

METHOD FOR INTEGRATED DESIGN USING A KNOWLEDGE FORMALIZATION

Alexandre Thibault, Ali Siadat, Régis Bigot, Patrick Martin
ENSAM, FRANCE
{alexandre.thibault, ali.siadat, regis.bigot, patrick.martin}@metz.ensam.fr

This paper presents a proposal for a method aimed to guide the designer within a context of product-process integration in forging design. The objective is not to generate but to assist the design of a product and its associated process planning. This method uses a decision support system called Ontoforge which is based on ontology for representing knowledge.

1. INTRODUCTION

The goal of this project is to conceive methods and computer tools to aid the design of a forged product by taking into account the competences and resources of the enterprise and the constraints related on the manufacturing process. For that, it is necessary to know not only the process knowledge but also the information concerning the resources (material, financial, human or of another nature), the product description during the design cycle and the production data.

The design aid computer tool called *Ontoforge* will be applicable to the forging domain, restricted to hot stamping, in order not to disperse into a too broad domain. However, this area could be widened in the future.

In the first section, the context of the project will be presented, before an example illustrates the kind of supported problems. In the third part are explained the mechanisms that intervene to supply the response to these problems. We discuss finally the benefits and drawbacks of the presented method.

2. SITUATION

The system proposed renders a service to the designer and/or to the manufacturer. It acts on the product data, the manufacturing process data and the resources data. Its aim is to make checks and proposals in respect to process knowledge and available resources.

2.1 Inputs

We place ourselves at the manufacturer point of view. The designer has already made some preliminary choices on the part design. That's why the design aid computer tool is supposed to intervene when the technical requirements have reached a certain level of precision: an incomplete specification of the part with mechanical functional characteristics already quantitatively defined. This specification is available for the geometry, the material or for other kind of data. The functions of the product won't be reassessed, only technical specifications could be changed in a certain range of value eventually defined by the designer. The precision level of this incomplete specification remains to be defined more precisely but we can already affirm that all the data necessary to do the morphological classification are required (see 3.1). The computer tool could also take into account a definition of the resources that are available in the manufacturer workshop (it can be the machines or the staff or any other type of resource).

2.2 Outputs

The goal is not to generate the manufacturing process like (Kim, 2000), but to assist the design of this one. In other words, the tool can make checks and proposals; the user has the possibility to consider or not those remarks. The checks and proposals are realized with the process knowledge and the available information that are filled in. Several sources are used to make it: the morphological classification (ADETIEF, 1974), the indicators presented by J.M. Arenas (Arenas, 2004), the availability of machines in the workshop, or some documents like norms.

3. A SIMPLE EXAMPLE

3.1 Introduction

The functioning of the application can be described with the following example: we consider a part whose draft geometry has been defined. First of all, in order to suggest some recommendations on the manufacturing process, the part is classified in a part family that is listed in (ADETIEF, 1974), this classification is oriented on the morphological point of view. A final processing is lead to know whether or not this part is manufacturable with the resources at disposition (the machines, staff...).

3.2 Automatic classification in a part family

A part is classified in a family according to the parameters enumerated on table 1.

Table 1 – Classification parameters

Name of the parameter	Type
Existence of a central boss	Boolean
Bent part	Boolean
Existence of an opening	Boolean
Part with slender shapes	Boolean
Existence of a plate	Boolean
Revolution part	Boolean

Complexity ratio	Real
Diameter/height ratio	Real
Diameter/Height of the boss (if exists) ratio	Real
Rectangularity ratio	Real
Section variation ratio	Real

After filling the parameters in, the recommendations are presented to the user, or if the manufacturing process is already partially defined, checks are realized that may cause alerts.

3.4 Automatic classification in relation to the available resources

This classification is realized with some parameters like material, size, etc... Two steps are required:

- 1) The user indicates the resources of the workshop, for example:
Press1 {Max length = 300 mm; Max power=...; availability: ok}
Press2 {Max length=400 mm; Max power =...; availability: out of order}
 The workshop is also able to manufacture only the parts that have max length less than 300 mm.
- 2) The user fills in the parameters of the part:
 If the parameter max length is equal to 305, the part would be classified in the category « parts not manufacturable by my workshop ». A proposal might be suggested: set the parameter max length at 300 mm (remember that we are in a design phase where the geometry isn't definitely set).

