Artificial Intelligence for Engineering Design, Analysis and Manufacturing (2009), *23*, 391–407. Printed in the USA.

doi:10.1017/S0890060409000146

Merged ontology for engineering design: Contrasting empirical and theoretical approaches to develop engineering ontologies

SAEEMA AHMED[1] AND MARIO ŠTORGA[2]

[1]Department of Management Engineering, Technical University of Denmark, Lyngby, Denmark
[2]Faculty of Mechanical Engineering and Naval Architecture, University of Zagreb, Zagreb, Croatia

(RECEIVED September 20, 2007; ACCEPTED September 11, 2008)

Abstract

This paper presents a comparison of two previous and separate efforts to develop an ontology in the engineering design domain, together with an ontology proposal from which ontologies for a specific application may be derived. The research contrasts an empirical, user-centered approach to developing the ontology engineering design integrated taxonomies (EDIT) with a theoretical approach in which concepts and relations are elicited from engineering design theories design ontology (DO). The limitations and advantages of each approach are discussed. The research methodology adopted is to map the ontology through examining each of the concepts and relations contained within each of the ontologies DO and EDIT with respect to the other. The comparison process results in an examination of both ontologies, with a few changes resulting from this. The importance of the two different approaches, one that is theoretically sound and another that is applicable, is recognized and argued. Finally, the merged ontology for engineering design is proposed as a template ontology that can be tailored by researchers and practitioners for a specific context.

Keywords: Design Theory; Empirical Research; Engineering Design Ontology; Indexing Knowledge; Ontology Mapping and Merging

1. INTRODUCTION

There is a long tradition in computer science and artificial intelligence that equates knowledge with facts in order to represent knowledge (Brewster & O'Hara, 2007). This tradition has led to the contemporary explosion of interest in ontology as a medium for knowledge representation occurring in many domains (Davis et al., 1993). An ontology can be described as an explicit specification of a shared conceptualization, which can be taxonomically or axiomatically based (Gruber, 1993). In general, an ontology consists of three parts: *concept* definitions, *attribute* definitions, and further *inference* definitions.

The *concept* definitions set up all the types of concepts in the domain. There can be three parts to the concept definitions:

1. Concept taxonomy is common to most knowledge representation languages, and through it is specified the nature of the categories in terms of generalization and specialization.
2. Concept defaults specify for each concept what the default values are for any attributes.
3. Concept restrictions specify the constraints on the values for each concept, for example, the types of values or number of values that are acceptable.

In the simplest case, an *attribute* for a concept just has a value but attributes may also express relationships between concepts. An attribute definition may have up to three parts as well:

1. The attribute taxonomy specifies the generalization/specialization between attributes.
2. Relational attribute inverses provide a form of inference allowing the addition of a relation in the opposite direction to the forward link between concepts.
3. The relational attribute maybe defined by attribute restrictions such that it can only appear between concepts of certain types (domain/range restrictions), or can only appear a specified number of times (cardinality restriction).

Reprint requests to: Saeema Ahmed, Department of Management Engineering, Technical University of Denmark, Nils Koppels Allé, DK-2800 Kgs. Lyngby, Denmark. E-mail: sah@mek.dtu.dk

The final part of an ontology is the specification of additional *inference* that the model provides. Examples of this are forward and/or backward chaining rules, path grammars, subsumption and/or classification, demons, and so forth.

As new applications of ontologies have appeared, it is evident that whereas some types of knowledge are eminently suitable to representation by ontologies (taxonomic information most obviously), others may not be. Furthermore, others argue ontologies as artefacts are unsuited to real-world applications once they are beyond a certain level of complexity; in other words, lightweight ontologies are acceptable, but there is a trade-off between expressivity and usability.

2. RESEARCH MOTIVATION AND AIMS

The motivation of this research is to understand the benefits and limitation of approaches of developing engineering design ontologies (DOs) and hence contribute to a general understanding of the engineering DO field. In particular, the approaches for two independent research studies leading to the development of two ontologies were examined: engineering design integrated taxonomies (EDIT; Ahmed, 2005) and DO (Štorga et al., 2008). EDIT employs an empirical and user-centered approach for its development, and DO is modeled from engineering design theories. EDIT was developed through a bottom-up approach developed in the context of the aerospace industry, with a focus upon a specific application of indexing design knowledge. DO was based upon a theoretical model and utilized a top-down approach, but the EDIT ontology was intended for a wide variety of applications in engineering design.

This research is concerned with the development of ontologies for supporting engineering design and has two main aims:

1. to examine and contrast two perspectives to developing ontologies in the engineering design domain, namely, theoretical and empirical approaches that were employed in previous research results (EDIT and DO), and to discuss their advantages and limitations of each of the results obtained;
2. to investigate if a combined approach may improve the shortcomings of each ontology.

Hence, a general merged ontology is proposed, termed merged ontology for engineering design (MOED), which can be tailored for a practitioner or researcher's particular needs. MOED was developed as an outcome of contrasting and mapping of EDIT and DO.

The next section describes a brief overview of ontologies in engineering design and approaches to developing an ontology in Section 3. Section 4 reports the research approach adopted for this research, and in Section 5, the results from contrasting the two approaches to developing ontologies and a proposal of a merged ontology are presented. The final section discusses the implications of the research together with conclusions.

3. BACKGROUND

A complete discussion of ontologies and their development isoutside the scope of this paper; however, a brief introduction to the engineering DO follows. The motivation for developing an ontology in the engineering domain includes knowledge sharing and developing a standard engineering language. For example, a structured basis for navigating, browsing, and searching engineering knowledge through the descriptions of the ontologies. This is particularly useful when engineers are unaware of the information available, and hence, they can retrieve documents by submitting natural language queries or navigating the ontology space. Another important motivation for building ontologies is the integration of knowledge models in different subdomains of the engineering process into a coherent framework (Uschold & Gruninger, 1996). Example applications include business process reengineering (an integrated knowledge model of the enterprise and its processes, organizations, products, goals, and customers is needed), distributed design among multicultural teams (where different participants need to communicate and solve problems), and in concurrent engineering and design. The application area of engineering ontologies can be divided into three main areas:

1. a foundation for the business/engineering processes formalization,
2. a foundation for achievement of full interoperability between different participants (humans and computer systems) of engineering process, and
3. a foundation for the effective implementation of engineering knowledge management methods and tools.

Previous research efforts in developing ontologies for the engineering design (Sections 3.1 and 3.2), in addition to the approaches undertaken by the authors for the development of ontologies from an empirical and theoretical viewpoint are described together (Sections 3.3 and 3.4) with the approach to integrate these efforts in the following sections.

