A conceptual model for the implementation of IKOES (an Inter-Knowledge Objects Exchange System) in automotive industry

Ludovic Louis-Sidney, Vincent Cheutet, Samir Lamouri, Olivier Puron, Antoine Mezza

A conceptual model for the implementation of an Inter-Knowledge Objects Exchange System (IKOES) in automotive industry

Ludovic Louis-Sidney[1,2], Vincent Cheutet[1], Samir Lamouri[3], Olivier Puron[2], Antoine Mezza[2]

[1]*LISMMA, EA 2336, Saint-Ouen, France*
[2]*Renault, Powertrain Technology and Design Department, Rueil-Malmaison, France*
[3]*Arts et Métiers ParisTech, LCPI, EA3927, Paris, France*

Corresponding author email: ludovic.louis-sidney@renault.com.

Abstract

The level of efficiency achieved in the diffusion of knowledge within an organisation is acknowledged to be a competitive advantage. As a result, various means have emerged to help share this knowledge. Among the various existing solutions, the use of parameterised files consistently encapsulating data of a refined level of granularity (e.g. figures, words, etc.) is a well-known practice. However, these files – which we will call Parameterised Knowledge Objects (PKOs) – often exhibit redundancy. This has led to a need for mapping, a fastidious work with no added value, which slows down the product redesign process. The goal of this article is to propose a conceptual model for the implementation of an automated tool to manage exchanges between these PKOs. The implementable nature of the model was validated by the development of a demonstrator, tested on an application provided by our industrial partner – the Renault Powertrain Technology Design Department.

Keywords: Parameterised knowledge objects, Redesign process, Data structuring, Data exchange.

1. Introduction

In the automotive industry, the success of a design project is mainly based upon the diffusion of knowledge - not only between the stakeholders contributing to the current design project, but also between stakeholders contributing to different projects. As a consequence, ***knowledge management systems are born to provide required knowledge at the right time, to the right agent, in the required form*** (Bernard and Tichkiewitch, 2008).

In this context, many approaches exist to manage and optimise the diffusion of knowledge and information during a product development process: ***personalisation-oriented*** approaches on one hand, and ***codification-oriented*** approaches on the other (MacMahon *et al.*, 2004). Personalisation-oriented approaches focus on the setup of communities of practices (Amin and Roberts, 2008). Stakeholders in the design process are able to share their knowledge by setting up organisational structures (sector meetings, project platforms, websites for sharing data, etc). Codification-oriented approaches focus on the formal expression of knowledge via models that are sufficiently complete and consistent for subsequent use. There are many types of solutions. They range from solutions aiming solely to take over the knowledge expressed by an individual (*e.g.* a book, an article) to those intended to give significant assistance to the individual in his task, or even replace him in carrying it out - *e.g.* knowledge-base engineering applications using Artificial Intelligence technology (Moka, 2000). The increase in data volumes in engineering (Triantaphyllou *et al.*, 2002) suggest that the solutions preferred today are those backed by digital technology.

In this article, we examine a digital solution which is halfway between those described above, and is simple to implement. It relies on the use of parameterised files which are frequently used in redesign processes - design processes encompassing routine and innovative design, as described by Gero (1990). Such solutions are not new; but to our knowledge, there is no dedicated name for them in a knowledge-centric view. Here; we will call them ***Parameterised Knowledge Objects*** (PKOs). This terminology is discussed in section 2.1.

Based on the state of the art presented in section 2, we point out the following facts for this study:

- Fact 1 (discussed in section 2.1): PKOs are widely used and encapsulate part of the company knowledge. This knowledge is not static but dynamic in nature. It is completed during the product development process itself.
- Fact 2 (discussed in section 2.2): KMTs (Knowledge Management Tools) and PDM (Product Data Management) only manage files by using metadata. The user is therefore totally responsible for their content: project stakeholders must open the file in order to access a required piece of knowledge. The level of granularity of these tools is not refined enough for a good use of these PKOs in a design project.
- Fact 3 (discussed in section 2.3): Knowledge expressed in PKO is redundant and few approaches exist to make them interact.

Based on these three facts, our motivation for this paper is to develop an **information system** enabling the **automated exchange of parameters between PKOs**. This system will save users from the fastidious, error-prone and time-consuming work of transferring information and knowledge between these objects. We propose a conceptual model, *i.e.* a model making it possible for its authors to bridge the gap between their mental representations of the solution their proposed and the solution itself (Ben-Ari and Tzippora, 2006). This conceptual model is characterised by the integration of a semantics-oriented static representation of all the parameters encapsulated in the PKOs, and of a dynamic representation of the exchange of these parameters in a heterogeneous project environment – since there exist numerous applications to process these files. The model aims to help with the implementation of ***IKOES***: an Inter-Knowledge Objects Exchange System.

Section 2 details the state-of-the-art summarized above. Section 3 is dedicated to presenting the conceptual model. The implementable nature of this model is validated in Section 4 through the presentation of a demonstrator tested on an application provided by our industrial partner, the Renault Powertrain Technology Design Department.

2. Knowledge objects and support tools

2.1. Knowledge and knowledge objects

The work of various researchers has dealt with the characterisation of knowledge in a conceptual approach (Tsuchiya, 1993; Ermine *et al*., 1996; Baizet, 2004). A consensus seems to have been reached, on the fact that knowledge is reflected by adding a semantics level and a contextualisation level to tangible or intangible data. The formal expression of knowledge therefore resides in a representation of the whole – data, semantics and contextualisation – that is as faithful as possible. To convey this representation, a support medium is required. Knowledge objects are one such medium. The notion of knowledge objects has already been developed in the literature. Baizet (2004), based on work by Prudhomme *et al*. (2001) describes them as artefacts that enable an individual to construct his own knowledge. In this work, a knowledge object is defined in a relatively generic manner. It can refer to a book, a calculation report, an equation, a simulation method, etc. More recently, following the same approach, Cacciatori (2008) has developed the notion of "memory objects". The author makes a distinction between two types of memory objects. On one hand, objects of a *static* nature containing "recommendations", as well as relatively fixed representations of knowledge that remain unchanged from project to project. Above all, their goal is to enable the individual to construct his knowledge without the systematic assistance of the people who hold this knowledge. On the other hand, objects of a *reconfigurable* nature are "constructed through the recombination of relatively immutable components". They do not replace dialogue between project stakeholders but are intended to provide them with a well-defined context. They "enable firms to build on experience, while maintaining the flexibility necessary to adapt to the specificities of each project." The author thus demonstrates how knowledge of a strategic nature (*e.g.* how to make a bid in a private finance initiative project) can be formally expressed and used by means of an Excel file.

In this article, we focus on these reconfigurable memory objects. However, Cacciatori (2008) specifies that the terms used ("memory", "remembering") have connotations. They are intended to represent the fact that the author is focussing on the processes of knowledge storage and extraction, not on those of creation and modification. On the other hand, we are interested in this dynamic aspect of knowledge. We will therefore use the term "knowledge object" which tends to represent the same concept from a different perspective. In light of the generic nature of this concept, we will make it more specific by using the term "parameterised". Parameterised Knowledge Objects (PKOs) are therefore defined as digital files that enable a predetermined set of data (or parameters) to be specified, in order to produce a result. The predetermined set of data and the expected result, together with the relationship between this data and this result constitute a representation of a certain part of the company's knowledge. These PKOs are used to specify the data and to obtain the result, in a given project, in the context of a product redesign process (as defined in Section 1).

