Knowledge representation, retrieval and reuse for product family design: an anti-logicist approach

Antonio Giovannini, Alexis Aubry, Hervé Panetto, Hind Bril El-Haouzi, Osiris Canciglieri Junior, Ludovic Pierrel

Knowledge representation, retrieval and reuse for product family design: an anti-logicist approach

A. Giovannini [a,b]. A. Aubry [a,b]. H. Panetto [a,b]
H. El Haouzi [a,b], O. Canciglieri Junior [c], L. Pierrel [d]

[a] CNRS, CRAN UMR 7039, France
[b] Université de Lorraine, CRAN UMR 7039, Boulevard des Aiguillettes
54506 Vandœuvre-lès-Nancy, France
(e-mail: {alexis.aubry; herve.panetto; hind.el-haouzi}@univ-lorraine.fr; ant.giovannini@gmail.com)
[c] Polytechnic School - PPGEPS, Pontifical Catholic University of Paraná (PUCPR), Brazil (osiris.canciglieri@pucpr.br)
[d] TRANE SAS, rue des Amériques, 88190 Golbey, France (ludovic_pierrel@trane.com)

Abstract: the product family design is a design approach to meet the demand of customisable products. This paper deals with the knowledge representation, retrieval and reuse supporting the design stage of product families. Usually, the methods in the literature do not focus on the retrieve and the reusability of the knowledge. In other words, they do not ensure if a non-expert user can effectively retrieve and reuse the represented knowledge. To cope with this point, here, the aim is to apply an anti-logicist approach for the unambiguous design-knowledge representation to support the unambiguous retrieval and the automatic reuse of the knowledge during a product family design stage. The retrieval is unambiguous because the link between the knowledge models and the requirements is based on a syntax comparison, e.g. intervals of numbers, units of measure. An algorithm for the automatic reuse has been developed: provided an unambiguous definition of the new requirements related to the product family, the algorithm's outputs are the functional and physical definitions of all the products included in the product families, i.e. performances and CAD files. The case study is a family of components of the HVAC (heating, ventilating and air-conditioning) systems sector. Finally the advantages and issues of a potential industrial implementation are discussed.

Keywords: Product family, Product platform, Knowledge representation, Knowledge reuse, Anti-logicism

1. INTRODUCTION

In the last decades, companies are increasing their efforts to reduce the cost and the lead time of the new products development while also meeting as much as possible the individual customers' requirements (Muffatto & Roveda, 2000). Companies are also stressing on the maximisation of the design assets reuse, e.g. components, modules, knowledge, for the development of new products (Moon, Simpson, & Kumara, 2010). The Product Family (PF) design is a strategy that allows to cope with those needs. Actually, a PF is a set of products that share common features. The set of features, components, parts and interfaces that are shared among the products in the same PF is named a Product Platform (PP) (Meyer & Lehnerd, 1997).

Many solutions have been developed to represent, retrieve and reuse knowledge about PPs and PFs. Most of the solutions in the literature are based on formal logics, e.g. ontologies, Formal Concept Analysis (FCA) (e.g. (Moon et al., 2010; Nanda, Simpson, Kumara, & Shooter, 2006; Seo & Ahn, 2009)). These approaches are intended to support the engineers during the new PF design stage.

However, there is a lack of studies about the retrieval and the actual reusability of the knowledge, i.e. is this knowledge representation approach efficient for making the *user* able to retrieve and reuse the knowledge required for the new PF design process? What about the knowledge retrieval if the user is not from the same area of expertise of the knowledge modeller?

The aim of this paper is to provide an approach to allow the retrieval and the reuse of the design-knowledge about PPs and PFs, even if the user of the knowledge is not from the same area of expertise of the modeller. In other words, the purpose is to propose an approach to link *unambiguously* the new customers' requirements and the represented knowledge about PFs and PPs.

The models of the customers' requirements and of the design-knowledge have been based on the anti-logicist approach for the design-knowledge representation (Giovannini et al., 2015). According to this approach, both the requirements and the knowledge are represented as *measurements* and mathematical relations between the *measurements* features, i.e. *what*, *where* and *when* to measure. Notice that, the refinement of the customers' requirements is out of scope of this paper. Actually, the input requirements of the algorithm are measurable, as in the definition of *requirement* of (ISO/IEC, 2007): a *statement [..] which is unambiguous, testable or measurable, and necessary for product or process acceptability*.

The links between requirements and knowledge about PPs and PFs are about functional and geometrical aspects of the products. Then, given a retrieved knowledge model and the requirements, these relations allow an algorithm to describe the resulting new PF from the functional and the geometrical point of view, i.e. performances and CAD files of the product in the PF.

The proposed knowledge representation approach can be seen as an enrichment of existing design models, e.g. CAD files and mathematical model representing the functional behaviour. The representation of the customers' requirements and of the knowledge about PPs is based on an appropriate link (discussed below) between mathematical models used to design the products (or components) and the geometrical representation of these products (or components). Therefore, the main effort required to represent the knowledge is mainly about the connection of these models.

The remainder of the paper is structured as follows: the second section discusses the related works; the section 3 is about an introduction of the anti-logicist approach for design knowledge representation; the section 4 is devoted to the description of our approach and the developed system to unambiguously retrieve and automatically reuse the knowledge about PPs; the section 5 describes the case study about the design of a *water coil* family; the section 6 discusses the main advantages and limits of the proposal.

2. RELATED WORKS

Research works about the representation, retrieval and reuse of knowledge for PFs design involve and combine techniques such as formal logics constructs, object oriented modelling, geometric modelling (i.e. CAD models), and mathematical modelling.

In this section the representation, retrieval and reuse of design knowledge are treated in separate subsections. Since this domain of research is wide, in this paper, only the works that aim (directly or not) to an automatic or semi-automatic knowledge reuse are reviewed.

2.1. Representation of knowledge about PPs and PFs

The techniques to model information about PPs and PFs for knowledge sharing are mainly based on *ad hoc* data structures or on ontologies, usually formalized in the Ontology Web Language (McGuinness & Van Harmelen, 2004), especially the OWL profile based on Description Logics (Baader, Horrocks, & Sattler, 2008). Ontologies are information models that, per definition (Gruber, 1993), provide an explicit and sharable knowledge representation. Appropriate algorithms help to explicit the knowledge in the ontological models, i.e. *reasoners*. These algorithms are based on the open world assumption (Atkinson & Kiko, 2008), roughly *if it is not represented in the model it is assumed as true*. This assumption improves the extensibility of the ontological models and so it fosters the collaboration on the represented knowledge.

In (Moon, Chang, Terpenny, Simpson, & Kumara, 2009; Moon et al., 2010; Nanda et al., 2006; Seo & Ahn, 2009), authors propose methods to build ontologies dedicated to the representation of products: the products in the PF are characterized as product components, modules, functions and/or related features. Approaches to build ontologies describing and merging different points of view have also been proposed: (Ge, Yang, Duan, & Chen, 2013) presents a method to link customer demand, demand forecast and product features in the PF; (Lim, Liu, & Lee, 2011) proposes a method to build a multi-domain ontology for integrating several points of view such as manufacturing, sustainability and economics; (Nanda, Thevenot, & Simpson, 2005) introduces a technique to build three ontologies (i.e. about customer needs, functional aspects and components features) then mapped with a matrix-based method.

Other data models not based on ontologies are presented in (Brière-Côté, Rivest, & Desrochers, 2010; Nomaguchi, Taguchi, & Fujita, 2006; Ong, Xu, & Nee, 2006; Siddique & Rosen, 2000; Tseng, Chang, & Chang, 2005; Youliang Huang et al., 2007; Zhang, Wang, Zhong, & Wan, 2005; Zha, Sriram, & Lu, 2004): also in these works the knowledge about PPs and PFs is intended as information about the functional aspects, modules or components features and constraints to define unfeasible components combinations. In detail, (Tseng et al., 2005) adds geometrical features to the PF models. (Zhang et al., 2005) proposes a technique to integrate PF and PP models to Product Data Management systems. (Ong et al., 2006) connects functional models and information about the orders history to CAD models. (Zha & Sriram, 2006) uses data models based on the STEP ISO standard (ISO, 2004).