3.1. Engineering ontologies

The importance of a sharable ontology for systematic exchange and management of knowledge in the engineering design field is recognized by many researchers. Early research projects include YMIR, which specifies a taxonomy of concepts for engineering design that define the semantic of design knowledge in multiple engineering domains (Alberts & Dikker, 1994). The concepts that YMIR represent are generalization of concepts used in the individual design domains, such as electrical engineering, mechanical engineering, and civil engineering. The same ontological basis was used for the integration of design synthesis knowledge and design standards in design process. The Rezgui (2006) ontology for knowledge management in construction industry was developed through eliciting concepts from documents, through the removal of stop words and summarizing the documents.

The ontology works together with profiles containing users preferences to support knowledge management. Two approaches based on formal ontologies to better organize information handled during the engineering design process are proposed by Gunderan et al. (2007): Process Specification Language as a key enabler for process-based communications and information exchanges and methods and tools facilitating communications through model-based transformation approaches. Darlington explores the process of developing ontologies for use in real-world problem solving and showed, by example implementation, how an ontology developed to capture suitable domain knowledge may be used for supporting engineering design requirement captures (Darlington & Culley, 2008). Hence, illustrating the general potential of ontologies to support the engineering design process support.

Ontologies have also been used to describe design activity. Sim and Duffy's (2003) ontology categorizes activities as design definition, evaluation, and management, which is based upon the literature. The ontology is seen as providing a consistent and coherent description of design activities upon which design education, system developers, and design researchers can extend further. The ontology of Sim and Duffy (2003) is extended with the use of design structure matrices to analyse information flows from activities for the generic design activity ontology of Kumar and Mocko (2007).

Other applications of ontologies in engineering design include the systematization of functional knowledge (Kitamura & Mitzoguchi, 2004). Kitamuara and Mitzoguchi applied their ontology for functional decomposition in the electric industry in Japan with successful results. A final application of ontology is the product family ontology development methodology (Nanda et al., 2006) that combines research in formal concept analysis, Semantic Web, and Web Ontology Language (OWL) in order to provide a structured methodology for product family ontologies. It aims to facilitate a shared, consistent, and traceable ontology development process within a diverse product development team.

A brief overview of ontologies and their applications in engineering design domain has been described, and the diagrammatic summary is provided below (Table 1).

The concept of warrants or the authority on which the ontology can be based is discussed in the following section to better understand the differences in origin an approach when building engineering design ontologies.

3.2. Approaches to developing ontologies

Warrants are the authority that is evoked by the classification to justify and verify decisions of the structure and choice of classes and concepts included in an ontology (Beghtol, 1986). Hence, the types of warrants that are possible and those employed by the two ontologies (DO and EDIT) are reviewed to obtain a greater understanding of the limitations of these approaches.

The two warrants referred to in the standards are literature and user. These and others found in literature are described by the following [National Information Standards Organisation (NISO), 1994]:

- Literary warrant: words and phrases from the literature determine the formulation of descriptors (NISO, 1994)
- User warrant: representation of the inclusion of a concept is due to frequent requests for information on the concept (NISO, 1994)
- Scientific warrant: the best philosophical and scientific consensual thinking (Beghtol, 1986)
- Cultural and epistemological warrant: "ensuring that concepts and semantic relations are dependent on the broader cultural context and incorporate different cultural views (Beghtol, 1986).

The vast majority of engineering design ontologies reviewed in Section 3 employs the literary warrant only. However, as the EDIT (further described in Section 3.3) approach originated through eliciting user concepts from interviews and the literature was consulted to identify taxonomies, it can be described as closest to the concepts of the literary and user warrant, although not quite the same. The users concepts were elicited from interviews that determined the choice of concepts rather than the users requesting the concepts. Further, the DO (described in Section 3.4) originated from eliciting concepts and their relations from the existing theoretical foundation in the engineering design domain, followed by derivation of terms and definitions based on the epistemological foundation of high-level ontologies [suggested upper merged ontology

Table 1. *Recent ontology development in engineering design*

Ontology Research Project	Author (Year)	Domain of Interest
YMIR: taxonomy of concepts for engineering design	Alberts & Dikker (1994)	General domain
Design activities ontology	Sim & Duffy (2003)	Process domain
Ontology for functional knowledge	Kitamura & Mitzoguchi (2004)	Functional domain
Engineering design requirements ontology	Darlington & Culley (2005)	Requirements domain
EDIT	Ahmed (2005)	General domain
Design ontology	Štorga et al. (2008)	Product domain
Rezgui ontology for knowledge management in construction industry	Rezgui (2006)	General domain
Process specification language	Gunderan et al. (2007)	Process domain
Product family ontology development methodology	Nanda et al. (2006)	Product domain

Table 2. *Contrasting EDIT and DO warrants*

Ontology	Warrant	Source	Purpose
EDIT	User	Industry interviews Industry documentation Scientific literature	Design knowledge indexing
DO	Literary	Design science theories available in scientific literature	Theoretical design models formalization

(SUMO)], can be described as based upon a scientific and epistemological warrant (Table 2).

3.3. EDIT

EDIT was developed through a systematic methodology aimed at gaining a cognitive understanding of engineering designers (Ahmed, 2005). The ontology was developed within the context of the aerospace industry, and its primarily application is in managing design documentation through the provision of a visible indexing structure for users to search for knowledge. EDIT was developed through a user-centered approach following the conceptual models of users, in this case engineering designers. Eighteen designers were interviewed to understand how designers described the process of designing of particular product from two companies. Hence, the root concepts of EDIT, shown in Figure 1 (together with the roots concepts of DO further described in Section 3.4.), were elicited from these interviews, through classifying the designers' descriptions of design process. From this analysis, four root concepts emerged:

1. The *design process* itself, that is, a description of the different tasks undertaken at each stage of the design process, for example, conceptual design, detail design, and brainstorming.
2. The physical *product* to be produced, that is, the product (component, subassemblies, and assemblies) using part of relations, for example, a cup or the handle of a cup. In the case of designers working on a subassembly or a component of the whole product, the components and assemblies that share a physical or functional interface with, what is being designed would also be considered. For example, when designing a turbine blade, the disk that holds the blade also needs to be considered to ensure that the interface between the disk and blade is appropriate.
3. The *functions* that must be fulfilled by the particular component or assembly. For example, one of the functions of a compressor disk is to secure the compressor blade or one of the functions of a cup is to contain liquid.
4. The *issues*, which are considerations the designer must take into account while carrying out the design process, for example, the unit cost or manufacturing considerations.