2.2. Knowledge Management Tools and Product Data Management Tools

PKOs encapsulate part of the company's knowledge in a distributed manner. Their flexible nature generates possibilities for the dissemination of this knowledge, both inside and outside the company. Tools are therefore required in order to maintain control over this asset in the context of product design, namely KMTs and PDM.

KMTs are dedicated to managing a company's knowledge assets. In view of the rapid development of this discipline since the Nineties (Grundstein, 2002), a particularly wide variety of technologies has been adopted. Nonetheless, authors who have carried out analyses of these tools all agree that they notably consist of databases in which the documents evidencing company knowledge can be stored and shared (Ngai and Chan, 2005; Joo and Lee, 2009; Vaccaro *et al.*, 2010). Joo and Lee (2009) endeavour to describe the causes of dissatisfaction of the users of such tools. One element of dissatisfaction is the lack of integration in terms of accessibility to the documents' internal data, which constitutes an acknowledged limit to these tools. They propose the use of Semantic Web technology to circumvent this limit, but they accept that present tools do not enable the automatic extraction of unstructured data from sources such as spreadsheet files.

PDM, originally dedicated solely to the management of CAD files, is now being used to manage all product data and related information (Hsu and Hwang, 2004; CIMdata, 2011). Nowadays, these tools appear to have the "ability to capture and manage enterprise intellectual assets throughout the product definition lifecycle" (CIMdata, 2011). It is therefore positions itself as a possible competitor for knowledge management tools. However, an analysis conducted by Pikosz and Malmqvist (1996) refers to the same limitations as those of knowledge management tools in terms of integration. These tools "can only see metadata, which is data about data (*e.g.* document ID, version, creator, status, etc.), whereas the document is treated as a black box". Our bibliographic study has not revealed any sources more recent than the work of Pikosz and Malmqvist (1996) mentioning these shortcomings of PDM. However, although their analysis is not very recent, our use of tools such as Windchill (PTC), Enovia (Dassault Systèmes) and TeamCenter (Siemens) has confirmed these limitations.

KMTs and PDM, although they are necessary to the management of PKOs, do not propose mechanisms that address optimisation of the use of the content of these files.

2.3. Working towards an inter-knowledge objects exchange system

The knowledge encapsulated in tangible objects such as PKOs is necessarily incomplete, heterogeneous and redundant (Grundstein et al., 2003). The incompleteness is due to their synthetic objective and their limited storage capacity. As a PKO often act as boundary objects between various professional sectors (Cacciatori, 2008), this leads to heterogeneity. And because of these two facts, redundancy appears when a same professional sector is involved with different PKOs. In this context, the more PKOs there are in the design project environment, the more it will be time-consuming and tedious to manually transfer information and knowledge between them. Nevertheless, there is currently no global approach to enable automatic sharing of knowledge expressed in these files. The scientific field of Knowledge-Based Engineering (KBE), which "concerns computerisation of processes associated with industrial products – usually routine design" (Moka, 2000), is appropriate to situate this approach. However, it should differ from approaches which aim to design actual KBE applications. Indeed, these application, developed according to the MOKA methodology (Skarka, 2007), the Common KADS methodology (Sutton and Patkar, 2009), or more *ad hoc* methodologies (Chapman and Pinfold, 2001; Yoshioka et al., 2004), aim to solve very specific problems and broadly lead to developing rich ontological models. Creating and maintain these models required the appropriation of a significant level of expertise, which does not match with the initial context of a need for simplification, which surrounds PKOs.

Moreover, literature mentions few tools whose goal is to share knowledge between PKOs. This type of approach appears to be taking shape in the work of Badin *et al*. (2010) in the context of the ADN (Alliance des Données Numériques) project. The authors propose a product model to interrelate files relating to CAD models and parameterised simulation models. This model should enable users to capitalise on parameters, mathematical relations, rules, limit conditions and discrete values according to several configurations of parameterised models. The implementation of this model is referred to but not in great detail. In addition, although the model is particularly complex, no consideration has been given to the stage at which the knowledge is formally expressed within the tool (this stage is viewed as being completed).

3. Proposed conceptual model

Our work positions itself on the grounds described in section 2.3 but in a broader perspective in terms of the PKO formats considered (Excel files in particular), and a more restricted perspective in terms of the complexity of the underlying model - it must be possible for information to be fed into the model using quickly accessible expertise. These requirements relate to the industrial context for the implementation of our work, in close collaboration with a major manufacturer.

In this article, we propose a CIM (Computational Independent Model)-type of conceptual model following an MDA (Model Driven Architecture) approach (OMGa, 2003). The purpose of this model is to describe a design solution from a design point of view. Specifically, it aims to be translated into a PIM (Platform Independent Model) and then into a PSM (Platform Specific Model), in order to allow the implementation of IKOES. Throughout the description of the proposed conceptual model we will use the product example of a screwed assembly, to make this description more concrete. This example is based on the parameters described in Figure 1 and is reused in section 4 (validation section).

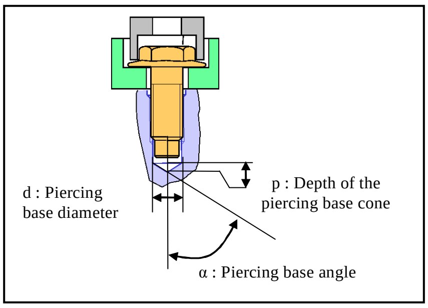

Figure 1. Description of three parameters of a screwed assembly

3.1. Conceptual model architecture

To make the automated exchange of parameters between PKOs possible, our conceptual model is based on four pillars (*cf* Figure 2) which enable:

- the representation of the data (***data*** pillar),
- the structuring of the data (***data structure*** pillar),
- the representation of the activities using this data (***activities*** pillar),
- the sequentialisation of these activities (***process*** pillar).

The data and data structure pillars develop a static focus to represent the parameters as well as the relations between parameters from a semantic point of view. The activity and process pillars develop a dynamic focus to represent exchanges of parameters between one PKO and another. This dynamic focus is essential, since it underlies the added value of the IKOES information system, which we wish to implement. It represents the actual work of cross-matching and copying parameters, carried out by PKO users. Relatively to this focus, the static focus is necessary.

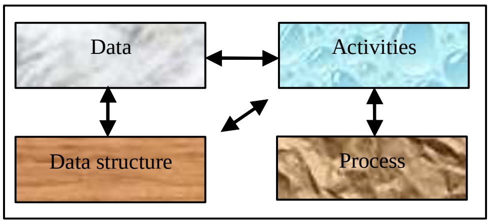

Figure 2. Conceptual model architecture

We will describe each of the pillars of this architecture in the following sections. In terms of completeness, the model has been evaluated, regarding whether it contained the basic bricks to enable exchanges. This evaluation is materialised by the development of a demonstrator tested on an industrial application presented in Section 4.

3.2. Data structure

The UML data structure model (OMGb, 2009) is shown in Figure 3. Its aim is to enable the specification of parameters by a user in such a way as to describe its semantics. It reflects a compromise between the rich ontological models which can be found in a KBE application, and classic descriptive methods such as classifications.

The *datum* class represents the parameter to be shared without any particular reference to semantics. It enables the qualitative nature of the characteristics of the parameter to be described (*e.g.* its type: “numerical” or “text”). The *datum definition* class represents the first level of semantic description associated with a datum (*e.g.* the parameter is a “diameter”). A *datum* can only be associated with one single *datum definition*. The aim of this constraint is to avoid the possibility of a misunderstanding. In this article we consider the set [*datum definition* / *datum*] as one piece of information. In order to complete this first level of semantics, a piece of information may be contextualised. To achieve this contextualisation, we use the principle of a faceted classification (Ranganathan., 1950). We consider three contextualisation facets: organisational, functional and structural. These facets are evidenced by means of context objects organised in a hierarchical form. Hierarchical structures are defined beforehand and cannot be modified by the user.

