INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED'07
28 - 31 AUGUST 2007, CITE DES SCIENCES ET DE L'INDUSTRIE, PARIS, FRANCE

ONTOLOGY-BASED MODELS FOR DESIGN RETRIEVAL AND ANALYSIS

Fabrice Alizon[1, 3], Jyotirmaya Nanda[2], Steven B. Shooter[1]* and Timothy W. Simpson[3]
[1]Bucknell University, Lewisburg, PA 17837
[2]Intelligent Automation, Inc., Rockville, MD 20855
[3]The Pennsylvania State University, University Park, PA 16802

ABSTRACT
For today's product designers, many new or improved design tools and methods have been developed to support product family design. As a result of these developments, descriptions of design data continually evolve and get revised, giving rise to compatibility and sharing issues between these tools and methods and the design repositories created to support them. Each time a design tool evolves, designers must manually modify the design repository to fit the tool's needs, and in this same context, designers must manually manage the sharing of design information. In this paper we propose an ontology-based model to partially automate the management of design tool evolution and the subsequent revision of the supporting design repository. The proposed ontology-based model assists designers in assessing design compatibility, sharing data, and ultimately ensuring "plug and play" capability for each specific design tool, leading to better design information retrieval and analysis. A case study based on a family of single-use cameras illustrates the proposed model.

Keywords: Design Repository, Ontology Development, Design tools, Information interoperability

1 INTRODUCTION
For today's product designers, many new or improved design tools and methods have been developed to support product family design. These new and improved design tools often require different information than previously needed, or they may require that the information be prepared in a different format. As a result, design repositories that support these tools often need new data descriptions. For instance, the product platform design field has recently shown an increasing number of tools and associated methodologies to improve the design of platform-based products. The resulting tools need new input to stratify new outputs [1], and design repositories need to be accessible in new and unanticipated ways. To manage this issue, most current approaches manually modify the design repository and adapt its data description (output) to fit the needs (input) of the design tools. Thus, it is necessary to specify new processes and revised models to improve the communication between tools and the repository. We believe that these models should be based on ontologies for both the data description (product) and design parameter descriptions (design tools). Ontologies developed for many fields to establish common vocabularies and capture domain knowledge have proven to be an advantageous paradigm over recent years [2]. The use of ontologies to capture the semantics of design parameter descriptions to perform semi-automated analysis of design information across multiple products will yield similar benefits. In this study, we propose a new process to automatically retrieve information from the design repository based on the need of each design tool. This process is based on matching two ontologies (design data and tool parameters) that enable designers to ultimately "plug and play" the different tools.

Section 2 presents background and related work. The proposed ontology-based model is introduced and detailed in Section 3. A case study is performed on a family of single-use cameras in Section 4. Finally, closing remarks are given in Section 5.

2 BACKGROUND AND RELATED RESEARCH

2.1 Design representation and repository

The ability to define design artifacts in a way that makes it easy to manage information about the product during all phases of the product life cycle is crucial for product-oriented organizations. Greer, et al. [3] propose a component basis as the framework for the development of a standard naming convention of mechanical parts. The component basis uses a lexical scheme to identify major categories to define classification terms for mechanical components. Stahovich, et al. [4] developed an ontology of mechanical devices by emphasizing common patterns of behavior of mechanical components over structural representation. Stahovich, et al. [5] developed a program called SketchIT that employs a paradigm of abstraction and resynthesis based on qualitative configuration space. Nahm and Ishikawa [6] describe an integrated product and process modeling framework for the collaborative product design. Kirschman, et al. [7] proposed a taxonomy of elemental mechanical functions that can be used with many decomposition techniques. The taxonomy proposed a common language for designers to refer to the same function but was limited in its vocabulary and was short of a neutral format of capturing the information. Iwassaki and Chandrasekaran [8] focused on the task of design verification using both knowledge of the structure of a device and its intended functions. The National Institute of Standards and Technology (NIST) is involved in development of an intelligent design repository based on Data Language and a Design Representation Language [9,10]. Shooter, et al. [11] presented a model for the flow of design information that is sufficiently formal to eventually support a semantics-based approach for developing information exchange standards. This model was then expanded to establish a foundation for interoperability in next-generation product development systems [12]. More recently, the design repository at the University of Missouri – Rolla (UMR), following NIST's approach toward neutral data exchange, has implemented a XML-based approach [12]. The UMR Design Repository, following NIST's approach toward neutral data exchange, has also implemented a XML-based approach to import and export the product knowledge from the design repository [13]. Commercial Software packages provide some kind of hierarchical decomposition of product structure. A detailed analysis of these commercial products can be found in Ref. [14].