The root concepts formed individual taxonomies within the ontology and were validated through indexing a set of 92 documents and through interviews. Individual taxonomies were either identified from literature, as was the case for the function taxonomy where the functional basis from Hirtz et al. (2006) is employed, or created if an appropriate taxonomy could not be found, as was the case for the remaining three taxonomies. The EDIT ontology consists of around 1000 classes. The EDIT ontology once populated with instances from the aeroengine is closer to 2000 terms, with the largest contribution from the product taxonomy. In addition to static relationships between classes, dynamic relations across taxonomies are created based upon a set of rules. Dynamic relationships between concepts are extracted as the ontology is populated with instances (described in Ahmed, 2006*a*, 2006*b*). The methodology employed during EDIT resulted in the development of a generic methodology to develop engineering DOs that can be found in Ahmed and colleagues (Ahmed, 2005; Ahmed et al., 2006). The process of

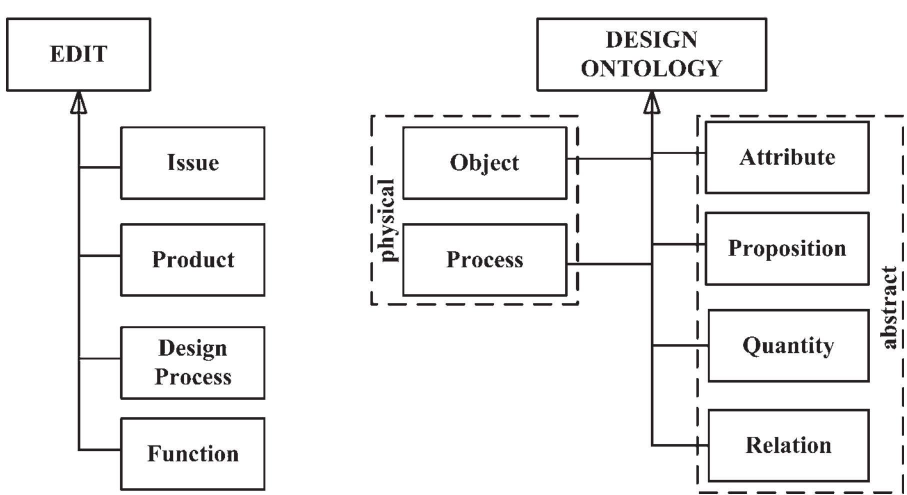

Fig. 1. The EDIT and DO root concepts.

https://doi.org/10.1017/S0890060409000146 Published online by Cambridge University Press

developing the ontologies was conducted in several stages. The stages were

1. identifying the taxonomies that form an engineering DO (referred to as the root concept of the taxonomy),
2. searching for existing taxonomies for each of the root concepts from the previous stage,
3. creating taxonomies if no existing taxonomy was found,
4. testing the taxonomies for the particular application (in this case for indexing design knowledge),
5. building a thesaurus for the integrated taxonomy, and
6. refinement of the integrated taxonomy.

Each of the six stages of the methodology has a clear output and at least one clear evaluation step, and is summarized in Figure 2. Each of the three columns illustrates the methodology; research methods employed, and evaluation procedure for each of the six stages. Each of the rows (excluding title row) represents the six stages of the methodology.

3.4. DO and SUMO

The DO project started with the recognition of the "design as a product" ontology as a main presumption for the successful knowledge management and exchange among different participants in product development process. In building a general DO, the domain description vocabulary was defined as the desired research result (Štorga et al., 2005, 2007, 2008). Theoretical literature related to engineering design and designing was analyzed with the purpose to understand terminology applied amongst different researchers in the domain. The main conclusion from this review was that the domain terminology is not unified, and definitions provided were not consistent or clear. Many relations between terms in theoretical literature were described as causal (if at all); these informal models cannot be utilized for any form of automatic reasoning. These findings motivated the research to build a formal model of the theoretical background: DO. The DO development process was conducted in six stages following

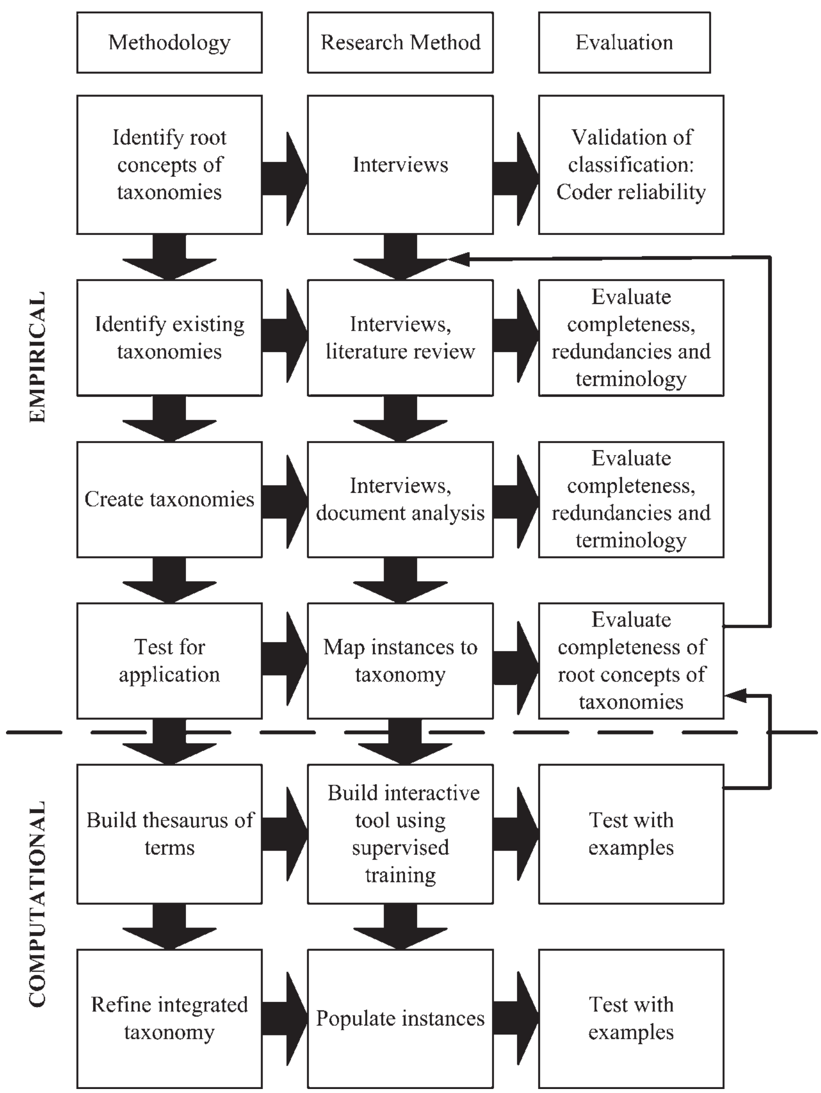

Fig. 2. The EDIT methodology for building an ontology.

https://doi.org/10.1017/S0890060409000146 Published online by Cambridge University Press

the previously mentioned EDIT methodology; however, the research methodology employed focused upon understanding engineering design theory as was mentioned before rather than the described empirical approach (Fig. 2). This phase included domain documentation analysis (theoretical models, industrial reports, and software documentation), identification of the key concepts and relations between them, and classification of the concepts and relations into taxonomies.

