In Proc. of the 5th International Seminar and Workshop Engineering Design in Integrated Product Development (EDIProD 2006), 21-23 September 2006, Gronów, Poland, pp. 59-69, 2006.

ROLES OF ONTOLOGIES OF ENGINEERING ARTIFACTS FOR DESIGN KNOWLEDGE MODELING

YOSHINOBU KITAMURA
The Institute of Scientific and Industrial Research, Osaka University
e-mail: kita@ei.sanken.osaka-u.ac.jp

Keywords: Product Knowledge Modeling, Ontology

Abstract: *Capturing design knowledge and its modeling are crucial issues for design support systems and design knowledge management. It is important that knowledge models are systematic, consistent, reusable and interoperable. This survey article discusses the roles of ontologies of engineering artifacts for contributing to such design knowledge modeling from a viewpoint of computer science. An ontology of artifacts, in general, consists of systematic and computational definitions of fundamental concepts and relationship which exist in the physical world related to target artifacts, and shows how to capture the artifacts. This article, firstly, discusses needs, types, levels and some examples of ontologies of engineering artifacts. Then, we discuss roles of the ontologies for design knowledge modeling such as modeling specification for consistent modeling, capturing implicit knowledge and basis of knowledge systematization.*

1. INTRODUCTION

Designing is a creative activity using several kinds of knowledge. The quality of design relies heavily on knowledge applied in the design processes. This is why capturing necessary knowledge and its modeling are recognized as crucial issues in the design support systems. Here, we concentrate on knowledge related to artifacts, especially, engineering products. Such product knowledge is widely dealt with by design support systems such as CAD, CAE, PDM, expert systems and knowledge management systems. For example, CAD systems deal with geometry data or form features of products. Expert systems deal with more conceptual knowledge such as generic rules on configuration of components. Typical knowledge management systems deal with design documents about products in a natural language.

In computer science research, ontology has been attracting a lot of attention [52]. While ontology research has begun in early 90's in the knowledge base community, the research activity has been accelerated and spread over the web technology community by the Semantic Web [1] movement in the last few years. It has been expected as a basis of knowledge modeling that it contributes to several issues of knowledge contents.

In the engineering domain, one of the typical expectations for ontologies is interoperability of knowledge about artifacts among engineering supporting systems, that is, a common data model, exchange and integration. In fact, one of the pioneering ontology research efforts in early 90's aims at product data exchange among engineering tools [11]. Practical product data exchange has been explored mainly for CAD data models in the projects such as STEP and CIM-OSA. Such practical efforts have been developed further to advanced data integration including ontological consideration such as EPISTLE [74].

This article discusses the roles of ontologies of artifacts in design knowledge modeling as well as product data exchange from a viewpoint of computer science. The roles discussed in this article include semantic constraints for modeling, capturing implicit knowledge, and a basis of knowledge systematization. First, we clarify scope of ontologies discussed in this article. Next, we discuss needs of ontologies for design knowledge modeling, types of ontologies and their levels. Following overviews of contents of some ontologies in the literature, the roles of ontologies of artifacts are discussed.

2. ONTLOGY OF ARTIFACTS

We start with a short look at general definition of "ontology". In philosophy, it means theory of existence. It tries to explain what is being and how the world is configured by introducing a system of critical categories to account things and their intrinsic relations. In the knowledge engineering community, a definition by Gruber is widely accepted; that is, "explicit specification of conceptualization" [23], where *conceptualization* is "a set of objects which an observer thinks exist in

[1] http://www.w3.org/2001/sw/

the world of interest and relations between them”[2] [21]. Gruber emphasizes that an ontology is used as agreement to use a shared vocabulary (*ontological commitment*) [23]. Here, we define compositionally an ontology as “a system (systematic, operational and prescriptive definitions) of fundamental concepts and relationships which shows how a model author views the target world, and which is shared in a community as building blocks for models”.

This article discusses ontologies of physical artifacts such as engineering products and engineering plants. Such artifacts are *target things* of design activities (called design targets). Many design support systems include a kind of a “model” of a design target. By a “model” of an artifact, we here mean intuitively a computer-operational description (assertion) about a specific design target (an individual or an instance). For example, a CAE tool has a geometric model of an artifact or a mathematical model based on a specific theory.

Recent Semantic Web systems cope with semantic (i.e., computer-operational) *metadata* annotated to web documents. Euzenat categorizes such annotation into *media-data* (such as encoding), *(media-)metadata* (authors), *indexes* (identifier), *content descriptor* (keywords) and *content representation* (structural abstract) [16]. The last “content representation” annotated to a product document can be viewed as a *model* of the product. In addition, we regard “content descriptor” as a kind of a (tiny) model of artifacts.

As well as a model of a concrete artifact, there is generic knowledge applicable to artifacts in a domain such as physical law. We view this kind of generic knowledge as *generic patterns* of instances.

An ontology of artifacts is knowledge for specifying contents of such models of artifacts. An ontology consists of *systematized* formal definitions of generic types of things which exist in the target world and relationships among them. By “systematized”, we mean that types (classes) in an ontology is defined using several “semantic links” to restrict meaning of concepts. Typical semantic links are “*is-a*” relation (called also as *is-kind-of*, *general-special*, *categorization*, *super-sub*, *subsumption* relations) and “*part-of*” relation (also *whole-part, has-part, decomposition, aggregation* relation). An ontology of artifacts generally aims at representation of the design targets themselves (such as structure and shape) and/or temporal changes of their physical attributes (so-called behaviour, process or function). For example, an ontology for electric circuits might define generic types of electric components such as transistor, connection relation among components, physical laws among physical quantities, and functions of components.

Such ontologies of artifacts are used for description of instance models of concrete artifacts as shown in Fig. 1. An instance model based on an ontology consists of instances of the classes and the relations defined in the ontology. An ontology provides a vocabulary for representation of instance models. The relations in an ontology play a role of restriction for combination (interpretation) of the concepts. The generic patterns provide useful common patterns of combination of the instances.