2.2 Ontology and Semantic Web

An ontology consists of a set of concepts, axioms, and relationships that describes a domain of interest. The concepts and relationships between them are usually implemented as classes, relations, properties, attributes, and values (of the properties/attributes) [15]. Ontology can be defined as [9]: "A formal, explicit specification of a shared conceptualization." Attributes "formal" and "explicit" enable the automatic machine-based interpretation of the conceptualization; "shared" enables the sharing, combination, and integrated use of ontological information [14]. To achieve this, ontologies need a language for semantic representation and reasoning. The W3C's Semantic Web initiative proposes a layered approach to a standard ontology language, OWL [16], which has been used in this paper for capturing ontologies. OWL, the Web Ontology Language, is designed for use by applications that need to process the content of information instead of just presenting information to humans. A detailed review of OWL and its applications can be found in Ref. [17].

Figure 1 illustrates a mapping between a design ontology and design artifacts. The development of common design ontologies helps capture the semantics and provides a standard vocabulary for creating and maintaining design artifacts within a product family. The inheritance-based representation using OWL helps in consolidating scattered information in a hierarchical structure and decreases the amount of information needed to describe design artifacts.

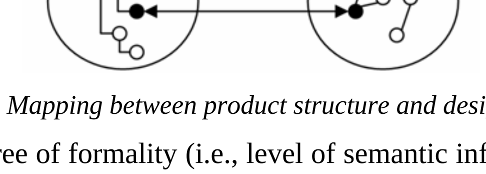

Figure 1: Mapping between product structure and design ontology

Figure 2 summarizes the degree of formality (i.e., level of semantic information) stored along with the data by a particular method of information storage and presents the technological maturity level of the methods. Ontologies, like OWL, are well suited for design knowledge representation by supporting reasoning outside the transaction context, i.e., avoiding a protocol specification to handle standard data format. Any knowledgebase expressed in OWL implicitly includes the axioms and definitions from OWL's ontology and facilitates greater machine interpretability. OWL is backed by Description Logic (DL) [16], enabling computers to interpret the semantics without human intervention. Also, software tools, irrespective of the subject domain, can provide support for the ontologies. Programming packages like Protégé [18] provide a graphical user interface for editing ontologies, and Jena [19] has application programming interfaces (APIs) for generic OWL manipulation and a rule-based inference engine.

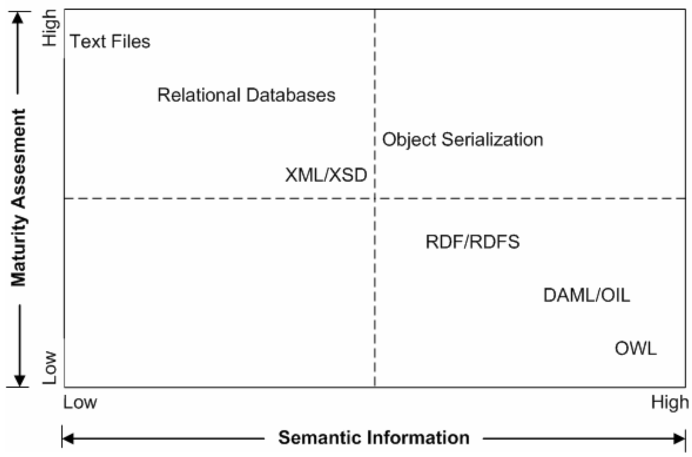

Figure 2: Methods of persistent information storage

2.3 Product Family Ontology Development Methodology

The Product Family Ontology Development Methodology (PFODM) [20], a novel methodology to develop formal product ontologies using the Semantic Web paradigm, is used here for development of both the product data and design tool ontologies. PFODM combines distinct, yet complementary, research in Formal Concept Analysis (FCA), Semantic Web, and Web Ontology Language (OWL). A structured methodology for product family ontology construction facilitates shared, consistent, and traceable ontology development within a diverse product development team. PFODM is useful for creating ontologies to support the sharing and analysis of design artifacts in a design repository.

2.4 Opportunities for this study

This new process and associated ontological model enable designers to:

- Automatically assess the compatibility between design repository data and parameters needed by design tools (output/input), ensuring that the tools can be used for design analysis;
- Share the same information through different design tools;
- Clarify the interface between data and tools to enable “plug-and-play” processes; and
- Automatically manage the evolution of needs (e.g., for new design tools).

Furthermore, this process also introduces ontology filtration (partial representation) in design to match data from the repository offer and design tools requirement.

3 PROPOSED ONTOLOGY-BASED MODELS

In this section the proposed ontology-based model combining design data and tool parameter ontologies is introduced and detailed. Product information is stored in the design repository, which then provides data for design tools that facilitate results that can be exploited by designers. The model focuses on the data description coming from a design repository and matching it against the parameter requirements of different design tools (see Figure 3). Currently, in a given system, the description of the same elements can be named differently. One of the aims in this new model is to avoid this non-interoperability issue. The development of an ontology can help to specify a generic interaction between a repository and tools.