The genetic design model system (GDMS; Mortensen, 1999) was selected as the main theoretical background for extracting the DO terms and definition. It was selected because it is built upon a strong theoretical background including the theory of technical systems (Hubka & Eder, 1988), theory of properties (Hubka & Eder, 1988), theory of domains (Andreasen, 1980), design process theory (Hubka, 1976; Pahl & Beitz, 1988), and theory of dispositions (Olsen, 1992; Table 3).

Knowledge about the product/design as the result of the development process is by Mortensen (1999) centered on four different conceptual model object or viewpoints:

1. The design: defines functional, organ, and part view on the design/product; inherent properties that are possessed by the design/product itself, that is, strengths, ductility, and so forth; and design/product view relevant for the different meetings during its life cycle.
2. The life cycle phases: technology model that defines the considerations during the product life, product life model, activity model describing intended and realized activities between the design and the operand/environment, and relational property model, that is, costs, lead time, quality, and so forth.
3. The life phase systems: the systems that gradually realize product life, that is, production, sales, and services with inherent properties.
4. The product assortment: a design normally belongs to the product family or product assortment that can be described by a plan that consists of an assortment/family elements structure and constraints between them.

After the extraction of vocabulary entities from theory, the main concepts were characterized and formally defined following Mekhilef et al.'s (2003) four levels of formalization procedure: epistemological-, domain-, application-, and project-modeling level. The SUMO (www.ontologyportal.org), an effort by IEEE (www.ieee.org) collaborators from the field of engineering, philosophy, and information science, was selected as an epistemological foundation for building the DO. SUMO originally concerned itself with metalevel concepts (general entities that do not belong to a specific problem domain). SUMO is aimed at creating a framework where ontology developers may utilize a common knowledge for derivation of the more domain-specific ontology (Niles & Pease, 2001). Some of the distinct advantages of the SUMO proposal in a comparison to the other high-level ontology efforts are described here (Niles & Pease, 2001):

Table 3. *Relating the GDMS to existing design model systems*

Model Types	Krause (1988)	Blessing (1994)	Anderl et al. (1991)	Salminen & Verho (1991)	Tomiyama et al. (1989)	Rosenman & Gero (1998)	ISO STEP (1997)	Meerkam (1995)
Model Object: The Life Cycle								
Technology model	■							
Transformational model	■			■		■		
Relation property model	■			■		■		■
Product life model	■							■
Model Object: The Design								
Working organ model		■	■	■	■	■	■	■
Function model	■	■	■	■	■	■		■
Inherent property model	■	■		■		■		
Part model	■	■	■	■	■	■	■	■
Model Object: The Life Phase System								
Life phase system model	■			■				
Inherent property model	■			■				
Model Object: The Product Assortment								
Product family plan								

Data are according to Mortensen (1999).

1. The SUMO is the working effort sponsored by open-source engineering community. This means that potentially users of SUMO can be more confident that this upper ontology will eventually be embraced by a large class of users.
2. The SUMO was constructed with reference to pragmatic principles. Hence, any distinctions of strictly philosophical interest have been removed.
3. The entire SUMO is mapped to the WordNet® lexicon (wordnet.princeton.edu), providing a link between the formal content expressed in SUMO and natural language, that is, paraphrasing the hard to read logical inscription of axioms into natural language.

Accordingly to the SUMO proposal, the vocabulary of the DO has been classified into six main subcategories as shown in Figure 1. At the top level of the SUMO hierarchy, the concept of *Entity* subsumes the concepts *Physical* and *Abstract*, where the former category includes everything that has a position in space/time, and the latter includes everything else. From the viewpoint of the DO research, the concept of *Physical* subsumes the disjoint concepts of *Object* and *Process*. The concept of *Object* is the most general concept of the *Entity* that exists in space. The concept of *Process* corresponds to any sustained phenomenon or one marked by gradual changes (space/time). Returning to the highest level distinction in SUMO hierarchy, the concept of *Abstract* subsumes four disjointed concepts relevant for the DO: *Attribute, Proposition, Quantity*, and *Relation*. The concept of *Attribute* includes all qualities, properties, and so forth, of an *Entity* that are not regarded as *Object*. The concept of *Proposition* corresponds to the notation of semantic or informational content. The *Quantity* concept is understood as a count independent of an implied or explicit measurement system together with a particular unit of measure. The concept of *Relation* is an abstraction belonging to or characteristic of ordered *Entity* tuples and connects two or more concepts. An example of a simple definition that was extracted and formally defined is shown below. The definition provided by GDMS was in the first step interpreted utilizing the terms included in the DO and then formalized as a DO definition.

The ontology was evaluated for reliability using the Cohen kappa coefficient of reliability following the EDIT methodology, which takes into consideration the agreement of the relevant experts in the researched field and subtract the percentage of the agreement that can be expected from chance (Bakemann & Gottam, 1997; Ahmed et al., 2007). In the final step of the research, a computer thesaurus has been created using the Ontoprise® ontology development environment (www.ontoprise.de). Using the thesauri, the knowledge evolved during a real product development case study was described, and the set of instances created were used for the ontology model to check consistency and for refinement.

4. RESEARCH METHODOLOGY

4.1. Ontology mapping: The state of the art

During the presented research on confronting the two ontologies, the authors were faced with an enormous diversity of work claiming relevance to ontology mapping and merging. For example, terms and works encountered in the literature include: alignment, merging, articulation, fusion, integration, morphism, and so forth. Given this diversity, it is difficult to identify problem areas and comprehend solutions provided. Kalfoglu and Schorlemmer (2003) scrutinized the literature and critically reviewed works originating from a variety of fields. They understand ontology mapping and merging as the task of relating the vocabulary of two ontologies that share the same domain of discourse in such a way that the mathematical and logical structure of ontological signatures and their intended interpretations, as specified by the ontological rules, are respected. From the literature reviewed, the difficulty in identifying transparent and repeatable procedures for mapping ontologies is evident. Even within the field of engineering design, ontologies may focus upon design activity or product relations, and hence, cannot easily be compared. Benchmarking, as employed in domains such as optimizations, is also difficult to apply as ontology is not as easily measurable as algorithms. Most approaches reviewed

GDMS definition:	"Function is ability of machine to deliver a purposeful effect."
Ontology building interpretation:	If ?MACHINE is an *Instance of Machine* and ?MACHINE is an *Instrument* of ?PRODUCTLIFECYCLEPHASE, then there exists ?FUNCTION so that *?PRODUCTLIFECYCLEPHASE Results* with purposeful ?EFFECT.
Formal ontology definition:	(=> (*and* (*Instance* ?*MACHINE Machine*) (*Instrument* ?*PRODUCTLIFECYCLEPHASE* ?*MACHINE*)) (*Exists* (?*FUNCTION*) (*Result* ?*PRODUCTLIFECYCLEPHASE*?*EFFECT*)))

https://doi.org/10.1017/S0890060409000146 Published online by Cambridge University Press

describe mapping between two ontologies where the process needs to be repeated for an additional mapping. The mapping procedure employed for this study is described in the following section. The merging of the two ontologies was an outcome of this activity, and is described in the Section 5.