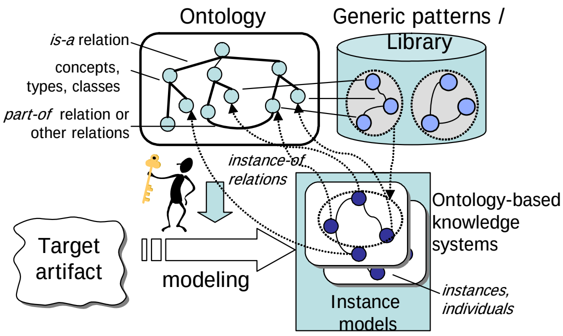

Fig. 1. *Ontology, generic patterns, and instance models of artifacts*

An ontology of artifacts discussed thus far can be regarded as a *domain ontology* for the design task. A *task ontology* for design is concerned with the process of design activities [50][52][58], while the domain ontology for design is concerned with design targets. Such design task ontologies have been explored as problem-solving method (PSM) research in knowledge engineering community [58] such as a design task ontology by Chandrasekaran [9]. In the design community, there are extensive efforts on generic models of design and design processes such as [20][33][67][72]. In a recent effort on design ontology [60], several design activities are defined in terms of types of knowledge as input/output. Here, we go into neither the details nor other work, because we concentrate on the domain ontology.

3. NECESSITY OF ONTOLOGY

One of the deep necessities of ontologies of artifacts for engineering design is, we believe, the lack of explicit description of *background knowledge of modeling*. There are many options for capturing a design target. A model of physical thing is usually a result of rather arbitrary decisions of such modeling options. The following shows examples of options.

- Terminology: Vocabulary used in the model.
- Abstraction level: A physical thing or physical phenomena can be described at several abstraction levels. For example, a model of an electric circuit can be at voltage level or at logical level.
- Size of granularity: The size of modeling primitives. Because an artifact and its behavior can be decomposed of elements whose size varies physically and temporally, respectively.
- Approximation: Any engineering models are results of some approximation such as ignorance of specific types of phenomena and (idealized) assumptions for environments or properties of artifacts.

[2] Contrary to this *extensional* definition, “conceptualization” here should be interpreted as *intensional* [52].

- Domain category such as mechanical and electrical.
- Domain theory: Each CAE system uses a physical theory to analyze the model.
- Engineering tasks such as design and diagnosis: A different (specialized) model can be used for each task.

Such dependence of model on modeling options is implicit in many cases. This causes (at least) the following three problems. The first is the difficulty in modeling to keep quality of models due to lack of modeling constraints. When one describes a model of a target thing, without explicit specification of the modeling options, the modeler is free to choose a modeling option in an ad hoc manner. Thus, the quality of models such as consistency of terminology depends on modelers. Our second concern is the difficulty in communications among computer (and human) agents, because agents cannot know if the modeling assumption is the same or not and exchange models in different assumptions. The third problem is the difficulty in reusing models for different systems, domains, or tasks. Without explicit modeling assumption, it is difficult to judge the applicability and translate the contents of the model.

Especially, in the case of knowledge about functionality of systems/components (so-called functional models) [7][10][33][54][62][69], functional modeling tends to be ad hoc because it is subjective rather than objective in comparison with structural or behavioral descriptions. For example, one may code "to weld metals" as a welding machine's function as Value Engineering does. It is, however, not a pure function but includes a certain way to achieve the goal, say, "that the metals' portion become fused". This example shows needs of an ontological consideration of function in order to restrict functional modeling.

An ontology of artifacts can be regarded as explicit specification of such background knowledge of modeling behind models. An ontology plays crucial roles in modeling, communications, reuse of models as discussed in Section 6.

4. LEVELS AND TYPES

Like modeling options for instance models, there are the following options for an ontology of artifacts.

- Aspects of artifact: What aspects of artifacts such as shape, geometry, behavior, function are covered by the ontology?
- Abstraction level of concepts: Does the ontology include upper-level concepts generalized from the concrete concepts which can be instantiated in the product model?
- Generality level of domain: Is the ontology specific to a target domain? Or is it generic to several domains?
- Generality level to tasks: Is the ontology specific to an engineering task such as design? Or is it generic to (independent of) tasks?
- Computational level of model: How computationally deep is the model intended to be described using the ontology? Typical computational levels include a complete model of target artifacts for computer operations such as simulation models, and a partial model for keyword- or annotation-based search.
- Description level of ontology: How is the description of ontology structured, formal and rigorous? Ontologies classified by this aspect might include labels only without structure (with natural language definitions), taxonomy with "*is-a*" hierarchy, structured concepts with rich relationships among concepts as intentional definitions of concepts, and first-order logic as axioms.
- Philosophical level: Does the concepts in the ontology have fidelity to (and/or be developed with consideration on) fundamental characteristics of reality in physical world? Does the ontology follow organizing principles developed in philosophy?

The first four determine the range of the target world (so-called *universe of discourse*) of the ontology and concepts captured in the ontology. The last three are properties of description of models or ontologies.

Ontologies are often categorized into *light-weight* and *heavy-weight* [52][70]. Using the options above, a typical light-weight ontology is set of concrete concepts, partial model, taxonomy only, not rigorous, and not philosophical. A typical heavy-weight ontology includes upper-level concepts, aims at complete models, has rigorous definition and rich relationships (and axioms), and is philosophical.

An ontology with higher levels of abstraction and generality is called as *upper(-level) ontology* which includes "concepts that are meta, generic, abstract and philosophical, and therefore are general enough to address (at a high level) a broad range of domain areas" [65]. In Ontological Engineering research, there are extensive efforts such as Sowa's [61], Guarino's [28], Cyc upper ontology [12], and SUMO (Suggested Upper Merged Ontology) [66] in the Standard Upper Ontology Working Group of IEEE P1600.1 [65]. Some generic types in product development are categorized based on SUMO [63].

5. EXAMPES OF ONTOLOGIES

5.1. Fundamentals

The fundamentals for capturing physical artifacts are what are primitives in the models and how the primitives form the whole. There are two major viewpoints: Device-centered and Process-centered views. The device ontology, e.g., one proposed by de Kleer and Brown [13], specifies the device-centered view of artifacts. Device-centered view regards any artifact as composition of *devices* which process input to produce output. For example, Gruber and Olsen implemented an ontology in Ontolingua as the Component-Assemblies Theory [25]. Fig 2 shows a part of its definition in

Ontolingua. It defines the ‘component’ class and two relations among the components, that is, *connected-components* for connections between two components and *subcomponent-of* for aggregation relations between a super-component and a sub-component. Such a device-centered ontology originates from the systems theory, and has been widely adopted in engineering task including design such as the German-style design methodology [54].

On the other hand, the process-centered view, e.g., one proposed by Forbus [18], concentrates on physical phenomena occurring in artifacts. Device ontology imposes a frame or viewpoint on phenomena, that is, a black box equipped with input and output ports. A device plays “actor role” in obtaining output (called *agent*), while the process ontology does not have such an agent but has *participants*. It this sense, the device ontology is a system of role-assignment for physical entities.