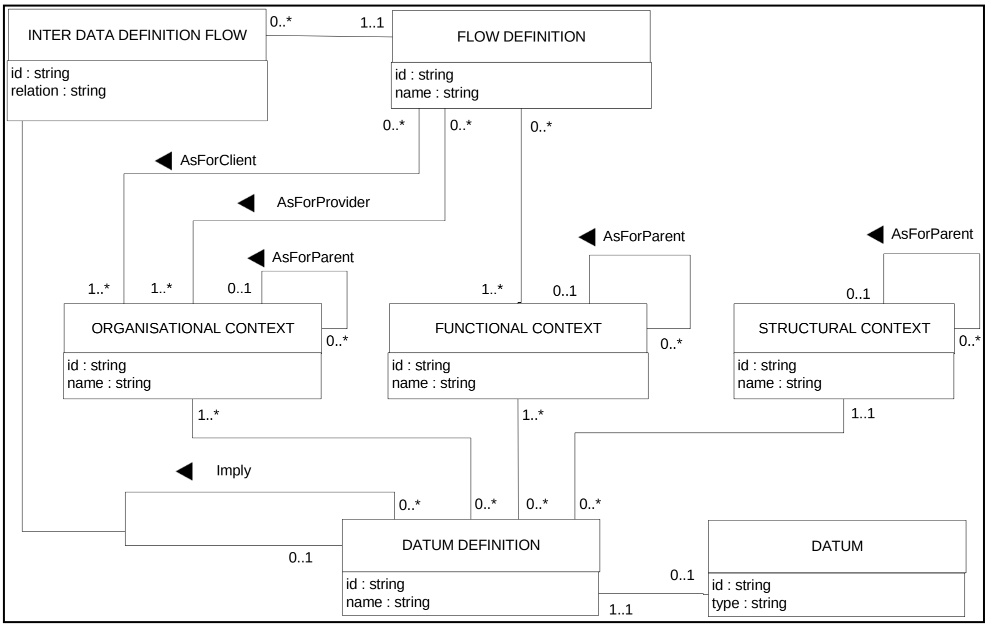

Figure 3. Data structure model

The *organisational context* enables the PKOs to which the information belongs to be specified (*e.g.*: a CAD screwed assembly model). This localisation viewpoint relates to the fact that our work takes place in a heterogeneous environment (*cf.* Section 1). The *functional context* enables information to be channelled into the professional sector environment where the information is used and generally refers to a group of project stakeholders sharing a common activity (*e.g.*: the static testing sector). The *structural context* is intended to specify which product element this information belongs to (*e.g.*: the cylinder head). It is possible to specify several *context* classes of the same nature for a piece of information, except for the *structural context*. The restriction is justified by the need to maintain control over the specification of synonymous parameters by partly curbing the possibilities for contextualisation. Before defining a parameter, the user only needs to check that it has not already been defined for the product element involved - the choice of the *structural context* as the exclusive context is linked to our product data-oriented working environment in the context of a redesign process. We consider the set [*contexts* [*datum definition* / *datum*]] as being knowledge representation (*cf* Section 2.1). Knowledge representation therefore consists of n *organisational contexts*, m *functional contexts*, one *structural context*, one *datum definition* and one *datum* with n ≥ 1 and m ≥ 1.

The flow objects enable relations between the parameters to be defined. These relations apply to the values of the parameters which are to be specified over the course of the project. They are therefore not known beforehand. As is the case for data objects, flow objects are defined by the user. A compromise has therefore been made in this model, between the possibility of defining complex relations, and the user's ability to model and subsequently manage these relations. Thus the *inter-data definition flow* class corresponds to the representation of a one-direction relation with only one resulting *datum definition* (parameter semantics). The simple nature and analysis possibilities of flows defined in this manner (*i.e.* simple association rules) have been the subject of several studies (Chen *et al.*, 2002; D'Enza *et al.*, 2008). This relation can be mathematical or rule-based (if structure). It will then become the subject of formal interpretation by a calculation tool which can execute this type of relation. As for data, a level of semantics and contextualisation must be added to a flows. The *flow definition* class represents the semantics level (*e.g.* "physical law"). It must be contextualised. The *organisational context class* allows us to specify which supplier PKOs and which client PKOs can be brought into play in the inter-parameter relation. All the relation entry parameters must be available for a given supplier PKO. The *functional context* class provides information on the professional sector environment in which this flow is defined. Structural contextualisation is not necessary since the information ([*datum definition* / *datum*]) is exclusively classified according to this context.

Figure 4 gives an example of an instance for flow-type knowledge representation. The dotted lines in the hierarchical structures express the fact that they are not fully represented for space and confidentiality reasons. It should be noted that this example highlights the fact that there is no notion of implicit inheritance relative to the hierarchical structures. Indeed, these structures use a subsumption relationship, the semantics of which are more conceptual (*i.e.* relative to a shared set-oriented logic) than formal. This inheritance must therefore be explicitly specified.

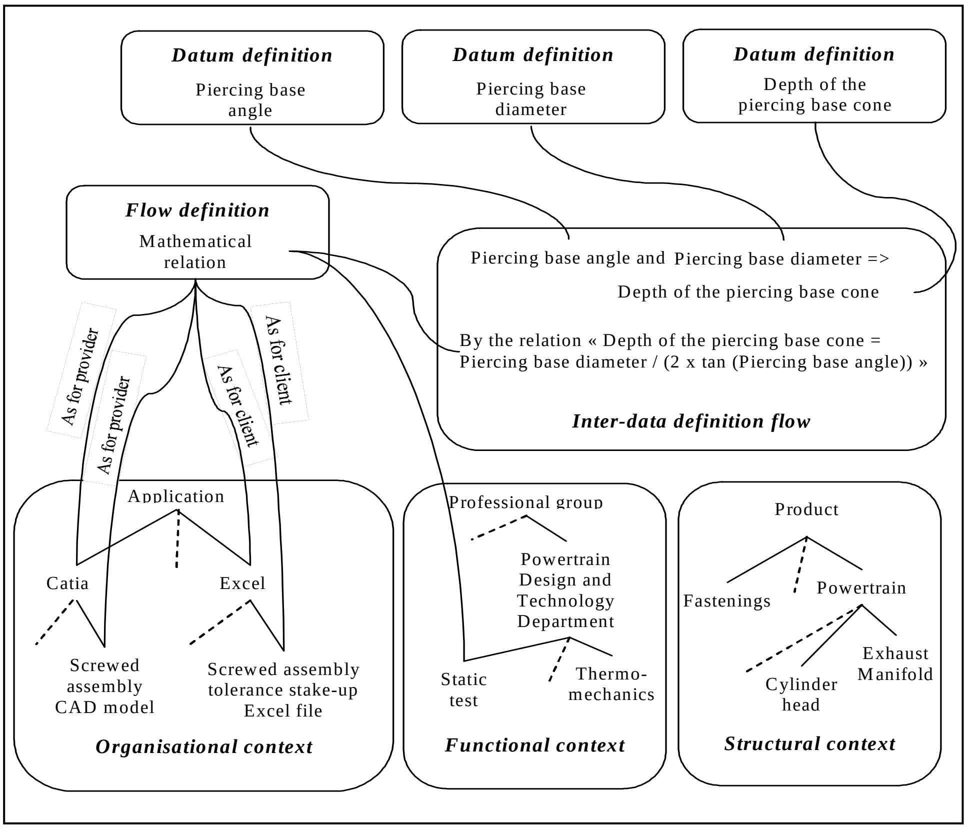

Figure 4. Example of an instance for flow-type knowledge representation between parameters

3.3. Data

The UML model relative to data is shown in Figure 5. The aim of this model is to accurately describe the parameters present in each PKO. It differs from the structure model, the aim of which is to reconcile the parameters present in a distributed manner in each PKO.

