

PRODUCT DEVELOPMENT ONTOLOGY IN TRACEABILITY IMPLEMENTATION FRAMEWORK

N. Pavković and M. Štorga

Keywords: ontology, traceability, design process, knowledge capture and reuse

1. Introduction

The recognition of the importance of the enterprise's knowledge identification, acquisition, development, dissemination, use, and preservation in the modern product development process, have addressed lot of research questions from either a organisational or a technological point of view [Štorga 2005]. The main challenge from the organisational viewpoint is in effective use of personal knowledge and the qualitative and quantitative adaptation of this knowledge toward a changing business environment. The technological viewpoint, by contrast, deals with questions about what information technology should be provided to support effective knowledge management. During our research on building product development ontology [Štorga et al. 2005] we have found that the effective knowledge management requires a hybrid solution, one that involves both people and technology. In this article we will argue a need for a design projects' repository as the core for supporting, sharing and reuse of individual and corporate knowledge [Pavković et al. 2006]. Built on the top of such repository, intelligent knowledge management services should actively provide the user working on a knowledge-intensive operational task with all the necessary and useful information (see Figure 1).

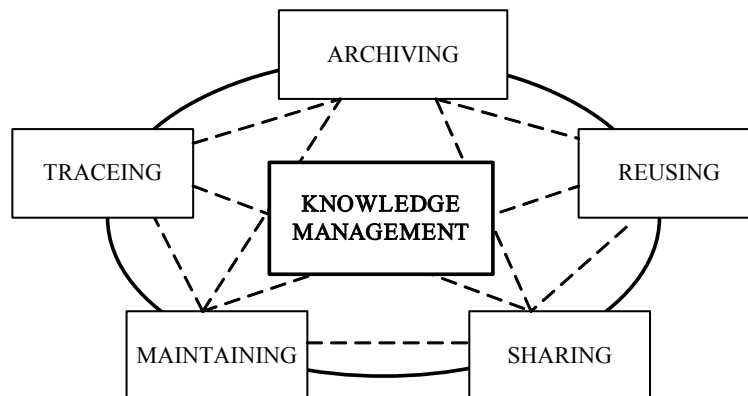


Figure 1. The basic knowledge-management activities

The proposed view of a design projects' repository partially grew out from our practical experiences and also conforms well to the results of our previous theoretical research. In order to enhance the enterprise's competitiveness by improving the way it manages its knowledge, the main efforts should

concentrate on knowledge capturing and preservation, which should be based primarily on explication of tacit knowledge by using the product development ontology [Štorga 2005.]. This process should allow participants of the product development projects to reuse, in a relevant way, the knowledge of a given domain previously stored and modelled, in order to perform new tasks [Pavković et al. 2006.], especially by the exploitation of existing documents, which are based primarily on natural language. In this article, some findings from the case-study implementation of the product development ontology are shown.

2. Levels of the product development ontology implementation

In order to make useful explication of tacit knowledge we propose a formalization of a domain knowledge structure by building the domain ontology. Literature proposes four levels of modelling that we have followed in presented case study: epistemological-, domain-, application-, and project modelling level (see Figure 2) [Štorga 2005.].

EPISTEMOLOGICAL LEVEL	GENERAL ONTOLOGY				
DOMAIN LEVEL	PRODUCT DEVELOPMENT ONTOLOGY				
	PRODUCT/DESIGN ONTOLOGY			DEVELOPMENT PROCESS ONTOLOGY	
APPLICATION LEVEL	TRACEABILITY	CONFIGURATION	ASSEMBLING	... ETC.	
PROJECT LEVEL	COMPANY 1	COMPANY 2	COMPANY n

Figure 2. Ontology building levels

The epistemological modelling level is established by defining the general set of vocabulary entities and possible associations between them in order to logically correct describe the situation in a domain of discourse on the highest level of abstraction. In particular research the Standard Upper Merged Ontology (SUMO) [Storga et al. 2005., Niles et al. 2001.] was chosen as a foundation base of the most basic and universal concepts that are generic, abstract and philosophical, and therefore are general enough to address (at a high level) a broad range of different domain areas. The domain modelling level is established by a set of formal informational structures that describe a situation in a particular domain of discourse. Domain level should be generic in a given area (product development domain in this case) and constrained by the content of the epistemological foundation. At this level, the knowledge about the product/design and the development process were considered from the different viewpoints as reasoning, synthesis, selection, documentation, business aspects, organizational responsibilities, workflow etc.

