

An approach to support Industry 4.0 adoption in SMEs using a core-metamodel

Michele Dassisti, Antonio Giovannini, Pasquale Merla, Michela Chimienti, Hervé Panetto

▶ **To cite this version:**

Michele Dassisti, Antonio Giovannini, Pasquale Merla, Michela Chimienti, Hervé Panetto. An approach to support Industry 4.0 adoption in SMEs using a core-metamodel. Annual Reviews in Control, 2019, 47, pp.266-274. 10.1016/j.arcontrol.2018.11.001 . hal-01920242

HAL Id: hal-01920242
https://hal.science/hal-01920242

Submitted on 13 Nov 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

An approach to support Industry 4.0 adoption in SMEs: a core-metamodel and applications

Michele Dassisti*, Antonio Giovannini **
Pasquale Merla *, Michela Chimienti ** and Hervé Panetto *****

*Politecnico di Bari, DMMM, VialeJapigia 182, 70126 BARI (Tel: 080-596.2747; e-mail: michele.dassisti@poliba.it).
**INRESLAB Scarl (e-mail: a.giovannini@inreslab.org,m.chimienti@inreslab.org)
***Ali6 Srl (e-mail: merla@ali6.it)
***** University of Lorraine, CNRS, CRAN, France (e-mail: Herve.Panetto@univ-lorraine.fr)

Abstract: **Despite the recent growing interest in the "factory smartness", still there are only few small and medium enterprises (SMEs) that adopt effective Industry 4.0 (I4.0) solutions. The main reasons can be related to the lack of formalized processes, lack of ICT knowledgeas well as low-cost commercial systems.To cope with these issues, this work focuses on the development and the application of an approach to provide SMEs with a multi-purpose, modular, knowledge-based system: the main aim is to provide a modular and extensible system that can be incrementally implemented without requiring huge initial investments. This system is based on a core design-knowledge meta-model. From this core meta-model, multi-purposes *modules* can be built: in this paper, we present modules for the traceability support, the AR-powered assembly support, themachine-to-machine control and the data analysis support.**

*Keywords:***knowledge-based systems, Industry 4.0,SME,multi-purpose systems, modular systems.**

1. INTRODUCTION

The recent growing interest in the "factory smartness" related to the availability of high-performing low-cost information and communication technologies gives the chance to rethink the old concept of manufacturing operations. The core of the Industry 4.0 (I4.0) is an announced revolution (Kagermann et al., 2013) with the potentialities of predicting the manufacturing outcomes by the profound knowledge of the processes available. The potentialities of I4.0 lie then in a better flexibility and scalability of manufacturing systems through information technologies and industrial automation (Brettel et al., 2014; Dassisti and De Nicolò, 2012). The "support wave" of a number of governments all around the world for I4.0 solutions is forcing this paradigm change, with a debatable success (e.g. Industrie 4.0, Manufacturing USA, Industrie du Futur, Industrial Internet of Things, Made in China 2025, Fabbrica intelligente-Industria 4.0)(Issa et al., 2017). The funding action is in fact unnaturally pushing the change, thus hurrying adoption of implementation solutions not adequately structured or even needed. The proof of this is that still few companies, particularly Small and Medium Enterprises (SMEs), are keen to adopt truly I4.0 solutions (Bosch, 2015; European Commission, 2015; Federmeccanica, 2016; Ubisense, 2014). If SMEs cannot align to I4.0 solutions, this can seriously affect the economic growth of a country (Kagermann et al., 2013; TELUS and IDC, 2014; Ubisense, 2014). SMEs tend to overestimate the complexity of I4.0 solutions and to underestimate their benefitsmainly because they believe that the I4.0 is a fad not suitable to them. In fact SMEs are often characterized by poorly formalised processes, by independent and/or legacy hardware and software systems and by smaller economical resources (against to the large companies). Furthermore, they typically lack internal IT competences and the related technological knowledge. Starting from this scenario, this work answers to the following question: *how to facilitate the adoption of I4.0 solutions by SMEs?*

This paper is an extension of the work proposed in (Dassisti et al., 2018). In this paper we provide details about the four industrial case implementations, about how the usage of a core meta-model can help to face the requirements described in detail in the Section 2 in favouring the adoption of I4.0 for SMEs. In section 3 we present in detail the core-metamodel. In section 4 we provide sound description of the core-metamodel implementation strategies. Finally, in Section 5 we provide some examples of implementation of the core-metamodel to industrial settings.

2. THE NEED OF A CORE-METAMODEL

To address the question put in the Introduction, several steps are required: an amazing simplification of the I4.0 ecosystem creation to extend their use to different knowledge domains (Bosch, 2015); a gradual introduction of the new technologies without asking to the SMEs to have large investments for replacing the existing systems (Federmeccanica, 2016); to make the use of the I4.0 solutions user-friendly (Bosch, 2015); to provide a simple and total interoperability with the existing legacy systems. The so-called Cyber-Physical Systems (CPS) refer to the combination and coordination between the physical assets and their computational capabilities. These hardware and software components are deeply intertwined and interacting with each other in diverse

ways that may change with context. Still these features do not provide the satisfaction of the following two main requirements for satisfying a SME-oriented I4.0 solution (SMEI4.0):

- *knowledge-based*: a SMEI4.0 should minimize the need of expert work to integrate, with a formalised knowledge, different returns of experience derived from not formalised process in order to favour its use from different interfaces;
- *extensible*: SMEI4.0 solutions must be flexible to allow a gradual implementation. They must ensure the possibility to reutilize the entire systems if other components are to be added.

The last requirements can be met only with a core-data model to be reused in all I4.0 application: a kind of meta-model for the knowledge representation of the "I4.0-enterprise". A similar need has already be expressed in theMIT*Solid* project (Mansour et al., 2016): it split the application and the used data. In our industrial scenario, this means having one core-meta-model and different applications gravitating around it with different aims.In other words, each application has a commondatabase.For a clear description of complex concepts, it is already clear that adequate reference models are required. Amongst all, cited in(Thoben et al., 2017), we mention the need of further research on "advanced knowledge representation mechanisms, supporting stakeholder communications and strategic standardization across manufacturing".