In (Zha & Du, 2006a, 2006b; Zha & Sriram, 2006), the authors propose a modular structure to model the PF. The modules are described by algebraic equation that define the functions. The modules are connected by means of parameters about functional aspects.

Examples of systems that implement these models about PP and PF are in: (Seo & Ahn, 2009), ontology-based system with a graphical interface and accessible from the web; (Ni, Yi, & Ni, 2011), an ontology-based repository for petrochemical data; (Jiao & Helander, 2006), author developed a system for product configuration that links aspects of design, manufacturing and supply chain; (Zha & Du, 2006a, 2006b), a web application that integrates CAD models, mathematical representation of the functional aspects of the PF and a genetic algorithm.

2.2. Impacts on the knowledge retrieval

Knowledge retrieval for databases and ontology is based on the formulation of appropriate queries. The main advantage of the ontological model is the possibility to launch an algorithm to make explicit all the represented knowledge in the model. The discussion about the other differences and similarities of the queries results for databases and ontologies is out of the scope of this paper.

The knowledge models presented in the previous section allow to retrieve knowledge about the features of the modules and of the components, about the functions that a component, module or product could provide and about the feasible combinations of components and/or modules. For instance, in (Ong et al., 2006; Zha & Du, 2006a, 2006b; Zha & Sriram, 2006), the connection of the represented knowledge with CAD models helps the concepts understanding in the knowledge models (i.e. databases or ontologies) and so the query formulation. In (Zha & Du, 2006a, 2006b; Zha & Sriram, 2006) also the mathematical characterization of the module functions could help the concepts understanding in the databases and so it can improve the knowledge retrieval by queries.

2.3. *Impacts on the knowledge reuse*

In this section, the focus is on the methods to help the deployment of the retrieved knowledge to modify an existing PF or to build a new one.

The retrieved knowledge from ontological models or databases represents features of components, modules and products developed during previous design processes. If these features are not connected to appropriate mathematical models, the effect of a required modification to the PF is not assessable. Let us consider a model describing the features of a product included in a PF. Suppose that this product is able to provide a power of 10kW. A new product with a power of 11kW is required. The represented knowledge is reusable only if the model contains enough information to calculate the effects of this requirements on all features of the product and/or of the module that compose it. In other words, the knowledge about *how to obtain a certain value of power* should be represented in the model.

Only few works (e.g. (Zha & Du, 2006a, 2006b; Zha & Sriram, 2006)) represent mathematical models of the modules behaviour to connect the features of the product. The systems implementing these models allow to appreciate the effects of an alteration of the value of one of these features. Therefore, when the knowledge is retrieved, it can be used to automatically alter the design of a PF or to design a new one. In some works (e.g. (Ong et al., 2006; Zha & Du, 2006a, 2006b; Zha & Sriram, 2006)) also an approach to link the functional features to the geometric features has been proposed.

2.4. *Discussion and problem statement*

In this section a discussion of the effect of the knowledge representation on the retrieval and the reuse of this knowledge on the PF design has been provided. From the literature reviewed in this section, it is possible to highlight two categories of approaches:

- Approaches that use structured data models representing abstract concepts by means of natural language, i.e. databases, ontologies; they provide methods to retrieve information about the previous designed PF and/or about the assets in the PPs;
- Approaches that combine data models with mathematical representations of the functional behaviour and of the geometric features; they provide a more complex and comprehensive representation approach; the retrieval is based on abstract concepts linked to measurable product features, e.g. performances, geometric features.

The methods in the first point lead to two main issues: 1) they do not allow an automatic knowledge reuse because they do not provide the knowledge (i.e. mathematical relations) to compute the appropriate product modifications to fulfil a new customer requirement; 2) the use of the natural language causes an inevitable source of ambiguity, i.e. the hypothesis of these methods is that the users relate the same semantic to the *words* that the modeller used to represent the knowledge; this hypothesis makes uncertain every retrieval effort (further details are in the next sections).

The methods in the second point solve the first issue. Regarding the second one, the use of natural language to link the pieces of mathematical models preserve the source of ambiguity for the knowledge retrieval.

In summary, there is no approach that guarantees the *unambiguous* correspondence between a query (e.g. a new requirement) and the retrieved knowledge. In other words, an approach that deals with the user's understanding of the concepts described in the database or in the ontology. Is the knowledge retrieval efficient even the *user* of the represented knowledge is not of the same domain of the modeller, e.g. an engineer from another area of expertise?

The aim of this paper is to support the new PF design process with an *anti-logicist* knowledge representation (Giovannini et al., 2015). This approach allows an unambiguous retrieval and an automatic reuse of the design-knowledge about PPs or previously designed PFs. The anti-logicist approach (described below) for design knowledge representation has been applied to provide *unambiguous* knowledge retrieval even if the *user* is not from the same area of expertise of the modeller. More specifically, the outputs of the knowledge retrieval and reuse stages are about the performances and the geometric features of all the product variants in the new PF.

3. FORMAL LOGIC CONSTRUCTS AND AMBIGUITY OF THE KNOWLEDGE MODELS

In (Giovannini et al., 2015), the authors propose a framework for the design-knowledge representation. The key feature of this framework is the *unambiguity* of the knowledge models that are conform to it. To clarify this point, the first part of this section is devoted to the definition of *ambiguity* of a knowledge model. Then, the characteristics that a representation should have to be *not ambiguous* are described. Finally, the *anti-logicist* approach is introduced and confronted with these features.

3.1. *About the ambiguity of a knowledge model*

Literally, the term *ambiguous* means *subject to more than one interpretation*[1]. In this paper, a knowledge model is considered *ambiguous* when the user of the model can associate the represented knowledge with more than one *meaning*, i.e. different users can perform different interpretation of the same model. Let us consider a general design-knowledge communication scenario. The modeller M represents in the model K his knowledge about how to solve a specific design problem X. According to M, Y is the solution (or a set of alternative solutions) of the problem X. Given a set of users U using only K to solve X, K is *not ambiguous* only if each member of U retrieves Y from K, i.e. like M would have done to solve X.

[1] http://www.oxforddictionaries.com/

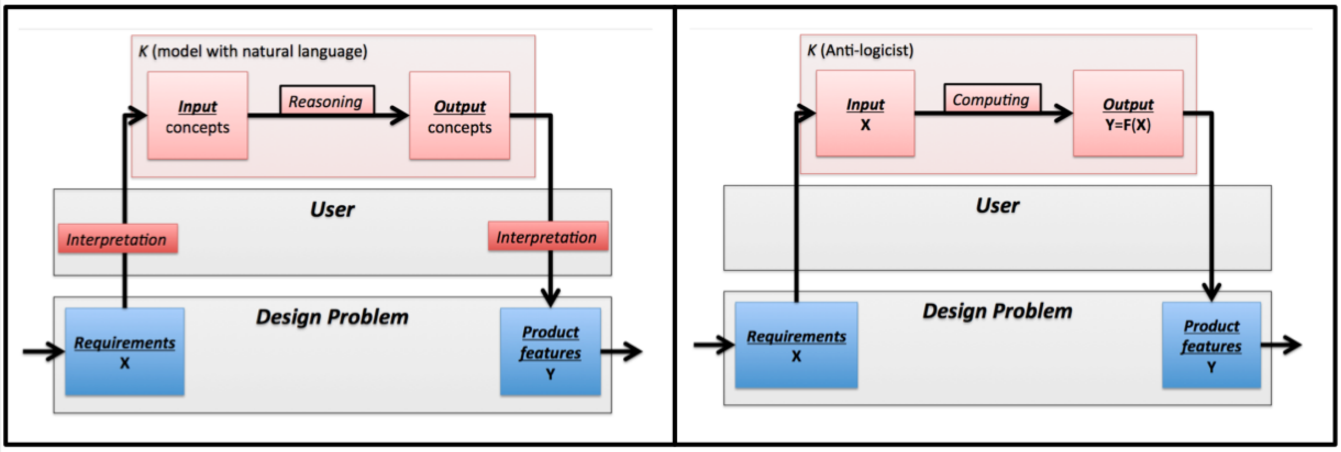


Fig. 1 – the use of logicist and anti-logicist knowledge models to solve a design problem, adapted from (Giovannini et al., 2015); the focus is on the *interpretation* required only on the left side to design the *product features* related to the *requirements* and using the knowledge model *K*.