Physical quantity is another primitive for description of physical world. Gruber et al. established the EngMath (Engineering Mathematics) ontology for description of physical quantities, units and physical laws [26]. A representation of physical quantities and units in RDF(S) is discussed in [15].

As a pioneering example of fundamental ontology, there is Hayes’s liquid ontology [30]. Rosenman and Simoff discuss relationships among substance-geometry model, feature model, and component-assembly model [56].

PSL (Process Specification Language) developed by NIST (and being standardized within ISO TC184/SC4) treats more general (discrete) “process” such as manufacturing process [34]. It includes core-concepts such as *activity*, *time point* and *objects* and relations such as *participates-in* and *before*. Definitions and axioms (especially for time) are well-formalized in KIF language.

5.2. Layered and/or Reused Ontologies

For real domain model, there are many ontological specifications. For reusability of ontologies, hierarchies of ontologies are commonly established. On the basis of fundamental ontologies mentioned above, additional detailed domain ontologies are developed. In fact, Gruber’s component ontology is published as a part of domain ontologies for VT task [27]. Borst et al. propose the PhysSys ontology as a sophisticated lattice of ontologies for engineering domain as shown in Fig. 3 [6]. It consists of meleology, topology, systems theory, component, process and mathematics for engineering domain (EngMath), which supports multiple viewpoints on a physical system. Mereology is a theory about relationship between whole and part. Topology adds (typed) connection relation among parts. The component ontology defines the ‘terminal’ concept and shows the structural aspect of artifacts. On the other hand, the process ontology shows the behavioral aspect of artifacts based on the bond theory. Lastly, EngMath [26] mentioned in the previous section is used as representation from the mathematical (physical quantities) aspect.

```
(define-class COMPONENT (?x)
 :def (individual-thing ?x)
 :axiom-def (and
      (domain-of component SUBCOMPONENT-OF)
      (domain-of component HAS-SUBCOMPONENT-OF)
      (domain-of component CONNECTED-COMPONENTS)))

(define-relation CONNECTED-COMPONENTS (?a ?b)
  :def (and (component ?a) (component ?b)
            (not (subcomponent-of ?a ?b))
            (not (subcomponent-of ?b ?a)))
  :axiom-def (and
      (irreflexive-relation connected-components)
      (symmetric-relation connected-components)))

(define-relation SUBCOMPONENT-OF (?sub ?super)
  :def (and (component ?sub) (component ?super))
  :axiom-def (and
       (irreflexive-relation subcomponent-of
       (antisymmetric-relation subcomponent-of)))
```

Fig. 2. *A part of the Component-Assemblies Theory* [25]

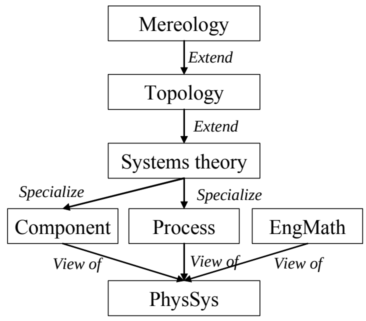

Fig. 3. *The structure of PhysSys Ontology* [6]

An ontology of airplanes has been developed in a similar way, i.e., by merging some existing ontologies [71]. An analysis of some existing (upper and generic) ontologies has been done in order to determine which is most appropriate for the manufacturing domain (As a result, Cyc was selected) [57]. A methodology for creating engineering ontologies reported in [1] includes reuse of existing taxonomies, test for application, and refinement of the integrated taxonomy. Shin et al. discusses generic types of design induced errors and a process of ontology development [59]. An ontology of chemical structures has been developed using the ontology-development methodology called METHONTOLOGY [47]. Horváth et al. [32] discuss design concept ontologies for comprehensive methodology for handling design concepts in conceptual design. Bachmann and Steffens discuss an ontology of materials such as metal and plastics based on an extension of EngMath ontology [3]. The FBSO ontology in [45] also adopts similar layered structure such as generic, application and domain layers. It includes ontologies for behavior, function, feature, components and mating relations.

5.3. Ontologies for Engineering Tools

CAD and CAE tools support drawing and analyzing artifacts in design processes. Each tool has a form of models of artifacts based on a viewpoint or theory. Thus, we can say each tool has an ontology of artifact. SHADE (Shared Dependency Engineering) [24] and PACT (Palo Alto Collaborative Testbed) project [11] aims at integration and exchange of engineering tools based on ontologies.

Tomiyama et al. explore ontological models of theories of engineering tools and their integration for KIEF (Knowledge Intensive Engineering

Framework) [73]. A physical concept ontology in the concept base of KIEF shown in Fig. 4 consists of basic concepts which are categorized into entity (800), relation (150), attribute (600) and physical phenomena (500) and physical law (300) (the numbers denote numbers of defined concepts or pieces of knowledge in each category). Using such concepts, *knowledge about a modeling system* is described for each CAE system. For example, the knowledge about a beam modeling system includes 'Beam', 'ConcentrateForce' and so on as *available concepts* used in the beam model. The available concepts are associated with *abstract concepts* such as 'Entity' and 'Force' as the abstract concepts of 'Beam' and 'ConcentrateForce', respectively. Using such mapping between generic concepts and the concepts in the tool-specific model, knowledge-level information sharing can be realized as discussed in Section 6.6.

5.4. An Ontology of Conceptual Knowledge: A Functional Ontology

Knowledge about functionality is at the conceptual level and thus tends to be ad hoc. Ontological considerations have been done in Engineering Design and Functional Representation [10][44]. Here, we show our own ontology of function [38][39]. Our ontology's main features are role-oriented definitions of ambiguous concepts and clear conceptualization of '*way of function achievement*'.