Different studies tried to address these issues. In (Pfrommer, 2015) a unified abstraction to describe the skills of production resources and the requirements of the product-specific manufacturing steps is proposed. Indeed, manufacturing skill modelling has been a wide field of research, being the skills the ability of a resource to implement a production process. In the same stream of *plug&produce* is the contribution from (Engel,2016) by semantically describing systems' characteristics of a manufacturing module to use as a criterion for orchestration. In this approach, the implicit recognition of a need is evident for a shared knowledge representation. In (Lee,2015) a unified system framework has been designed for general applications where the stress is on the inference of meaningful information from the data. In (Lavdim,2016) it is clearly stated the need for techniques and standards for representing and exchanging information, data and knowledge. The future cooperative scenarios, made also of frequent mutations and adaption, is tacked here using an administrative shell, based on semantic knowledge representation formalisms. Later, more complex approaches were developed. For instance, (Harrison et al., 2016) proposed the "vueOne" development environment for implementing a cooperative SME oriented platform. The system is based on a common knowledge model that supports all the activities along the product life cycle.In (Huang et al., 2013)the idea of manufacturing resource and capability sharing based on cloud computing for SME is faced introducing a SME-oriented cloud manufacturing service platform. This cloud-manufacturing platform is a momentous tool to exploit and share the manufacturing resource, promote design of products, operation management, and manufacture ability, and enhance the comprehensive abilities for the SMEs. The model behind this approach is by layering the architecture and clustering resources within each layer by interfacing them using transaction agents. Still it is a prototype platform that has to prove the flexibility to self-adapting to varying conditions, provided no knowledge sharing model has been adopted. The centre of the cloud-manufacturing model is always knowledge (Tao et al., 2011) with its layering approach that supports each manufacturing steps, always relying on an appropriate platform: from design up to maintenance and integration. Finally, in (Schlegel et al., 2017) the new approach «Linked Factory» developed by Fraunhofer Institute for Machine Tools and Forming Technology is discussed. This approach relies on a common hub sharing knowledge and data. The problem highlighted is, however, that collected data is mostly used and analysed only accordingly to the original reason of acquisition. A crucial aspect, the common available production-related data often contains hidden interdependencies that are relevant to manufacturing. This interesting point again stresses the missing of an adequate and general modelling of the manufacturing scenario.

Despite the number of technical and scientific works facing the adoption of I4.0 solutions by the SMEs, none of these works clearly states a final solution of how the above mentioned requirements can be practically addressed. In the next sections, we propose the extension of a knowledge model based on the concept of *measurement*. The aim is to put it as a core model for the I4.0 scenario for SMEs.

3. A COREMETA-MODEL FOR I4.0 APPLICATIONS IN SMES

Since the I4.0 principles are about automations and M2M (machine-to-machine) interaction, the knowledge formalisation framework should be able to represent the sensors capabilities and to make action on machine's actuators. So, a core meta-model is necessary for formalising the knowledge of experts and minimizing the integration issues. This matter has been faced in (Giovannini et al., 2015): here the authors propose a conceptual framework to formalise the knowledge for minimising the ambiguities.

3.1. Concepts of the knowledge formalisation based on measurements

The conceptual model showing the basic concepts of the framework in (Giovannini et al., 2015) is presented in **Erreur ! Source du renvoi introuvable.**(Highlighted section~~blue classes and black relationships~~). This framework was intended to formalise design knowledge for minimising the ambiguities. In the same paper, the authors use the concepts of *measurement* to limit the ambiguities (i.e. a content is subject to more than one interpretation) when several modellers instantiate the knowledge-base. In other words, the disambiguation is based on the measurement system, i.e. the unit of measure, the range of measured values,

the error, the time and place of measurement. The semantics of the main concepts in the proposed framework is the following.

A *measurement* is the characterisation of the act of perceiving. This concept can identify a sensor with its time and place of measurement. For instance, the temperature perceived in a sensor positioned in a volume of air at a time *t*.

A vector with *space*, *time* and *shape (S, T, S)* characterises each measurement. For instance, the temperature can be the *shape* of the *measurement* (i.e. *what* to measure), the volume of air can be the *space* of the *measurement* (i.e. *where* to measure) and the *time* represents *when* the temperature is measured. Each one of these three elements is defined as *property*.

Each *property* is involved in one or more *transformations*: a *transformation* is a mathematical relationship between a set of *properties*. For instance, the relationship between the temperature and the time in which it is measured is the *transformation T=f(t)*, where *t*is the time of the *measurement*, *T* is the temperature and *f* is the mathematical relationship that links the two *properties*. Each *property* has to be detailed by a *range of values*, a *UOM* (unit of measure) and a *tolerance* of the *measurement*. For instance, the *shape* of the *measurement* can be measured in Celsius degrees (*UOM*) between 25°C and 50°C (observed*range of values*) with a *tolerance* of ±0.1 (i.e. the error of the measurement system).

A set of mathematical relationships between the properties of a set of *measurements* is defined *experience*. An *experience* is a kind of report of experiment: the constant parameters (*properties* that are constant during all the observations) and the observed variables are all formalised as mathematical relationships between *measurements*, i.e. as *transformations*. As an example, let us consider an experiment to test the validity of the ideal gas law *PV=nRT*(an instance of *transformation*). The instance of the *experience* has to capture the mathematical relationships between all the tuples (S, T, S) of the *measurements* (e.g. the pressure, the temperature, the volume of the gas considered) that describe the environmental conditions where the law holds. In other words, the *experience* instance describes a system behaviour and in which conditions this behaviourshould be observed.