3.2. About the characteristics of an unambiguous design knowledge representation

To obtain the unambiguity, K should constrain the users' interpretations to be unique and equal to the M's one. In other words, constraining the users' interpretations means that there is (at least) a *N-to-1* relation between the representation and the knowledge that M wanted to communicate. The verification of this N-to-1 relation is necessary to assess the unambiguity of a knowledge model.

To do so, let us consider the nature of the knowledge to represent. The design-knowledge concerns the links between requirements (representing the design problem) and the product features (representing the solution). Both of them are represented by measurements: requirements are measurable per definition (see (ISO/IEC, 2007)); product features are obviously testable and so measurable. Therefore, if the aim of the knowledge model is to be reused for a new design stage, the model should be linked to the measurements about the requirements and the product features, respectively *X* and *Y* at the bottom of the Fig. 1.

When the model is not connected to X and Y, the *user* is obliged to bridge the *gaps* 1) between the problem X and the input of the knowledge model (i.e. the concepts used in the knowledge model to describe X) and 2) between the model outputs (i.e. the concepts used in the knowledge model to describe the Y) and the measurable product features, i.e. the solution Y. In other words, to constrain the user interpretation, a knowledge model should be formalised by measurable representations of the requirements and of the product features. Therefore all the previously reviewed methods that use natural language (even if in data structures, e.g. databases, ontologies – at the left of the Fig. 1) are not constraining the users' interpretations, i.e. the user has to bridge the gaps between the *concepts* and the *X* and *Y*. In the previous section, it has been shown that all the reviewed methods are at least partially based on *concepts*, e.g. function, module, feature.

Thus, in the next section, the *anti-logicist* approach is introduced for representing knowledge. This approach is based on mathematical connection between X and Y, therefore it does not involve the user interpretation (at the right of the Fig. 1). It is also explained how this approach allows to constrain the user interpretation and so to respect a N-to-1 relations between the representation and the knowledge of the modeller.

3.3. The anti-logicist approach for knowledge representation

In the anti-logicist approach proposed in (Giovannini et al., 2015), the *measurement* is the key concept that allows to avoid the user interpretation and so the consequent ambiguity in the knowledge representation and retrieval.

The conceptual model showing the basic concepts of this approach is shown in Fig. 2. The semantics of the main concepts is the following.

- A *measurement* is the characterisation of the act of perceiving, by mean of a measurement device, a certain measure (*what*) in a certain place (*where*) at a given time (*when*). For instance, the temperature (*what*) perceived in a room (*where*) during the morning (*when*) can be an instance of *measurement*.
- Each measurement is identified univocally by a vector (*STS*) with three components: shape (*what*), time (*when*) and space (*where*) characterisation. For instance, the temperature is the *shape* of the measurement (i.e. what to measure), the volume of air in the room is the *space* of the measurement (i.e. where to measure) and the time interval of the measurement represent the *time*. Each one of these three elements is a *property*.
- Each *property* is involved in one or more transformations: a *transformation* is a mathematical relation between a set of properties' values. For instance, the relation between the temperature and the time when the temperature is measured is the transformation $T = f(t)$, where t is the *time* of the measurement, T is the temperature (*shape*) and f is the mathematical relation that links the two properties.
- A set of mathematical relations between the properties of a set of measurements is defined as an *experience*.

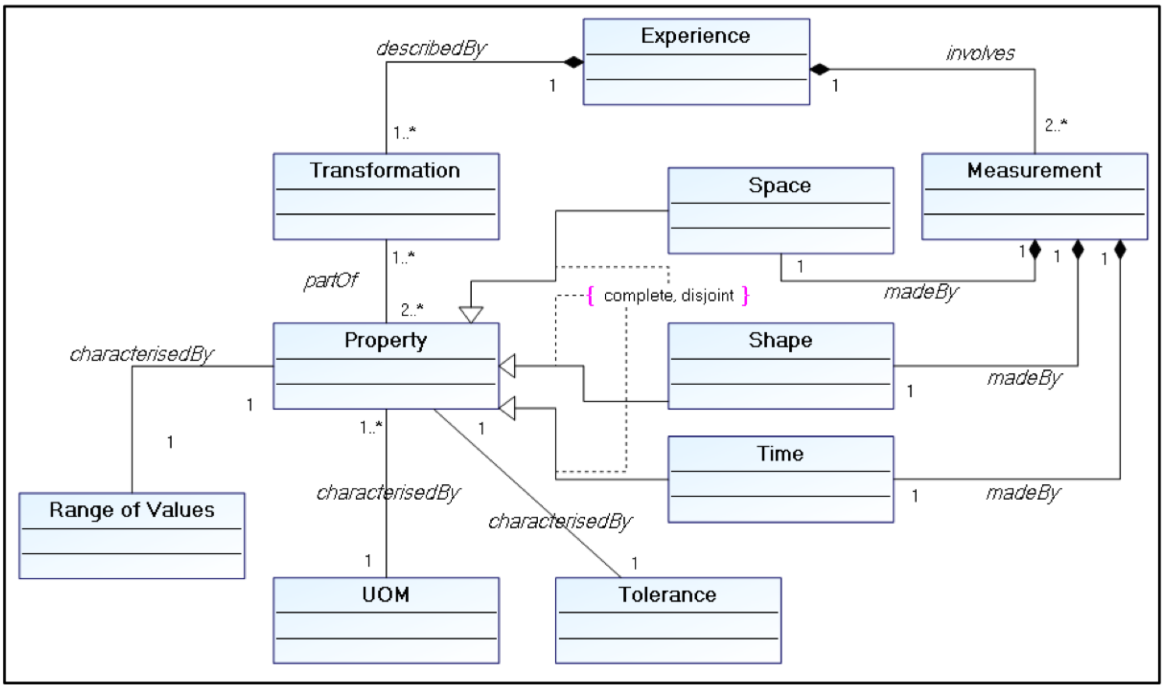


Fig. 2 – conceptual model representing the concepts of the anti-logicist approach, from (Giovannini et al., 2015)

- 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 (*range of values* for the validity of the experimental law) with a *tolerance* of ±0.1. The *tolerance* represents the uncertainty of the measurement (relative to the measurement error of the instrument), e.g. for 25°C±0.1 the temperature value is greater than 24.9° and lower than 25.1°

The ten concepts in this approach can be used to represent every kind of object by measurements. For instance, a copper cable should be described as a cylinder (*space*) in which it is possible to measure some *shapes* that should define the interaction of the copper with the rest of the observed system, e.g. the conductivity of the copper if the behaviour of an electric power system is the object of the observation or the thermal conductivity if the knowledge to be represented is about an air conditioning system. In this way, the object is characterized only by the measurements that define its impact in the system behaviour.

Thus, natural language is not necessary to represent design-knowledge. Moreover, every part of the knowledge model can be connected to the set of measurements representing the problem (*X*) and with the set of measurements representing the solution (*Y*).

If every part of the model can be represented as relations of measurements, an algorithm can be designed that compares the *properties* (*range of values*, *UOM*, *tolerance*) of the *space*, the *time* and the *shape* of each measurement that represents a problem X and a solution Y (more details in (Giovannini et al., 2015)). In other words, this algorithm can retrieve the knowledge related to the measurements that define the requirements only by comparing numbers (i.e. *range of values* and *tolerances*) and the syntax of the *UOM*. If an algorithm can be used to retrieve the knowledge useful to solve the problem X, then all the users and the modeller using this algorithm are constrained to one unique interpretation of the knowledge, i.e. the algorithm proposes the same Y for all the users.

Moreover, since equivalent geometries and/or equivalent equations can be modelled in several different ways (e.g. different volume intersections, different transformations of the mathematical relation), the relation between the knowledge representation and the usage of it is an *N-to-1* relation.

4. ANTI-LOGICIST APPROACH FOR SUPPORTING PRODUCT FAMILY DESIGN

In this section, the aim is to apply the anti-logicist approach (Giovannini et al., 2015) for the formalisation of the knowledge about the PPs and PFs. The knowledge representation and the impact of it on the retrieval and the reuse of the design-knowledge have been analysed in separate sections:

1. How to represent the knowledge about the existing PFs according the approach in the previous section?
2. How to unambiguously retrieve the knowledge related to a set of requirements from a knowledge base composed by *experiences*?
3. How to automatically reuse the retrieved knowledge for the computation of the geometrical and functional features of the new PF?