4.2. Mapping approach in presented research

For the presented research, the methodology for mapping the two engineering DOs approaches derives from different perspectives, as it was difficult to apply any *one* proposed approach, because of previously described differences in the ontology scope and warrant. Similarly applying any automatic tool to do this mapping/merging was not considered for the same reasons, and both of the proposed ontologies were not semantically strong enough to apply formal mathematical and logical methods. Hence, the most relevant approaches for the particular project were ideas from several approaches that were intuitively combined.

The starting point of the research is aligned with the idea of integrating ontologies based on taxonomic features and detection of synonymous concepts in the two ontologies as described by Fernandez-Breis and Martinez-Bejar (2002). The difference between the applied procedures is that it was not possible to consider the attributes of concepts (because they are not defined). Therefore, it was not possible to define a typology of equality criteria for concepts for automatically integration.

The research methodology adopted was for the authors to examine each of the concepts and relations contained within one of the ontologies, DO, with respect to the other, EDIT. Because not all of the concepts and relations contained in EDIT are contained in DO, this process was then repeated but starting with the concepts and relations in EDIT. This process involved understanding the different terminologies that may have been used to describe the same concept. Because the authors are also the originators of the two contrasted ontologies, it was easier to ensure that mapping went beyond terminology than if this were not the case. In addition to contrasting the concepts and relations of ontologies, the structures of the ontology, that is, placement of concepts at different levels, the parents and siblings of each concept were also examined.

The ontologies were examined in order to do the following:

- to identify concepts that were in common, which may have had different labels;
- to identify concepts that were only present in one of the two ontologies;
- to identify relations employed between concepts, for those that were common between the two ontologies, and to understand the relationships between the different concepts within each of the ontologies. This is important, as even if both ontologies contained the same concepts, their placement within the particular ontology could be different because of the relations employed between them;
- to compare the placement of common concepts in each of the two ontologies.

Each ontology was already evaluated during its development. Hence, the evaluation here focused upon the following:

1. What is missing and is redundant from DO?
2. What is the applicability of DO? Identifying concepts that are too abstract for a specific purpose.
3. What is missing and redundant from EDIT?
4. Evaluate the theoretical background of EDIT, thus moving from a concrete and specific case study to a generic ontology.

5. RESULTS

The comparison of the two approaches and their results brought an understanding of the main differences between them: the starting point of the DO is to describe "design as a product," and of EDIT is to describe "design as an activity," incorporating both product and process. Realizing this difference was the key to understanding the nature of the ontology's concepts and relations to characterize the overlapping and mapping between the two ontologies. A second difference was identified: the hierarchical structure of the DO vocabulary represents the *is a kind of* relationships, highlighting a taxonomy of general and more specific concepts and relations of different kinds. In contrast, the structure of the EDIT concepts contains different kinds of relationships between concepts: *part of, type of*, and *has a*. Because of this understanding, the authors decided to consider separately the nature of the concepts and those of the relations to ensure mapping at the same level.

5.1. Mapping of the concepts

At the start of the research, it was recognized that mapping all the concepts directly from one ontology to another was not expected. After the preliminary research, it was concluded that it was relatively easy to map the top level concepts as their definitions are easily understandable and similar from both the theoretical and the practical viewpoint (see Fig. 3). Figure 3 illustrates the mapping of concepts at the top level, illustrating how each are mapped together and also the relations between concepts at this top level. For the concept on the lower levels, the situation was not so obvious, as concepts of the same kind in one ontology maybe placed differently in the second.

https://doi.org/10.1017/S0890060409000146 Published online by Cambridge University Press

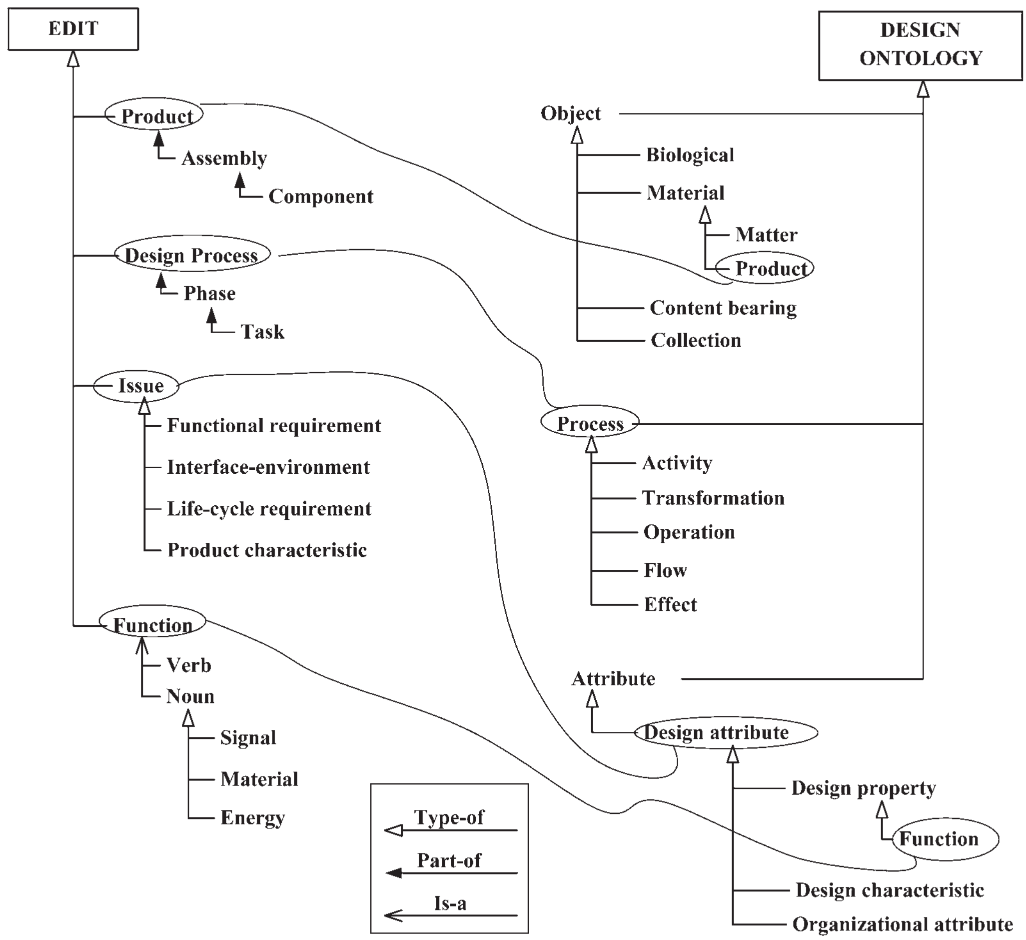

Fig. 3. The EDIT and DS mapping at the top level.

