

The International Journal of Digital Curation

Issue 4, Volume 3 | 2009

Enabling Product Design Reuse by Long-term Preservation of Engineering Knowledge

Jörg Brunsmann,
Wolfgang Wilkes,
Fernuniversität Hagen, Germany

Abstract

In the highly competitive engineering industry, product innovations are created with the help of a product lifecycle management (PLM) tool chain. In order to support fast-paced product development, a major company goal is the reuse of product designs and product descriptions. Due to the product's complexity, the design of a product not only consists of geometry data but also of valuable engineering knowledge that is created during the various PLM phases. The need to preserve such intellectual capital leads engineering companies to introduce knowledge management and archiving their machine-readable formal representation. However, archived knowledge is in danger of becoming unusable since it is very likely that knowledge semantics and knowledge representation will evolve over long time periods, for example during the 50 operational years of some products. Knowledge evolution and knowledge representation technology changes are crucial issues since a reuse of the archived product information can only be ensured if its rationale and additional knowledge are interpretable with future software and technologies. Therefore, in order to reuse design data fully, knowledge about the design must also be migrated to be interoperable with future design systems and knowledge representation methods. This paper identifies problems, issues, requirements, challenges and solutions that arise while tackling the long-term preservation of engineering knowledge.¹

¹ This paper is based on the paper given by the authors at the 5th International Digital Curation Conference, December 2009; received November 2009, published December 2009. The 5th International Digital Curation Conference taking place in London, England over 2-4 December 2009 will address the theme *Moving to Multi-Scale Science: Managing Complexity and Diversity*.



Introduction

Product Life Cycle Management (PLM) systems manage the entire lifecycle of a product from its conception through design and manufacturing to service and disposal. PLM systems integrate people, data, processes and systems and provide a product information backbone for companies. The usage of PLM systems is common in industries like electrical or mechanical engineering. In such industries, newly created products tend to be complex and are developed and serviced in an international and collaborative environment. In addition, products are developed by a great number of engineers that use knowledge-intensive processes.


Engineering Knowledge

Although the debate about the data-information-knowledge-wisdom hierarchy is still ongoing (Fricke, [2009](#)), we regard knowledge as personal and public know-how and know-that. A classification of such engineering knowledge is performed by Ahmed ([2007](#)). The following list gives examples of engineering knowledge:

- During **service and operation**, knowledge about the behaviour of a product is created that is useful for a product redesign or variation.
- An engineer designs a product using his knowledge. Existing approaches to capture such **design rationale** are compared in Regli, Hu, Atwood and Sun ([2000](#)). Other approaches include capturing the design rationale by means of ontologies (Medeiros, Schwabe & Feijo, [2005](#)) or by the use of a design rationale editor (Ahmed, Bracewell & Kim, [2005](#)).
- Innovations in product designs are created during an **innovation process**. Most of these innovative designs are not manufactured at all, but remain important intellectual property of the company. Such intellectual property needs to be archived for possible future use.
- The **design history** (provenance) is also relevant knowledge. For instance, designs are derived from previous designs or they are created because of failure reports or new requirements.
- During **collaboration sessions**, engineers in the mechanical and electrical CAD domain exchange design constraints which represent knowledge.
- During a **design review**, decisions, arguments, ideas and justification are exchanged by experienced expert designers.
- In mechanical and electrical engineering, references to standard **product classifications** are created which also represent knowledge.

All this knowledge, necessary in order to fully understand a product design, represents a valuable resource to any company.

Today, in competitive industries, a major aim for companies is to be first on the market with innovative products while simultaneously cutting their costs. One key approach to support such shorter development cycles is to *reuse* product designs and associated knowledge. One example of such knowledge reuse is the application of experience acquired during product service (e.g., failures which occurred during product operation) to avoid similar errors and failures. Another example of knowledge reuse is the consideration of past design rationales that include the reasons behind a design decision, the justification for it, alternatives that were considered, the tradeoffs evaluated and the argumentation that led to the design decision. Such design rationales are useful in accelerating the design process for a product variation.