3.2. Extending the meta-model for the human-machine interface

The conceptual model described in the previous section was intended to formalise knowledge for an automatic or semi-automatic reuse. This implies the absence of human readable descriptions of the formalised instances. In this paper we extend this conceptual model to allow an easier user knowledge formalisation. The ambiguity of the instances is not expected to increase, i.e. each one of the contents should be disambiguated on the basis of the measurement system. In other words, we need to:

- Introduce new concepts to allow a non-expert user interaction anda multi-perspective descriptions of thebasic concepts about the systems behaviours (e.g. products, manufacturing processes) described in (Giovannini et al., 2015).
- Introduce concepts to improve the accessibility of the knowledge base content at multiple levels of detail; especially, a non-expert user should be able to reach the desired concept by filtering the results not only on the basis of the *numerical* concepts, as in (Giovannini et al., 2015).In **Erreur ! Source du renvoi introuvable.**, the conceptual model extensions are shown~~: the blue relationships are modified ones; the green artefacts are added ones~~.

3.3. The abstraction of the concepts of UOM and space

To improve the accessibility of the content, we studied different ways to group the basic concepts of *measurement* and so on. Since the *measurement* is a composition of *properties*, the research in the knowledge base should be based on:

- *properties* (S,T,S):
  - UOM;
  - UOM and tolerance;
  - UOM range of values;
  - UOM, tolerance and range of values.
- *experiences*.

The *time* (i.e. when the measurement is captured) has not been considered as basis for a content research. In fact, the time of the measurement is composed of two terms: the start and the duration. These ones are numbers in a scale that is unique for the experience. Therefore, those values make no-sense if not considered in the context of an experience.

All the other concepts in the previous list were experience-dependent in model in (Giovannini et al. 2015): *UOM*, *range of values* and *tolerance* existence depends on the *property*; the *property* existence depends on the *measurement*; the existence of the latter depends on the *experience*. The first adaptation of the model in (Giovannini et al. 2015) dealt with the "abstraction" of concepts originally part of a UML composition or a 1-to-1 relationship: the UOM and the *space* has been identified as important *objects* that can ease a content research. Let us consider two system behaviours: the first one describing the light diffusion in a room; the second one describing the heat exchange in a room. The concept of "window" is in both cases involved in the systems. Without the abstraction of the concept of *space* there is no chance of finding the two experiences by querying the word "window".

Therefore, the relationships that linked UOM with *property* (1-to-1) and *space* with *measurement* (composition) to the other classes have been changed (blue relationships in **Erreur ! Source du renvoi introuvable.**) into1-to-many relationships.

Other concepts like *time*, *range of values* and *tolerance* have not been abstracted because they are too related to *experience* and *UOM*.

3.1. The definition of two concepts: coefficient and assembly

As discussed above, the system behaviour is described by means of mathematical relationships between properties.The abstraction of the concepts of *space* and UOM has made possible a research for the measurements. Instead, it is still not possible to discriminate the experiences by the described behaviour. To do so, we defined the concept of *coefficient*.

In the mathematical models characterising an experience, there are parameters related to the *measurements* (i.e. direct measurements) and other parameters (e.g. heat exchange coefficient) that are related to indirect measurement and therefore not represented in(Giovannini et al., 2015). The *coefficient* concepts are here used to describe how the measurements relate in an experience. Therefore, the coefficients are intended to express the type of behaviour we want to characterise in the knowledge base.

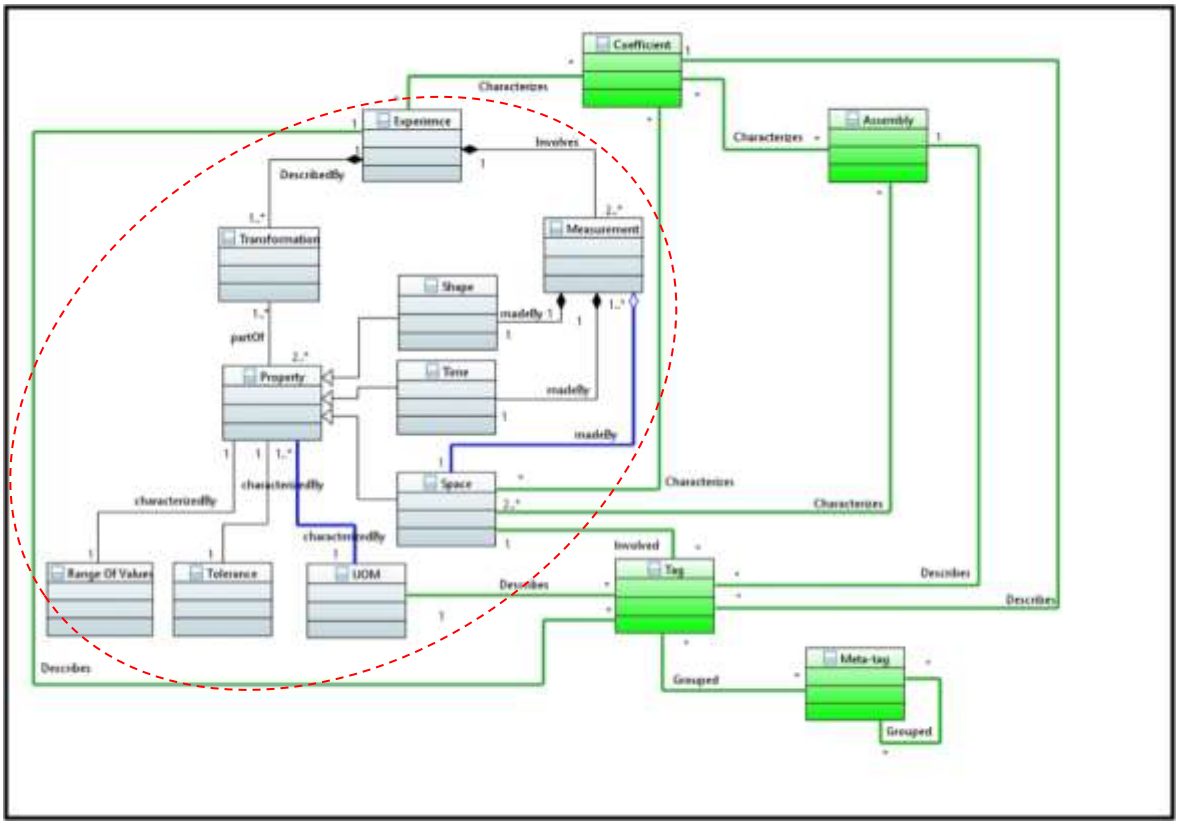


Fig. 1– The extension of the meta-model in (Giovannini et al., 2015)

Since coefficients describe relationships between measurements, they are linked with the *experience* class in **Erreur ! Source du renvoi introuvable.**. The coefficient class is also related to the *space* concept because the coefficient can relate measurements with the space. However, when a coefficient relates measurements with different spaces, a new concept is required because, in the original model, there were no space aggregations.