Our framework for functional knowledge [38][39] is shown in Fig. 5 as layers of ontologies, knowledge and instance models like PHYSYS mentioned in Section 5.2. Basically, knowledge or a model in a certain layer is described in terms of more general (and/or fundamental) concepts in the upper layer. Firstly, the extended device ontology is role-oriented extension of the conventional device ontology mentioned in Section 5.1. The functional concept ontology provides a taxonomy of generic functions (called functional concepts). We define *function* of a device as a role played by *behavior* to achieve a specific goal under a context of use, while behavior is objective temporal changes of physical quantity. Such teleological interpretation relation

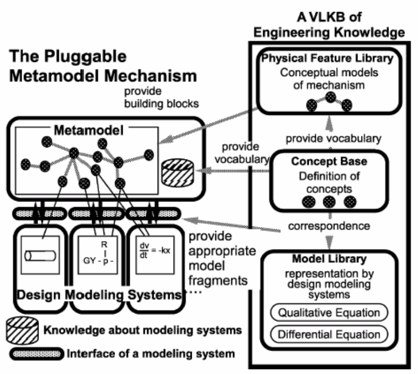

Fig. 4. *Knowledge Intensive Engineering Framework* [73]

shares its idea in literature such as F-B relationship [69] and "aims-means" [33]. We ontologically define function, related concepts, and related relationships (i.e., *is-a* and *part-of* relations). In comparison with conventional functional taxonomy such as [31][33][54], our taxonomy has clear operational relationship with objective behavior of a device. It is also clearly distinguished from "how to achieve a function", which we conceptualize into "*way of function achievement*" as the background knowledge of functional decomposition such as physical principles. Although the feature of function decomposition is also captured as "means" in [7][49], those works focus mainly on the function decomposition tree of a specific product. We focus on systematization (categorization) of general knowledge as discussed in Section 6.9.

6. ROLES IN DESIGN KNOWLEDGE MODELING

Uschold and Jasper elaborate ontology application scenarios, which are neutral authoring, common access to information, and indexing and concept search from repository [70]. Mizoguchi categorizes the roles of an ontology into a common vocabulary, data structure, explication of what is left implicit, semantic interoperability, explication of design rationale, systematization of knowledge, meta-model function, and theory of content [52]. Here, we discuss roles of ontology of artifacts mainly in

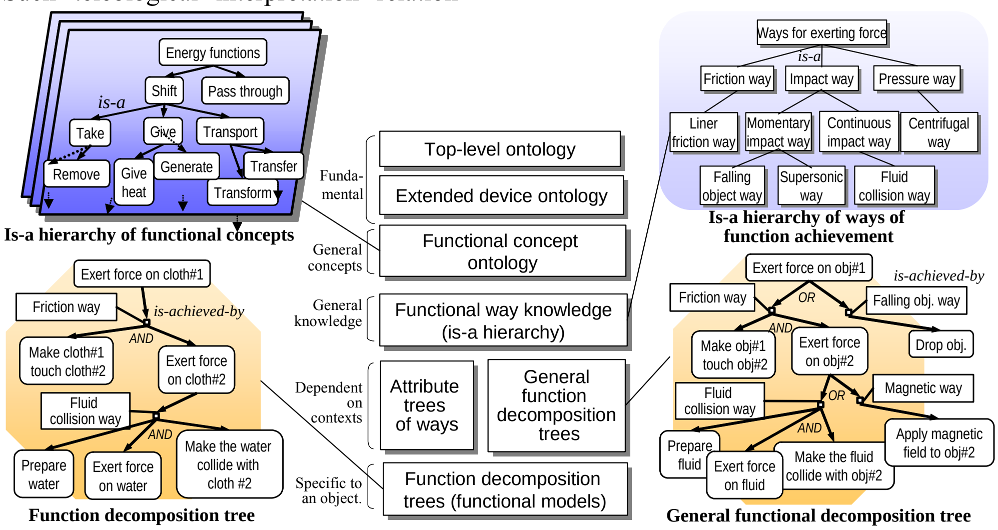

Fig. 5. *A layered framework of ontologies, knowledge, and models of functions* [38][39]

knowledge modeling.

6.1. Shared Vocabulary and Taxonomy

An ontology of artifacts can provide common "controlled vocabulary" for knowledge related to artifacts. Terms in models (or in texts) are lexical labels associated with classes defined in the ontology. It realizes lexical uniformity in models or in keywords. In this case, just a set of labels of classes is used. However, if an ontology has rich relationships, it makes the prescriptive meanings of the concepts in the ontology clearer in order to avoid misinterpretation of vocabulary.

If an ontology has an *is-a* hierarchy, it enables authors to select an appropriate abstraction level. A piece of knowledge can be indexed or categorized using the hierarchy as well. For example, a product ontology for car sheets in [35] includes simple taxonomies of functionality, safety features and seat naming conventions. It aims at providing common vocabulary for requirements management in order to have shared understanding between OEM and system suppliers in the car manufacturing industry. An ontology for a product family in [53] aims at capturing the unique, variant and common component of the product family.

6.2. Conceptual (Standard) Data Schema

Some ontologies of artifacts aim at providing a conceptual data schema (and generic/standard data model) for product data management (PDM). It enables us to realize exchange, integration and repository of data. Many of them mainly provide object-oriented data structure without rich *is-a* hierarchy. Welty and Guarino pointed out the difference between such 'conceptual model' with engineering trade-off of running application and implementation-independent 'ontology' [75].

STEP (ISO 10303) is a pioneering standard for product data exchange with many Application Protocols (APs). Guarino et al. pointed out also that STEP lacks clear meaning of modeling entities and necessity of a formal ontology [29]. EPISTLE for process plants (AP 221 in STEP) includes well-sophisticated ontological concepts [74]. The INTEREST Information Model (IIM) has been developed for a data warehouse of product data (called Virtual Repository) [5]. It is based on EPISTLE and adopts the principal entities and modeling principles. The Design Repository Project at NIST includes a generic information model [64] and a taxonomy of function [31].

6.3. Metadata Schema for Documents

In the Semantic web context, an ontology can be used as a schema for semantic metadata for web documents as shown in Fig. 6. Semantic metadata is annotation to documents or terms in documents, and organized as semantic network with relationships (called 'property') in RDF[3]. Metadata are described

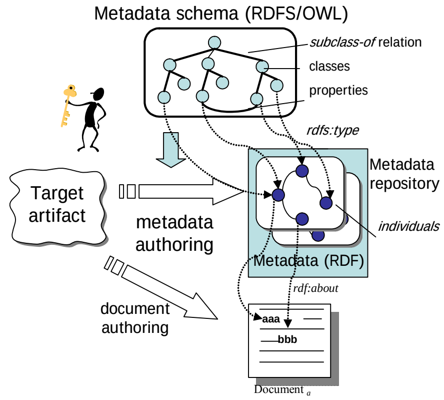

Fig. 6. *Ontology as metadata schema and metadata*

as instances of classes defined in a specific schema represented in RDFS[4] or OWL[5]. According to the categorization in Section 4, while *light-weight* schema provides just vocabulary for metadata, *heavy-weight* schema provides authors with rich semantic constraints on authoring (see Section 6.4) and interoperability (Section 6.5).