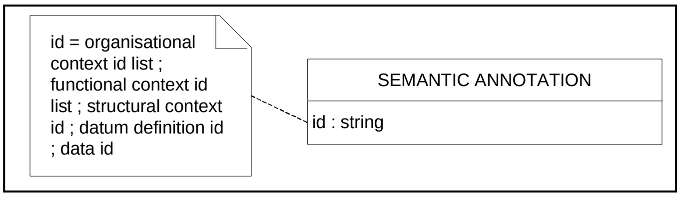

Figure 5. Data model

This model uses the semantic annotation concept. This is a concept known from the Semantic Web (Gomadam *et al.*, 2010). Its goal is to represent the semantics of a concept defined within an ontology, by means of a string of characters. An instance of a semantic annotation model is thus linked to a well-defined numerical parameter and evidenced within a PKO. It is made up of an ordered series of identifiers with separators taken from the data structure model. This series of identifiers is selected in such a manner as to model the semantics and the exact level of contextualisation of the selected parameter. Thus a semantic annotation is composed of h *organisational contexts* (since the selected parameter is located in an identified environment), p *functional contexts* (since the parameter can be shared by several professional sector environments), one *structural context* (since the data structure model is exclusive to this context), one *datum definition* and one *datum with* $n \geq h \geq 1$ (n being the number of *organisational contexts* associated with knowledge representation that have the same *datum definition* in the data structure model) and $m \geq p \geq 1$ (m being the number of *functional contexts* associated with knowledge representation which have the same *datum definition* in the data structure model). It should be noted that a semantic annotation is less contextualised than the knowledge it refers to. This is due to the fact that a knowledge object has been constructed in a particular context and, this context does not encompass all the semantics reconciled in the data structure model which corresponds to that of several PKOs. A semantic annotation can thus be seen as a fragment of knowledge representation. In certain cases, a PKO parameter may be generic over several structural contexts. This is the same PKO parameter that can be used to specify different products (e.g. a diameter parameter which corresponds to the diameter of part 1 or part 2, depending on the ongoing study). In view of the exclusive nature of the structural context (cf section 3.2), this parameter can not be described by just one semantic annotation concept. We propose to semanticise this type of parameter by defining several parameters on different structural contexts in the data structure model and then, use several semantic annotations. This method enables us to compensate for the descriptive limit of our data structure model.

Figure 6 shows an example of a semantic annotation instance. This example is applied to the diameter parameter of a well-identified, parameterised CAD model.

Semantic annotation

Screwed assembly CAD model ; Static test ; Cylinder head ; Screw passing diameter ; Numerical

Figure 6. Example of a semantic annotation instance

3.4 Activities

In a dynamic approach, activities are intended to represent the operations that enable exchanges between knowledge objects based on our static approach (data structure model, data model). They are represented by means of BPMN models (OMGc, 2011) in Figure 7. In these models, preparation activities for the exchange, publication and management of annotations require the assistance of a user. The others are automatic.

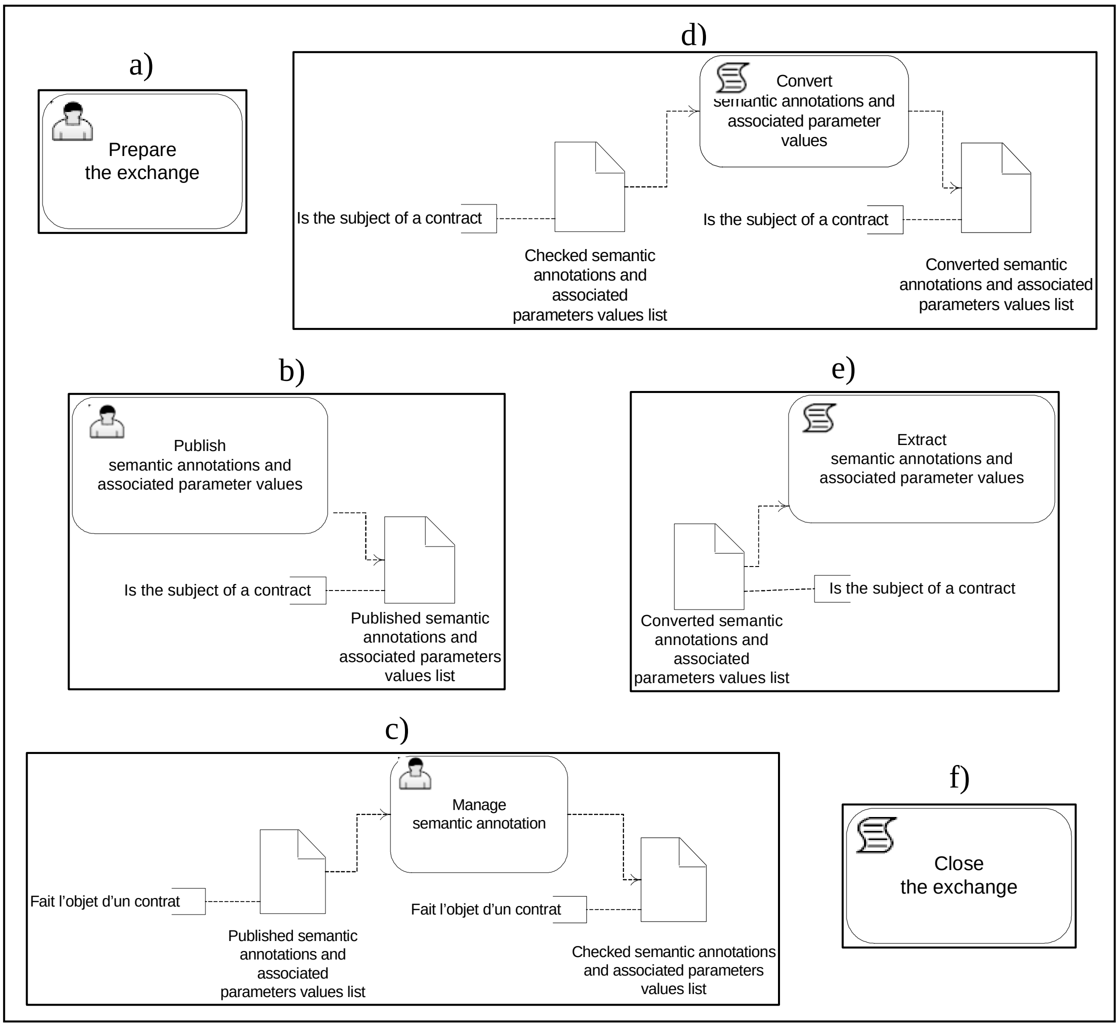

Figure 7. Inter-knowledge objects exchange activities model

The exchange preparation activity (a) enables the user to select the supplier's organisational context, the client's organisational context and the functional context in which the exchange is to take place.

The publication activity (b) enables the user to select a list of parameters to be exchanged. It enables the recovery and publication of a list of semantic annotations and related parameter values to be supported, from a PKO.