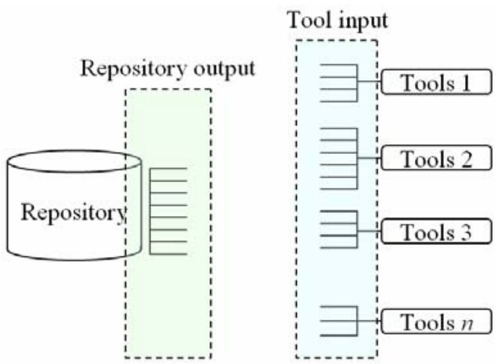

Figure 3. Illustration of the input/output network

The ontology development for design data description as well as for the design parameters is done using PFODM [20]. Figure 4 represents the generic OWL description of the class *Control Parameters*, which is composed of the necessary parameters needed to ensure the coherence between the repository offering and the tools' input requirements.

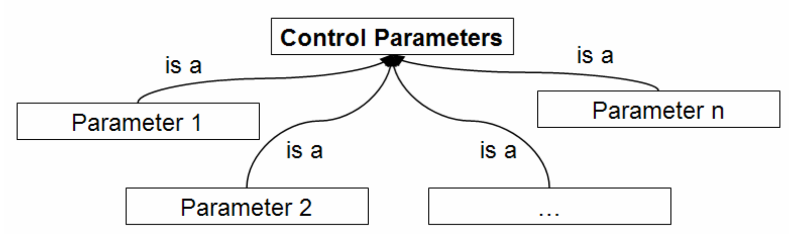

Figure 4. OWL Parameters for commonality analysis

The class hierarchy of the tools is shown in Figure 5. This Tools ontology is based on Input parameters, Output parameters, and Description plus all of the specificities of each tool considered as part of this study: Commonality versus Diversity Index (CDI) [21], Reuse Existing Unit for Shape and Efficiency (REUSE) Method [22], Product Differentiation Index (PDI) [23], and Family Differentiation Index (FDI) [23].

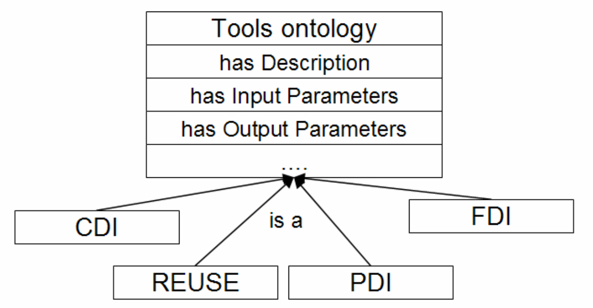

Figure 5. OWL Class hierarchy in Tools

Figure 6 represents the information architecture of the system: Tool Ontology - Parameter Template - Repository Ontology. This figure illustrates the location of the control parameters in the Parameter Template and in the Repository ontology at each level of abstraction (Family, Products, Components, and Functions).

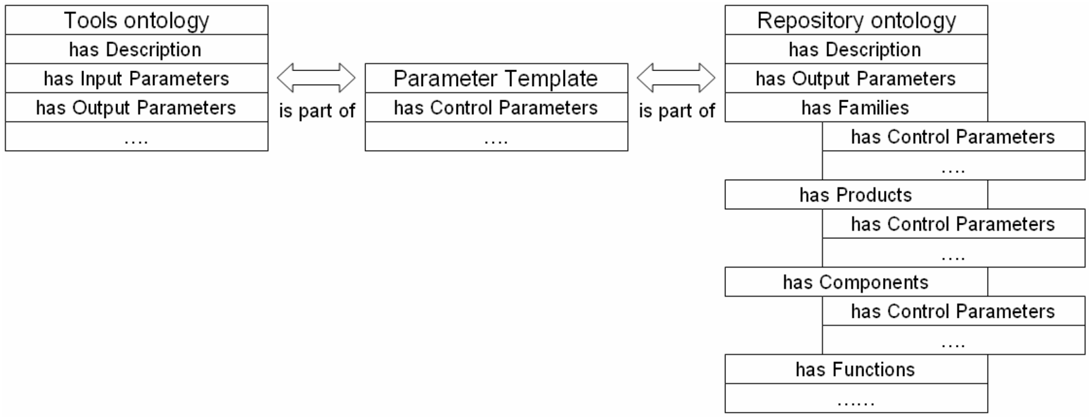

Figure 6. Mapping of control parameters

Figure 7 represents the ontological model with two main aspects: the design tools (left) and the product designs (via the design repository) on the right hand side of the figure. Based on the PFODM, two ontologies are generated: Ontology 1 is based on the tools and the generic tools ontology, and Ontology 2 is created by using the product design data and the repository ontology. Then the two branches (tools and products) merge using the parameters template for implementation activity. Both ontologies (1 and 2) are first validated via the control parameters to ensure that the information required by the tool is in the design repository. Finally, designers get the results of the product family design analysis based on the selected tool. In the long-term, new tools (top left corner) and new products (top right corner) will enrich the model, and PFODM will update Ontologies 1 and 2.