The application modelling level should be a reuse and extension of a domain level and should be specific for an application in particular domain (for example traceability, configuration design, sustainable design, design for assembling etc.). Terms at this level are organized to characterize specialization of common features specific for application of a domain model in particular use case. In presented research we have extended the proposed product development ontology with terms necessary for achievement of the traceability during a specific design episode in product development. The project level extends an application domain modelling level to include knowledge found within specific implementation project, depending on the situation and requirements of the concrete product

development scenario. These additional concepts could arise from the specific synonyms for the general defined terms that are used in particular company, customer specification of design policies, company internal procedures (organizational, safety and confidential tasks, quality standards), company procedures related to implemented PDM or ERP systems, etc.

3. Traceability in product development process

During product development designers need traceability carried by traces of the design routes, because they want to reuse existing design knowledge along meaning, reasons, arguments, documentations (proofs, specifications), choices, critique, consequences etc. One of the major problems in modelling of design knowledge on design routes is in finding an appropriate set of formalized concepts and rules that the knowledge should refer to - product development ontology [Štorga 2004.]. Therefore, the overall goal of presented research was to satisfy a need for models and tools to effectively support the formal representation, capture and exchange of product development related data, information and knowledge that are unavailable in traditional CAE tools (mostly non-geometric knowledge as information about development process, physical, behavioural or functional decompositions of product, specifications, and various kinds of relationships among these entities...).

3.1 Practical requirements

In recent years, we have performed several case studies, prototype developments, and evaluations concerning knowledge based systems for supporting complex tasks in product development domains. Discussions with industrial partners and our growing understanding of their particular needs turned us away from the approach centered around the idea of an autonomous problem-solver. We have moved instead to the idea of a three-tier product development working environment, which emphasizes the active and intelligent assistance of the designers by different mechanisms for providing, maintaining, and distributing relevant information and knowledge [Štorga 2005.].

Our experience as well as experiences from the similar projects showed that the following requirements are crucial for their success of knowledge management systems in industrial practice [Kühn, 1997.]:

- Capture and systematic organization of the knowledge from various sources.
- Minimization of the knowledge engineering for the outermost users.
- Exploiting user feedback for knowledge maintenance and evolution.
- Integration into existing working environment.
- Active presentation of relevant knowledge.

3.2 Proposed implementation architecture

Following the literature [Abecker et al., 1998., van Elst et al., 2002.], the possible traceability and knowledge management implementation architecture is depicted on the Figure 3. This approach models and executes development process and tasks on the application layer. When a user in product development process recognizes need for the knowledge within the actual flow of work, a semantic query to the description level must be derived. This query is instantiated and constrained as specifically as possible on the basis of the actual user needs. In the opposite way, the user can also store new knowledge created within a given working situation in a contextually enriched form. One of many possibilities for realizing the proposed architecture application layer can be conventional business-process systems and workflow-management systems.

The description layer serve for the mapping from the application layer specific information needs to heterogeneous source layer via a uniform access and utilization method on the basis of a rule-based product development ontology. Actual work context is mapped onto expressions in the traceability knowledge base resulting with the appropriate assertions and queries. The vocabulary for the description layer comes from the product development ontology, extended by attributes for applying traceability – such as timelines, sources, evolution links, specific dependencies between design objects etc. Because description layer relies substantially on existing knowledge, the source layer is characterized by a variety of sources, heterogeneous with respect to several dimensions concerning

form and content properties. This layer comprises manifold information and knowledge sources, ranging from machine-readable formal representations to human-readable informal representations.