The aim of the semantic annotation management activity (c) is to handle the processing of simple and multiple semantic annotations (*cf* Section 3.3 on the generic parameters relative to several structural contexts). In the case of simple annotations, it reduces the number of context identifiers in the annotations published. For the organisational contexts, only the supplier's organisational context specified in the preparation task is retained. For the functional contexts, only the functional context, specified in the preparation task, is retained. Should multiple annotations exist, user intervention is requested to specify the current structural context. Based on this information, only the semantic annotation corresponding to the specified structural context is retained. This semantic annotation is then processed in the same way as a simple semantic annotation.

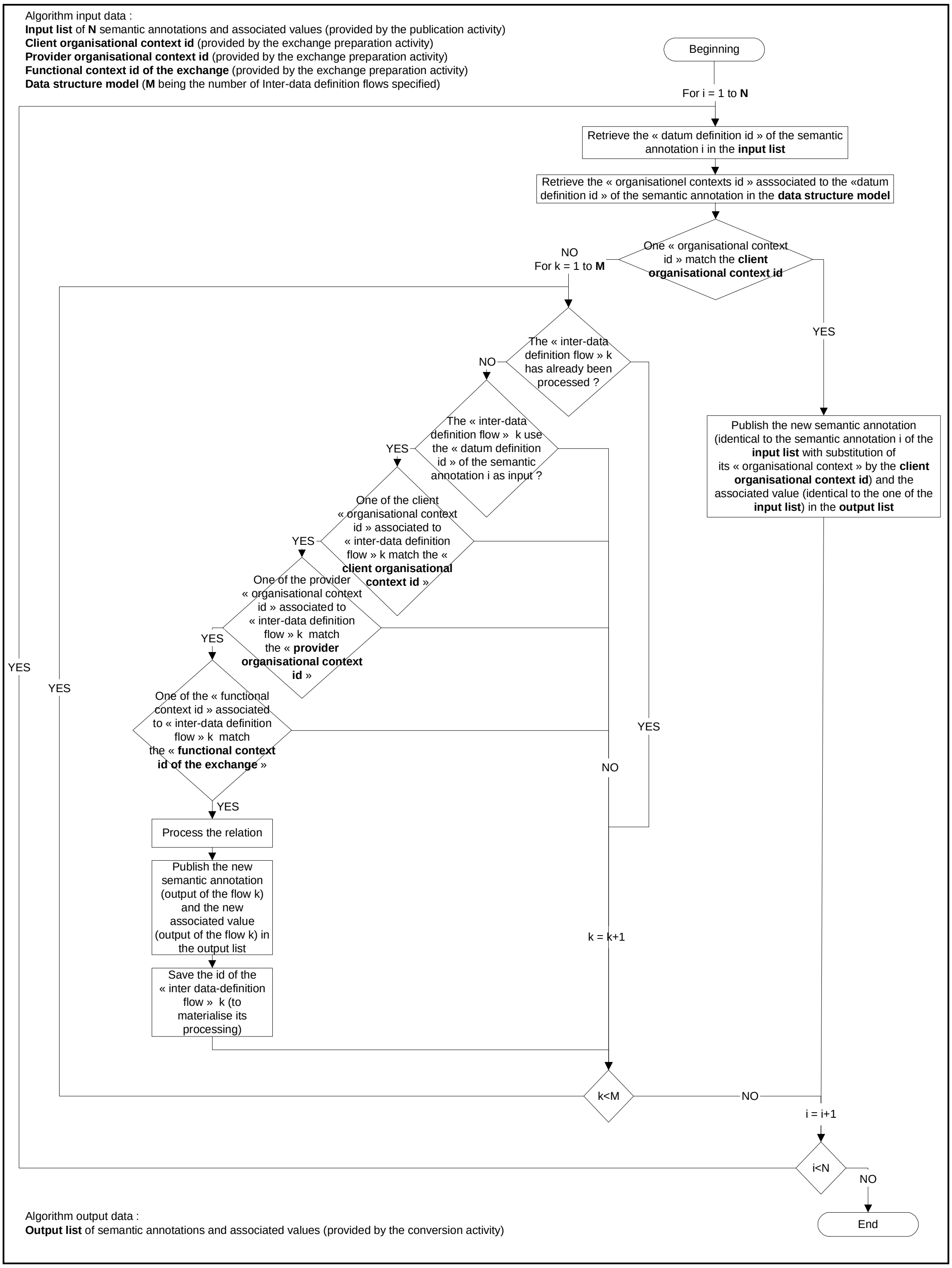

Figure 8. Conversion algorithm for semantic annotations and related parameter values

The purpose of the conversion activity (d) is to convert a list of semantic annotations and parameter values from one PKO to another. This activity takes into account the relations between the parameters specified in the data structure model by means of flow objects. It brings into play a conversion mechanism, described in Figure 8.

The aim of the extraction activity (e) is to update a list of parameter values within a PKO, from a list of semantic annotations and related values. A simple mapping mechanism between the identifiers contained in the semantic annotations is used: for a given semantic annotation in the list, the annotation for the matching PKO parameter must contain at least the same identifiers. It can contain more (*cf* definition of a semantic annotation in Section 3.3).

The close-off activity (f) enables us to indicate the end of the exchange.

The model indicates that the entry and/or output list artefacts for the first three activities are the subject of a contract. This contract relates to the fact that, although the model enables a certain level of semantics for a parameter to be taken into account, it does not cover the value of this parameter, which is specified during the project. This implicit contract requires that the value associated with the semantic annotation in the list artefact correctly matches the value related to the parameter in the PKO.

3.5 The process

The objective of the process model is to describe how activity sequences are established and to attach them to an executing application. Its implementation must enable user interventions to be kept to a minimum during an exchange between PKOs. Figure 9 shows the BPMN model of the process.

The exchange process starts with an initialisation event triggered by the user from his/her professional application interface. The user then specifies the exchange conditions by means of the “prepare the exchange” activity in the IKOES central module. From this point, one application takes on the role of the client, and the other, that of the supplier. These roles can be reversed in subsequent exchanges. Activities are subsequently rolled out in sequence by means of events with the assistance of the user, when required (*cf* Section 3.4).

This model proposes a distributed vision of IKOES, the system we aim to implement. It supposes the need to customize professional sector applications and does not facilitate control over the life cycle of semantic annotations. Uren *et al*. (2006) have already proposed that the centralised management of annotations should enable greater control over the life cycle of these annotations. However we have endeavoured to propose this solution for the following reason: access to PKOs in a project context is frequently regulated by means of PDM or KMTs. By using semantic annotations supported by PKOs we do away with difficulties such as: managing access rights to annotations (which are assumed on files by PDM or KMTs) and maintaining links between documents and their annotations (which would be fastidious depending of the life cycle of the file itself). In addition, giving the application which handles PKOs the role of publishing and extracting these semantic annotations and the value of the associated parameters, enables us to limit interoperability problem – since the IKOES central module does not have to carry out the task of searching for these elements inside the file.