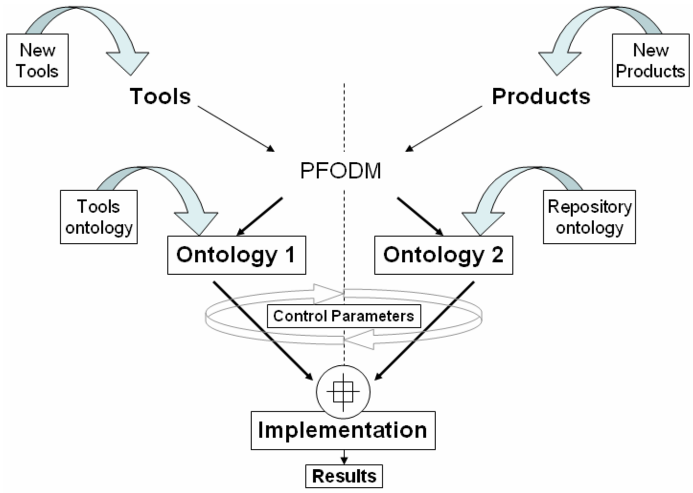

Figure 7. Over all model to manage tools and data

Each ontological tool has a parameter template associated with it. Based on the parameters needed by a particular tool, a subset of the repository's design parameter is queried. The parameter template has all the parameters that are part of the control parameters class. The repository also uses the control parameters to define the products, functions, and components etc.

4 CASE STUDY USING A SINGLE-USE CAMERA FAMILY

The ontology-based model is applied to the family of single-use cameras shown in Figure 8.

Figure 8: Existing designs, from the left to the right: Zoom, ADVANTIX Switchable, Black & White, Fun Saver, Max High Definition, Max High Quality, Outdoor, +Digital, Max Power, Water & sport

Product information is gathered in the software Protégé [18]; the ontology is also applied in this environment. Four tools are used to implement the ontological-based model:

1. Commonality vs. Diversity Index (CDI) [21]: Based on the tradeoff between commonality and diversity tradeoff, this index helps designers to reach the best tradeoff.

2. Reuse Existing Unit for Shape and Efficiency (REUSE) Method [24]: This method filters existing designs from a repository based on their similarity and efficiency.
3. Product Differentiation Index (PDI) [23]: This index assesses the differentiation between two products based on the functionality they provide.
4. Family Differentiation Index (FDI) [23]: This index assesses the differentiation between two families of products from a functional perspective.

4.1 Example

The needs of the REUSE, PDI, CDI, and FDI are given in Figure 9, Figure 10, Figure 11, and Figure 12, respectively. These models follow the OWL Class hierarchy in Tools. The repository information is represented in Figure 13. This decomposition also follows the OWL Class hierarchy in Repository.

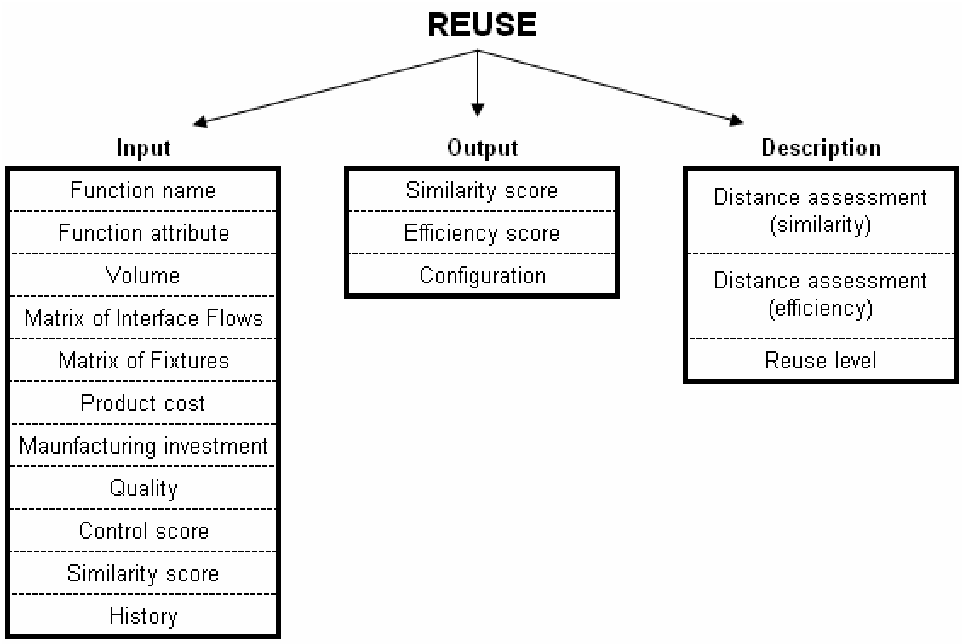

Figure 9. OWL Class hierarchic in REUSE

For the REUSE Method, the Tool Ontology description is given by "Distance assessment (similarity)", "Distance assessment (efficiency)", and "Reuse level". This description is unique for this tool and requires inputs and outputs, e.g., "Function name" for inputs and "Similarity score" for output.