4. HOW CAN IT WORK

After explaining what is an ontology and what can be expressed with, we'll see how we can formalize the knowledge for our needs.

4.1 What is an ontology?

An ontology enables:

- To describe a knowledge field (ex: to belong to a certain part family, the part must have certain characteristics and this family implies some recommendations about the manufacturing process).
- To store a data structure and also the data (ex: in the ontology can be described a class named « Machine » that have for attributes: "name", "power", "availability", etc. and the user can fill in the fields with the corresponding data for his workshop).
- To define a common vocabulary for the field (in the form term/definition).

We must observe that the reasoning applied on the ontology rests on the open world assumption, i.e. all that is not asserted isn't assumed to be false. It implies that for certain instances of classes, the system cannot determine where to classify (we'll see it later).

4.2 Automatic classification

We'll illustrate the classification with the previous example (determination of the manufacturability of the part by the workshop). The ontology is expressed in OWL (Ontology Web Language) (Antoniou, 2004) with the OWL *Protégé* plug-in. Figure 1 presents the taxonomical hierarchy (based on the is-a relation) and the relations between the classes.

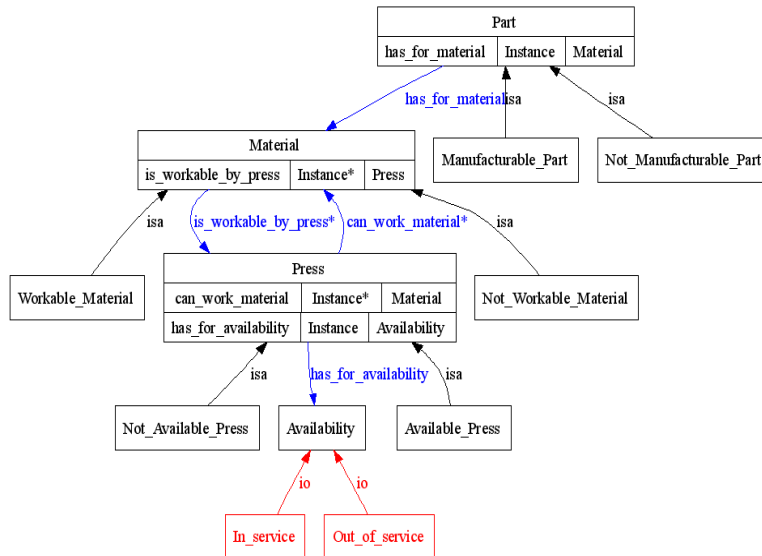


Figure 1 – Relations between the classes

After having defined the taxonomy as well as the relations between the classes, it is possible to assert some restrictions on those relations (and then we can really talk about “knowledge”). These restrictions are used to define necessary and/or necessary and sufficient conditions to be instance of a class.

The Material, Press and Part classes are primitive classes, i.e. they have only necessary conditions. The other classes are defined classes: there exists for each of those ones at least one necessary and sufficient condition. Thanks to that, the classification in those classes is conceivable. On table 2 are presented the necessary and sufficient conditions for the defined classes.

Table 2 – Necessary and sufficient conditions on the defined classes

Workable_Material	<input type="checkbox"/> Material \Leftarrow is_workable_by_press Available_Press
Not_Available_Press	<input type="checkbox"/> Press \int has_for_availability {Out_of_service}
Available_Press	<input type="checkbox"/> Press \int has_for_availability {In_service}

Manufacturable_Part	<input type="checkbox"/> Part <input checked="" type="checkbox"/> has_for_material Workable_Material
---------------------	---

Explanation for Available_Press:

Press: means that an instance of the Available_Press class must be also an instance of the Press class.

has_for_availability {In_service}: means that if an instance of the Available_Press class has a relation has_for_availability then this relation is necessarily associated to the instance "In_service".

If an instance verifies the two previous conditions then it is an instance of the Available_Press class.

It is important to remark that the properties has_for_availability and has_for_material are functional i.e. they accept a single instance for the range (for example a unique instance of the class Availability concerning the property has_for_availability). On the contrary, the property is_workable_by and its inverse property are not (actually a press can work with several materials and a material is workable by several presses).