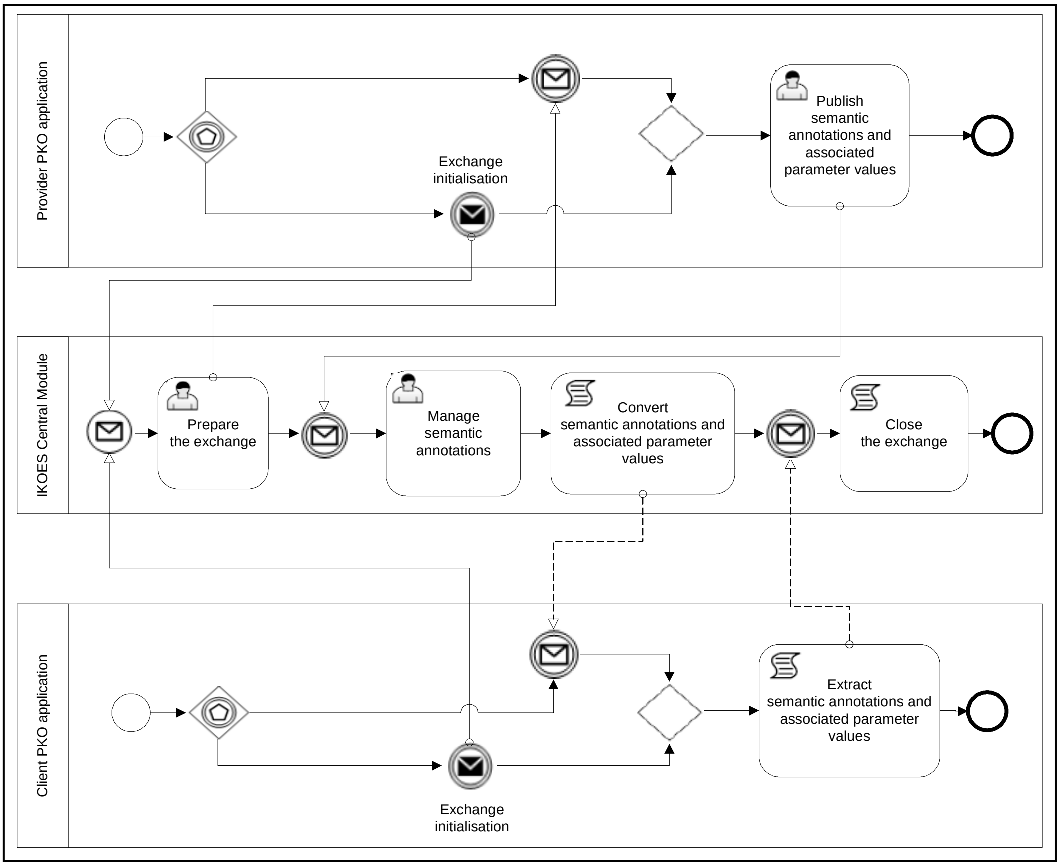

Figure 9. Inter-knowledge objects exchange process model

4. Validation of the conceptual model

4.1. Demonstrator architecture

To validate the implementable nature of our conceptual model, we have produced a demonstrator, the distributed architecture of which is shown in Figure 10 (the textures of the different boxes in Figure 10 are linked to those in Figure 2).

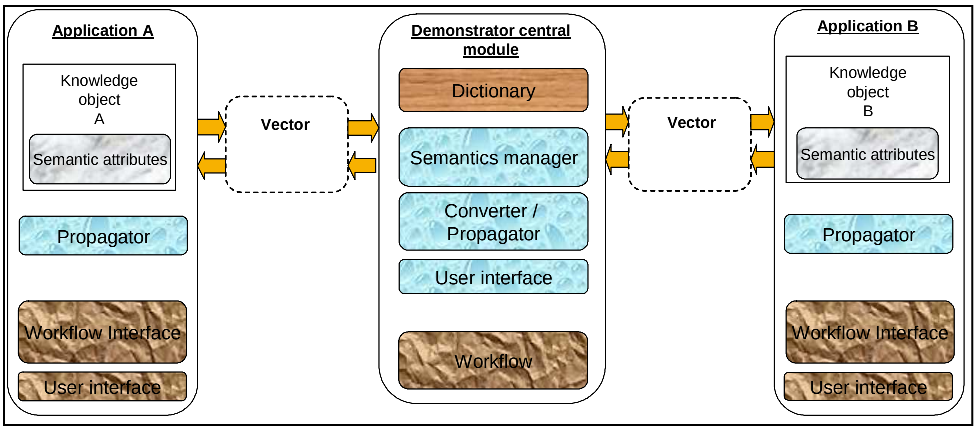

Figure 10. Demonstrator architecture

4.2. Details of demonstrator elements and implementation technology

Table 1 shows the demonstrator elements, their relationship with the conceptual model, and the implementation technology used.

Demonstrator elements	**Relation with the conceptual model**	**Implementation technology**
Dictionary	This element encapsulates the data structure model.	Excel file (the hierarchical structure of concepts was smoothed for ease of representation reasons).
Semantic attributes	This element corresponds to the container of the data model's semantic annotations.	Field for comments on the parameters of the studied PKO (*cf* Figures 11 and 12)
The semantics manager	This element encapsulates the semantic annotations management task of the activity model. In the context of the demonstrator, it also carries out syntactical verification of the annotations contained in the vector, with respect to the dictionary. The purpose of this verification is to overcome the fact that they can be easily modified by users.	VBA module.
Propagators	This element encapsulates the publication and extraction task of the activity model.	Visual Basic for Application Module (VBA).
Vector	This element supports the list of entry data for activity tasks in the activity model.	Excel file.
Converter	This element encapsulates the activity model conversion task in interaction with the data structure model.	VBA Module (This module consults the dictionary according to the algorithm indicated in Section 3.4).
User interface of the demonstrator central module	This element encapsulates the activity model "prepare the transfer task".	VBA Module.
User interface - applications, Workflow and workflow interface	This element encapsulates the process model.	VBA class modules for application events. Excel events for demonstrator central module events. An Excel worksheet for global process management.

Table 1. Description of the demonstrator elements

4.3. Demonstrator tests

We implemented and tested this demonstrator at Renault Powertrain Technology Design Department, our industrial partner for this work. Details of the application case study are given below.

The PKOs proposed are as follows:

- a parameterised CAD model, representing a screwed assembly (*cf* Figure 11),
- a parameterised Excel file to calculate tolerance stake-up for a screwed assembly (*cf* Figure 12).

The applications used which enabled the processing of these PKOs were an off-the-shelf CAD application (CATIA V5) and the Excel application.