Specification of metadata schema consists of specification of 'metadata elements' and that of 'metadata values'. The former specifies kinds of characteristics (attributes) of the target. The latter specifies a controlled vocabulary for concrete values of the attributes. For example, the Dublin Core Metadata Element Set specifies 'language' as a metadata element and recommends RFC3066/ ISO639 as its controlled value (such as 'en' for English). In the EPISTLE project, the core model and the reference data library (RDL) correspond to elements and values, respectively.

There are many projects aiming at metadata schema in engineering domain. ScadaOnWeb project [14] aims at building a common platform for the controlling task of engineering plants based on an ontology about physical quantities. It gives semantics to numeric data of controllers and sensors by annotation. A functional modeling framework for the Semantic Web has been proposed in [43]. It is based on Functional Basis [31] and is represented in the description logics (DL) for repository reasoning tasks. In [68], design rule knowledge including situation of design targets is described in RDF based on ontologies for rich semantics. We are developing a framework for annotation about functionality of engineering products based on the functional ontology shown in Section 5.4 [41].

6.4. Semantic Constraints for Modeling

When an ontology of artifacts has rich relationships among concepts, it can be used as semantic constraints for modeling. Ontological definitions of concepts and relationships such as restrictions of the use of concepts and the cardinality of relationships specify necessary (and possible) structures of models. It can act as rules or guidelines for how to

[3] Resource Description Framework, http://www.w3.org/RDF/

[4] RDF Vocabulary Description Language, http://www.w3.org/TR/rdf-schema/

[5] Web Ontology Language, http://www.w3.org/2004/OWL/

associate concepts each other. Such semantic guidelines make authoring easier and contribute to improvement of the consistency and reusability of models. Such function can be called "meta-model function" [52]. Moreover, using the axioms of concepts, a reasoner can provide a logical reasoning to fulfill missing information.

This kind of use can be realized as modeling-support system or knowledge acquisition/authoring tool so-called "ontology-aware authoring tools". The Protégé system [6] is an ontology-based knowledge acquisition tool as well as an ontology development tool. An example of an ontology-based KA system using Protégé 2000 is shown in [8] for control and monitoring expert systems.

The ontologies shown in Section 5 can be used for this kind of usage. De Kleer's device ontology aims at reusability of component models for different systems by enforcing the No-Function-In-Structure principle [13]. The PHYSYS ontology is used for building a model library [6][55].

6.5. Generic Knowledge and Patterns

Some ontologies include generic pattern knowledge which are commonly used as building-blocks in instance models. Generic pattern knowledge can be distinguished from the semantic constraints discussed above which aim at defining (restricting use of) concepts. For example, OpenCyc includes many pieces of 'commonsense knowledge' and facts in the world [12]. As shown in Fig. 5, we call generic patterns of the way of function achievement as function way knowledge. Some ontological systems include physical laws or generic rules in the physical world. KIEF [73] includes a library of physical law and phenomena. A knowledge-rich catalog system for engineering components reported in [36] uses axioms representing relations among components (e.g., motor's brushes) and physical phenomena (e.g., explosion) in order to make mapping user requirements into part types and attribute constraints.

6.6. Interoperability and Integration

An ontology shows background knowledge of modeling as discussed in Section 3 and semantics of the vocabulary in a model. For example, CML is an ontology-based modeling language for the compositional modeling, in which component models are with explicit modeling assumptions [17].

Such explicit knowledge of modeling enables computer-systems to realize interoperability and exchange among models with different forms or modeling assumptions. One of the typical frameworks for interoperability shown in Fig. 7 uses a reference ontology for mapping two target ontologies. Each target ontology shows the background knowledge of modeling of instance models such as a modeling viewpoint and modeling assumptions. The reference ontology includes generic (upper) types of things, to which the concepts in the target ontologies are categorized.

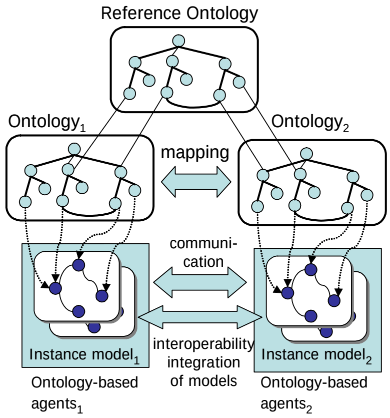

Fig. 7. *Interoperability of models and Ontologies with A Reference Ontology*

In KIEF mentioned in Section 5.3, the tool-specific models are linked via *metamodels* using concepts defined in the physical concept ontology [73]. When new information is added into a tool-specific model, the information is linked into the metamodel (using the *knowledge about a modeling system* mentioned in Section 5.3) and then is accessible from other tool-specific models. Thus, the information is shared by different engineering tools.

Agent-approach research efforts such as SHADE [24], PACT [11], FIPA and DAML aim at distributed data-exchange based on rather partial commitment. An ontology (rather a data model) of engineering analysis models reported in [22] includes properties such as limitation, idealization and resolutions, which are similar to the properties discussed in Section 3. An ontology-based product data exchange between two CAD/CAE tools in cutting domain is discussed in [14]. In the Semantic Web, this kind of usage has been explored. For example, in [48], an ontology mapping enables information integration of product data.

We developed an ontology-based knowledge transformation system for integration of knowledge about function and fault [42]. A reference ontology of function is being developed for interoperability among functional taxonomies [40][41] (like Fig. 7).

PSL [34] aims at exchange of manufacturing process data among application systems. PSL provides a common vocabulary (with rich formal axioms without *is-a* hierarchy) for defining terminology in each application (similar to the way shown in Fig. 7) and for avoiding inconsistent interpretations among the applications. Liu proposed a two-level model based on the device ontology and the charge-carrier ontology that focuses on the movement of free electrons (or holes) in electronic devices at the microscopic level [46].

One of the aims of task ontology research is reuse of domain ontology by different tasks [27]. For example, reuse of model and integration of ontologies for diagnosis and planning for electrical network are reported in [4].

[6] http://protege.stanford.edu/

6.7. Communication Support and Querying

An ontology can provide a basis for communication not only among computer-agents but also between computer and human (user) or among humans. Thus, it contributes to explanation or Q&A functions. For example, an ontology-based Q&A system for chemical domain [2] has been developed. In [51], explanation for plant operators has been developed based on a rich plant ontology. Based on role conceptualization, it enables context-sensitive explanation of equipment in an oil refinery plant.

Querying databases can be viewed as a kind of human interfaces. This type of usage is common with the roles discussed thus far especially for data schema in Section 6.2 and integration of models in Section 6.6. Ontology can be used to expand or generalize user's query [5][36] or to realize personalized query [45].