To associate a coefficient describing the interaction of two measurements related with two different spaces, we defined the concept of *assembly*. In **Erreur ! Source du renvoi introuvable.**, the *assembly* class is related with the *space* and the *coefficient* classes.

3.2. The definition of tags and meta-tags

In the previous sections, we introduced concepts to group elements and to increase the expressiveness and reachability of concepts like *UOM* and *space*. However, the research of the content is still based on numbers and units of measure. Originally, the conceptual model was intended to formalise a machine-readable knowledge. To make the interface more user-friendly, descriptions of the models and spaces – and generally of the numerical values– should be provided. These descriptions must not introduce ambiguity in the model.

To cope with this point, we should provide a 1-to-N relationship between the contents and the related description. This cardinality allows us to express the points of view of more users without generating ambiguity. In fact, a description is related only to one content, so the disambiguation is transposed to the meta-model in (Giovannini et al., 2015): i.e. a description-based research results in a unique artefact of the meta-model in (Giovannini et al., 2015).In Fig. 1, the descriptions of the contents are defined *tags*. The *tag* concept is a one-word description associated with all the researchable

concepts of the model, i.e. *UOM*, *space*, *coefficient*, *assembly* and *experience*.

When a user performs a tag-based research, the results are directly connected with the original concepts that are intended to disambiguate the knowledge on the basis of the measurement system. Consequently, the *tag* introduction does not alter the ambiguity of the formalised models.

To speed up a research based on descriptions and to associate related descriptions, a concept to aggregate the *tags* is required: in Fig. 1 the *metatag* concept has been added. A *metatag* can be related to more *tags* or *metatags.* These relationships allow to build several levels of aggregations.

4. THE METAMODEL IMPLEMENTATION

4.1. How the meta-model faces the requirements

So far, we introduced concepts around the semantic of *measurement*. The aim still remains to meet the requirements defined in section 2, i.e. *knowledge-based* and *extensible*.

As concerns the *knowledge-based* requirement, we can link different spaces and assemblies to the same experience, as well as it is possible to refer the same spaces to different experiences. Different users can formalise knowledge about different knowledge domain, which refers to the same module. As an instance, a technician performing the maintenance of the plant may link an experience that refers to its maintenance to the knowledge related the plant. Another technician managing the production process, may formalise an experience about the production process that refers to the formalization of the same plant.

The extensibility is thus ensured, provided that the formalisation here adopted for the knowledge uses the same meta-model, thus reaching the unambiguity. In this way all M2M or HM interfaces can use all formalised knowledge.

Since it is necessary that even an inexpert user can formalise and query the knowledge according to the proposed meta-model (to ensure the unambiguity and in this way to satisfy both the requirements), it is necessary to create a DB according to the meta-model and an interface that allows to easily upload the knowledge into this DB.

4.2. MySQL database

The first step here performed was to implement the proposed meta-modelwith a MySQL database (Fig. 2). The DB is accessed from the web. The aim was to allow the concurrent DB instantiation.

The *spaces* are mainly represented with CAD software. When CAD models are not available, sketches or even images can be uploaded. The aim was to represent where the measurements are performed, i.e. where the sensors/actuators are installed. To make it easier the introduction of related spaces, it is appropriate to upload a CAD assembly model. An interface has been developed to read the assembly model and build the related *assembly* and *space* instances in the DB.

The *transformations* that represent the mathematical relationships between the measurements' *properties* and/or coefficients are formalised in files with scripts or MINLP models to be launched with a solver.

All I4.0 applications will use the same DB. In this way different applications in different domain can refer to the same experience.