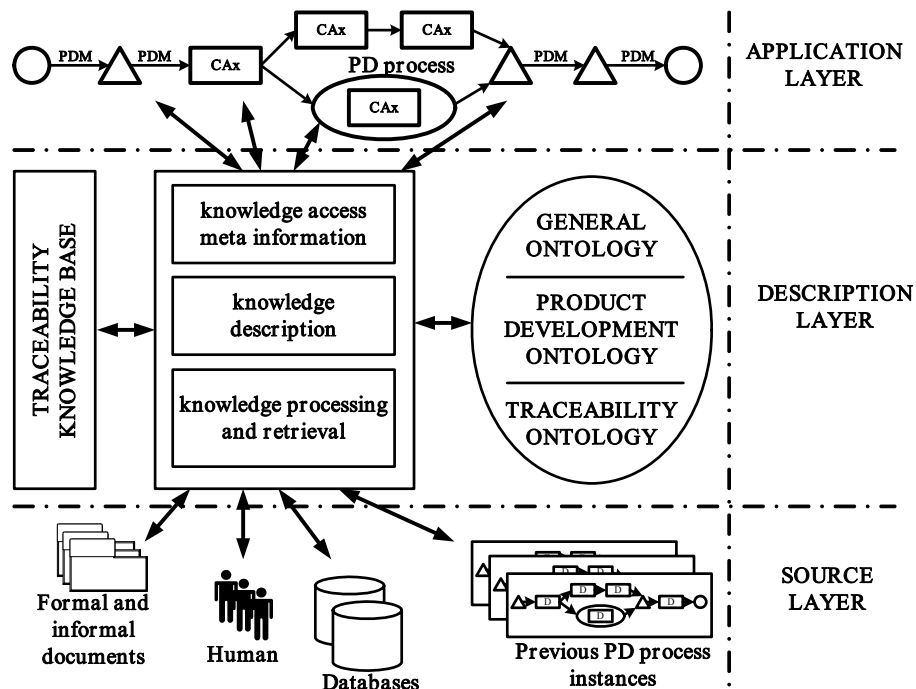


Figure 3. Proposed implementation architecture

4. Conceptual issues of process ontology implementation

This section of paper discusses the conceptual and development issues of implementation of proposed architecture (figure 3).

4.1 The role of design process ontology

The proposed implementation architecture built around product development ontology provide a feature that can help in solving of the existing difficulties for achieving traceability:

- ontology provide unique formal syntax and semantic definitions;
- different ontology abstraction levels can be used for avoiding the situated nature of traceability;
- semantic rules and deduction mechanisms can resolve complexity of traces among design routes;
- semantic interoperability based on ontology can be a communication medium between heterogeneous design object sources.

4.2 Initial concepts for the design process ontology implementation

The following chapter describes the first development phase of software tool for design process workflow management which should implement the proposed product development ontology application layer (figure 3). Initially, four entities make the core of the proposed workflow model: design plan, design process step, operation and document. Each of these entities is categorized to enable implementation of different behaviour for various kinds of entity instances. Short definitions and descriptions of those basic entities are given as follows:

Design plan

The term "design plan" encompasses the data structures and operations that constitute the model of design process execution (progression) flow in the real time. Design plan is represented with directed graph. The design plan is a formulated program of steps and operations. Execution of the plan leads to the desired goal if and only if the preconditions for the plan execution are fulfilled. Operation in the plan is an activity that generates effects if the preconditions for the operation execution are satisfied.

Design process step

In the proposed model, the design process step is defined as a combination of set of operations A_j which transforms an object O , being in a state S^O_i , in the next (new) state S^O_{i+1} :

$$E = \{A_n : S_j (A_j (S^O_i) = S^O_{i+1})\}$$

This definition can be extended to a set of objects, i.e. a set of operations transforms a state of each particular object in a new state. In such a process operation may generate new objects in the domain of the design process model, and furthermore, operations may change the state of newly generated objects. The design plan step includes: checking of preconditions, list of operations, checking the "postconditions" and deciding about the next step.