An ontology-based information extraction proposed in [37] is based on the Engineering Design Integrated Taxonomy (EDIT) including names of product instances [1]. It is used for lexical level analysis of documents for identifying named entities.

6.8. Capturing Implicit Knowledge

When an ontology provides a clear definitions of concepts (or relationships) which have been confused in the domain by engineers, it helps them distinguish these confusing concepts.

For example, the conceptualization of "way of function achievement" in our ontology shown in Section 5.4 helps us detach "how to achieve" (way) from "what is intended to achieve" (function). For example, "to weld something" mentioned in Section 3 should be decomposed into the "joining function" and "fusion way". This increases generality and capability of a functional model which accepts wide range of ways such as the bolt and nut way as an alternative way of achievement.

6.9. Basis of Knowledge Systematization

Realization of knowledge systematization requires to place a piece of knowledge appropriately in relationships (structure) of related knowledge by a solid basis of fundamental concepts. Ontological engineering provides us with the basis on which we can build knowledge and with computer-interpretable vocabulary in terms of which we can describe knowledge systematically and consistently. An ontology can provide fundamental concepts to capture the target world and their clear relationships as a basis for making relations of different pieces of knowledge clearer.

The EPISTLE ontology [74] can be regarded as ontology as basis of knowledge systemization. GENIAL project within the Global Engineering Networking (GEN) initiative also discusses ontologies in the Common Semantic Models for knowledge systematization [19]. GNOSIS - Knowledge Systemization for Post-Mass-Production Manufacturing project within the IMS program which we have been involved in also aims at knowledge systematization.

Our functional ontology shown in Section 5.4 also aims at systematic organization of functional knowledge based on fundamental concepts. We organize the functional way knowledge based on clear conceptualization of relationships; *is-a* relation and *part-of* relation of the way of function achievement. The *is-a* relation of ways represents generalization (or specialization) of physical principles for function achievement. For example, the *fluid collision way* for exerting force can be categorized into the *impact way*. This *is-a* relation of ways of function achievement is different from that of functions. Moreover, we decompose a way of function achievement for increasing reusability. For example, the *arc welding way* for joining should be decomposed into the *fusion way* for joining and the *arc way* for generating heat. Such relationships, we believe, enables us systematic organization of reusable functional knowledge.

7. Practical Benefits in Deployment

Here we explain the practical benefits of ontology-based description of functional knowledge which are found in successful deployment of our functional ontologies in a manufacturing company [39]. Our framework has been deployed since May, 2001 into the plant and production systems engineering division of Sumitomo Electric Industries, Ltd. (hereinafter referred to as SEI).

One of the uses of the function decomposition tree shown in Fig. 5 is to clarify functional knowledge and to share it with other engineers. The experiential evaluation by SEI's engineers was unanimously positive. Such explicit description of intentions is useful especially in the design review activity, where a team of designers double-check the original design and explore possible alternatives. For this, the role of the ontology discussed in Sections 6.4 and 6.8, that is, semantic constraints for modeling and capturing implicit knowledge are crucial. The way of function achievement enables engineers to show alternative ways for achieving functions exhaustively for each (sub) function, their features in comparison, and reasons for adopting a specific way, or not, in one figure. The number of times the design reviews had to be done was reduced to one third after adopting our framework.

Writing a function decomposition tree gives designers the chance to reflect on possible alternative ways, which leads them to an in-depth understanding of the equipment. It contributes to redesign and solving problems concerning equipment. In the deployment, a redesign of a manufacturing machine (a polishing machine) has been done by reusing a functional decomposition tree of a wire-saw, in which the reused knowledge provides the engineer with stimuli to make him aware of an implicit function. It can be viewed as effects of the common vocabulary (Section 6.1) and the semantic constraints for modeling (Section. 6.4).

Comparing design candidates as different ways to achieve functions contributes to patent analysis and patent applications. In communications between engineers and patent attorneys in applying for a new patent, it is difficult to determine the product's originality and to make appropriate claims. When the general function decomposition tree has been adopted as the regular document format for patent application, the period was reduced to just one week from three or four weeks. This is the role of ontology for human communication support discussed in Section 6.7. Moreover, the number of the patent claims was increased, doubled in some cases, since the attorneys found extra differences with other patents by checking at each level of function decomposition.

Generic knowledge about ways of function achievement helps designers search ways to achieve a function and/or alternatives in existing products. In the deployment, a novice engineer developed an inspection machine in three days by systematically consulting a library of generic ways of shedding light. Such development usually requires experts two weeks. The discrimination of the ways of function achievement has helped engineers avoid a great deal of confusion (Sections 6.5 and 6.9).

8. CONCLUDING REMARKS

This article has discussed ontologies of artifacts for engineering design; especially, their roles in product knowledge modeling. As well as data exchange and integration, such an ontology contributes to semantic constraints for modeling, capturing implicit knowledge, knowledge systematization and so on. In summary, an ontology provides us with "a theory of content" to enable research results on "content of knowledge" to accumulate.

Acknowledgements: This article is a result of extensive discussion with Prof. Riichiro Mizoguchi. Figure 4 appears by courtesy of Prof. Tetuo Tomiyama and Prof. Masaharu Yoshioka. Special thanks go to Dr. Masayoshi Fuse, Mr. Masakazu Kashiwase, Mr. Shuji Shinoki of Sumitomo Electric Industries, Ltd. for their cooperation with the deployment.

References

[1] Ahmed, S., Kim, S., Wallace, K. M., *A Methodology for Creating Ontologies for Engineering Design*, In Proc. of ASME 2005 DTM, 2005, DETC2005-84729.

[2] Angele, J., Moench, E., Oppermann, H., Staab, S., Wenke, D.: *Ontology-based Query and Answering in Chemistry: OntoNova @ Project Halo*, In Proc. of ISWC 2003, 2003, pp. 913-928.

[3] Bachmann, B., Steffens, M.: *An Ontology of Materials*, Proc. of ECAI '94 WS W13: Comparison of Implemented Ontologies, 1994, pp. 17-33.

[4] Bernaras, A., Laresgoiti, I., Corera, J.: *Building and Reusing Ontologies for Electrical Network Applications*, In Proc. of ECAI 96, 1996, pp. 298-302.

[5] Brown, D., Leal, D., McMahon, C., Crossland, R., Devlukia, J., *Virtual A Web-enabled Virtual Repository for Supporting Distributed Automotive Component Development,* Advanced Engineering Informatics, Vol.18, 2004, pp.173-190.

