” *IFAC Proc. Vol.*, vol. 46, no. 7, pp. 281–287, May 2013.

[3] [3] M. Barrett, E. Davidson, J. Prabhu, and S. L. Vargo, “Service Innovation in the Digital Age: Key Contributions and Future Directions,” *MIS Q.*, vol. 39, no. 1, pp. 135–154, Jan. 2015.

[4] [4] P. Muñoz and B. Cohen, “Mapping out the sharing economy: A configurational approach to sharing business modeling,” *Technol. Forecast. Soc. Change*, 2017.

[5] [5] S. Choudary, “Platform Thinking Blog - The New Rules of Business in a Networked World,” 2016. [Online]. Available: http://platformed.info/.

[6] [6] G. Landolfi, A. Barni, S. Menato, F. A. Cavadini, D. Rovere, and G. Dal Maso, “Design of a multi-sided platform supporting CPS deployment in the automation market,” in *2018 IEEE Industrial Cyber-Physical Systems (ICPS)*, 2018, pp. 684–689.

[7] [7] M. Luca, “Designing Online Marketplaces: Trust and Reputation Mechanisms,” Cambridge, MA, Sep. 2016.

[8] [8] P. C. Evans and A. Gawer, “The Rise of the Platform Enterprise- A Global Survey,” 2016.

[9] [9] S. Cicero, “That’s Cognitive Capitalism, baby.,” 2017. [Online]. Available: https://medium.com/@meedabyte/that-s-cognitive-capitalism-baby-ee82d1966c72.

[10] [10] J. Cenamor, D. Rönnberg Sjödin, and V. Parida, “Adopting a platform approach in servitization: Leveraging the value of digitalization,” *Int. J. Prod. Econ.*, vol. 192, pp. 54–65, Oct. 2017.

[11] [11] F. Vendrell-Herrero, O. F. Bustinza, G. Parry, and N. Georgantzis, “Servitization, digitization and supply chain interdependency,” *Ind. Mark. Manag.*, vol. 60, pp. 69–81, Jan. 2017.

[12] [12] S. Lenka, V. Parida, and J. Wincent, “Digitalization Capabilities as Enablers of Value Co-Creation in Servitizing Firms,” *Psychol. Mark.*, vol. 34, no. 1, pp. 92–100, Jan. 2017.

[13] [13] A. Bettoni, A. Barni, M. Sorlini, S. Menato, P. Giorgetti, and G. Landolfi, “Multi-Sided Digital Manufacturing Platform Supporting Exchange of Unused Company Potential,” in *2018 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC)*, 2018, pp. 1–9.

[14] [14] H. W. Chesbrough, *Open Innovation: The New Imperative for Creating and Profiting from Technology*. 2003.

[15] [15] U.S. Department of Transportation, “Data Integration Glossary,” 2001.

[16] [16] M. R. McCaleb, “A conceptual data model of datum systems,” *J. Res. Natl. Inst. Stand. Technol.*, vol. 104, no. 4, p. 349, Jul. 1999.

[17] [17] B. Thalheim, *Entity-Relationship Modeling*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000.

[18] [18] J. Bézivin and P.-A. Muller, Eds., *The Unified Modeling Language. «UML»’98: Beyond the Notation*, vol. 1618. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999.

[19] [19] International Standard Organisation, “Product data representation and exchange -- Part 11: Description methods: The EXPRESS language reference manual,” in *ISO 10303-11:2004: Industrial automation systems and integration*, 2004.

[20] [20] A.A.V.V., “IDEF - Family of Methods – A Structured Approach to Enterprise Modeling and Analysis,” 1999. [Online]. Available: http://www.idef.com/.

[21] [21] P. Bernus, K. Mertins, and G. Schmidt, Eds., *Handbook on Architectures of Information Systems*. Berlin/Heidelberg: Springer-Verlag, 1998.

[22] [22] M. Hepp, P. Leenheer, A. Moor, and Y. Sure, Eds., *Ontology Management*, vol. 7. Boston, MA: Springer US, 2008.

[23] [23] W3C, “Semantic Web,” 2015. [Online]. Available: https://www.w3.org/standards/semanticweb/. [Accessed: 08-Feb-2018].

[24] [24] X. Chang, “Ontology Development and Utilization in Product Design,” 2008.

[25] [25] F. Furini, R. Rai, B. Smith, G. Colombo, and V. Krovi, “Development of a Manufacturing Ontology for Functionally Graded Materials,” in *Volume 1B: 36th Computers and Information in Engineering Conference*, 2016, p. V01BT02A030.

[26] [26] F. Ameri and D. Dutta, “An Upper Ontology for Manufacturing Service Description,” in *Volume 3: 26th Computers and Information in Engineering Conference*, 2006, vol. 2006, pp. 651–661.

[27] [27] S. Lemaignan, A. Siadat, J.-Y. Dantan, and A. Semenenko, “MASON: A Proposal For An Ontology Of Manufacturing Domain,” in *IEEE Workshop on Distributed Intelligent Systems: Collective Intelligence and Its Applications (DIS’06)*, pp. 195–200.

[28] [28] H.-K. Lin, J. A. Harding *, and M. Shahbaz, “Manufacturing system engineering ontology for semantic interoperability across extended project teams,” *Int. J. Prod. Res.*, vol. 42, no. 24, pp. 5099–5118, Dec. 2004.

[29] [29] P. Leitão and F. Restivo, “ADACOR: A holonic architecture for agile and adaptive manufacturing control,” *Comput. Ind.*, vol. 57, no. 2, pp. 121–130, Feb. 2006.

[30] [30] “Semantically-enriched framework for analysis and design of dynamically evolving shop floor operations,” 2006.

[31] [31] Q. Tong, F. Zhang, and J. Cheng, “Construction of RDF(S) from UML Class Diagrams,” *J. Comput. Inf. Technol.*, vol. 22, no. 4, p. 237, 2015.

[32] [32] A. Belghiat and M. Bourahla, “Transformation of UML models towards OWL ontologies,” in *2012 6th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT)*, 2012, pp. 840–846.

[33] [33] S. Viademonte and C. Zhan, “Deriving OWL Ontologies from UML Models: an Enterprise Modelling Approach.,” 2010.

[34] [34] D. Gašević, D. Djurić, and V. Devedžić, “MDA-based Automatic OWL Ontology Development,” *Int. J. Softw. Tools Technol. Transf.*, vol. 9, no. 2, pp. 103–117, Mar. 2007.

[35] [35] “Apache Jena.” [Online]. Available: https://jena.apache.org/. [Accessed: 08-Feb-2018].

[36] [36] “FOAF – Friend of a Friend Ontology,” *Paddington Edition*, 2014. [Online]. Available: http://xmlns.com/foaf/spec/. [Accessed: 08-Feb-2018].

[37] [37] L. Deshayes, S. Foufou, and M. Gruninger, “An Ontology Architecture for Standards Integration and Conformance in Manufacturing,” in *Advances in Integrated Design and Manufacturing in Mechanical Engineering II*, Dordrecht: Springer Netherlands, 2007, pp. 261–276.

[38] [38] Eclipse Foundation, "Emf Uml2," *Eclipse Foundation*, 2013. [Online]. Available: http://www.eclipse.org/modeling/mdt/?project=uml2. [Accessed: 08-Feb-2018].

[39] [39] M. Gruninger; and M. S. Fox, "Methodology for the Design and Evaluation of Ontologies," 1995