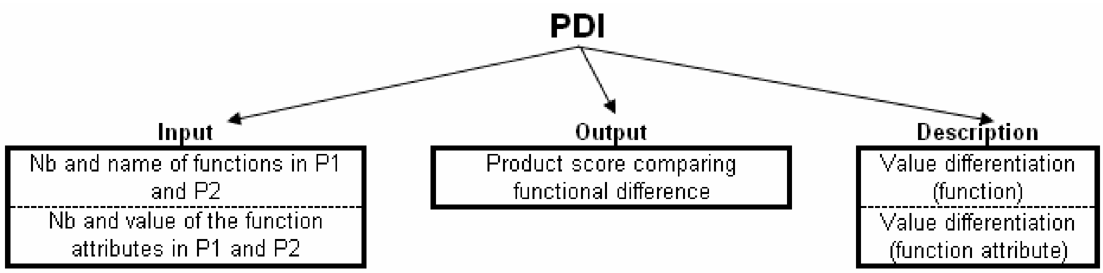

Figure 10. OWL Class hierarchic in Product Differentiation Index

The ontology of the PDI tool is composed of Value differentiation (function) and Value differentiation (function attribute). Inputs are "Nb and name of functions in P1 and P2" (NB: number; P1 and P2: product 1 and 2) and "Nb and name of function attributes in P1 and P2" (NB: number; P1 and P2: product 1 and 2). The output is "Product score comparing functional difference".

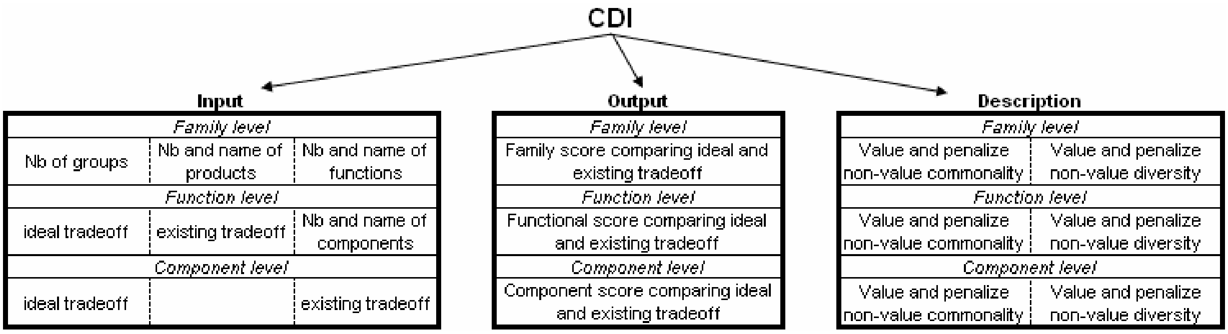

Figure 11. OWL Class hierarchy in Commonality vs. Diversity Index

Regarding the CDI tool, its ontological definition is composed of three categories "Family level", "Function level", and "Component level" with two sub-categories for each "Value and penalize non-value commonality" and "Value and penalize non-value diversity". Inputs are also divided in three same categories; the family level input is split in "Nb of groups" (group being the group of components having the same tradeoff), "Nb and name of products", and "Nb and name of functions". The function level has three sub-inputs: "Ideal tradeoff", "Existing tradeoff", and "Nb and name of components". The Component level for inputs is composed of "Ideal tradeoff" and "Existing tradeoff". Regarding the outputs, Family level is composed of "Family score comparing ideal and existing tradeoff"; Function level is described by "Functional score comparing ideal and existing tradeoff"; and component level is "Component score comparing ideal and existing tradeoff".