This last remark has some issues in relation to the open world assumption; considering a class Not_Workable_Material defined as follows: an individual is an instance of the Not_Workable_Material if and only if it is an instance of the Material class and all the relations of type is_workable_by are pointing on instances of the Not_Available_Press class. But because that property isn't functional, it cannot be determined how many relations of this type really exist. Because it is assumed that what is not asserted may be true, it cannot be inferred that all the relations of this instances complies with the restriction. Then, it cannot be classified in this class (in the case of a functional property, there is no problem).

Example:

The instance "Steel" have only a single relation with « Press1 » which has_for_availability "Out_of_service". It cannot be inferred that this instance is of the Not_Workable_Material class; the contrary also cannot be inferred.

However, it is possible that some instances are classified in the Workable_Material class. It suffices that the instance has a relation is_workable_by with an instance of the Available_Press class.

Figure 2 presents a summarization of the situation.

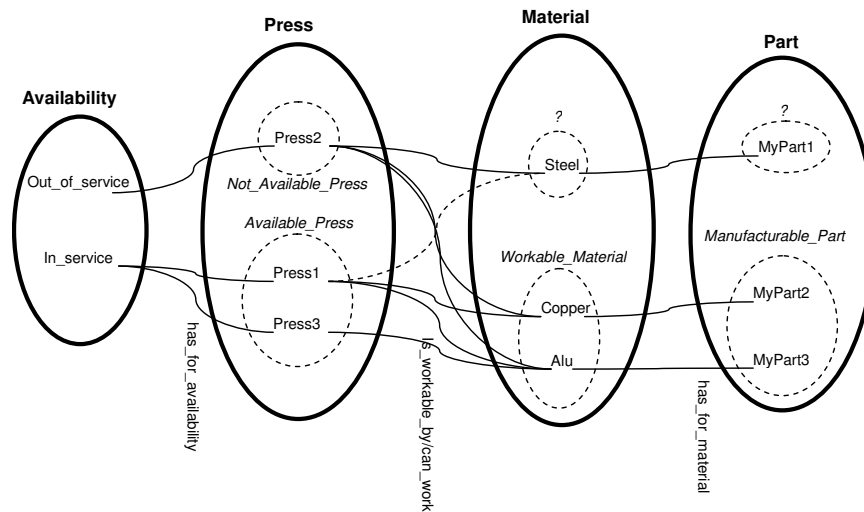


Figure 2 – Illustration of the classification

The relation between Steel and Press1 isn't asserted but according to the open world assumption, it can exist, then it cannot be concluded that Steel is an instance of the Workable_Material class. MyPart1 has_for_material Steel, it cannot either be concluded something on its type apart that it's an instance of the Part class.

Fortunately the classes that are particularly interesting, that show the availability of the press, the workability of the material, and the manufacturability of the part are fully exploitable.

5. ADVANTAGES OF THIS METHOD

The information extracted from the above processing is the result of a classification. This result would of course be attained by a simple algorithm which would process the data. The interest for the use of an ontology is that it is not only consistent but also upward compatible and easily modifiable. Changing an algorithm implies coding, compiling, updating whereas the change of an ontology can easily be made with an ontology editor like *Protégé*. For instance, if some new parameters must be taken into account for a check on the part, we need just to add the appropriate structures in the ontology.

An ontology would permit to avoid the juxtaposition of several systems such as a xml database for the field terminology, an object database for representing the information concerning the manufacturing process, the product or the resources, and an expert system with a rule base for the field knowledge. Actually, an ontology can be used as an object database, a class diagram in UML can be transcribed in OWL (Falkovych, 2003; Colomb, 2004).

6. INCONVENIENCES

6.1 Database performance

Few similar projects have been carried out until now; we can mention the Kasimir project (D'Aquin, 2004) or also On-To-Knowledge and other different projects (Fensel, 2004; Davies, 2003; Sugumaran, 2002), then we don't know a priori what are the behaviour and performance of a database implemented on an OWL ontology. We can note however that there exists viable solutions for XML database and OWL is based on XML. Moreover an OWL ontology can be mapped in a relational database, so the problem can be less serious than we could think. In addition, we can cite a method for benchmarking Semantic Web knowledge base systems with respect to use in large OWL applications (Guo, 2005).

6.2 Some blocking points...

Among the inconveniences, we can cite some limits of this recent technology: there is especially a blocking point, solvable but relatively embarrassing: the use of literals.