Finally (in section 4.4), a prototype that performs the automatic reuse has been presented.

4.1. Knowledge representation

The application of this approach for the PF design support requires to formalise the knowledge needed to design a new

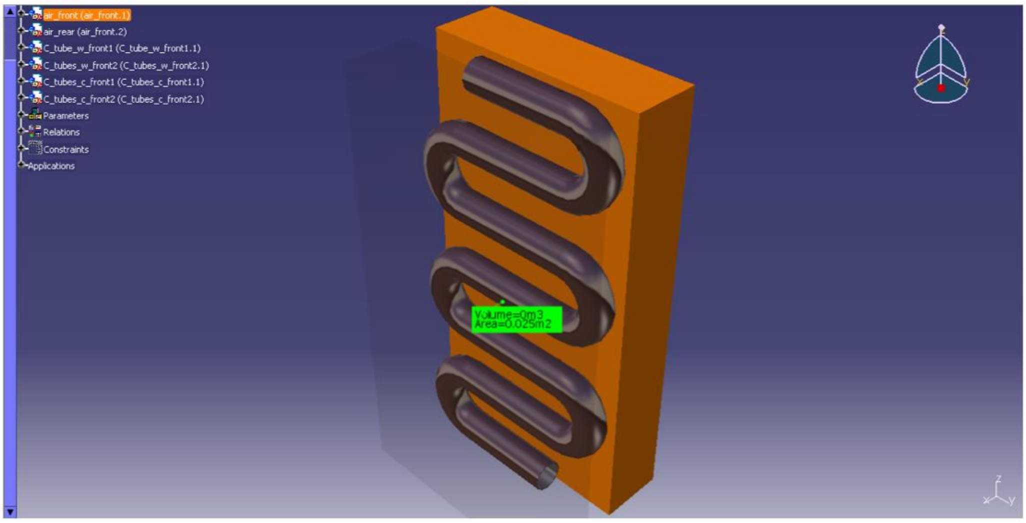


Fig. 3 - CAD model of a bare tube coil including the representations of the flows of water and air.

PF or to modify an existing one. In this paper, this design-knowledge has been represented as an instance of the *experience* concept. An *experience* should capture the variety about the components features that can be combined to design a product satisfying the customers' required performances. The components features and the customers' required performances have been represented as instances of *property*. All the *properties* of a *measurement* are represented in CAD software.

- The *spaces* are volumes in the CAD model. They represent the geometrical features of the components (e.g. the profile of a fan) and the places where the customers' required performances (e.g. the room where the fan generates the air flow) should be observed.
- The *shapes* are *properties* linked to these volumes. They represent the customers' required performances (e.g. the speed of the air flow) and the components features that impact the achievement of these performances (e.g. the fan required energy).
- The *times* are also linked to the volumes. They represent the intervals of time when the *shapes* must be observed in the volumes. All the *times* of the measurements in the same *experience* are related. In other words, the *time* the measurement *X* defines how much time later or before than the other measurements *X* is observed. For instance, the time characterisation of room temperature with the fan switched off (measurement X_1) and the room temperature after one hour of air ventilation (measurement X_2).

For each one of these *properties*, a *UOM*, a *range of values* and a *tolerance* must be specified. The *transformations* that involve the *properties* are represented in a mathematical model that describes the behaviour of the system.

For the sake of simplicity, here, let us consider an instantiation of the concepts in Fig. 2. The knowledge related to a bare-tube coil (i.e. an example less complex than the one in case study) has been considered. A bare tube coil provides a tool to exchange heat between a water flow and an air flow. A given volume of water flows into the coil to cool the airstream. The represented behaviour is a sensible heat transfer (ASHRAE, 2012). The captured variety is about how the coil material and geometry impacts the temperature of an airstream.

The bare-tube coil is the *experience* to be represented. As explained above, an experience is a set of mathematical relations (i.e. *transformations*) between *measurements*. The representation of its *shape*, *time* and *space* univocally define a *measurement*.

The instances of the ***measurement*** concept are (referred to the Fig. 3):

- the temperatures of the air before and after the heat exchange, T_A1 and T_A2;
- the temperatures of the water before and after the passage in the coil, T_R1, and T_R2;
- the speed of the air, V_A;
- the density and the specific heat capacity of the air and the water, RO_A, RO_W, C_P, C_R;
- the film coefficient of the heat transfer between the air and the external coil surface, F_A;
- the film coefficient of the heat transfer between the water and the internal coil surface, F_R.

The ***space*** representation has been performed on CAD software: parameters are added to the CAD files and linked to the volumes that should provide the representation of *where* the measurement is performed. The CAD model of the bare tube coil is shown in Fig.3: the volumes represent the copper (i.e. tubes), the water (i.e. that flows in the tubes) and the air (i.e. in front of and behind the coil).

The relative ***shapes*** are the *ranges of values*, the *UOM* and the *tolerance* related to the measurements of the temperatures, the speeds, the densities and the heat transfer coefficients, e.g. from 10 °C to 30 °C, ±0.1 °C. All measurements are related to specific volumes or surfaces (the *space* characterisation). For instance, in Fig. 3, the volume related to the C_P, RO_A and T_A1 measurements is the orange volume behind the coil. The link of the measurement spaces with the volumes represented in the CAD model is performed adding a specific parameter and associating it to the measurement of the relative volume. All the parameters about the *shape* are connected with the parameters about the *space* (i.e. dimensional parameters of the geometries in the CAD) by means of rules that should be coded in the CAD. When parameters about *shape* change, the CAD should use the rules adapt properly the parameters about the

space. (details in the case study sections and in (Giovannini et al., 2015)).
The observed system is stationary and thus all measurements are ***time***-invariant. In the case of dynamic behaviour, the shapes (i.e. variables such as temperature, pressure) and the spaces (i.e. volumes) can be expressed as functions of the time variable. The associations of the measurements to the volumes provide a characterisation of the properties related to the volumes that are involved in the system behaviour. For instance, the association of the C_P and RO_A with the air volumes characterises the physical properties of the air that are involved in the behaviour of the system, i.e. the sensible-heat transfer.
The ***transformations*** that relate the *properties* of the measurements are algebraic equations that constrain the properties' values. These equations include a group of variables not related to the above cited measurement:

- Q_T is the heat exchange rate;
- W_A is the air mass flow;
- W_R is the water mass flow;
- DELTA_T is the mean temperature difference between the airstream and the water;
- U_0 is the overall coefficient of heat transfer for sensible cooling (without dehumidification).

Actually, the computation of these values is based on the values of the above-defined properties; e.g. the DELTA_T is calculable from the water and air temperature values.
The other variables not cited in the transformations are:

- A_A that is the front surface of the coil;
- A_0 is the external surface of the coil;
- A_I is the internal surface of the coil.

These three properties are directly related and calculated on the basis of the characterisation of the coil geometry in the CAD model. Actually also F_A and F_R can be calculated on the basis of the coil geometry, but the detailed instantiation is out of the scope of this section.
As discussed above, the abstract concept of air is formalised by means of the *properties* of the C_P and the RO_A. In this way this framework provides a representation of the object without using the natural language. The abstract concepts that should define an object are replaced by the *shapes* of the *measurements* in the volume occupied by the object in a certain *time*. Because of any knowledge is defined by measurements, no interpretations are needed to connect a certain measured reality with a modelled system behaviour. Therefore no *ambiguity* is possible when the formalised knowledge will be retrieved and interpreted for designing a new PF.
There is no actual limit to the PF and PP variety that can be captured in one knowledge model. The higher is the variety represented in one model the lower is the amount of models required to represent all the variety of the components' features. However, the larger is a knowledge model the more complex is the computation of the related mathematical model. Thus, the resources required to manage the mathematical model should limit the size of the knowledge model. For instance, let us consider the water coil case (discussed below). In the case study, the relative positions of the pipes are represented with a parametric rectangle: the centres of the pipes' sections are at the vertexes of rectangles. The parameters about the relative pipes' positions are related to the sides of the rectangle. The literature (ASHRAE, 2009) shows that the centres of the pipes' sections can be arranged at the vertexes of other shapes, e.g. triangle, hexagon. All this variety of geometries can be represented in a unique knowledge model. However, this strategy will increase the complexity of the related mathematical model about the heat exchange. Therefore, since the representation is not ambiguous (i.e. the fragmentation of the knowledge does not affect the interpretation), the choice about fragmentation of the knowledge representation depends only on the computational resources required to manage the mathematical model.