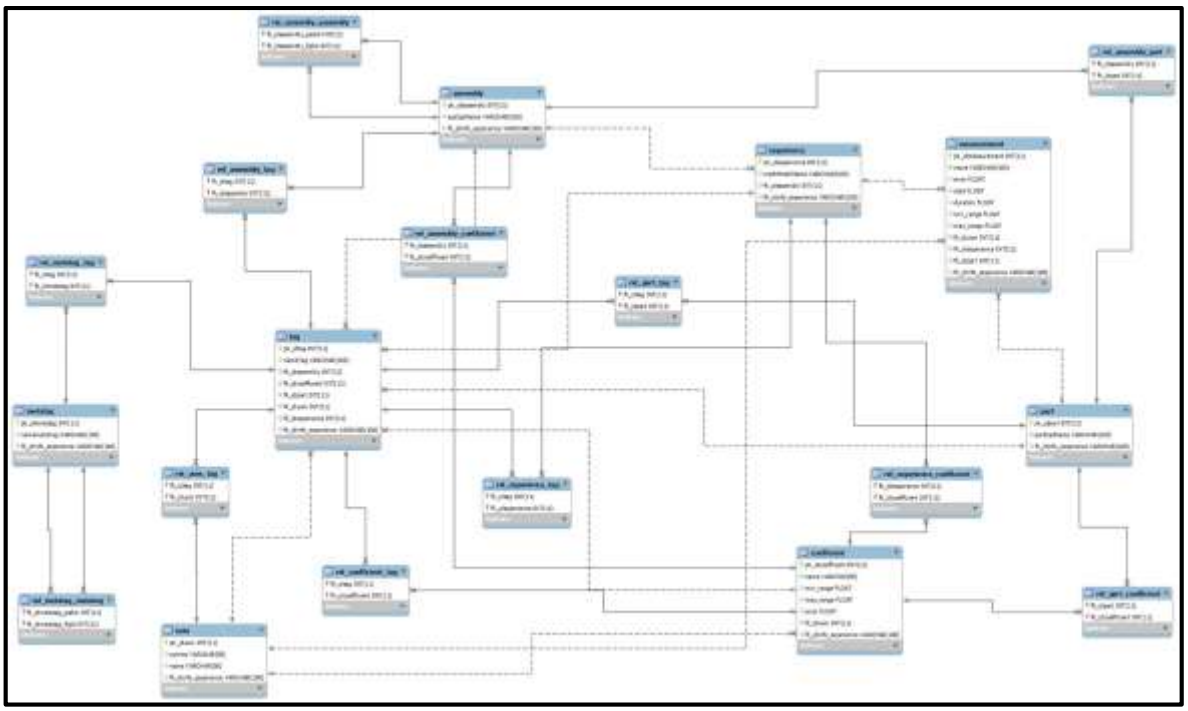


Fig. 2 – The meta-model implementation in MySQL DB

4.3. Web interface

The previous sections presented the meta-model for the knowledge formalization and-its implementation in the resulting DB. This section describes-the web interface created to upload the knowledge into the DB.

The idea is to have an interface that supports the understanding of implemented knowledge. We decided to represent the different instances of the DB (i.e. *spaces*, *assemblies*, *measurements*, *coefficients*, etc.) and their relations as a network in which each instance is represented as a node with different colour (i.e. blue for experiences, yellow for parts, etc.). In this way all users can simply identify the inserted instances and the relations between them.

The first step in the formalization of an *experience* is the loading of the *space(s)* to which the *transformations* of the *experience* refer. To create such a *space(s)* the user should load an XML file that contains the structure of a CAD model of an assembly. The structure of the XML file is as follows: 1) a parent node corresponding to the *assembly* 2) *n* child nodes corresponding to the *parts* that compose it.

After the XML insertion, the user should to insert: 1) the CAD files corresponding to the structure of the *assembly* in the loaded XML; 2) a report that contains the list of *coefficients* and *measurements* involved in the *experience*;3) a mathematical model which contains the *transformations* between the *measurements* and the *coefficients* just loaded. The modeller will have to define, for each *coefficient* and *measurement* (in report in step 2), the parameters that characterize it (i.e. *range of values* and *tolerances*). The user will have also to establish the relationships that exist between the instances by connecting, for example, the *measurements* to a *part* and to a *UOM*. Creating a relationship should fulfil all required modelling constraints (i.e. the requirement that a *measurement* must have a *shape* characterized from a *UOM*). Only after completing the previous steps the modeller can insert *tags* and *meta-tags*. He can create new tags and meta-tags or use existing ones. Fig. 3 shows the example of inserting an *experience e* where there are: 1 *assembly* (in red) and 9 *parts* connected (in yellow). There are 2 *measurements* (in purple) and 3 *coefficients* (in pink) linked to a *UOM* (in light blue). 3 *tags* (in light green) and 2 *meta-tags* (in dark green) have been added (1 new and 1 existing).

The modeller can confirm the insertion of the new *experience*, that will automatically link to the existing ones that share *tag*, *meta-tag*, *UOM*, *assemblies*, etc. All inserted *experiences* will be part of the DB for the knowledge formalisation, even if they refer to different knowledge domain, since they are formalized according to the same meta-model.Fig. 4shows the representation in web interfaces of all inserted *experiences* (blue nodes). You can see how different experiences share *tags, meta-tags, UOM, measurements, coefficients, assemblies* etc.



Fig. 3 – A screenshot of the web interface: *experience* insertion

Some of the *experiences* showed in Fig. 4 refers to the applications described in the following section.

Using this web interface allows to satisfy both the requirements before mentioned (i.e. *knowledge-based* and *extensibility*), because all formalized knowledge uses the meta-model that satisfies these requirements.

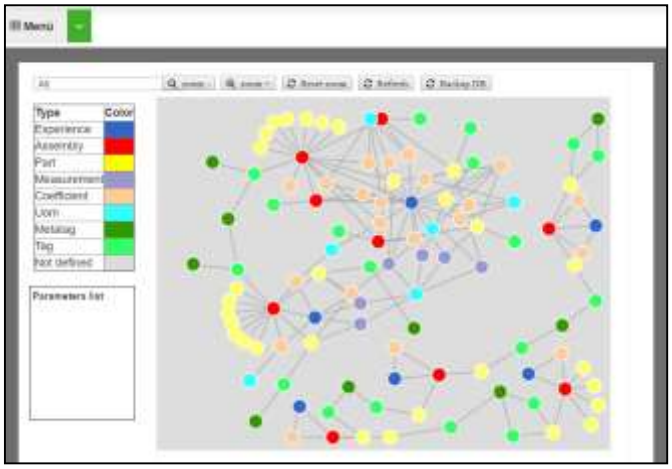


Fig. 4 – The web interface: representation of all the *experiences* in the DB.

5. CASE STUDY: 4 INTERFACES USING THE SAME METAMODEL

In this section, we describe 4 applications that were developed based on the implemented core meta-model. Each application uses the formalised knowledge to query the model and to fetch the required data to pursue its goal. The DB structure and instantiation mechanisms remain the same. An update on the data in the DB has effects on all the linked applications. The four examples here described are only a limited example of the possible applications sharing the same enterprise knowledge. The abstraction of the concepts in the meta-model (e.g. *measurement*) as well as their connection to the physic of the measurement systems allows the design of several other applications using the knowledge, structured in the same way.

5.1. Smart manufacturing traceability app

The aim of this application is to monitor the coherence of the materials traceability information in a shop floor. To maintain this information the application needs updated data about the manufacturing processes and the process parameters related to the ordered products.