It has to be noted, that a future engineer is only able to reuse a product design if the current design and all relevant knowledge can be found, retrieved and understood. Although archiving, searching, displaying and understanding such knowledge sounds easy, one must bear in mind that the whole product lifecycle is accompanied by knowledge-intensive processes. Frequently this knowledge is not captured at all, because engineers tend to be unconvinced that the effort of documenting knowledge is justified. Even when knowledge is captured, engineers often apply hidden and implicit knowledge which is neither captured nor archived. This is not regarded as problematic while the engineers who retain this knowledge are available. However, on a long-term basis, the probability increases with time that engineers retire, leave the company or simply forget their design intent. If knowledge has not been captured and archived over certain PLM phases it is lost forever.

As a consequence, a present-day engineer needs to document and long-term archive not only the product data but also other relevant engineering knowledge at real time. This knowledge has to be described by a formal representation language (e.g., ontology) in order to be interpretable (e.g., searching, reasoning) by computers. Even if this knowledge is archived, it is in danger of becoming unusable, because it is not unlikely (given the product longevity) that the knowledge representation (syntax) or the knowledge meaning (semantics) will change over time. Unfortunately, this evolution raises difficulties: in the case of reuse, knowledge archived in the past may no longer be understandable because it no longer matches current ontology versions. Therefore, the archived knowledge needs to be preserved in order to remain interoperable in the future, for example for future reuse.

Requirements for Capturing, Archiving and Preservation of Engineering Knowledge

For such archiving and preservation of knowledge, the following requirements in respect of engineering knowledge management can be derived:

- **Real-time knowledge capture:** engineering knowledge has to be captured and archived at that point in time when it is initially created. Otherwise the knowledge is lost forever since it cannot be recreated.
- **PLM-based knowledge capture:** relevant engineering knowledge needs to be created in all phases of PLM.
- **Non-intrusive capture:** since knowledge capture is costly, the process should not interfere with normal engineering activities and knowledge extraction should be automated to minimize intrusion.
- **Formal knowledge capture:** if a product design has to be found in the future, formal knowledge capture is necessary because good search results can only be achieved if product designs are archived accompanied by their knowledge in a formal representation. The formal representation of knowledge can be done in different ways, for example with ontologies.
- **Knowledge preservation:** captured knowledge has to be preserved with special preservation methods so that it remains interpretable in the future.

This paper describes solutions to the problem of losing semantic interoperability and is structured as follows: in the next section the long-term preservation of captured knowledge is described. This section also examines whether existing approaches tackle the long-term preservation of knowledge. After that, challenges and potential solutions for the migration of knowledge are identified.



Long-term Knowledge Preservation

The need for long-term preservation of product data is attracting increasing attention in the engineering industry. Both legal (legislation) and business reasons (reuse) motivate companies to archive their product data (Heutelbeck, Brunsmann, Wilkes & Hundsdörfer, [2009](#); Wilkes et al, [2009](#)). While it might be sufficient just to visualize the engineering data and knowledge for legal reasons, for the purposes of reuse, it is necessary to use the archived data with contemporary software and techniques. Some projects and initiatives address the long-term preservation problem of engineering data (Brunsmann & Wilkes [n.d](#)). But, in fact, most of the existing projects in this realm (LOTAR², MOSLA³, VDA Recommendation 4958⁴) focus on the preservation of geometry information which is transformed and migrated via vendor-neutral file formats like STEP (Standard for the Exchange of Product model data). By contrast, the following projects also observe engineering knowledge:

Knowledge and Information Management through Life Project (KIM)

The KIM Project (Ball, Patel, McMahon, Culley & Green, [2006](#)) proposes a framework of Lightweight Models with Multilayered Annotations (LiMMA) that combines vendor-neutral, lightweight CAD formats with annotations. These annotations (knowledge) are collected throughout the life of a product and are archived as informal augmentations to geometric data. Another feature of KIM is the Registry/Repository of Representation Information of Engineering (RRoRIE) which is a representation information registry for engineering specific file formats.

Digital Engineering Archives

Design repositories (Kopena, Shaffer & Regli, [2006](#)) archive heterogeneous function, behaviour and rationale of a 3D design. These design aspects (knowledge) are captured and classified by the use of taxonomies of standardized terminologies. STEP and the Ontology Web Language (OWL) are used as a standard representation in order to support easier design retrieval and reasoning in design repositories.

Sustaining Engineering Informatics

This NIST (National Institute of Standards and Technology) project (Lubell, Rachuri, Mani & Subrahmanian, [2008](#)) focus on methods and metrics in the curation of engineering data using the concept of “3Rs”: reference, reuse and rationale. Reference is the ability to view the data, reuse means to modify or to reengineer the data and rationale (“Why questions”) is the ability to display information such as construction history or design intent. In this case, rationale is regarded as knowledge.