Operation

"Operation" is here considered and defined in the context of generating, transforming and representing of information (only in the context of informatics), but not in the context of methodology of design and design procedures. Therefore, the primary criterion for proposed operation classification is the type of "affecting" on design process model objects. The operations are being created and planned before the design process, being realized in the duration of the design process. Operations mainly operate on product documentation objects.

Document

Entity "product document" models particular kind of set of written information about product being designed. This entity models the "existence" of document, i.e. encapsulates all notions and events from the life cycle of a particular document that contains a set of information about product.

5. Design process workflow model as a part of ontology implementation framework

The proposed design process workflow model is based on authors' previous research experience in the area of script-based planning of design process. Very interesting considerations about planning the design process may be found in [Giaopulis et al., 1995]. The important research question is to what extent a design process could be planned? Real design process environments generate dynamic situations - they can change while the plan is being executed in a way that makes the plan invalid [Pollack, 1992]. As if this is not enough, real environments may also change in a way that do not invalidate a current plan, but instead, offer new possibilities for action. However, we believe that the concept of script-based plans is a promising approach in building design process workflow model able to fulfil the requirements for ontology implementation framework. In most cases designers wish to reuse captured design knowledge to adapt past solutions and apply these to current problems, and novice designers may wish to understand lessons from previous experiences (Ahmed S., 2005).

When starting a new design project, the designer opens the particular design plan skeleton containing a set of predefined steps, operations and their sequence. A design plan skeleton is the structure that should represent, collect and organize every type of knowledge and data that is or should be known prior to start of a new design project. The term "skeleton" is used to emphasize that this is an initial structure for building the complete recording and documentation of finalized design project. Relying on the previous research work [Pavkovic, Marjanovic, 2001] we assume that it is possible to create skeletons of design plans for adaptive and variant designs. The most experienced designers should create skeletons of design plans for particular classes of design projects in particular design office. This way experienced designers should generate the knowledge about how some particular project

should be solved by their own experience. As previously stated in this stage the main task is to define what (kind of) objects should be traced and what (kind of) links are needed between those objects. Figure 4 shows schematically the process of knowledge flow from generation to storage.