4.2. Knowledge retrieval

In this section, it is discussed how to represent requirements about a PPs or PFs by means of the *antilogicist* representation. For numerical examples, see the case study.
When the knowledge representation follows the anti-logicist approach, the retrieval is based on the comparison of the properties' values (values for the *shape*, *time* and *space*) of the requirements representation with the represented knowledge models of the PPs or PFs. The customers' requirement should be compared with each *experience* to find the components related-knowledge that can help to meet the requirements. Therefore, the number of *experience* models has effects also on the knowledge retrieval, i.e. on the number of comparison between requirements and knowledge models.
Let us consider some customers who require the air quality control of a room. For instance, they specify the desired temperature and the humidity to be constant (*shape* and *time* characterization). To retrieve the *experience* model containing the appropriate knowledge, the user (e.g. a customer, an engineer) should compare the temperature and humidity values with the *ranges of values* (of the *shape* and *time*) included in the model (if the *experience* model deals with this measurements). Moreover, the *space* of the measurement has to be compared, e.g.; the size and shape of the room should be compared with the orange area in Fig. 4. More specifically, the room has to be compared with all the values the parameters (associated to the CAD volumes) that describe the variety of sizes and shapes of the orange area can accept. When all the measurements of the requirements have corresponding measurements (corresponding *space*, *shape* and *time* characterizations), the knowledge in the *experience* can be used to design the products in the new PF.
This retrieval is only based on the comparison of number intervals (*ranges of values*, *tolerances*) and on the syntax of the *UOM* of the measurements. Therefore, an algorithm can be developed to automatically retrieve the knowledge, i.e. the knowledge models are not *ambiguous*. This algorithm is currently under development but the discussion about its advantages and limits is out of the scope of this paper.
The output of the retrieval process is the correspondence of all the *properties* used to define the requirements representation with the *properties* used to represent PP knowledge model. Since a knowledge model has been retrieved, other constraints can be defined to limit the values that the *properties* in the model can accept. For instance, the bare tube coil *experience* is retrieved for cooling an airstream. A further customer requirement can constrain the temperature of the water

entering and leaving the coil, or the generated noise, or the air temperature observed just before the coil and so on.

4.3. *Knowledge reuse*

This section shows how the *transformations* are mathematically represented to be computed for the knowledge reuse.

At the end of the retrieval stage, the customers' requirements are constraining the values of the *properties* in an *experience* model. The knowledge reuse is based on the computation of feasible solutions of mathematical model represented with Mixed-Integer Non-Linear Programming (MINLP - (D'Ambrosio & Lodi, 2013)). The constraints in these models are the algebraic equations (*transformations*) and the customers-related limits on the *properties'* values. For each customer, when a solution of the model exists, this solution represents a set of values of the components-related *properties* that can meet the requirements. When the customers' requirements are represented in continuous intervals, e.g. 10-15°C, to identify the individual customers, this interval should be discretized as follows.

The presence of a value for the *tolerance* allows the discretisation of the properties related to the customers' requirements. For instance, consider that the range of temperature T_1 identifies the customer requirements. The T_1 has values from 10 to 15 (*range of values*) °C (*UOM*) with a *tolerance* of ±1. From the discretisation of the requirements interval T_1, six different customers' requirements are identified as follows:

$$\begin{cases} 9 < T_1 < 11 \\ 10 < T_1 < 12 \\ 11 < T_1 < 13 \\ 12 < T_1 < 14 \\ 13 < T_1 < 15 \\ 14 < T_1 < 16 \end{cases} \quad (1)$$

The general MINLP mathematical model to compute the values for the components-related *properties* is the following:

$$find\ \bar{x}, \bar{y}, \bar{k} \quad (2)$$
$$s.t.$$
$$L_1 < \bar{x} < U_1 \quad (3)$$
$$L_2 < \bar{y} < U_2 \quad (4)$$
$$L_3 < \bar{k} < U_3 \quad (5)$$
$$f_j(\bar{x}, \bar{y}, \bar{k}) = 0, j \in [1, P] \quad (6)$$
$$g_w(\bar{x}, \bar{y}, \bar{k}) \leq 0, w \in [1, Q] \quad (7)$$

where:

- the P functions f_j and the Q functions g_w are the linear and/or non-linear constraints representing the *transformations*;
- the vector $\bar{x}$ has dimensions representing the *properties* related to the customers' requirements (e.g. the air temperature of the room);
- the vector $\bar{y}$ as dimensions representing the *properties* related to the components (e.g. diameter of the tubes of the coil);
- the vector $\bar{k}$ has dimensions representing the *properties* involved in the *transformations* but not related neither to the customer requirements nor to the components (e.g. the turbulence of the water flow);
- L_i and L_i are the vectors with the lower and upper bounds respectively for each dimension of $\bar{x}, \bar{y}, \bar{k}$; there bounds represent the uncertainty of the measurements (as discussed in the previous section); these bounds of $\bar{x}$ are calculated as for the discretization discussed above; concerning $\bar{y}, \bar{k}$, the bounds represent the range of values that the properties can accept.

A MINLP solver is used to compute the solution (if existing). When a solution exists, the considered components having the computed values for the *properties* can be designed and combined to achieve the customers' required performances.

4.4. *Prototype for the knowledge reuse*

In this section a tool to collect the constraints, generate the MINLP problems and compute the best PF is presented.

The data about the customers' requirements are structured in a XML file. In this file, each customers-constrained *property* is characterised by three parameters: 1) the *tolerance* that is accepted by the customer; 2) the *tolerance* that is related to the *property* in the *experience;* 3) the values or the range of values (if a discretization is needed) that constrain the *property*. The solver does not accept < and > constraints. Therefore, to formalise the value of the *properties* bounds, L and U (discussed in the previous section), both the *tolerance* values are required. For instance, let us consider the temperature, T, of the air exiting the coil with a value of 11.5. Since the customer *tolerance* is ±0.1 and the *tolerance* in the *experience* is ±0.01, the constraint that has to be represented in the solver for the customer $11.4 < T < 11.6$ is the following:

$$11.31°C \leq T \leq 11.49°C,$$

where the lower bound is equal to

11.5°C (nominal value) - 0.1°C *(customer-related tolerance) + 0.01*°C *(tolerance in the experience)*

and the upper bound is equal to

11.5°C (nominal value) + 0.1°C *(customer-related tolerance) - 0.01*°C *(tolerance in the experience).*

Even if the focus of this paper is the knowledge reuse and not an approach to provide "the *optimal"* PF design, a simple mechanism to orient the solution toward the optimum solution has been provided.

- To each feasibility problem (related to a customer), an objective function to minimize the costs of the product variant has been associated: e.g. related to the usage of raw materials.
- Moreover, a cost for the variety can be defined, i.e. a cost to take into account how many values for each *module* (e.g. length of coil tubes) are used to meet the requirements of all the customers. Therefore, in the XML file, also

constraints related to the components-related *properties* can be expressed. These constraints can be used to limit the *property* to accept only some standard values, e.g. standard tube length, diameters. In other words, all the product variants are constrained to use only some specific components, here called *modules*. Defining three possible values for the coil tube length means that there is one module and three possible values for it.

- The amount of MINLP models to generate is equal to the product of the number of customers (*N*) and the number of values defined for the modules (*Q*). The solver is invoked to calculate a solution for each problem. The results provide information about the achievable customers' requirements and the related components-related *properties* values, i.e. the values that optimise the provided objective function.