In this app, the knowledge formalized in the DB refers to a manufacturing process of a steel pipe. The *experiences* correspond to different steps in manufacturing process (i.e. cutting, painting, welding). The knowledge formalization must be done by a user with skills in the application domain. In this case, an *assembly* is linked to each *experience* that corresponds to the production resource performing the production step. Other *assemblies* or *parts* that are linked to this *assembly*, corresponding to inputs and outputs of the production step (e.g. raw materials, products, half-processed products), as well as other tools (external to production resource) that are involved in the production step (e.g. detergent for cleaning). The *measurements* and the *coefficients* correspond to the parameters involved in the same production step, and the *transformations* refer how the measurements and coefficients are related.

For example, Fig. 5shows two *experiences* that refers to two different production steps. The first (a) refers to the painting of a metal sheet. The *assembly* corresponds to the production resource that perform the painting. The four *parts* linked to the *assembly* corresponds to possible input and output of this production step. The second experience (b) refers to the cleaning process. Even in this example, the *assembly* represents the production resource that performs this production step.Compared to the paintingcase, in addition to four parts that corresponding to input and output, there is a *part* that refers to detergent used in cleaning. In both the examples, to each part is linked at least a *tag* or *meta-tag*, that represent a user-friendly description (name) that allows to technician to univocally refer to *part* (input and output) during production step.

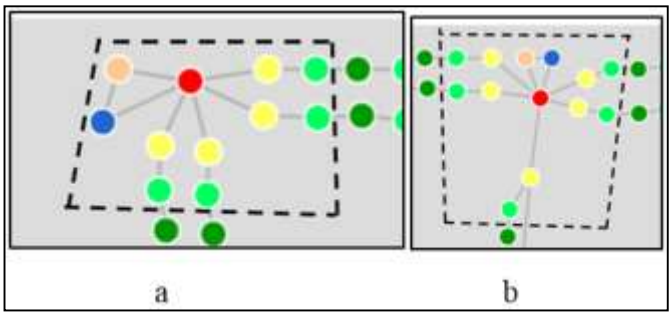


Fig. 5 – Representation of the *experience* corresponding to painting process (a), and to cleaning process (b).

This application uses the knowledge about the products described in the DB. When a product order is introduced, the name of the product is queried in the DB. The product (i.e. the *space* in an *experience*) is linked to a production step (i.e. an *experience* which have CAD files tagged as outputs that corresponds to the ordered product) and the related process parameters, that the traceability app requires to trace. Starting from the *parts* or *assemblies* in input for this production step, the app queries to the DB the production steps where the same *parts* or *assemblies* are the output. The app proceeds backwards to identify along the production process until all the inputs are raw materials and then stops.

In this app, for each product to perform, one can go back to the related manufacturing process, the input and the output of each single production step and what is necessary to trace to ensure the traceability of each component of each product. When the order of a new product is issued, the app synchronises the internal information with the DB and suggests what to trace.

The use this application does not require an expert user of traceability best practices, because by means of the formalized knowledge, is possible to support the users, suggesting what to trace.

5.2. CPS for the optimisation of M2M interaction

The aim of this app is to optimise the scheduling in real-time to exploit energy saving opportunities. We developed a prototype that implements the hybrid architecture in (Dassisti et al., 2017) for the M2M interaction. In the proposed architecture a central unit configures groups of production resources (PRs) based on each production order entry. This O 1) gathers periodically the states of the PRs in the groups 2) compute the optimal process parameters for all the PRs in the group and 3) provide to the PRs the results of the optimization. Details are in (Dassisti et al., 2017).

Fig. 6 shows the prototype created for this application. In this prototype we propose the simulation of three PRs (each PR is represented to one motor linked to one driver). The proposed hardware interacts with PLCs that simulates the industrial controller of each PR. The control unit of each PR corresponds to the three electronic devices in the lower part of Fig. 6. These control units are connected to central unit.

The simulated PRs correspond to the three machines (grinding, winding, and tempering) that needs to produce a steel spring in our case. As in the traceability app, the knowledge formalized in the DB refers to steps in manufacturing process linked to PRs in production plant. In this case more different *transformations* (alternative production cycles) are linked to the same *experience* referring to different way of performing the same production step. T*ransformations* differ because of different execution time and consumption of the resources.

At each order entry, the central unit, according to products to be processed and to the state of PR in the plants, creates the OGs and for each OG it assigns to a PR the role of the O. To this point the central unit 1) sends to all PR involved in the production their own agenda 2) synchronizes the manufacturing process knowledge with the DB to update the alternative production cycle applied by the PRs, and 3) sends to the O of each OG all the possible alternative production cycle for each PR of their own OG. During performing the O chooses for the PRs of its own OG the best alternative production cycle to save energy, based on available time.

Even in this app, the user doesn't need to know details about formalised knowledge (the alternatives of the production steps in manufacturing processes). The system will choose the best solution for him.

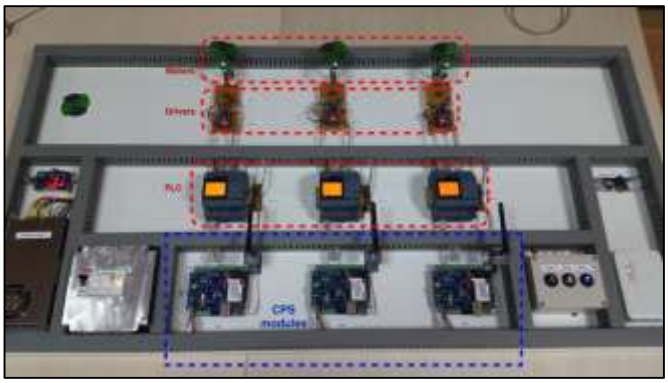


Fig. 6- The CPS prototype: three brushed motors are controlled by the PLCs; each PLC is connected with the proposed hardware to make them communicate; details in (Dassisti et al., 2017).

5.3. Augmented realty assembly support app

The aim of this app is to support the assembly stage of small assemblies in a scenario of high product variety. The app generates automatically an approximated assembly sequence by improving the algorithms in (Dini and Santochi, 1992). At each assembly request of a new product, the app synchronises with the DB to download the CAD data of the product. The generated sequences are sent to the cloud and read by an AR holographic device that reproduces the sequence on demand. For example, Fig. 7shows the third step of the assembly sequence of a flanged pipe.