NIST Core Product Model

The NIST Core Product Model (CPM) (Rachuri, [2007](#)) is another approach to capturing knowledge. CPM is an abstract model that is defined as an UML class diagram and is based on the form, function and behaviour of products. Extensions to the CPM like the Product Semantic Representation Language (PSRL) use a formal representation of product information.

² LOTAR, *Long-term archiving and retrieval of digital technical product data in the aerospace community*. Retrieved August 5, 2009, from <http://www.prostep.org/en/project-groups/long-term-archiving-lotar/the-lotar-standard.html>

³ MOSLA, *Maturity of standard for long-term CAD data archiving*. Retrieved August 5, 2009, from <http://www.mosla.org/>

⁴ VDA Recommendation 4958. Retrieved August 5, 2009, from <http://www.prostep.org/en/downloads/recommendations-standards.html>

Methodology and tools Oriented to Knowledge-based engineering Applications

MOKA (Methodology and tools Oriented to Knowledge-based engineering Applications) (Klein, 2000) is a framework for structuring and representing engineering knowledge and aimed to develop a methodology of knowledge modelling in design and engineering. MOKA described a way of collecting, structuring and formalizing engineering knowledge associated with designs.

As already mentioned, it is required to use a formal machine-readable mechanism to represent knowledge. While using such formal representations, one needs to bear in mind that engineering knowledge has external dependencies such as public or private semantic languages that give meaning to data. For example, if knowledge is specified by referencing an ontology, archival of knowledge instances also requires archiving the reference to this specific state (version) of the ontology. It is quite likely that the ontology will change over time or another representation language will be used; this will mean that the previously archived knowledge instances will be unusable in combination with a new version of the ontology. This evolution of semantics and syntax has not come within the scope of existing projects and will be discussed in the next section.

Knowledge Migration Management

This section describes the problem of engineering knowledge preservation in more detail, lists system requirements, and presents an information architecture for knowledge preservation by means of knowledge migration.

Statement of Problem

The motivation for preservation of engineering data is twofold:

- **Legal reasons:** data need to be recovered for viewing and inspection (e.g., accident investigation). Here, the basic goal of preservation is to keep the existing state of data and knowledge which may be achievable by an *emulation* of the old system environment or by using a viewer that is able to interpret old data models. Migration of data is not the focus in this case.
- **Business reasons:** first, the previously archived data need to be integrated into a contemporary system environment for *reuse*. Thus, for the purposes of interoperability with future models, terminologies and technologies, it is necessary to ensure that the data is *migrated* into a model which can be handled in a future system environment. Second, archived knowledge needs to be available for *searching*. For example, the product designs that were created during innovation processes and that were not manufactured, need to be retrievable for future reuse. If searching is done via metadata that are described by ontologies and those ontologies evolve, it must be assured that an archived design still remains retrievable via *mediation*.

As described, for these motivations the preservation methods are different (emulation, migration, mediation). This paper concentrates on the business scenarios in which the long-term archived knowledge is in danger of becoming obsolete due to:

- **Semantic evolution:** Today, knowledge will become obsolete sooner than ever before and this knowledge change is reflected by newer versions of the same ontology or by a new conceptualization that both reflect a new interpretation of the real world.



- **Syntactic evolution:** The representation of knowledge might change over time, for example if a system change from one software vendor to another occurs, or new syntactic features of the same language are invented.

In such evolutionary occurrences, the long-term archived knowledge has to be migrated in order to remain interpretable. In the first semantic evolution case, the knowledge is migrated to a new version of the same representation. In the second syntax representation change case, the migration takes place from one representation to another representation, both on a syntactical and semantic level which makes it even more complicated. In both cases, the migration process could be executed via ontology mapping (Kalfoglou & Schorlemmer, 2003). Such mappings are difficult to build and the mapping process cannot be fully automated since models frequently fail to contain the complete information, while other information is hidden in heads and software systems.