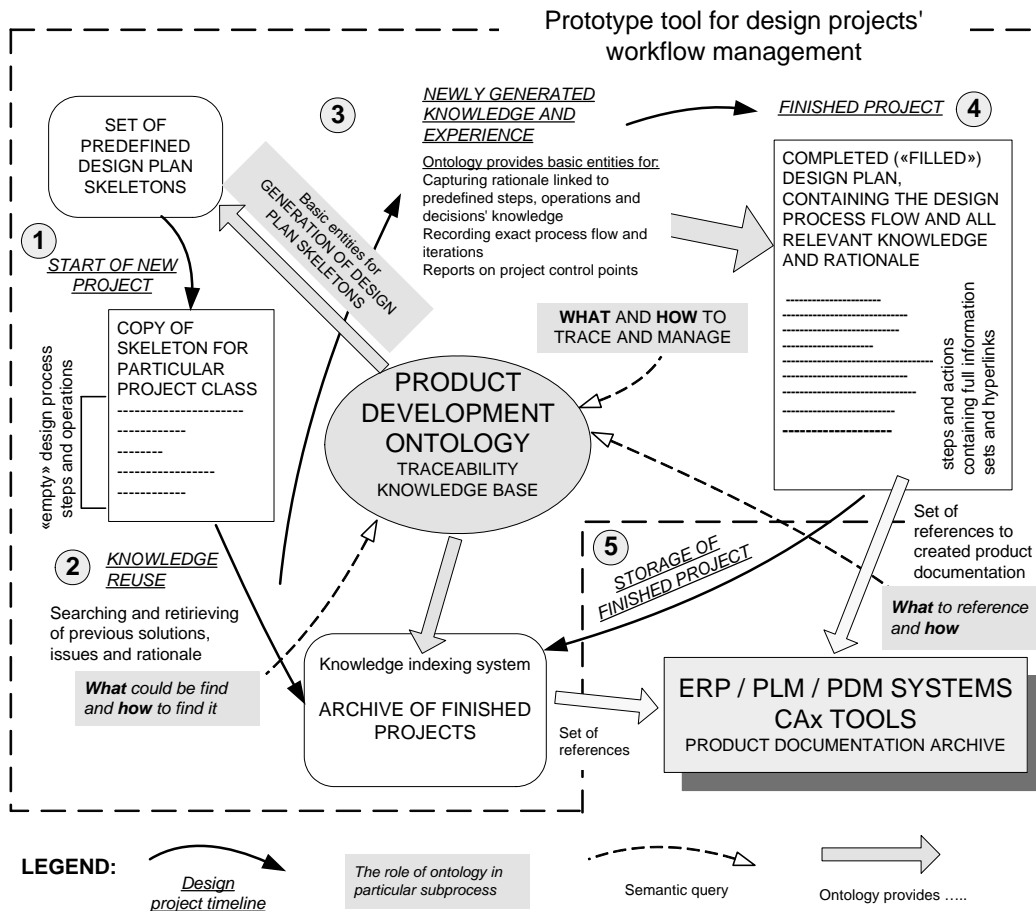


Figure 4. The flow of stored and newly generated knowledge in the process of the proposed system usage

5.1 Traceability achievement

Through the timeline of design project realisation, the initial design plan skeleton is being updated, upgraded and "filled" with created traces that are result of product development activities, designers' operations, decisions, reasoning, events, etc. At the end of the design project skeleton is transformed to completed design plan with explicit design items and routes recorded and documented. Such design plan is a complete description of a finished design project. Finished design plans are being stored in archive.

5.2 Knowledge reuse

By choosing the appropriate design plan skeleton, the designer accesses and reuses expert knowledge and experience. The other way of reusing knowledge is by extraction of design items from archive of finished projects. Extraction could be done by following, querying, and navigating through recorded and documented traces as also by querying and accessing linked documentation in digital (or some other form). Predefined and organized reuse of stored knowledge in a proposed system becomes an

integral part of the design project solving process. Predefined and organized reuse is achieved through special items in design plans that tells the designer what is (or should be) available for searching and where this knowledge should be found. Some simple examples of these situations could be: changes on existing solutions, reusing of the existing solutions in new projects, viewing problems and solutions on previous similar projects, analysing testing reports, etc.

6. Issues of proposed framework usage in practice

An essential issue for the proposed system usage is to find a good balance between amount of information that a designer should record in system and the time he spends for this purpose. To achieve this goal, two components are of primary importance – interfaces and the product development ontology. A lot of research work and evaluation and validation remains to be done to develop good and intuitive interfaces (and procedures) that would save designers' valuable time. *The role of ontology is to provide design community with platform for better communication. There should be no misunderstandings about what is being recorded or what is being searched for.*

While developing the model structure, it becomes evident that a test implementation will require a lot of efforts in coordinating the perception of model elements and structure among potential users. The other important question is how to ensure the designers will write down the appropriate explanations (rationale) when they are necessary. One of the key requirements for the proposed system is that it should take as little as possible of the designer's time. Previously mentioned facts affect many issues in preparing the terminology, the procedure of system usage and the interfaces.

7. Conclusion and further work

The proposed approach to the achievement of the traceability in product development is at a conceptual stage. There are still many open questions and some unclear points to be clarified in a future. The aim of the authors was to make a first step into the complex problem area with the new ideas, for getting response and advises for further work. It is believed that future research in this field will lead to interesting clarifications in more different areas: design rationale patterns for representing common elements of design reasoning structures in a form suitable to trace and reuse, distributed traceability mechanisms focused on making context among heterogeneous systems reusable via semantic mapping and context management, and active traceability knowledge bases with mechanisms that should note engineers if their area of interest is affected by any new developments.