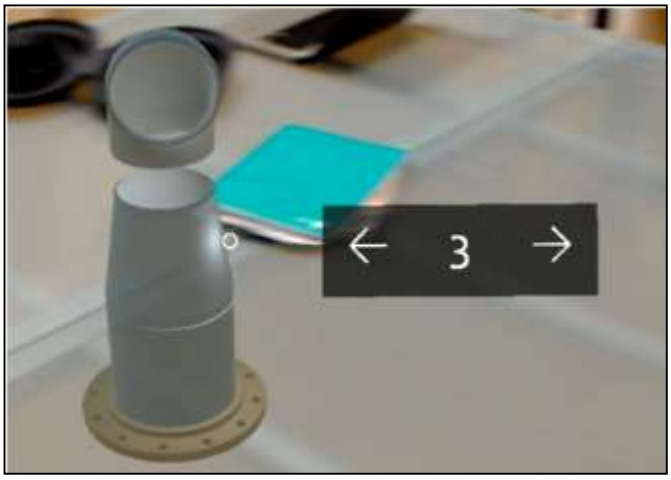


Fig. 7 – example of assembly step in AR holographic interface

In this cases the formalized knowledge exclusively refers to the product to assembly. The *transformations* contain the constraint between *parts* of an *assembly*. On the basis of an API interacting with the CAD model in the DB, the constraints between the *parts* in the *assembly* are retrieved and sent to the assembly sequence generator.

Even in this case to use this app the user doesn't need to be an expert in assembly process. The user only has to start the app and he can visualize the assembly sequence created starting from the knowledge formalized in the DB directly on the AR holographic device.

5.4. App to support the real-time data interpretation

The aim of this application is to support the real-time analysis of a monitored system. This application allows a user interested in monitoring a system, to receive notifications about the occurrence of anomalies for the parts of the plant and the processes of his interest. The knowledge formalization must be done by an expert user of the plant and related processes.

For this app we uploaded two *experiences* that refer to the efficiency and cost of chemical treatments of the water in evaporative cooling tower. The two *experiences* refer to the same *assembly* that represents an evaporative cooling tower. The *parts* connected to this *assembly* are components of the evaporative cooling tower, from which efficiency and cost of chemical water treatment depend. In this case, the *parts* linked to the evaporative cooling tower (*assembly*) refers to its components, since the main goal of this app is to highlight anomalies on components of the plant. Fig. 8 shows the efficiency *experience*. Since only for some components it was necessary to report anomalies, only some *parts* (in the highlighted section), in addition to the *assembly*, were linked to *tags* and *meta-tags*. In this way it is easier to report on a HM interface the components for which an anomaly has occurred.

As concerns the use of the knowledge in DB, it can be done by any registered user with an HM interface. During registration, the apps synchronizes with DB and it shows the user all *tags* and *meta-tags* related to components for which is possible to report the anomalies. The user must select among all the proposed *tags* and *meta-tags*, those of his interest. Just start the app to ensure that the user periodically receives notifications about prediction anomalies if the user's interest *tags* or *meta-tags* are within the limit distance associated with the *measurements* of the components for which the app predicted the occurrence of the anomaly. For anomaly predictions, the app uses data obtained from measurements made directly on the plant. These data are inserted into the mathematical models (*transformations*) performed using a mathematical solver. The results are compared with the correct operating range inserted in the DB.

For a correct use of this app it is not necessary the user to know the content of the DB, how the knowledge has been formalized, which measurements and coefficients are involved, and the transformations that allow the prediction of anomalies.

5.1. How the apps face the requirements?

The previous sections presented four different applications based on the implementation of the meta-model, which referred to four different knowledge domains.

Regarding the first requirement (knowledge based), although the knowledge to be formalized was extremely different and the way in which the meta model instances were used was different (e.g. input and output of production steps, components of a plant), it was possible to formalize all the knowledge necessary to use the four applications using always: 1) the same meta-model, 2) the same DB, and 3) the same interface, without the need for intervention of an expert. Moreover, it is no necessary that the user know how the knowledge was formalized, the measurements and the coefficients involved in the different experiences to use the proposed apps. Each interface is able to correctly understand the formalized knowledge and support the user.

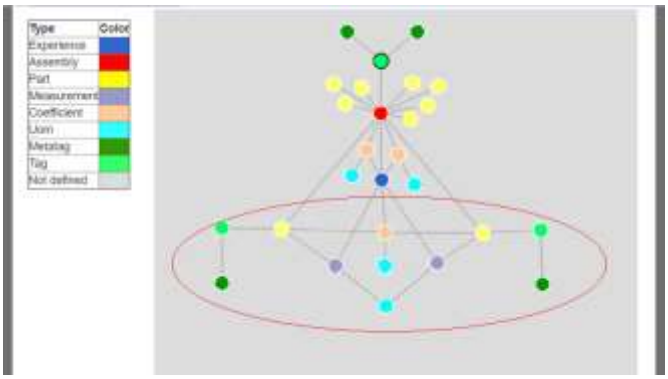


Fig. 8 – Representation in web interfaces of efficiency *experience*

On the other hand, concerning the second requirement (*extensibility*), the knowledge formalized in the same way has been used by completely different interfaces, both M2M and HM. So potentially, the same knowledge module can be used by different interfaces. This shows how the proposed way to formalize the knowledge also satisfies the requirement of *extensibility*.

6. DISCUSSION AND CONCLUSIONS

In Section 2 we have identified two main requirements that must be satisfied to promote the I4.0SMEs solutions: *knowledge-based* and *extensible*.

As shown in the applications presented in Section5, it is not required that the users have a deep knowledge about the application domain because a system that use the knowledge formalized with the proposed meta-model is able to properly support the users.

About the last requirement, all cases described in Section 5 use the same meta-model to formalise different knowledge without change anything about the formalization rules. So, the same DB can structure knowledge for different domains that can be used for different applications.