System Requirements

From what has been said so far, we can derive some requirements for digital preservation of PLM data and related knowledge:

- **Different kinds of data and knowledge need to be preserved:** In addition to the pure CAD product data (geometry, netlist), design knowledge also needs to be preserved and made available for future access.
- **Different kinds of data and knowledge need to be preserved independently from each other:** Different data and knowledge have to be archived differently. For some of them, standardized formats may exist which can be used for long-term archiving (e.g., STEP for geometry data); others may need to be stored in native form because no globally agreed standard model exists.
- **Models and ontologies need to be preserved themselves:** Data and knowledge are always captured and stored according to a certain data model. This is particularly true of design data which are highly structured and can be interpreted only in the context of a data model. These models change over time, either in an evolutionary way or by a replacement with a new model.
- **Use of public knowledge in standardized ontologies:** To ensure the understanding of the meaning of data, the reference to external dictionaries, classifications and ontologies should be supported and recommended.

Information Architecture

Based on these requirements, we propose the following architecture for the information models of a PLM system (see Figure 1) which handle all PLM data and design knowledge in the same way; we regard them as data which are based on a specific model. This model may be of any kind, for example a relational database model, an XML schema or DTD, an OWL ontology or a PLIB product ontology (ISO 13584-42, 1998). PLM data are frequently described by a proprietary model in a PLM system. In Figure 1, the models are represented by double-sided rectangles and the corresponding data are represented by database symbols.

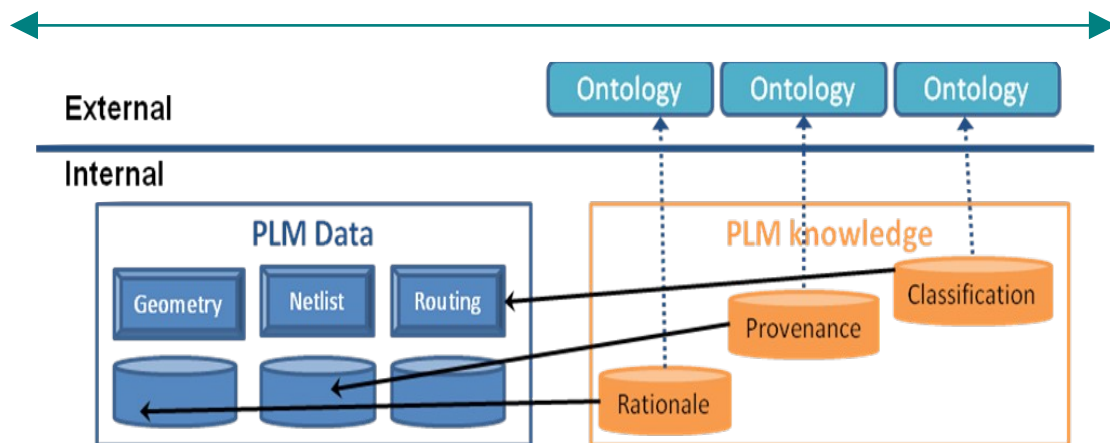


Figure 1: Architecture of PLM information models.

Thus, different representations of PLM data (e.g., geometry, netlist, routing) are independently represented and modelled. In the same way, different kinds of design knowledge like design rationale or provenance information are independent from each other and this knowledge is attached to PLM data. The references to public knowledge (which may be represented by global ontologies) are depicted as dashed arrows and represent a very important aspect of the architecture. These ontologies will have some standard status and have been built in a consensual process by several people, companies and organizations. It is assumed that these “public models” will have a much longer lifespan than local models; and, in particular, that they are understandable and usable by third-party tools which is not guaranteed for the proprietary models of software vendors. Design knowledge can be related in various ways to such external ontologies:

- **Annotation of meaning to PLM objects:** In this case, single data elements are related with concepts in the external model. For instance, it might be useful to annotate a set of geometrical instances with the information that these objects represent a punching hole in the complete geometry of a complex mechatronic printed circuit board. This situation is illustrated in Figure 1 by the link between knowledge instances (provenance and rationale knowledge as an example) on the right side and the PLM data representation. Actually, it would be implemented by creating an instance from the external ontology concept in the system and relate this instance object with the (set of) objects which are supposed to be annotated.
- **Mapping of concepts in internal models with concepts in external models:** In this case, concepts of the internal model are directly related to concepts of the external model. Thus, we provide a mapping between the different models. For such a mapping, different kinds of links may be used for expressing the relationship between the concepts in the different models (e.g., equivalence, subsumption, etc.).