The mapping of the concepts from the EDIT to the concepts in DO could be done as follows:

- The *Product* (EDIT) could be mapped to the concept of the *Product* (DO) as they both represent the physical result of the product development process.
- The *Design Process* (EDIT) could be mapped to the concept of the *Process* (DO) because both represent the technical process as a chain of activities that should be completed to define the physical product.
- The *Issues* (EDIT) could be mapped to the concept of *Design Attribute* (DO) because both represent the considerations that should be addressed by the engineers during the product development process to describe their solutions.
- The *Function* (EDIT) could be mapped to the concept of *Function* (DO) as they both represent the functions or the expected purpose that each product (including component and assemblies) should address.

Mapping of the concepts in the opposite direction, from the DO to the EDIT, brought out more problems:

- In DO, *Content Bearing Objects* exists explicitly as a physical object that is bearing some informational content (e.g., document). Conversely, *Document* is not explicitly defined as a concept in EDIT but one that utilizes the ontology to be indexed, and it is an instance to which any number of concepts may be linked.
- *Operation* is defined in the DO as the smallest single step of the *Activity*, but it is outside of the scope of EDIT, it is too prescriptive, and therefore was not mapped. The *Transformation* defined in the DO could be mapped to the *Phase* of the *Design Process* in EDIT, and the *Activity* from the DO responds to the single *Task* defined as a smallest part of the phase in EDIT.
- In DO, *Organizational Attributes* are related to each concept and relation (e.g., time of the creation, i.d., the creator, time of the last change, etc.). In EDIT, they are implicit and linked to the particular *Document* as source of the concepts and relations between them.
- The concept of the *Flow* from DO could be mapped to the *Energy Flow* function in the EDIT, and the DO con-

cept of *Effect* could be mapped to the *Energy Effort* function in the EDIT.

- DO abstract *Propositions* like *Idea*, *Fact*, *Principle*, *Plans*, and so forth, in EDIT exist only implicitly and are described in *Documents*, so they cannot be directly mapped.
- The DO concept of *Collection*, including concepts of *Group, Assortment*, and *Family*, cannot be mapped into EDIT, because these concepts were not anticipated by EDIT. This also applies to the concept of *Quantities*.

5.2. Mapping of the relations

Mapping of the relations was only possible on a general level. The reason for this is that in EDIT all the relations besides those mentioned *part-of, type-of*, and *has-a* are dynamic (Ahmed, 2005). They are not part of the EDIT definition, are generated dynamically as the result of a search through knowledge sources (documents) based upon prescribed rules, and differ from case to case. In contrast, DO specifies the relation taxonomy as a static structure, and the instances of relations between the concepts could be defined based upon these rules.

The three main relations that are part of the *EDIT* taxonomies definition could be mapped to DO as follows:

1. The *part-of* (EDIT) relation could be mapped to the *Compositional* relation as is defined in DO, describing the relation between the complex entities and its constituent.
2. The *has-a* (EDIT) relation is utilized by taxonomies that are part of the EDIT and could be mapped to the class of the *General* relations (DO) describing that an *Entity* is characterized by another *Entity* (e.g., the function is characterized by verb and noun).
3. The *type-of* (EDIT) relation is used in both proposals as a main relation that is utilized for building the taxonomies: in DO for the whole concepts' taxonomy in a form of the *is-a* relation, and in EDIT for describing the *Issues* and *Function* taxonomies.

The seven main classes of the relations described in DO could be mapped to EDIT as follows:

1. *Compositional* relations (DO) could be mapped to the *Part-of* relation (EDIT) as is described earlier.
2. *Spatial* relations (DO) could be mapped to the physical relations that could be derived between the *Product* and another *Product* in EDIT, representing the physical connection that exists between the *Products* and *Issues–Product Characteristic–Geometry–Geometric interface*.
3. *Case role* relations (DO) represent the role of an *Entity* in a *Process*, and therefore could be mapped to the relations that could be derived between the *Design Process* and *Issues* domains (EDIT).
4. *Dependency* relations (DO) could be mapped to the functional relations that could be derived between the *Product* and another *Product* (EDIT), representing the abstract connection between the two products.
5. *Influence* relations (DO) could be mapped to the relations that could be derived between the *Issue* and another *Issue* (EDIT) and also between the components, representing the abstract connection between them.
6. *Temporal* relations (DO) could be mapped to the *Phase* structure in design process taxonomy (EDIT), representing the time line of the *Design Process*.
7. The *General* relation (DO) could be mapped to the relations that could be derived between the four main taxonomies (i.e., top four root concepts) contained within EDIT.

5.3. Evaluation

The ontologies were examined in contrast to the methodology from which they were developed; DO, which is based upon a design theory, was examined for its applicability to an applied industrial context, whereas EDIT, which is empirical, derived and for a particular application in mind, was evaluated for its theoretical background. The evaluation described here is to understand the limitations of the approaches to develop ontologies, which may result in changes in each of the ontologies. The evaluation does not focus on evaluating each of the ontologies, as this has already been carried out. EDIT employed the *EDIT* methodology described earlier and found in detail in Ahmed et al. (2007). The methodology has an evaluation at each of its six stages; a modified version of this was applied for the DO (Štorga et al., in press).

5.3.1. Evaluation of DO

During the process of comparing the ontologies, some of the concepts within the DO were reevaluated. These changes were made as a result of the comparison with EDIT; the changes were made if the original concept (or its position within the ontology) was inconsistent, or if it was beyond the limits of an ontology for engineering design. One of the difficulties with an ontology that has a theoretical basis is setting the limits and boundaries; by having a particular purpose (i.e., a concrete application) it is easier to evaluate whether a concept is necessary, and to understand its positioning.

The *Object* domain should be reconsidered to understand if/how it differs from the concept of *Material* domain. In EDIT, *Material* is used as part of the *Function* taxonomy together with the concepts of *signal* and *energy*. It is important

https://doi.org/10.1017/S0890060409000146 Published online by Cambridge University Press

to understand that both approaches refer to the same kinds of material, signal and energy, because taxonomy of the functions used in EDIT is originated in the work of Hirz et al. (2001) based on the same background as used in DO. *Object* in EDIT is embedded much deeper, that is, at a lower lever than in DO. The thinking behind this is related to literature that describes energy, material, and signal as the three main concepts that pass through a technical system. Similarly, the *Energy* (DO) should be reconsidered to be moved from the *Process* domain into the *Functional Qualities*, representing the amount of something that could be measured by standard units.

The concept of *Symbol* was removed from the ontology, as was *Abstract–Propositions–Element*, as it was believed to be beyond the boundaries of the ontology. In addition, the following concepts were moved within the DO:

- *Flow* and *Effects* were moved from *Process* to *Abstract–Quantities–Functional Quality* domain,
- *Abstract–Propositions–Behavior* were moved to *Attributes*, and
- the concept of *Signal* as a physical *Content Bearing Object* was reconsidered as *Abstract*.