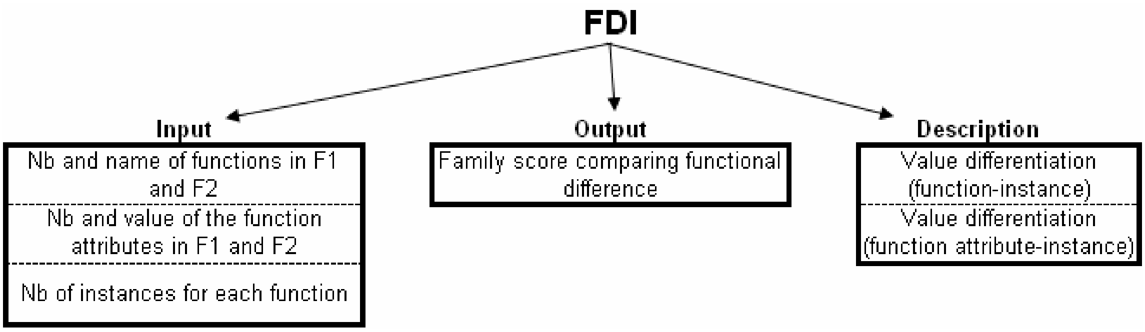

Figure 12. OWL Class hierarchic in Family Differentiation Index

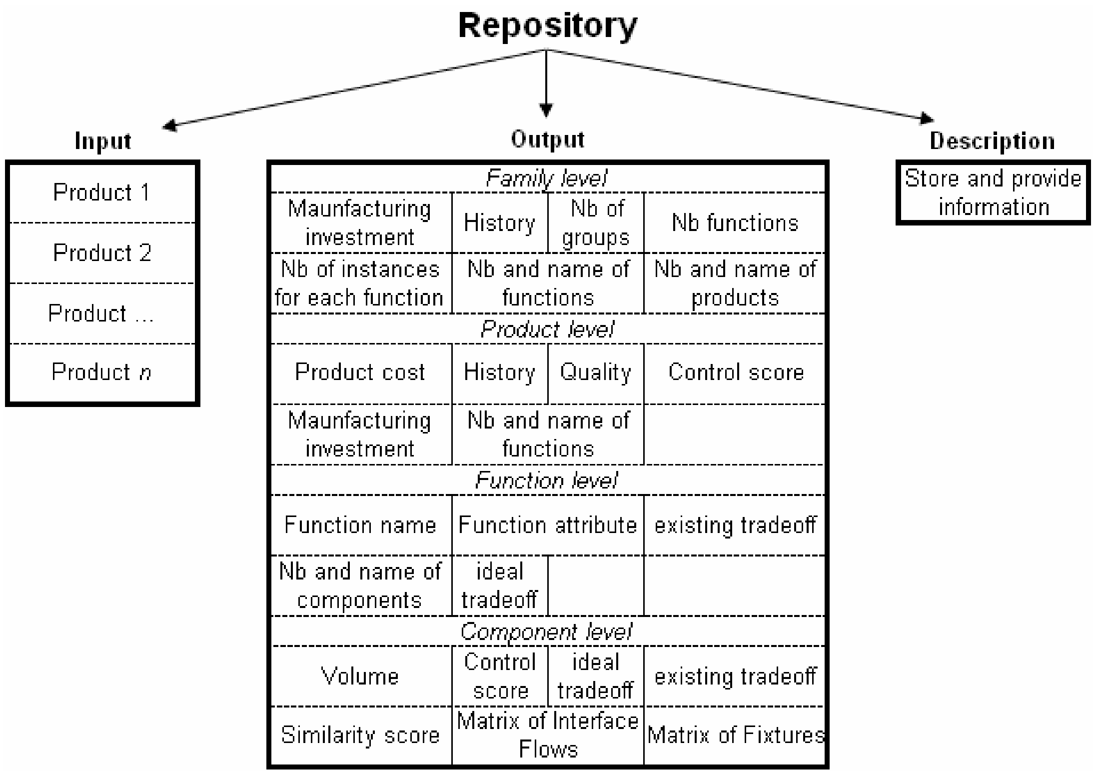

Figure 13. OWL Class hierarchic in Repository

The FDI ontology is the “Value differentiation (function instance)” and Value differentiation (function attribute-instance). Inputs are the same as the PDI but this time for the family plus the “NB of instances for each function” (NB: number). The output is the Family score comparing functional difference. The ontology chosen for the Repository (see Figure 13) is in input all the products. Its outputs are composed of four levels: “Family”, “Product”, “Function”, and “Component”. The Family level is composed of “Manufacturing investment”, “History” (history of the family: success, failure, modification, etc), “Nb of groups”, “Nb of functions”, “Nb of instances for each function”, “Nb and name of functions”, and “Nb and name of products”. The Product level includes “Product cost”, “History”, “Quality”, “Control score” (level of control of this product based on the number of maturity of this product [22]), “Manufacturing investment”, “Nb and name of functions”. The function level is composed of “Function name”, “Function attribute”, “Existing tradeoff” (between commonality and diversity), “Nb and name of components”, and “Ideal tradeoff”. Finally the component level is described by “Volume”, “Control score”, “Ideal tradeoff”, “Existing tradeoff”, “Similarity score”, “Matrix of interface flows”, and “Matrix of fixture”. Finally the ontological description of the Repository is “Store and provide information”. Figure 14 shows an image of the repository using the proposed ontology-based model within Protégé. Each tool has its own representation as well as the repository (embedded in Protégé), which is described as mentioned before. As a result, interoperability between tools and the information in the repository is ensured. Tools can be applied in this repository, and inputs and outputs can be stored and managed directly in the repository.