Whereas in the first case (annotation) such a link specifies a fact for a specific instance of a concept, the fact is valid for all occurrences of the local concept in the second case. Both forms of relationships add additional semantics to the model or data elements which give them a publicly recognized meaning. Two example scenarios help to understand the problems that arise if such additional semantics are archived:



Reuse Use Case Example

An entity of a product design is annotated with product semantics. The entity is a screw which is selected from a product catalogue. The product catalogue itself is based on an abstract product classification that is written down as an ontology. The screw has an abstract property named *length* (originating from the product classification) and a concrete value of 5cm (originating from the product catalogue). This semantic information is attached to the design object and archived. Five years later, the semantics of the product classification standard is refined. The property *length* is divided into two properties named *bodyLength* and *headLength*. Given this development, the question inevitably arises as to which meaning was associated with the archived value of 5cm. It might be the length of the screw including the screw head or it might be the length of only the screw body without the head.

Search Use Case Example

An entity of a product design is annotated with design rationale. The ontology employed allows for the specification of *arguments* either in favour or against this design (Medeiros et al, 2005). The arguments and the specific ontology version are then archived. Five years later, the ontology has evolved; both *counter-arguments* and *arguments* are now captured. Based on the current ontology version, the contemporary application software allows searching for product designs with specific *arguments* and *counter-arguments*. However, it is also necessary that those product designs be searched even though they were archived five years ago based on the old ontology. For example, if a query is executed for a counter-argument that includes the term “costs”, this query also needs to be mediated for archived arguments.

In both cases, semantic evolution causes interoperability issues and results in semantic obsolescence. It is necessary to tackle this problem since knowledge evolution is very likely to occur both in the near term and the long.

Digital Preservation on the Basis of the Proposed Architecture

Besides all the aspects of physically retaining data in a processable form which are dealt with in other projects (LOTAR, MOSLA, VDA), on the logical side, digital preservation of engineering data should support the reuse of a design in an unknown future design system. Accordingly, the data and design knowledge produced by a PLM system today have to be captured in a way that it can be integrated into a system environment of a future system. Here we really have to “communicate with the future” (Mois, Klas & Hemmje, 2009); or we can also express this as: “We have to interoperate with the future.”

Under the assumption that a future PLM system will fit our basic information architecture, the basis of such an integration are mappings from the old models (of today) to the new models (of tomorrow). These mappings specify in which way concepts in the source model are related to concepts in the target model. The mappings have to be specified in a way that they can drive the migration of instances and they may even define some consistency rules which can be used to check whether such a migration has been performed successfully ([ASD-STAN PREN 9300](#)). The mapping process (Euzenat & Shvaiko, 2007) is depicted in Figure 2 where a matcher produces alignments that are used by a translator to migrate ontology instances.

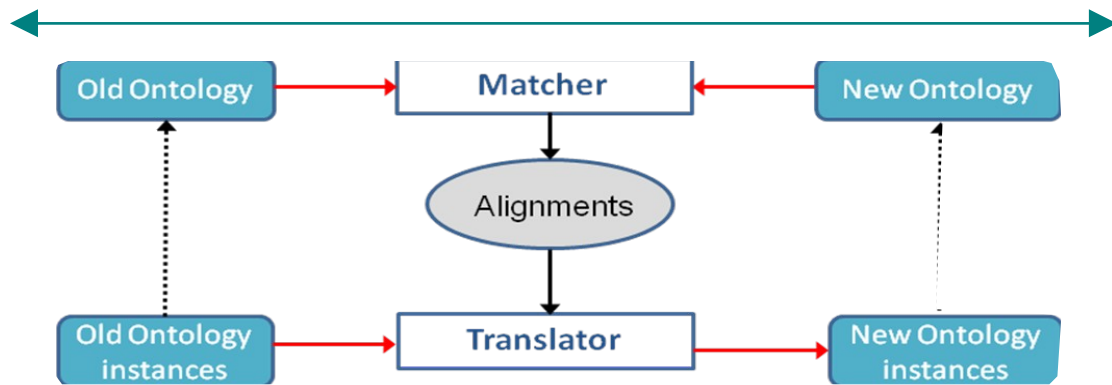


Figure 2: Ontology mapping via ontology alignments processing.