5.3.2. Evaluation of EDIT

The comparison of DO with EDIT resulted in the addition of the concepts *family* and *assortment* related to *product*. Because EDIT is created primarily to provide a visible browsing and navigational structure when searching for knowledge, and as an ontology to index engineering knowledge, differences between the treatment of *Material, Energy, and Signal* within EDIT and DO became apparent; these are treated as an abstract within EDIT, which is not the case within DO. *Function* is an abstract concept, that is, the *Function* that a *Product* (component or assembly) needs to fulfill may exist before a concept or a product exists. The *Function* taxonomy within EDIT uses combination of verbs and nouns; the nouns are not all abstract concepts, for example, under *Material* there is *Material–Solid Object*. However, the use of them as a combination to represent a function means that the concept is now abstract. As a noun independent of a *Verb–Noun* combination (describing a function) *Material–Solid Object* is physical, and similarly *Human* (part of *Function–Noun–Material–Human*) maps directly to DO *Physical–Object–Biological–Human*. This difference is related to the application of the ontology: *Material*, including *Human* and *Material Object* are physical, but the use of them as part of a combination of verb–noun is abstract. If *Material* were place as physical (and therefore not part of the function taxonomy), it would be difficult for a user (engineering designer) to locate, for example, a solid object when trying to describe a *Function*, as the concept will be located away from *Function*. Hence, for pragmatic reasons, the position was not changed in EDIT.

5.4. Discussion on mapping and evaluation of two approaches

It was found that a theoretical view point may ensure that the concepts and relations are mapped correctly; however, there is a difference between a theoretically consistent ontology and one that is accessible for engineering designers to use in a specific context. For example, the concept *Product*, if the concept of the product is to be placed consistent to the theoretical approach (as employed by DO), it would appear as part of the attributes in *Entity–Physical–Object–Material Object* and hence would be embedded very deep within the ontology or in EDIT would be part of the *Function–Noun–Material–Solid Object*. In the application of EDIT, and indeed many engineering DOs, the physical product (or service) is a central view for the users of the ontology; for example, if searching for knowledge, documents related to other similar documents are very relevant. Hence, there is a strong argument for *Product* to be at a much higher level in the ontology than it would otherwise be. Therefore, there is a need for two different views for ontologies: one that is theoretically sound, and contributes to engineering design theory and understanding, whereas another in a view that is applicable. For each new application new classes may be needed below the top level of the EDIT ontology; however, the theoretical ontology does not necessarily need these.

The two ontologies may exist with different structures but overlapping concepts. In addition, as users' conceptual models are different depending on their roles or their organization context, that is, they understand the *same* concept with different terminology, each ontology derived for a particular purpose may use different labels for concepts in different contexts and for different users; however, the concepts, relations, and structures stay the same.

5.5. MOED proposal

Based upon the previously described evaluation and mapping, the concepts and relations from both ontologies were merged into single proposal: MOED (www.cadlab.fsb.hr/moed). MOED is aimed to be a starting template for researchers and practitioners alike that can be tailored to build ontologies for their particular needs and context. To ensure backward compatibility with the high-level ontologies, SUMO in this case, the four resulting taxonomies (object, process, attribute, and relation) classified between the physical and abstract domains, were generated, merging the knowledge and understanding gained from the previous research work on building two ontologies from different perspectives. Figure 4 shows the concepts on the first two levels of the MOED taxonomies and indicates the relations defined between them.

The definitions of the terms in the first two levels of the MOED taxonomies are shown in the following tables. The

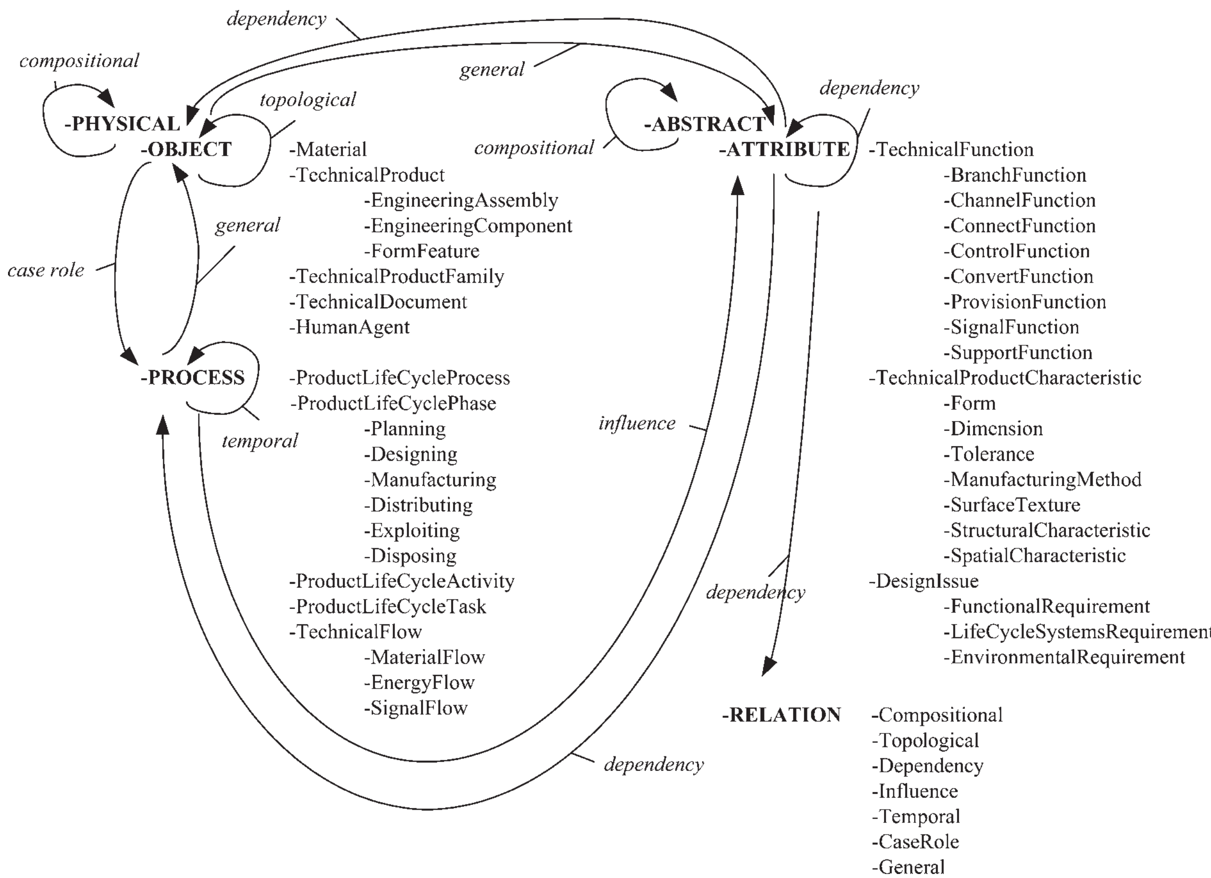

Fig. 4. The MOED concepts and relations at the top level.

level of the MOED vocabulary term among specific domains is indicated by the number of dots in the prefix of every particular term and shown in the left column of each table. The definitions of the concepts and relations behind terms, derived based on EDIT, DO, SUMO, and WordNet inputs, are quoted in the right column, as well as definition of the SUMO term that is specialized in the MOED term to ensure compatibility (the SUMO concepts are written in *italic* style). The prefix "&%" in definitions refers to the MOED term defined in one of the taxonomies (Tables 4–7).