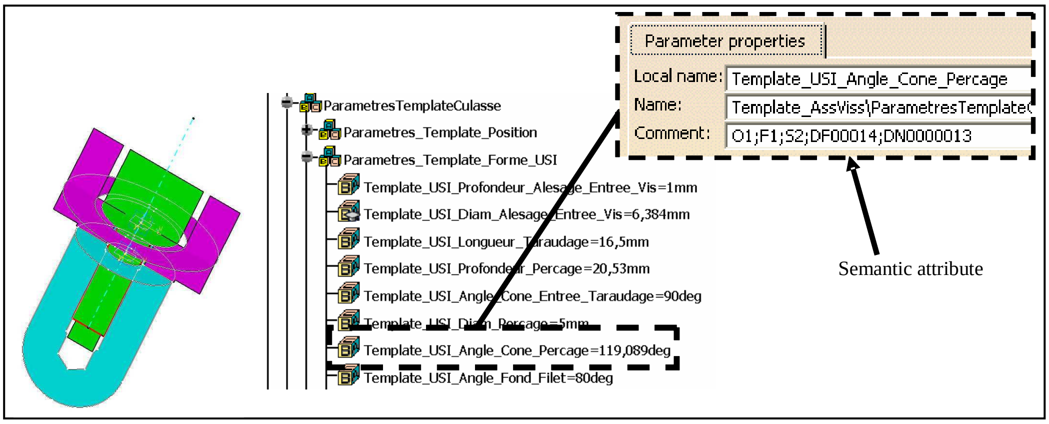

Figure 11. CAD PKO

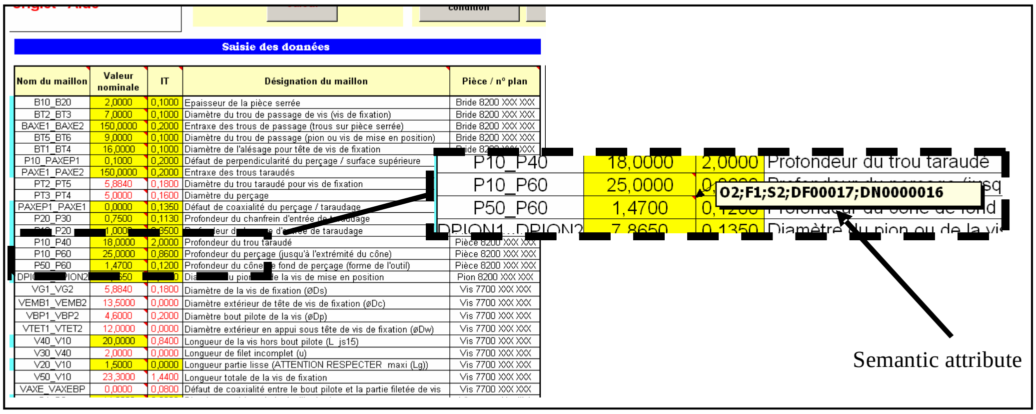

Figure 12. Excel PKO

The first stage of our work involved the formal expression of knowledge in the dictionary, based on the following scenario:

- a PKO manager manually completes the PKO parameters in the Excel dictionary file. The contexts defined are as follows: two organisational contexts (a CAD screwed assembly template, a screwed assembly dimension chain file), three structural contexts (screw, exhaust manifold and cylinder head) and a functional context (static test). Figure 13 shows an extract from this file where O1, O2 ,F1 , S2 are respectively the identifiers for CAD screwed assembly template, screwed assembly dimension chain file, static test and cylinder head,

- the PKO manager manually completes the formulas (in the form of VBA methods) in the converter then enters the identifiers for these formulas in the Excel dictionary file - i.e. in the “relation” attribute of the *inter-data definition flow* class (*cf* Figure 3),

- the PKO manager constructs the semantic annotations by retrieving the identifiers of elements from the dictionary and fill in the PKO semantic attributes (*cf* Figures 11,12).

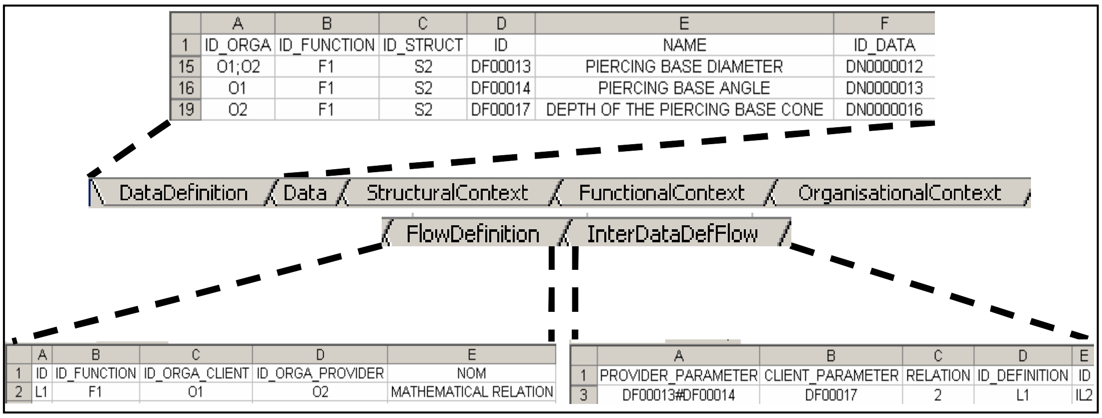

Figure 13. Extract from the Excel dictionary file

The second stage of the work consisted in the actual specification of the PKO parameter values. This work constitutes an every day task of a designer in a redesign process. The following design loop was implemented:

- a designer sets the dimensions for the screwed assembly in the CAD model,
- the designer automatically transfers the defined parameters (nominal dimensions) via the IKOES demonstrator to the tolerance stake-up file,
- the designer calculates the tolerance stake-up in the appropriate PKO and thus balances the nominals,
- the designer automatically transfers the tolerance stake-up file parameters to the CAD model, via the IKOES demonstrator.

Following this stage, some research gaps have been highlighted for our conceptual model:

- ***The first research gap relates to the data structure model***. The simultaneous transfer of several instances of the same parameter is not supported. This case occurs when a PKO is able to process several configurations of a parameter with the aim of retaining only one of them.
- ***The second research gap relates to the data structure model***. The use of several syntaxes for the same datum is not supported (*e.g.*: value x of the diameter parameter can be written as “x” or “Øx”, depending on the PKO). In fact, our data structure model allows a *datum definition* to be associated with only one *datum*, *cf* Figure 3.
- ***The third research gap relates to the activity model***: The algorithm for the conversion of semantic annotations and the exchange preparation activity can be optimised in order to manage the definition of relations, whose input parameters are distributed between several PKOs - as may be the case in an assembly.
- ***The fourth research gap relates to the process model***. At the time of an exchange, the considered parameters are necessary “pushed” from the supplier PKO to the client PKO. If the client PKO is not well-known by the user, it is therefore possible to transmit more parameters than is strictly necessary. Although the converter implements a filter (via the conversion algorithm, *cf*. Section 3.4) to ensure that only parameters that can be understood by the PKO are transmitted, this exchange is not optimal. A possibility for improvement of this point would be to define a parameter request process from the client PKO.

5. Discussion

Relatively to our motivation expressed in section 1, the feedback from professional sector groups was positive. An information system of this type does not modify the initial objective of PKOs, which is to act as boundary objects and adaptable memory devices across projects (Cacciatori, 2008). As shown in the case study, it does not imply any strong modification in the way knowledge is formulated and structured in the file due to the parameter approach. It is therefore possible to reuse existing PKO's. Moreover by reducing the lead time and by making it less tedious to transfer parameter values between PKOs, this system should enable PKOs to be used further upstream in redesign projects.

Beyond these goals, the IKOES data structure model spawns elicitation of knowledge. Indeed, its enables the construction of a map on interrelated parameters; at level of granularity which is not usually tackled by off-the-shelf tools (PDM and KMTs). This cartography supposes the preliminary identification of the PKOs sharing same parameters or parameters that can be deduced from one another following simple association rules. The complete identification of these files does not seem trivial to our industrial partner. Tools used to manage these files today, do not always show a sufficient descriptive quality to be used in this intent. For example, it can be difficult to extract from such tools, the overall parameterised files concerning a particular product part (these files likely having common or linked parameters). The works of Joo and Lee (2009), which mention the limitations of KMTs in terms of search capacity, tends to generalise this point. In addition, setup interviews in specific professional sectors can be particularly long. Indeed, as the use of these files is strongly fragmented, they cannot be identified by only a few individuals. A progressive supply of parameters in the system seems necessary (this would contribute to justify the simplicity requirement of the underlying model). Nevertheless, this last approach seems to be moderate insofar as the profitability associated with IKOES implementation has to be proven by first elements of cartography: to assess the overall gain in terms of lead times, a study is in process at Renault Powertrain Technology Design Department to map relevant PKOs. The time and resource-intensive nature of this work can be seen as an obstacle to the development of a system aiming at parameters management and probably justify that there is currently not much work done on this topic. Nevertheless, we believe that it constitutes an important step for industrial organizations, enabling them to achieve a global approach of their knowledge considering a granularity level often out of control.