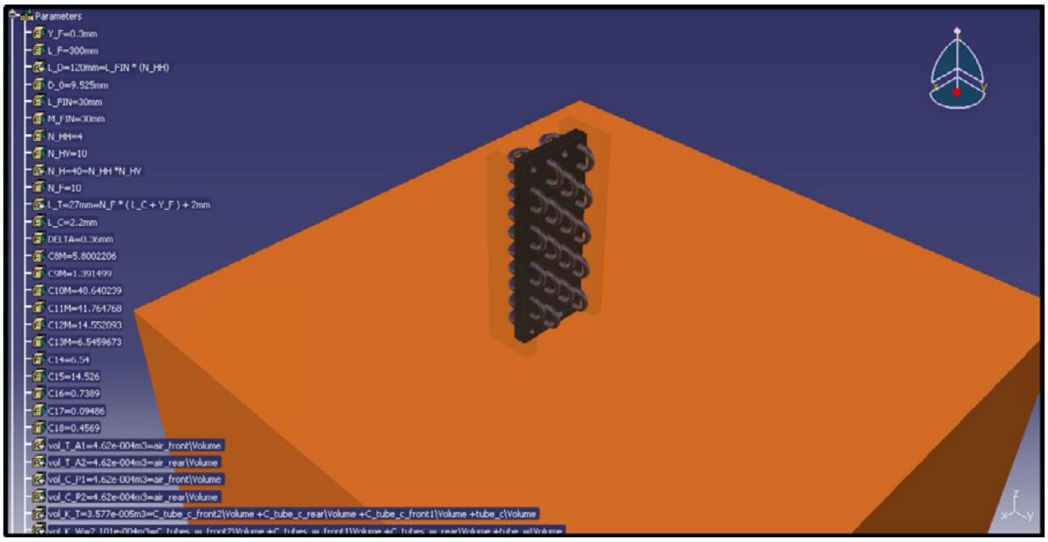


Fig. 4 - CAD file representing the *spaces* of the water coil platform.

To take into account the costs and the variety minimization, the following optimization problem is generated:

$$min\ \sum_{m=1}^{M}(CV_m \sum_{v_m=1}^{V_m} U_{mv_m}) +\ \sum_{q=1}^{Q}\sum_{n=1}^{N} Z_{nq}C_{nq} \quad (8)$$

$s.t.$

$$\prod_{m=1}^{M} V_m = Q \quad (9)$$

$$\forall n,\ \sum_{q=1}^{Q} Z_{nq} = 1 \quad (10)$$

$$\forall m, \forall v_m,\ BU_{mv_m} - \sum_{n=1}^{N} Z_{nq_{v_m}} \leq 1 \quad (11)$$

$$\forall m,\ U_{mv_m} \in \{0,1\} \quad (12)$$

$$\forall n,\ Z_{nq} \in \{0,1\} \quad (13)$$

where

- N is the number of customers, M the number of modules, V_m is the number of values that a module m accepts, Q is the number of combinations of modules' values, B is a number greater than N,
- Z_{nq} is the binary variable that defines if the product variant q for the customer n is part of the best PF;
    - $Z_{nq_{v_m}}$ is the binary variable that defines if the product variant q (that uses the module m with the value v_m) for the customer n is part of the best PF;
- U_m is the binary variable that defines if the value v_m of the module m is used in the best PF;
- C_{nq} is the cost of the product variant using q for the customer n; this cost is the result of the optimization of the other product variant *properties* according the defined objective function;
- CV_m is a parameter representing the estimated cost for the variety management of one module.

The limits and the improvement perspectives about the optimization mechanism are discussed in the conclusions.

In summary, the developed tool allows:

(1) to collect the data about the customers constrained *properties*; the data are structured in a XML file;
(2) to discretize the eventual requirements represented by ranges of values;
(3) to generate a MINLP model for each customer;
(4) to invoke the solver Lingo[2] for the solution of the mathematical models;
(5) to model an optimization problem to compute the best PF to meet the customers' requirements.

5. A CASE STUDY: WATER COIL FAMILY

The aim of this section is to apply the knowledge representation discussed in the previous sections and to simulate the knowledge retrieval and reuse for supporting the design of a water coil PF.

[2] http://www.lindo.com/

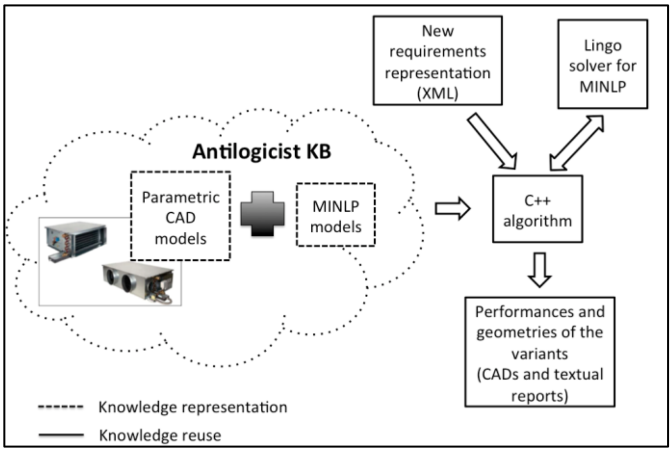


Fig. 5 - abstract representation of knowledge representation and reuse.

5.1. *Knowledge representation*

A water coil is a component of the *fan coil* (Fig. 6). A *water coil* is made of a set of punched aluminium fins. The fins are assembled on a coil of copper tubes that contains a flow of water. A motor activates a fan that generates and orients an airstream towards the coil for the heat transfer. The usual *fan coil* implementation is used in hotel rooms, universities, offices and so on.

Fig. 6 - images of the Trane fan coil.

The formalised knowledge about the *water coil* behaviour is from the following standards for the HVAC domain:

- The chapter 1 in (ASHRAE, 2009), about *psychrometrics*, has been used to model the air temperature and humidity relations;
- The chapter 4 in (ASHRAE, 2009) about *heat transfer* has been used to model the relations between heat transfer and fins geometry;
- The chapter 22, about *air-cooling and dehumidifying coils*, in (ASHRAE, 2012) has been used to model the water coil system behaviour;
- The AHRI standard on *Forced-Circulation Air-Cooling and Air-Heating Coils* (Air-conditioning, Heating & Refrigeration Institute, 2001) has been used to model the relations between the coil geometry and the heat transfer.

The knowledge about the water coil is formalised on a CAD software according to the framework in (Giovannini et al., 2015). The ***spaces*** represented in the model are about: the coil components, i.e. the tubes, the fins and the tube connections; the components geometrical features include numbers of tubes, tube length, number of holes in the fins, number of fins, fin thickness, tubes internal and external diameters, tube distance, number of circuits; also the flows of water and air and the air in the room have been represented. The ***shapes*** represented are about: customers' required performances such as the air temperature and humidity of a room have been modelled; the room is represented as a volume (orange in Fig. 4) in which the temperature, the humidity, the room usage (expressed as a power) and the air heat exchange coefficients are measured; the measurements about the customers' required performances are related to the heat exchange performance of the coil expressed as characterisations of the materials (i.e. aluminium and copper); also the water and air flows are modelled by means of *properties* such as the flow rates, the air pressure.
The represented behaviour is stationary therefore there is not a characterization of the ***time***.
The resulting model has 145 algebraic equations (i.e. *transformations*) that describe the mathematical relations between the measurements of performances and the coil components features.

5.2. *Knowledge retrieval*

To simulate the retrieval, the following data relative to the customers' requirements have been considered:

- A temperature from 21 to 25°C with a tolerance of ±1 °C;
- Two values for the humidity, 45% and 50%, with a tolerance of ±0.5%;
- Three values for the room usage, 2kW, 5kW and 10kW, with a tolerance of ±0.5kW.

As discussed above, this representation allows the knowledge retrieval with a comparison of numeric intervals and UOM. The three requirements represent three *shapes* of three measurements observed in the same *space*, i.e. the room. Therefore, the knowledge retrieval can be performed as follows: 1) find the CAD model that represents a volume with the same size and shape of the room; 2) check if there are three measurements in this volume and that they have the *shapes* corresponding with the customers' requirements.
When a corresponding model is retrieved other *properties* can be constrained. Concerning the water coil, the temperature of the water entering and leaving the coil has been constrained. The considered values are the following:

- The entering temperature goes from 6 to 9°C with a tolerance of ±1 °C;
- The leaving temperature goes from 10 to 13°C with a tolerance of ±1 °C.