In this paper, we presented a meta-model based on the concept of measurement (related to sensors/actuators). The proposed meta-model represents the extension of the another ones, previously presented in (Giovannini et al., 2015). We implemented the new meta-model to favour the adoption of the I4.0 solution in the SMEs that are characterized to the lack of structured processes, expertise and resources, that make the I4.0 implementation even harder. The meta-model has been implemented in a DB and a web interface and tested on four prototype-apps. Each app interacted with the DB to query information to perform the required functionalities. The knowledge formalised in the DB and the querying process in the apps allow an easier implementation of apps for real-time data analysis, M2M interaction optimisation, smart traceability and AR supported assembly process.

The limits of the proposal are still related to the knowledge mapping in the DB. Ideally, a CAD files comparison can increase the integration of knowledge modules and their reachability. Moreover, other apps should be tested to perform a knowledge flow toward the DB: i.e. machine learning applications. This flow would allow to autonomously update the knowledge base of all the apps linked to the DB.

ACKNOWLEDGEMENTS

The scientific contents described in this paper are disclosed with the permission of MCI Srl, In. Tech. Srl, RG Impianti Srl, and Technoacque Srl companies, which committed research projects to InResLab scarl.

REFERENCES

Bosch, 2015. Industry 4.0 market study: demand for connected software solutions [WWW Document]. URL https://www.bosch-si.com/media/en/bosch_software_innovations/media_landingpages/market_survey_industry_4_0/20150928_industry4_0_market_study_en.pdf (accessed 11.24.16).

Brettel, M., Friederichsen, N., Keller, M., Rosenberg, M., 2014. How virtualization, decentralization and network building change the manufacturing landscape: An Industry 4.0 Perspective. Int. J. Sci. Eng. Technol. 8 1 37 44.

Dassisti, M., De Nicolò, M., 2012. Enterprise Integration and Economical Crisis for Mass Craftsmanship: A Case Study of an Italian Furniture Company., in: OTM Workshops. pp. 113–123.

Dassisti, M., Giovannini, A., Merla, P., Chimienti, M., Panetto, H., 2017. Hybrid Production-System Control-Architecture for Smart Manufacturing, in: OTM Confederated International Conferences" On the Move to Meaningful Internet Systems": OTM 2017 Workshops. Springer.

Dassisti, M., Giovannini, A., Merla, P., Chimienti, M., & Panetto, H. (2018, June). An approach to support I4. 0 adoption in SMEs: a core-metamodel and applications. In 16th IFAC Symposium on Information Control Problems in Manufacturing, INCOM 2018.

Dini, G., Santochi, M., 1992. Automated sequencing and subassembly detection in assembly planning. CIRP Ann.-Manuf. Technol. 41, 1–4.

Engel, G., Greiner, T., & Seifert, S. (2016, February). Two-Stage Orchestration Approach for Plug and Produce Based on Semantic Behavior Models. In Semantic Computing (ICSC), 2016 IEEE Tenth International Conference on (pp. 258-261).

IEEE.

European Commission, 2015. Annual Report on European SMEs 2014/2015.

Federmeccanica, 2016. Industria 4.0 in Italia: l'indagine di Federmeccanica.

Giovannini, A., Aubry, A., Panetto, H., El Haouzi, H., Pierrel, L., Dassisti, M., 2015. Anti-logicist framework for design-knowledge representation. Annu. Rev. Control 39, 144–157. https://doi.org/10.1016/j.arcontrol.2015.03.013

Harrison, R., Vera, D., Ahmad, B., 2016. Engineering Methods and Tools for Cyber #x2013;Physical Automation Systems. Proc. IEEE 104, 973–985. https://doi.org/10.1109/JPROC.2015.2510665

Huang, B., Li, C., Yin, C., Zhao, X., 2013. Cloud manufacturing service platform for small-and medium-sized enterprises. Int. J. Adv. Manuf. Technol. 1–12.

Issa, A., Lucke, D., Bauernhansl, T., 2017. Mobilizing SMEs Towards Industrie 4.0-enabled Smart Products. Procedia CIRP 63, 670–674.

Kagermann, H., Helbig, J., Hellinger, A., Wahlster, W., 2013. Recommendations for implementing the strategic initiative INDUSTRIE 4.0: Securing the future of German manufacturing industry; final report of the Industrie 4.0 Working Group. Forschungsunion.

Lavdim, H., Collarana, D., &Hoffmeister, M. Towards a Semantic Administrative Shell for Industry 4.0 Components. In 2016 IEEE Tenth International Conference on Semantic Computing (ICSC).

Lee, J., Bagheri, B., & Kao, H. A. (2015). A cyber-physical systems architecture for industry 4.0-based manufacturing systems. Manufacturing Letters, 3, 18-23.

Mansour, E., Sambra, A.V., Hawke, S., Zereba, M., Capadisli, S., Ghanem, A., Aboulnaga, A., Berners-Lee, T., 2016. A demonstration of the solid platform for social web applications, in: Proceedings of the 25th International Conference Companion on World Wide Web. International

World Wide Web Conferences Steering Committee, pp. 223–226.

Pfrommer, J., Stogl, D., Aleksandrov, K., Navarro, S. E., Hein, B., &Beyerer, J. (2015). Plug & produce by modelling skills and service-oriented orchestration of reconfigurable manufacturing systems. at-Automatisierungstechnik, 63(10), 790-800.

Schlegel, A., Langer, T., Putz, M., 2017. Developing and harnessing the potential of SMEs for eco-efficient flexible production. Procedia Manuf. 9, 41–48.

Tao, F., Zhang, L., Venkatesh, V.C., Luo, Y., Cheng, Y., 2011. Cloud manufacturing: a computing and service-oriented manufacturing model. Proc. Inst. Mech. Eng. Part B J. Eng. Manuf. 225, 1969–1976.

TELUS, IDC, 2014. Internet of Things Study 2014 – The Connected Canadian Business.

Thoben, K.-D., Wiesner, S., Wuest, T., 2017. "Industrie 4.0" and Smart Manufacturing–A Review of Research Issues and Application Examples. Int J Autom. Technol. Vol 11.

Ubisense, 2014. 2014 Smart Manufacturing Technologies Survey.