6. Conclusion

In large manufacturing companies, optimising the diffusion of knowledge is a lever towards improving the success of product redesign processes. Various approach strategies exist to contribute to this optimisation. Among them, the use of parameterised knowledge objects is a simple, flexible and often-used practice.

In this article, we have proposed a conceptual model to implement IKOES: an Inter-Knowledge Objects Exchange System. This system enables users to avoid the fastidious work of copying or adapting parameter values from one PKO to another and thus reduces the resulting lead times and risks of error. Our conceptual model was evaluated by implementing a demonstrator and testing it in the context of a scenario proposed by our industrial partner, Renault Powertrain Technology Design Department.

Various prospects for future work have been proposed for this conceptual model. The aim of this work will be to enable the model to support the simultaneous exchange of several parameter instances, the management of several syntaxes for the same parameter, and the integration of a “pull processing” parameter exchange.

Acknowledgment

This work is part of a PhD thesis project jointly undertaken with our industrial partner Renault (Powertrain Technology Design Department).

References

AMIN Ash, ROBERTS Joanne. Knowing in action: Beyond communities of practice, Research Policy, Volume 37, Issue 2, Pages 353-369, 2008.

BADIN Julien, MONTICOLO Davy, CHAMORET D, GOMES Samuel. Moving toward a product model to manage data, information and knowledge for mechanical design and simulation, Proceeding of KARE'10 (Knowledge Acquisition, Reuse and Evaluation), 2010.

BAIZET Yoan. La gestion des connaissances en conception-Application à la simulation numérique chez Renault-DIEC, PhD thesis of INP Grenoble (France), 2004.

BEN-ARI Mordechai, TZIPPORA Yeshno. Conceptual models of software artifacts, Interacting with computers, Volume 18, Issue 6, Pages 1336-1350, 2006.

BERNARD A., TICHKIEWITCH S. Methods and Tools for Effective Knowledge Life-cycle-management. Springer, 2008.

CACCIATORI Eugenia. Memory objects in project environments: Storing, retrieving and adapting learning in project-based firms, Research Policy, Volume 37, Issue 9, Pages 1591-1601, 2008.

CHAPMAN Craig B, PINFOLD Martyn. The application of knowledge based engineering approach to the rapid design and analysis of an automotive structure, Advances in Engineering Software, Volume 32, Issue 12, Pages 903-912, 2001

CHEN Guoqing, WEI Qiang, LIU De, WETS Geert. Simple association rules (SAR) and the SAR-based rule discovery, Computers & Industrial Engineering, Volume 43, Issue 4, Pages 721-733, 2002.

CIMdata. cPDm – The Key to Harnessing Innovation in an E-Business World, http://www.cimdata.com/plm/definition_cpdm.html (Accessed February 2011), 2011

D'ENZA Alfonso Iodice, PALUMBO Francesco, GREENACRE Michael. Exploratory data analysis leading towards the most interesting simple association rules, Computational Statistics & Data Analysis, Volume 52, Issue 6, Pages 3269-3281, 2008.

ERMINE Jean-Louis, CHAILLOT Mathias, BIGEON Philippe. Méthode pour la gestion des connaissances, Ingénierie des systèmes d'information, AFCET-Hermès, Vol. 4, n° 4, pp. 541-575, 1996.

GERO John S. Design Prototypes: A knowledge Representation schema for design; AI Magazine, American Association for Artificial Intelligence, 11/4:26–36, 1990.

GOMADAM . K., RANABAHU. A., SHETH. A. SA-REST: Semantic Annotation of Web Resources, http://www.w3.org/Submission/SA-REST (Accessed August 2010), 2010

GRUNDSTEIN Michel, ROSENTHAL-SABROUX Camille, PACHULSKI Alexandre. Reinforcing decision aid by capitalizing on company's knowledge: Future prospects, European Journal of Operational Research, Volume 145, Issue 2, Pages 256-272, 2003.

HSU Ching-Chih, HWANG Sheue-Ling. A study of interface design improvement in an engineering data management system on the world wide web, Computers & Industrial Engineering, Volume 47, Issue 1, Pages 31-43, 2004.

JOO. J., LEE. S.M. Adoption of the semantic web for overcoming technical limitations of knowledge management systems, Expert systems with applications, Volume 36, Issue 3, Part 2, Pages 7318-7327, 2009.

McMAHON Chris, LOWE Alistair, CULLEY Steve; Knowledge management in engineering design: personalization and codification. Journal of engineering design, Volume 15, Number 4, p389-403, 2004.

MOKA: A framework for structuring and representing engineering knowledge. Final synthesis, http://web1.eng.coventry.ac.uk/moka/kbe.htm (Accessed February 2011), 2000.

NGAI E.W.T, CHAN E.W.C. Evaluation of knowledge management tools using AHP. Expert Systems with Applications, Volume 29, Issue 4, P889-899, 2005.

OMGa, MILLER Joaquin, MUKERJI Jishnu. MDA Guide Version 1.0.1, http://www.omg.org/cgi-bin/doc?omg/03-06-0 (Accessed February 2011), 2003.

OMGb. OMG Unified Modeling Language Superstructure version 2.2, http://www.omg.org/spec/UML/2.2/ (Accessed February 2011), 2009.

OMGc. Business Process Model and Notation version 2, http://www.omg.org/spec/BPMN/2.0 (Accessed February 2011), 2011.

PIKOSZ. P., MALMQVIST. J. Possibilities and Limitations when using PDM Systems to support the product development process, Proceedings of NordDesign'96, s.165-176, Helsinki, 1996.

PRUDHOMME G., BOUJUT J.F. et POURROY F. Activités de conception et instrumentation de la dynamique des connaissances locales, Ingénierie des Connaissances, Plate-forme AFIA, Presses Universitaires de Grenoble, pp. 41-60, 2001.

RANGANATHAN S.R., Classification, coding and machinery for search, UNESCO/NS/SL/3, Paris, 1950.

SKARKA W., Application of MOKA methodology in generative model creation using CATIA, Engineering Applications of Artificial Intelligence, Volume 20, Issue 5, Pages 677-690, 2007.

SUTTON D., CommonKADS analysis and description of a knowledge based system for assessment of breast cancer, Expert Systems with applications, Volume 36, Issue 2, Pages 2411-2423, 2009.

TRIANTAPHYLLOU. E, LIAO T.W, IYENGAR S.S. A focused issue on data mining and knowledge discovery in industrial engineering, Computers & Industrial Engineering, Volume 43, Issue 4, Pages 657-659, 2002.

TSUCHIYA S. Improving Knowledge Creation Ability through Organizational Learning, Proceedings of International Symposium on the Management of Industrial and Corporate Knowledge ISMICK'93, Compiegne, 1993.

UREN V., CIMIANO P., IRIA J., HANDSCHUH S., VARGAS-VERA M., MOTTA E., CIRAVEGN F. A Semantic annotation for knowledge management : Requirements and a survey of the state of the art, Web Semantics: Science, Services and Agents on the World Wide Web, Volume 4, Issue 1, Pages 14-28, 2006.

VACCARO Antonino, PARENTE Ronaldo, VELOSO Francisco M. Knowledge Management Tools, Inter-Organizational Relationships, Innovation and Firm Performance, Technological Forecasting and Social Change, Volume 77, Issue 7, Pages 1076-1089, 2010.

YOSHIOKA. M., UMEDA. Y., TAKEDA. H., SHIMOMURA. Y., NOMAGUCHI. Y., TOMIYAMA. T. « Physical concept ontology for knowledge intensive engineering framework », Advanced Engineering Informatics, Volume 18, Issue 2, Page 95-113, 2004.