After the discretisation of the intervals, the number of customers is equal to 480, i.e. 5 values for the room temperature, 2 for the humidity, 3 for the usage, 4 for the entering temperature and 4 for the leaving temperature.
In the developed prototype, the requirements are an input represented in XML, as in Fig. 7.

```xml
<?xml version="1.0"?>
<!--empty file to fill in with customer requirements and constraints.-->
<Requirements>
    <Attribute Name="X1" Type="real" Interval="continuous" Description="***" Tolerance="1" Error="0.01" Lower="10" Upper="13"/>
    <Attribute Name="X1" Type="real" Interval="continuous" Description="***" Tolerance="1" Error="0.01" Lower="6" Upper="9"/>
    <Attribute Name="X3" Type="real" Interval="continuous" Description="***" Tolerance="1" Error="0.01" Lower="21" Upper="25"/>
    <Attribute Name="X4" Type="real" Interval="discrete" Description="***" Tolerance="0.05" Error="0.001">
        <Value>0.45</Value>
        <Value>0.50</Value>
    </Attribute>
    <Attribute Name="X5" Type="real" Interval="discrete" Description="***" Tolerance="0.5" Error="0.01">
        <Value>2</Value>
        <Value>5</Value>
        <Value>10</Value>
    </Attribute>
    <Module>
        <Attribute Name="Y1" Type="real" Interval="discrete" Description="***" Tolerance="0.01" Error="0.001">
            <Value>20</Value>
            <Value>30</Value>
        </Attribute>
        <Attribute Name="Y2" Type="real" Interval="discrete" Description="***" Tolerance="0.01" Error="0.001">
            <Value>12</Value>
            <Value>15</Value>
            <Value>17</Value>
        </Attribute>
    </Module>
</Requirements>
```

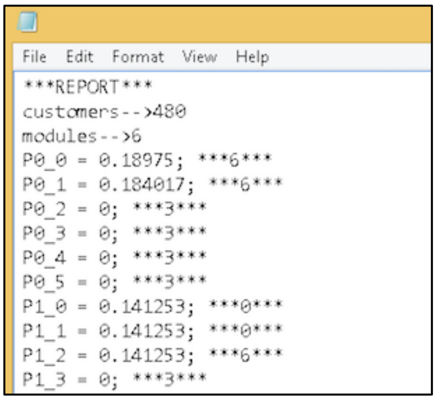
Fig. 7 - XML to represent the requirements.

5.3. *Knowledge reuse*

In this case study only a module has been considered. It is related to the fin holes position. Actually, this geometrical feature impacts the costs related to the punching tools and to the setup times for the punching process. Four values have been considered, i.e. two values for each side of a rectangle with the hole centres at its vertexes:

- The first side can be 12, 15 or 17 mm with a tolerance of ±0.01 °C;
- The second side can be 20 or 30mm with a tolerance of ±0.01 °C;

The number of generated MINLP models is equal to 2880. The execution time of the solver is limited to 10s in our case. Since the infeasibility result can require too much time to the solver, the problems for which the solver do not find a feasible solution in less than 10s are considered infeasible. The limit has been fixed at 10s after tests with random values: for almost all the launched models, the solver found the solution in less than 10s. The report of a feasible solution contains the value for each variable. Even if the model is strongly non-linear, the solver has been able to give great parts of the feasible solutions in few seconds. The objective function of each problem has been related to the quantity of aluminium and copper used for the coil manufacturing. The coefficient *CV* related to the variety costs has been considered equal to $3*10^3$. The values are selected only for demonstration purposes.
The aggregated report of the feasibility of all the tested combination is presented in Fig. 8: in the text file are reported the number of customers, the number of modules and, for each combination (e.g. P0_0), the value of the variable optimized and if the solver achieved a valid solution (0= global optimum; 6=local optimum; 1 and 3= infeasible or not determined).
The results show that the knowledge formalised did not allow to meet the requirements of all the 480 customers. For the 17,9% of the customer there is no available solution. Moreover, the 45% of the customer are fulfilled with two modules, the 12mmx20mm (24,8%) and the 17mmx30mm (20,2%). All the modules are employed in the designed PF because one of the constraints (the (11)) obliges to fulfil the customer when possible and in some cases there was only one alternative for one customer.



```
File Edit Format View Help
***REPORT***
customers-->480
modules-->6
P0_0 = 0.18975; ***6***
P0_1 = 0.184017; ***6***
P0_2 = 0; ***3***
P0_3 = 0; ***3***
P0_4 = 0; ***3***
P0_5 = 0; ***3***
P1_0 = 0.141253; ***0***
P1_1 = 0.141253; ***0***
P1_2 = 0.141253; ***6***
P1_3 = 0; ***3***
```

Fig. 8 - excerpt of the aggregated report of the feasibility of all the tested combination.

6. DISCUSSION

In this section, some issues related to the implementation of the proposed approach are discussed.
The knowledge representation based on the anti-logicist approach can be considered much more complex than the one using databases or ontologies. However, it is based on the connections of CAD files with mathematical models describing how the products' features impact the required performances. Both these kinds of models are required during the design stage of the PFs. Therefore, this approach requires only small modifications and an effort of connections of these existing models. The payback of this effort is the high level of reusability. In fact, an *experience* model allows to merge fundamental knowledge (e.g. hydraulics) with the customers' requirements and the geometrical features of the products' components.
Also the representation of the customers' requirements in such an accurate way can be a controversial point. As explained in the introduction, the requirements definition used in this paper is the one in (ISO/IEC, 2007). Moreover, a design stage cannot be based on ambiguous requirements, i.e. ambiguous objectives.
A possible obstacle to the implementation could be the computational complexity of the MINLP problems. Actually, the algebraic equations that should have been formalised are the ones used in every design stage. The higher complexity is

relative to the degree of freedom that a solver can manage. For the sake of simplicity, let us consider an equation system of 5 equations and 7 variables. To solve it, we should fix the values of two variables. Instead, the solver can provide a feasible solution without fixing any values. In summary, if this higher flexibility causes a not affordable computational cost, the degrees of freedom of the models can be limited.

7. CONCLUSIONS AND FUTURE WORKS

The main purpose of this paper was the application of an unambiguous knowledge representation to support the PF design stage. The knowledge representation approach permitted to capture the variety (functional and geometrical) of products' components and link them with the customers' required performances. The geometrical features of the components were linked to the mathematical model that describes the impacts of their values on the performances. The proposed implementation uses CAD software to formalise the knowledge about the geometrical features and to link them to the mathematical model. The knowledge is formalised following the framework proposed in (Giovannini et al., 2015). This allows retrieving and connecting knowledge without ambiguities, i.e. the models can be used even if the user is not of the same area of expertise of the modeller.

In the case study, we showed how a PF design could be performed for a *water coil*. Formalising the customer profiles, the product and the process features as *measurements* (e.g. temperatures, diameters, and features of the punching tools) permitted to build a knowledge model that consider simultaneously customers' requirements and components' features. For the retrieval and the connection of the knowledge needed to satisfy the customer requirements, no-human interpretation was necessary, because of the unambiguity of the knowledge models.

Even if the optimization was not the focus of this paper, a mechanism to show how to optimize the result has been included in the developed software tool. The main limit of this optimization approach is the number of MINLP problems to generate when the number modules and the related values is too high. An heuristic inspired to (Fujita & Yoshida, 2004; Khajavirad, Michalek, & Simpson, 2009) can represent an ideal solution. Another perspective can be to take into account the works in (Daaboul, Da Cunha, Bernard, & Laroche, 2011) to extend the knowledge representation to the process variety. Therefore, the PF and the related manufacturing processes should constrain simultaneously the MINLP models. Moreover, the possibility to characterise *unambiguously* more than one production system can be deployed to address the issues about the manufacturing interoperability (Panetto, Dassisti, & Tursi, 2012). Finally, future works can address an interface to make easier the anti-logicist knowledge representation, i.e. a hybridization with a formal logic approach preserving the unambiguity.

8. REFERENCES

Air-conditioning, Heating & Refrigeration Institute. (2001). AHRI Standard 410-2001 with Addenda 1, 2 and 3: Forced-Circulation Air-Cooling and Air-Heating Coils.