[6] Borst, P. Akkermans, H., and Top, J, *Engineering Ontologies*, Human-Computer Studies, 46(2/3), 1997, pp. 365-406.

[7] Bracewell, R.H., Wallace, K.M., *Designing a Representation to Support Function-Means based Synthesis of Mechanical Design Solutions*, In Proc of ICED 01, 2001.

[8] Chan, C. W., Peng, Y., Chen, L-L., *Knowledge Acquisition and Ontology Modelling for Construction of a Control and Monitoring Expert System*, Int'l J. of Systems Science, 33(6), 2002, pp. 485-503.

[9] Chandrasekaran, B., *Design Problem Solving: A Task Analysis*, AI Magazine, 11(4), 1990, pp. 59-71.

[10] Chandrasekaran, B, Josephson, J.R., *Function in Device Representation*, Engineering with Computers, **16**(3/4), 2000, 162-177.

[11] Cutkosky, M.R., et al., *PACT: An Experiment in Integrating Concurrent Engineering Systems*, *Computer*, January, 1993, pp. 28-37.

[12] Cyc, OpenCyc, http://www.opencyc.org/

[13] de Kleer, J., Brown, J. S.: *A Qualitative Physics Based on Confluences*, Artificial Intelligence, Vol.24, 1984, pp.7-83.

[14] Dartigues, C., and Ghodous, P., *Product Data Exchange Using Ontologies*, In Proc. of Artificial Intelligence in Design (AID2002), 2002, pp. 617-637.

[15] Dreyer, T., Leal, D., Schroder, A., Schwan, M., *ScadaOnWeb - Web Based Supervisory Control and Data Acquisition*, In Proc. of ISWC 2003, 2003, pp. 788-801.

[16] Euzenat, J., *Eight Questions about Semantic Web Annotations*, IEEE Intelligent Systems, 2002, pp. 55-62.

[17] Falkenhainer, B., Farquhar, A., Bobrow, D., Fikes, R., Forbus, K., Gruber, T., Iwasaki, Y., Kuipers, B.: *CML: A Compositional Modeling Language*, KSL-94-16, KSL, Stanford University, 1994.

[18] Forbus, K. D.: *Qualitative Process Theory*, Artificial Intelligence, 24, 1984, pp. 85-168.

[19] Gausemeier, J., Seifert, L., Joosten, M., Kesteloot, P. and Monceyron, J. L., *GENIAL: The logical Framework for the Systematisation of Internal and External Engineering Knowledge*, In Proc. of ICED 97, 1997.

[20] Gero, J.S., Kannengiesser, U., *The Situated Function-Behaviour-Structure Framework*, In Proc. of AI in Design '02, 2002, pp. 89-104.

[21] Genesereth, M. and Nilsson, N. *Foundation of Artificial Intelligence*, 1987.

[22] Grosse, I. R., Milton-Benoit, J. M., Wileden, J. C., *Ontologies for Supporting Engineering Analysis Models*, AIEDAM, 2005, 19, 1-18.

[23] Gruber, T. R., *What is an Ontology?*, http://www-ksl.stanford.edu/kst/what-is-an-ontology.html

[24] Gruber, T. R., Tenenbaum, J. M., Weber, J. C.: *Toward a Knowledge Medium for Collaborative Product Development*, In Proc. of AI in Design '92, 1992, 413-432.

[25] Gruber ,T, Olsen, G.,Theory *Component-Assemblies*, Ontology Server. http://www-ksl.standford.edu, 1994.

[26] Gruber T. R., Olsen, G. R.: *An Ontology for Engineering Mathematics*, In Proc. of Fourth Int'l Conf. on Principles of Knowledge Representation and Reasoning, 2004.

[27] Gruber, T. R., Olsen, G. R., Runkel, J.: *The Configuration Design Ontologies and the VT Elevator Domain Theory*, Int. J. of Human-Computer Studies, 44, 1996, 569-598.

[28] Guarino N.: *Some Ontological Principles for A Unified Top-Level Ontology*, Proc. of AAAI Spring Symposium on Ontological Engineering, 1997.

[29] Guarino, N., Borgo, S., Masolo, C., *Logical modelling of Product knowledge: Towards a well-founded Semantics for STEP*, In Proc. of European Conf. on Product Data Technology, 1997.

[30] Hayes, P. J.:Naive physics I: *Ontology for liquids*, Formal theories of the commonsense world, 1985, 71-107, Ablex.

[31] Hirtz, J., Stone, R.B., McAdams, D.A., Szykman, S., Wood, K.L. *A Functional Basis for Engineering Design: Reconciling and Evolving Previous Efforts*. Research in Engineering Design, 13, 2002, pp. 65-82.

[32] Horváth, I., Kuczogi, Gy., Vergeest, J. S. M.: *Development and Application of Design Concept Ontologies for Contextual Conceptualization*, In Proc. of ASME DETC/CIE, DETC-5701, 1998

[33] Hubka, V., Eder, W.E., *Theory of Technical Systems*, 1988, Berlin: Springer-Verlag.

[34] ISO TC184/SC4/JWG8 (2003). *Process Specification Language*, http://www.tc184-sc4.org/SC4_Open/SC4_Work_Products_Documents/PSL_(18629)/.

[35] Kerr, C. I., Roy, R., Sackett, P. J., *A Product Ontology for Automotive Seat Specification,* In Proc. of ASME DETC 2004, DETC2004-57071, 2004.

[36] Kim, J., Will, P., Ling, R., Neches, B., *Knowledge-rich Catalog Services for Engineering Design*, AI EDAM, 17, 2003, pp. 349-366.

[37] Kim, S., Ahmed, S., Wallace, K., *Improving Document Accessibility through Ontology-based Information Sharing*, In Proc. of TMCE 2006, 2006.

[38] Kitamura, Y., Sano, T., Namba, K., and Mizoguchi, R., *A Functional Concept Ontology and Its Application to Automatic Identification of Functional Structures*. Advanced Engineering Informatics **16**(2), 2002, 145-163.

[39] Kitamura, Y., Kashiwase, M., Fuse, M., Mizoguchi, R., *Deployment of an Ontological Framework of Functional Design Knowledge*, Advanced Engineering Informatics, **18**(2), 2004, 115-127.

[40] Kitamura, Y., et al. *Towards a Reference Ontology of Functionality for Interoperable Annotation for Engineering Documents*, In Proc. of posters of ESWC 2006, 2006.

[41] Kitamura, Y., Washio, N., Koji, Y., Mizoguchi, R., *An Ontology-Based Annotation Framework for Representing the Functionality of Engineering Devices*, In Proc. of ASME IDETC/CIE 2006, DETC2006-99131, To appear.