Unfortunately, the production of mappings is not something that can be done completely automatically; much research activity in database integration and ontology alignment (Kalfoglou & Schorlemmer, 2003) indicates that some human intervention is inevitable. Therefore the alignment with public ontologies is of utmost importance because it can support the reuse of engineering data in many ways:

- **Explication of hidden information:** Very often, the engineering data themselves only contain part of the whole story because some information may not feature in the data at all and remains hidden. Examples are specific assumptions about data which are only implemented in the application software but which cannot be derived from the pure data. Similarly, people very often know more about a design than is represented in the data and the design knowledge (e.g., why has it be done that way, what was the underlying reason for taking such a decision, and what does that mean for the modification of a design). By annotating engineering data and design knowledge, this hidden information may become more explicit and thus be saved for reuse in the future.
- **Facilitation of mappings:** If external ontologies are in wider use because they are approved by a big community, then there is a good chance that some mappings exist of such a public resource to other publicly available models of a similar nature. Therefore, if the models or their instances in a PLM system are related to publicly available ontologies, then the likelihood is much bigger that reusable mappings to other public ontologies may exist. Such a mapping could be used to facilitate the integration of the old model into the new system.

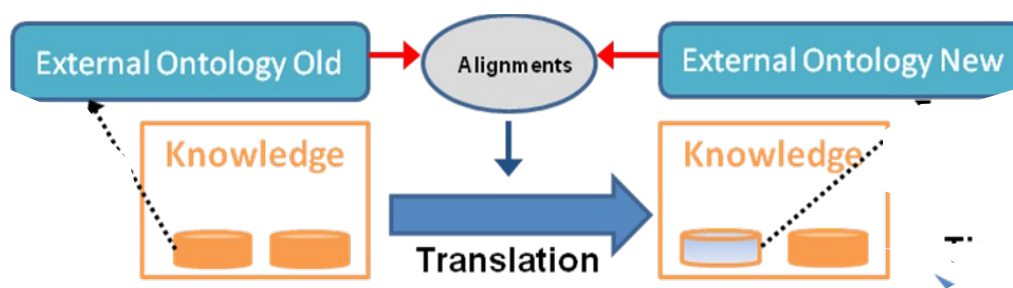


Figure 3: Mapping between knowledge of today and tomorrow with external ontologies.



In Figure 3, the knowledge of the old system on the left has to be migrated to the future system on the right by using ontology-matching functionality. Consequently the various knowledge instances have to be migrated to instances which comply with the corresponding future model. In some cases nothing has to be done because the knowledge does not reference ontologies. In other cases, the references to external ontologies have been changed and the knowledge needs to be migrated. If the old model is related to an external ontology, then the two following possibilities might help in building the mapping:

1. The external ontology is still valid and the corresponding model in the future system is also related to this ontology.
2. The external ontology has evolved over time and the corresponding model in the future system is related to a new conceptualization or to the successor of the original ontology. In this case, information about the changes probably exists and can be used.

In the second case it is possible to generate a (usually incomplete) initial mapping by connecting the old ontology to the new ontology in order to migrate the knowledge instances. It must be possible to extend or modify such mappings in order to meet company-specific needs, because company-specific knowledge was applied.

If a relationship to an external ontology does not exist, then it becomes more difficult to create such a mapping. The greatest likelihood is that a software vendor who provides the software to deal with the specific PLM data or design knowledge has changed the model in an upward-compatible way, or provides some reorganization software which migrates the old data to the new model. In the case of replacement of the system by another vendor's system this might not exist – and if the models are proprietary, then there is very little chance of creating the mapping and the migration rules. It is essential therefore to transform product designs to vendor-neutral formats like STEP and to attach knowledge to such standardized formats in order to migrate archived knowledge instances in the case of reuse.

Conclusion and Outlook

Engineering knowledge that is appropriate for archiving is created during several product lifecycle phases. This paper has investigated issues that arise when product designs are archived which are annotated with engineering knowledge and which need to be reused by future engineers. External semantic representations like ontologies that formally describe knowledge change over time. This semantic evolution results in a loss of knowledge since this knowledge is only consistent with specific versions of the semantic model. The solution to this threat of semantic obsolescence is a migration of knowledge instances via ontology mapping. Such knowledge migration functionality has to be embedded into existing archiving architectures like the OAI preservation planning functionality. Furthermore, it has to be investigated whether the ontology mapping documents themselves need to be archived since they too evolve over time. Future work will also include the identification of an appropriate ontology-matching approach for the engineering industry and the application of an ontology change language that defines the transformation rules for the implementation of knowledge migration.