Until now, there is no way to use literals in restrictions. For instance, the following condition isn't expressible with OWL:

Relation: has_for_mass between an instance of the Part class and a numeric value

Condition: If x is-a Manufacturable_Part then x has_for_mass y with $y < 150$

However, this problem can be solved by creating interval classes (D'Aquin, 2004), for example: Mass_less_than_150, Mass_greater_than_150. Hence, if the user inputs the value 78 on the interface, the application would create an instance of the Mass_less_than_150 class that would be classifiable.

On the other hand, it is possible to express some knowledge with the Semantic Web Rule Language (Horrocks, 2005), that could accept the use of literals but this alternative hasn't been taken under consideration because until now there isn't any reasoner that can exploit the rules expressed with this language.

7. CONCLUSION AND PROSPECTS

We have seen on a simple and concrete example how to use an ontology for storing and exploiting knowledge. This use is realized with the help of an elementary processing: classification. We must be aware of some subtleties such as the open world assumption or the use of literals in order to control this classification. In addition, reflections must be made about the idea of using an OWL ontology like an object database.

Until now, the developments concerned:

- The creation of several demonstration ontologies to illustrate classification
- The creation of an ontology gathering the classes of a knowledge field (forging in our case) with their designation terms on several languages, and their definition (AFNOR, 1984)

- The integration in the previous ontology of the classes allowing the representation of a manufacturing process

The follow-up of this work consists essentially in developing a plug-in program in *Protégé*, that would permit to exploit an ontology according to our needs: creating instances of parts, defining the resources of the workshop, realizing checks on the parts, proposing some recommendations in relation to the manufacturing process. *Protégé* would remain the means to maintain and develop the ontology.

8. ACKNOWLEDGMENTS

This project is carried out in the common laboratory ENSAM-CETIM within the framework of a PhD Thesis financed by the CETIM organization. We thanks particularly Mr. Patrick Marchand and Mr. Pierre Ravassard for their active collaboration.

4. REFERENCES

1. ADETIEF. "Classification morphologique des pièces estampées", 1974.
2. AFNOR. "Forge par estampage et matriçage", NF E 82-000, 1984.
3. Antoniou G, Van Harmelen F. "A Semantic Web Primer", The MIT Press, ISBN 0-262-01210-3, 2004.
4. Arenas JM, Sebastian MA, Gonzales C. "Parametric classification based on technico-economic approaches to optimize the design and manufacturing of hot forged parts", *Journal of Materials Processing Technology*, N°150, 2004, pp. 223-233.
5. Colomb B, Hart L, Chang D, Kendall E. "OWL Full and UML 2.0 Compared", 2004, available at: www.itee.uq.edu.au/~colomb/Papers/UML-OWLont04.03.01.pdf.
6. D'Aquin M, Brachais S, Lieber J, Napoli A. "KASIMIR: Gestion de connaissances décisionnelles en cancérologie", *Proceedings of 2ème colloque du groupe de travail C2EI*, december 2004.
7. Davies J, Fensel D, Van Harmelen F. "Towards the semantic web", John Wiley & sons, Ltd, ISBN 0-470-84867-7, 2003.
8. Falkovych K, Sabou M, Stuckenschmidt H. "UML for the Semantic Web: Transformation-Based Approaches", in *Knowledge Transformation for the Semantic Web. Frontiers in Artificial Intelligence and Applications*, IOS Press, vol. 95, 2003, pp. 92-106.
9. Fensel Dieter. "Ontologies: a silver bullet for knowledge management and electronic commerce", Springer-Verlag, ISBN 3-540-00302-9, 2004.
10. Guo Y, Pan Z, Heflin J. "LUBM: A benchmark for OWL knowledge base systems", *Journal of web semantics*, article in press, 2005.
11. Horrocks I, Patel-Schneider PF, Bechhofer S, Tsarkov D. "OWL rules: A proposal and prototype implementation", *Journal of web semantics*, N°3, 2005, pp. 23-40.
12. Kim DY, Park JJ. "Development of an expert system for the process design of axisymmetric hot steel forging", *Journal of Materials Processing Technology*, N°101, 2000, pp. 223-230.
13. Sugumaran V, Storey VC. "Ontologies for conceptual modelling: their creation, use, and management", *Data & Knowledge Engineering*, N°42, 2002, pp. 251-271.