The research projects in the area of computer-based modelling of product development process were mainly focused on issues that are islands in the area (sometimes isolated). The attempts of development of general integrated frameworks were numerous, but it seems that there are still no good results, i.e. widely acceptable and usable systems. It is obvious that such a task is very huge and complex. It is difficult to believe that isolated research teams could come to widely acceptable solution. Using ontology, a distributed design community could easily collaborate in developing the design process model. Maybe this fact could initiate the process of creating a theoretical framework for integrated computer design process support that will be widely accepted in a design community.

Acknowledgements

This research is part of funded project "Models and methods of improving the computer aided product development" supported by the Ministry of Science and Technology of the Republic of Croatia. A part of conceptual issues and the practical evaluation of the presented research are performed in co-operation with KLEX d.o.o., Zagreb, a mechanical engineering design office.

References

- Abecker, A., Bernardi, A., Hinkelmann, K., Kuhn, O., Sintek, M.: "Toward a technology for organizational memories", *IEEE Intelligent Systems, IEEE computer Society, IEEE Inc., 1998., pp 40-48*
- Ahmed, S., "Encouraging reuse of design knowledge: a method to index knowledge", in *Design Studies – article in press*

Giaopolis A., Schlüter A., Ehrlenspiel K., Günther J., (1995) "Effizientes Konstruieren durch Generierendes und Korrigierendes Vorgehen", *Proceedings of 10th ICED conference in Praha, Volume 2, Schriftenreihe WDK 23, Praha, pp. 477-483.*

Kühn, O., Abecker, A.: "Corporate Memories for Knowledge Management in Industrial Practice: Prospects and Challenges", *Journal of Universal Computer Science, Special Issue on Information Technology for Knowledge Management, Vol. 3, No. 8, 1997, pp. 923-954*

Mortensen, N.H.: "Design modelling in a Designer's Workbench – Contribution to a Design Language"; *PhD thesis; IKS 00.02.A; DTU; 1999.*

Niles, I., Pease, A.: "Origins of the IEEE Standard Upper Ontology"; *Working Notes of the IJCAI-2001 Workshop on the IEEE Standard Upper Ontology; Seattle; pp. 37-42; 2001.*

Pavković N., Klarić I., Štorga, M., "Design projects' repository – a tool for design process guiding, knowledge capturing and reusing", *Proceeding of the TMCE 2006, Millpress, Rotterdam, 2006.*

Pavković N., Marjanović D., "Considering an object-oriented approach to the design process planning", *International Journal of Technology Management, Vol. 21, No. 3/4, 2001.*

Pollack, M.E. (1992) "The uses of plans", *Artificial Intelligence, Vol 57, No1, pp. 43-68.*

Štorga, M., Marjanović, D., Bojčetić, N., "Considerations on IT systems interoperability in product development", *Proceeding of the TMCE 2004, Millpress, Rotterdam, ISBN, 2004.*

Štorga, M., (2004), "Traceability in product development", *Proceedings of 8th International design conference – DESIGN 2004, Dubrovnik, 2004.*

Štorga M., Andreasen M. M., Marjanović D., "Towards a Formal Design Model Based on a Genetic Design Model System", *Proceedings of ICED 05, Melbourne 2005.*

Štorga, M., "The Design Ontology – Contribution to the Design Knowledge Exchange and Management", *PhD Thesis, University of Zagreb, 2005.*

Van Elst, L., Abecker, A., "Ontologies for information management: balancing formality, stability, and sharing scope", *Expert Systems with Applications 23, pp. 357-366, 2002.*

Ass. Prof. Neven Pavković
 University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture
 Product design and development chair
 Ivana Lučića 5, 10000 Zagreb, Croatia
 Tel.: +385 1 6168 545
 Fax.: +385 1 6168 284
 Email: neven.pavkovic@fsb.hr
 URL: <http://www.cadlab.fsb.hr>