ASHRAE. (2009). *2009 ASHRAE Handbook: Fundamentals*. American Society of Heating, Refrigerating & Air-Conditioning Engineers, Incorporated.

ASHRAE. (2012). *2012 ASHRAE Handbook: heating, ventilating, and air-conditioning systems and equipment*. Atlanta, Ga.: American Society of Heating, Refrigerating & Air-Conditioning Engineers, Incorporated.

Atkinson, C., & Kiko, K. (2008). *A Detailed Comparison of UML and OWL* (Working Paper No. 1898). Retrieved from http://ub-madoc.bib.uni-mannheim.de/1898

Baader, F., Horrocks, I., & Sattler, U. (2008). Description logics. *Handbook of Knowledge Representation*, *3*, 135–179.

Brière-Côté, A., Rivest, L., & Desrochers, A. (2010). Adaptive generic product structure modelling for design reuse in engineer-to-order products. *Computers in Industry*, *61*(1), 53–65.

Daaboul, J., Da Cunha, C., Bernard, A., & Laroche, F. (2011). Design for mass customization: Product variety vs. process variety. *CIRP Annals - Manufacturing Technology*, *60*(1), 169–174. http://doi.org/10.1016/j.cirp.2011.03.093

D'Ambrosio, C., & Lodi, A. (2013). Mixed integer nonlinear programming tools: an updated practical overview. *Annals of Operations Research*, *204*(1), 301–320. http://doi.org/10.1007/s10479-012-1272-5

Fujita, K., & Yoshida, H. (2004). Product variety optimization simultaneously designing module combination and module attributes. *Concurrent Engineering*, *12*(2), 105–118.

Ge, J., Yang, C., Duan, T., & Chen, Y. (2013). Research on method of construction and configuration of product family based on ontology. In *2013 International Conference on Measurement, Information and Control (ICMIC)* (Vol. 02, pp. 1436–1440). http://doi.org/10.1109/MIC.2013.6758228

Giovannini, A., Aubry, A., Panetto, H., El Haouzi, H., Pierrel, L., & Dassisti, M. (2015). Anti-logicist framework for design-knowledge representation. *Annual Reviews in Control*, *39*, 144–157. http://doi.org/10.1016/j.arcontrol.2015.03.013

Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition*, *5*(2), 199–220.

ISO. (2004). ISO/TS 10303, 2004, STEP modules related to Product Data Management. Industrial automation systems and integration - Product data representation and exchange. Geneva.

ISO/IEC. (2007). *Systems and Software Engineering - Recommended Practice for Architectural Description of Software-Intensive Systems. ISO/IEC 42010:2007*. Geneva, Switzerland: International Organization for Standards (ISO)/International Electrotechnical Commission (IEC).

Jiao, J. (Roger), & Helander, M. G. (2006). Development of an electronic configure-to-order platform for customized product development. *Computers in Industry*, *57*(3), 231–244. http://doi.org/10.1016/j.compind.2005.12.001

Khajavirad, A., Michalek, J. J., & Simpson, T. W. (2009). An efficient decomposed multiobjective genetic algorithm for solving the joint product platform selection and product family design problem with generalized commonality. *Structural and Multidisciplinary Optimization*, *39*(2), 187–201.

Lim, S. C. J., Liu, Y., & Lee, W. B. (2011). A methodology for building a semantically annotated multi-faceted ontology for product family modelling. *Advanced Engineering Informatics*, *25*(2), 147–161. http://doi.org/10.1016/j.aei.2010.07.005

McGuinness, D. L., & Van Harmelen, F. (2004). OWL web ontology language overview. *W3C Recommendation*, *10*, 2004–03.

Meyer, M. H., & Lehnerd, A. P. (1997). *The power of product platforms*. Simon and Schuster.

Moon, S. K., Chang, X., Terpenny, J., Simpson, T. W., & Kumara, S. R. (2009). Towards a Knowledge Support System for Product Family Design. *Research in Mass Customization and Personalization: Strategies and Concepts-Applications and Cases*, 297.

Moon, S. K., Simpson, T. W., & Kumara, S. R. (2010). A methodology for knowledge discovery to support product family design. *Annals of Operations Research*, *174*(1), 201–218.

Muffatto, M., & Roveda, M. (2000). Developing product platforms:: analysis of the development process. *Technovation*, *20*(11), 617–630.

Nanda, J., Simpson, T. W., Kumara, S. R., & Shooter, S. B. (2006). A methodology for product family ontology development using formal concept analysis and web ontology language. *Journal of Computing and Information Science in Engineering*, *6*(2), 103–113.

Nanda, J., Thevenot, H. J., & Simpson, T. W. (2005). Product family design knowledge representation, integration, and reuse. In *Information Reuse and Integration, Conf, 2005. IRI-2005 IEEE International Conference on.* (pp. 32–37). IEEE. Retrieved from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1506445

Ni, J., Yi, J., & Ni, S. (2011). A Practical Development of Knowledge Management Model for Petrochemical Product Family. In *2011 International Conference on Information Management, Innovation Management and*

Industrial Engineering (ICIII) (Vol. 3, pp. 197–200). http://doi.org/10.1109/ICIII.2011.330

Nomaguchi, Y., Taguchi, T., & Fujita, K. (2006). Knowledge model for managing product variety and its reflective design process. In *ASME 2006 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference* (pp. 327–340). American Society of Mechanical Engineers. Retrieved from http://proceedings.asmedigitalcollection.asme.org/proceeding.aspx?articleid=1588086

Ong, S. K., Xu, Q. L., & Nee, A. Y. C. (2006). Design Reuse Methodology for Product Family Design. *CIRP Annals - Manufacturing Technology*, *55*(1), 161–164. http://doi.org/10.1016/S0007-8506(07)60389-8

Panetto, H., Dassisti, M., & Tursi, A. (2012). ONTO-PDM: Product-driven ONTOlogy for Product Data Management interoperability within manufacturing process environment. *Advanced Engineering Informatics*, *26*(2), 334–348. http://doi.org/10.1016/j.aei.2011.12.002

Seo, K.-K., & Ahn, B. (2009). A Product Family Knowledge Management Framework using the Semantic Web., *11*(3), 105–112.

Siddique, Z., & Rosen, D. W. (2000). Product family configuration reasoning using discrete design spaces. *AMSE DETC, Baltimore, MD*. Retrieved from http://www.srl.gatech.edu/publications/2000/dr.siddique.product.family.configuration.2000.pdf

Tseng, H.-E., Chang, C.-C., & Chang, S.-H. (2005). Applying case-based reasoning for product configuration in mass customization environments. *Expert Systems with Applications*, *29*(4), 913–925. http://doi.org/10.1016/j.eswa.2005.06.026

Youliang Huang, Ng, W. K., Liu, H., Lu, W., Song, B., & Li, X. (2007). Semantic Modeling and Extraction for Cross-family Product Configuration. In *2007 International Conference on Service Systems and Service Management* (pp. 1–6). http://doi.org/10.1109/ICSSSM.2007.4280186

Zhang, J., Wang, Q., Zhong, Y., & Wan, L. (2005). Configuration-oriented product modelling and knowledge management for made-to-order manufacturing enterprises. *The International Journal of Advanced Manufacturing Technology*, *25*(1), 41–52. http://doi.org/10.1007/s00170-003-1871-z

Zha, X. F., & Du, H. (2006a). Knowledge-intensive collaborative design modeling and support: Part II: System implementation and application. *Computers in Industry*, *57*(1), 56–71. http://doi.org/10.1016/j.compind.2005.04.006

Zha, X. F., & Du, H. (2006b). Knowledge-intensive collaborative design modeling and support: part I: Review, distributed models and framework. *Computers in Industry*, *57*(1), 39–55.

Zha, X. F., & Sriram, R. D. (2006). Platform-based product design and development: A knowledge-intensive support approach. *Knowledge-Based Systems*, *19*(7), 524–543.

Zha, X. F., Sriram, R. D., & Lu, W. F. (2004). Evaluation and selection in product design for mass customization: A knowledge decision support approach. *AI EDAM: Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, *18*(01), 87–109.