[42] Koji, Y., Kitamura, Y., Mizoguchi, R, *Ontology-based Transformation from an Extended Functional Model to FMEA*, In Proc. of ICED 05, 2005, 264.81.

[43] Kopena, J. B., Regli, W. C., *Functional Modeling of Engineering Designs for the Semantic Web*, IEEE Data Engineering Bulletin, **26**(4), 2003, pp. 55-62.

[44] Kumar, A. N. and Upadhyaya, S. J.: *Component-Ontological Representation of Function for Reasoning about Devices*, Artificial Intelligence in Engineering, 12, 1998, pp. 399-415.

[45] Li, Z., Anderson, D. C., Ramani, K., *Ontology-based Design Knowledge Modeling for Product Retrieval*", In *Proc. of ICED 2005*, 2005, 462.1.

[46] Liu, Z.: *Integrating Two Ontology for Electronics*, In Recent Advances in Qualitative Physics, MIT Press, 1992, pp. 153-168.

[47] López, M. F., et al. *Building a Chemical Ontology using Methontology and the Ontology Design Environment*, IEEE Intelligent Systems, 14(1), 1999, pp. 37-46.

[48] Maier, A., Schnurr, H. P., Sure, Y.: *Ontology-based Information Integration in the Automotive Industry*, In Proc. of ISWC 2003, 2003, pp. 897-912.

[49] Malmqvist, J., *Improved Function-means Trees by Inclusion of Design History Information*, Journal of Engineering Design, **8**(2), 1997, pp. 107-117.

[50] Mizoguchi, R. et al., *Task ontology for reuse of problem solving knowledge*, Proc. KB&KS95, 1995.

[51] Mizoguchi, R., et al., *Construction and Deployment of a Plant Ontology*, In Proc. of EKAW 2000, 2000, 113-128.

[52] Mizoguchi, R., *Tutorial on ontological engineering - Part 1, 2, 3*. New Generation Computing, 21(4), 2003, pp. 365-384, 22(1), 2004, pp. 61-96, 22(2), 2004, pp. 193-220.

[53] Nanda, J., Thevenot, H. J., Simpson, T. W., Kumara, S. R. T., Shooter, S. B., *Exploring Semantic Web Technologies for Product Family Modeling*, In *Proc. of ASME DETC 2004*, 2004, DETC2004-57683.

[54] Pahl, G., and Beitz, W., *Engineering Design - a Systematic Approach*. The Design Council, 1998.

[55] Pos, A., et al.: *Automatic Revision of Simulation Models*, IEEE Intell. Systems, 13(2), 1998, pp. 75-81.

[56] Rosenman, M., Simoff, S. J.: *Some Conceptual Issues in Component-assembly Modeling*, Artificial Intelligence in Engineering, 15, 2001, pp. 109-119.

[57] Schlenoff, C., Denno, P., Ivester, R., Libes, D., Szykman, S., *An Analysis and Approach to Using Existing Ontological Systems for Applications in Manufacturing*, AI EDAM, 14, 2000, pp. 257-270.

[58] Schreiber, G., Akkermans, H., Anjewierden, A., de Hoog, R., Shadbolt, N., Van de Velde, W. and Wielinga, B. *Knowledge Engineering and Management - The Common-KADS Methodology*, 2000, MIT Press.

[59] Shin, I. J., Busby, J. S., Hibberd, R. E., McMahon, C. A., *A Theory-based Ontology of Design Induced Error*, In Proc. of ICED 05, 2005.

[60] Sim, S. K., Duffy, A. H. B., *Towards an ontology of generic engineering design activities*, Res Eng Design, 14, 2003, pp. 200–223.

[61] Sowa, J. F. *Top-level ontological categories*, Int'l Journal of Human-Computer Studies, 43(5/6), 1995, pp. 669–685.

[62] Stone, R. B., Chakrabarti, (eds.), *Special Issues: Engineering applications of representations of function*, *AI EDAM*, **19**(2 and 3), 2005.

[63] Štorga, M., Andreasen, M. M., Marjanović, D., *Towards a Formal Design Model based on a Genetic Design Model System*, In Proc. of ICED 05, 136.385.

[64] Sudarsan, R., Fenves, S.J., Sriram, R.D. , Wang, F.., *A product information modeling framework for product lifecycle management*, Computer-Aided Design, 37, 2005, pp. 1399–1411.

[65] Standard Upper Ontology Working Group (SUO WG), IEEE P1600.1, http://suo.ieee.org/

[66] Suggested Upper Merged Ontology, http://ontology.teknowledge.com/

[67] Takeda, H., Veerkamp, P., Tomiyama, T., Yoshikawa, H. *Modeling design processes*, AI Magazine, 11(4), 1990, pp. 37-48.

[68] Takeda, H., Yoshioka, M., Shimomura, Y., Fujimoto, Y., Morimoto, K., Oniki, W., *An Architecture for Designer's Support Systems with Knowledge-embedded Documents*, In Proc. of ICED 2005, 120.81.

[69] Umeda, Y., Ishii, M., Yoshioka, M., Shimomura, Y., and Tomiyama, T., *Supporting conceptual design based on the function-behavior-state modeler*, *AI EDAM,* 10, 1996, pp. 275-288.

[70] Uschold, M., and Jasper, R., *A Framework for Understanding and Classifying Ontology Applications*, In Proc. of the IJCAI99 Workshop on Ontologies and Problem-Solving Methods, 1999

[71] Valente, A., Russ, T., MacGregor, R., Swartout, W.: *Building and (Re)Using an Ontology of Air Campaign Planning*, IEEE Intelligent Systems, 14(1), 1999, pp. 27-36.

[72] Yoshikawa, H., *General design theory and a CAD system*, Man-machine Communication in CAD/CAM, eds. Sata, T. and Warman, E.A., 1981, pp. 35-58.

[73] Yoshioka, M., Umeda, Y., Takeda, H., Shimomura, Y., Nomaguchi, Y., Tomiyama, T., *Physical Concept Ontology for the Knowledge Intensive Engineering Framework, Advanced Engineering Informatics,* **18**(2), 2004, 95-113.

[74] West, M. *Some Industrial Experiences in the Development and Use of Ontologies*, EKAW 2004 WS on Core Ontologies in Ontology Engineering, 2004, pp. 1-14.

[75] Welty, C., Guarino, N., *Supporting Ontological Analysis of Taxonomic Relationships*, Data Knowledge Engineering, Vol. 39, 2001, pp.51,74.