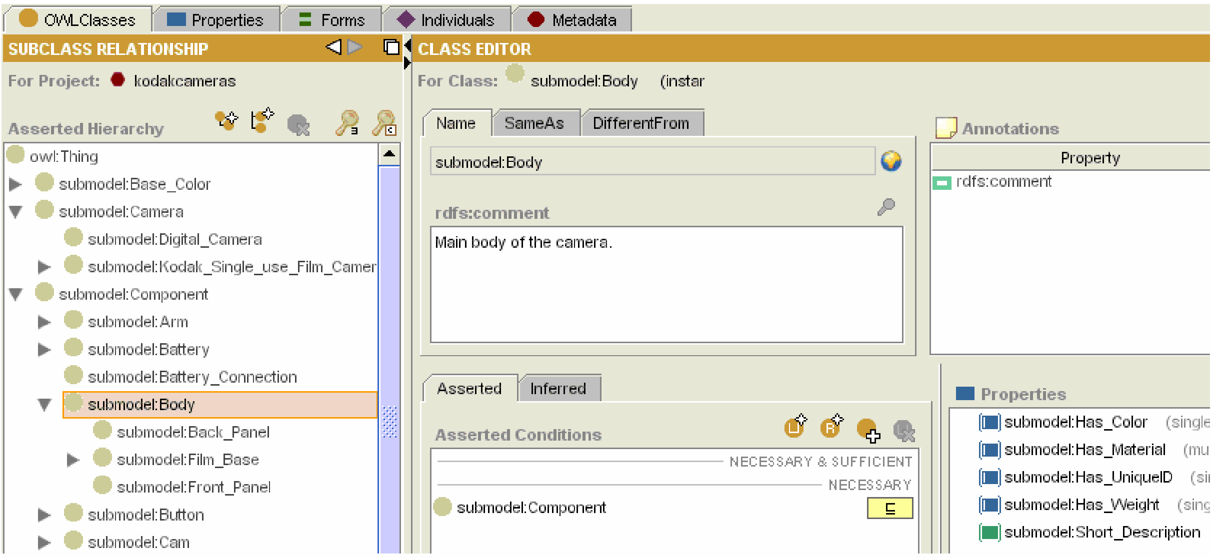

Figure 14. Extraction of the Repository

In this extraction, all of the cameras and components (arm, battery, battery connection, body, button, cam, etc.) are illustrated. In this example, the Component *Body* has the properties (submodel: Has_Color, submodel: Has_Material, submodel: Has_Unique ID, submodel: Has_Weight, and submodel: Has_Description).

The ontology-based model has helped formalize and interface the repository offerings (outputs) and tools requirement (inputs). An evolving model has been presented, which enables designers to validate first that tool requirements fit repository offer, and to facilitate sharing information between repository and tools. The example highlights the generic aspect of the model and its interoperability.

5 CLOSING REMARKS

In this paper, an ontology-based model is introduced to match the repository data (output) and the needs of analysis tools (input). We use Web Ontology Language (OWL-DL) to specify the semantics of the tools template and repository template. This model enables designers to automatically check the compatibility between repository and tools, manage interface between tool and repository via data description, and manage the evolution of the needs (new tools). This model also introduces ontology filtration (partial representation) in design to match the needed and existing data.

ACKNOWLEGEMENT

We gratefully acknowledge the National Science Foundation through Grant Nos. IIS-0325321 and IIS-0325402. Any opinions, findings and conclusions or recommendations presented in this paper are those of the authors and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

[1] Hansen, L., Shooter, S. B., Alizon, F., Terpenny, J., Simpson, T. W., Stone, R. B. and Kumara, S. R. T., 2006, "Experiences with an Inter-University Collaborative Undergraduate Research/Learning Experience for Product Platforming Planning," *Proceedings of the ASEE International Conference on Engineering Education*, Chicago, IL.

[2] van der Vegte, W. F., Kitamura, Y., Mizoguchi, R. and Horváth, I., 2002, October 10th, "Ontology-based Modeling of Product Functionality and Use - Part 2: Considering Use and Unintended Behavior," *EdiPROD Conference*, Poland, pp. 115-124.

[3] Greer, J. L., Stock, M. E., Stone, R. B. and Wood, K. L., 2003, "Enumerating the component space: First steps toward a design naming convention for mechanical parts," *ASME Design Engineering Technical Conference*, Chicago, IL.

[4] Stahovich, T. F., Davis, R. and Shrobe, H., 1993, "An Ontology of Mechanical Devices," *National Conference on Artificial Intelligence (AAAI-93)*, Washington, DC.