Acknowledgements

This paper is supported by the European Union in the 7th Framework within the IP SHAMAN.

References

- Ahmed, S. (2007). An industrial case study: Identification of competencies of design engineers, *Journal of Mechanical Design*, Vol. 129.
- Ahmed, S., Bracewell, R., & Kim, S. (2005). Engineering knowledge management. In *Symposium in Honour of Ken Wallace*. Cambridge, UK, Retrieved August 5, 2009, from <http://redigering.sitecore.dtu.dk/upload/institutter/ipl/sektioner/2005kenchapter.pdf>
- ASD-STAN PREN 9300. *Long-term archiving and retrieval of digital technical product documentation such as 3D, CAD and PDM data within the aerospace industry* “AECMA-STAN LOTAR” PART 100, Part 200: DATA DOMAIN SPECIFIC PARTS.
- Ball, A., Patel, M., McMahon, C., Culley, S., & Green, S. (2006). A Grand Challenge: Immortal Information and Through-Life Knowledge Management (KIM). *International Journal of Digital Curation*, 1(1), pp.53–59.
- Brunsmann, J., & Wilkes, W. (n.d.). *State-of-the-art of long-term archiving in product lifecycle management*, submitted for publication
- Euzenat, J., & Shvaiko, P. (2007). *Ontology matching*, Springer-Verlag
- Fricke, M. (2009). The knowledge pyramid: A critique of the dikw hierarchy. *Journal of Information Science*, 35(2), April 2009, pp.131–142.
- Heutelbeck, D., Brunsmann, J., Wilkes, W., & Hundsdörfer, A. (2009). Motivations and challenges for digital preservation in design and engineering. *Proceedings of First International Workshop on Innovation in Digital Preservation*.
- ISO 13584-42. (1998). *Industrial automation systems and integration -- Parts library - Part 42: Description methodology: Methodology for structuring part families*.
- Kalfoglou, Y., & Schorlemmer, M. (2003). Ontology mapping: The state of the art. *The Knowledge Engineering Review*, 18 (1), pp. 1-31.
- Klein, R. (2000). Towards an integration of engineering knowledge management and knowledge based engineering. In M. Spitalni (Ed.), *Proceedings of the CIRP Conference on Advanced Engineering Information Systems*, Haifa, Israel.



- Kopena, J., Shaffer, J., & Regli, W.C. (2006): CAD archives based on OAIS. *Proceedings of ASME Computers and Information in Engineering Conference*, DETC2006-99675, Philadelphia.
- Lubell, J., Rachuri, S., & Mani, M., & Subrahmanian, E. (2008), Sustaining engineering informatics: Toward methods and metrics for digital curation, *The International Journal of Digital Curation*, 3(2), pp. 59-73.
- Medeiros, A.P., Schwabe, D., & Feijo, B. (2005). Kuaba ontology: Design rationale representation and reuse in model-based designs. In *Proceedings of the 24th International Conference on Conceptual Modelling*, Klagenfurt, Austria, *Lecture Notes in Computer Science 3716*, pp 241-255, Springer, October 2005, ISBN 3-540-29389-2.
- Mois, M., Klas, C.P., & Hemmje, M. (2009). Digital preservation as communication with the future. In *DSP 2009, Proceedings of the Fifth International Conference on Web Information Systems and Technologies*.
- Rachuri, S. (2007). Long-term sustainment of digital information for engineering design: Model driven approach. Presentation at *Atlantic Workshop on Long Term Knowledge Retention*, February 2007. Bath, UK. Retrieved August 5, 2009, from UKOLN Website:
<http://www.ukoln.ac.uk/events/ltr-2007/programme.html>
- Regli, W. C., Hu, X., Atwood, M., Sun, W. (2000). A survey of design rationale systems: Approaches, representation, capture and retrieval. *Eng. Comput. (Lond.)* 16(3-4), pp.209-235.
- Wilkes, W., Brunsmann, J., Heutelbeck, D., Hundsdörfer, A., Heidbrink, H-U., & Hemmje, M. (2009). Towards support for long-term digital preservation in product life cycle management. In *iPres 2009, October 2009, SanFrancisco*.