[5] Stahovich, T. F., Davis, R. and Shrobe, H., 1996, "Generating multiple new designs from a sketch," *National Conference on Artificial Intelligence*, Portland, OR, Part 2, **2**, pp. 1022-1029.

[6] Nahm, Y.-E. and Ishikawa, H., 2004, "Integrated Product and Process Modeling for Collaborative Design Environment," *Concurrent Engineering: Research and Applications*, **12**(1), pp. 5-23.

[7] Kirschman, C. F., Fadel, G. M. and Jara–Almonte, C. C., 1996, "Classifying Functions for Mechanical Design," *ASME Design Engineering Technical Conference*, Irvine, CA, DETC/DTM–1504.

[8] Iwassaki, Y. and Chandrasekaran, B., 1992, "Design Verification through Function and Behavior-Oriented Representations: Bridging the gap between Function and Behavior," *International Conference on Artificial Intelligence in Design*, Pittsburgh, PA.

[9] Gruber, T. R., 1993, "A Translation Approach to Portable Ontology Specifications," *Knowledge Acquisition*, **5**(2), pp. 199-220.

[10] Murdock, J. W., Szykman, S. and Sriram, R. D., 1997, "An Information Modeling Framework to Support Design Databases and Repositories," *ASME Design Engineering Technical Conferences*, Sacramento, CA, DETC97/DFM-4373.

[11] Shooter, S. B., Keirouz, W. T., Szykman, S. and Fenves, S. J., 2000, "A Model for the Flow of Design Information in Product Development," *Engineering with computers*, **16**(3), pp. 178-194.

[12] Szykman, S., Fenves, S., Keirouz, W. and Shooter, S. B., 2001, "A Foundation for Interoperability in Next-Generation Product Development Systems," *Journal of Computer Aided Design*, **33**, pp. 545-559.

[13] Bohm, M., Stone, R. and Szykman, S., 2003, "Enhancing Virtual Product Representations for Advanced Design Repository Systems," *Journal of Computer Information Science in Engineering*, accepted for publications.

[14] Hyvönen, E., 2001, October 2, "The Semantic Web - The New Internet of Meanings," *Semantic Web Kick-Off in Finland - Vision, Technologies, Research, and Applications*, Helsinki, Finland, Helsinki Institute for Information Technology Publications, pp. 3-26.

[15] Daconta, M. C., Obrst, L. J. and Smith, K. T., 2003, *The Semantic Web: A Guide to the Future of XML, Web Services, and Knowledge Management*, Wiley Publishing, Inc., Indianapolis, Indiana.

[16] Baader, F., Calvanese, D., McGuinness, D., Nardi, D. and Patel-Schneider, P., 2003, *The Description Logic Handbook: Theory, Implementation and Applications*, Cambridge University Press, Cambridge, UK.

[17] Nanda, J., Thevenot, H. J., Simpson, T. W., Kumara, S. R. T. and Shooter, S. B., 2004, September 28-October 2, "Exploring Semantic Web Technologies for Product Family Modeling," *ASME International Design Engineering Technical Conferences - Computers and Information in Engineering*, Salt Lake City, UT, ASME, Paper No. DETC2004/CIE-57683.

[18] Noy, N. F., Sintek, M., Decker, S., Crubézy, M., Fergerson, R. W. and Musen, M. A., 2001, "Creating Semantic Web Contents with Protege-2000," *IEEE Intelligent Systems,* **48**(2), pp. 60-71.
[19] McBride, B., 2002, "Jena: A Semantic Web Toolkit," *IEEE Internet Computing,* **6**(6), pp. 55-59.
[20] Nanda, J., Simpson, T. W., Kumara, S. R. T. and Shooter, S. B., 2006, "A Methodology for Product Family Ontology Development Using Formal Concept Analysis and Web Ontology Language," *Journal of Computing and Information Science in Engineering,* **6**(2), pp. 103-113.
[21] Alizon, F., Shooter, S. B. and Simpson, T. W., 2006, "Assessing and Improving Commonality and Diversity within a Product Family," *International Design Engineering Technical Conferences,* Philadelphia, PA, DETC2006/DAC-99499.
[22] Alizon, F., Shooter, S. B. and Simpson, T. W., 2006, "Reuse of Manufacturing Knowledge to Facilitate Platform-based Product Realization," *Journal of Computing and Information Science in Engineering,* **6**(2), pp. 170-178.
[23] Alizon, F., Shooter, S. B. and Simpson, T. W., 2006, "Assessing and Increasing Product and Family Differentiation in the Market," *International Design Engineering Technical Conferences,* Philadelphia, PA, DETC2006/DAC-99538.
[24] Alizon, F., Shooter, S. B. and Simpson, T. W., 2006, "Knowledge Reuse from Existing Product Designs," *Computer-Aided Design,* under review.