The axioms and rules that already exists in SUMO are inherited from the terms defined in SUMO and specialized for the MOED terms. To characterize relations usefully, they are, in addition, defined by axioms considering their logical properties of symmetry, reflectivity, and transitivity that enable inference of the new facts/knowledge among the existing model.

The criteria for evaluating the ontology proposal should include ontological completeness, clarity, and coherence (Wand & Weber, 1993; Uschold & Gruninger, 1996). The evaluation of the MOED ontological completeness is based on the presumption of involving two different ontologies, empirical and theoretical based, that have been evaluated previously. Therefore, the MOED ontology is sufficiently expressive in eliciting the shared meaning of the concepts and relations describing the phenomenon of the engineering design process and its results. Ontological clarity is concerned with the interpretation of meaning of the engineering design concepts and relations applied between them. The aim of MOED has been to derive a shared understanding of the meaning of the concepts and relations identified by categorizing the concepts and relations from engineering design research and design practice according to the high-level ontology. This has been achieved by comparing and contrasting the descriptions given by two different ontological approaches and resolving any ambiguities that arose. As a result, a theoretically consistent and application-sound categorization and definition of each concept and relation are derived. Hence, MOED should address the shortcomings of both DO (which is theoretical) and EDIT (which is for a specific application of indexing knowledge). For example, an ontology to support planning in the product development process in the aerospace industry, could take MOED as a starting point, where the physical process (Table 6) forms the main body, together with relevant concepts from the physical–object domain (to describe the projects that the product development processes refer to) and concepts such as human relations (Table 5). In contrast to the previous ontologies, EDIT did not include human relations, as this was not relevant for the specific application of indexing design knowledge, and DO would be structured from a theoretical

https://doi.org/10.1017/S0890060409000146 Published online by Cambridge University Press

Table 4. *Taxonomy of the MOED physical-object domain*

Term	Definition
.Material	"A tangible substance that goes into the makeup of a &%TechnicalProducts." (subclass Material *Substance*)
.TechnicalProduct	"A &%Product that in &%ProductUsing &%ProductLifeCyclePhase &%realizes necessary effects that &%satisfies the user requirements." (subclass TechnicalProduct *Product*)
..EngineeringComponent	"A &%Product that is one of the individual parts with specific task in realization of the &%TechnicalFunction, and of which an &%EngineeringAssembly is made up." (subclass EngineeringComponent TechnicalProduct)
..EngineeringAssembly	"A group of &%EngineeringComponents that fit together to form a self-contained structural and functional unit of &%TechnicalProduct." (subclass EngineeringAssembly TechnicalProduct)
..FormFeature	"An individual &%partOf an &%EngineeringComponent's &%Form." (subclass FormFeature TechnicalProduct)
.TechnicalProductFamily	"A &%Collection of different variants of the same kind of &%TechnicalProduct." (subclass TechnicalProductFamily *Collection*)
.TechnicalDocument	"A &%Object that contains information about &%TechnicalProduct or &%ProductLifeCycleProcess." (subclass TechnicalDocument *ContentBearingObject*)
.HumanAgent	"Someone that could take the role of an &%operand or &%operator in the different &%ProductLifeCyclePhases of the &%ProductLifeCycleProcess." (subclass HumanAgent *Agent*)

Table 5. *Taxonomy of the MOED physical-process domain*

Term	Definition
.ProductLifeCycleProcess	"A series of &%ProductLifeCyclePhases through which an &%TechnicalProduct passes during life." (subclass ProductLifeCycleProcess *Process*)
.ProductLifeCyclePhase	"An individual &%IntentionalProcess that is &%partOf a &%ProductLifeCycleProcess." (subclass ProductLifeCyclePhase *IntentionalProcess*)
..ProductPlanning	"An &%IntentionalProcess of drawing up the &%DesignIssues and &%Plans for development of a &%TechnicalProduct." (subclass ProductPlanning ProductLifeCyclePhase)
..ProductDesigning	"An &%IntentionalProcess of working out the &%TechnicalProductCharacteristics based on the required &%TechnicalFunction, and by solving &%DesignIssues resulting with the full description of a &%TechnicalProduct in &%TechnicalDocumentations." (subclass ProductPlanning ProductLifeCyclePhase)
..ProductManufacturing	"An &%IntentionalProcess of making &%EngineeringComponents from raw &%Material and assembling them together into &%EngineeringAssemblys of &%TechnicalProduct." (subclass ProductPlanning ProductLifeCyclePhase)
..ProductDistributing	"An &%IntentionalProcess of transporting, selling and installing &%TechnicalProduct from a producer to a customer." (subclass ProductDistibuting ProductLifeCyclePhase)
..ProductUsing	"An &%IntentionalProcess of putting &%TechnicalProduct into service and make it work for a particular purpose of fulfilling its &%TechnicalFunction." (subclass ProductUsing ProductLifeCyclePhase)
..ProductDisposing	"An &%IntentionalProcess of processing used &%TechnicalProduct for use in creating new &%TechnicalProduct." (subclass ProductDisposing ProductLifeCyclePhase)
.ProductLifeCycleActivity	"Any procedure that is &%partOf execution of a &%ProductLifeCyclePhase." (subclass ProductLifeCyclePhase *IntentionalProcess*)
.ProductLifeCycleTask	"A specific piece of work required to be done as a &%partOf a &%ProductLifeCycleActivity." (subclass ProductLifeCycleTask *IntentionalProcess*)
.TechnicalFlow	"The continuous flow of &%Entities in &%ProductLifeCyclePhase." (subclass TechnicalFlow Process)
..MaterialFlow	"The continuous flow of &%Material in &%ProductLifeCyclePhase." (subclass MaterialFlow TechnicalFlow)
..EnergyFlow	"The continuous flow of &%Energy in &%ProductLifeCyclePhase process." (subclass EnergyFlow TechnicalFlow)
..SignalFlow	"The continuous flow of &%Signal in &%ProductLifeCyclePhase process." (subclass SignalFlow TechnicalFlow)

https://doi.org/10.1017/S0890060409000146 Published online by Cambridge University Press

Table 6. *Taxonomy of the MOED abstract-attribute domain*

Term	Definition
.TechnicalFunction	"A what &%TechnicalProduct is manufactured and used for." (subclass TechnicalFunction *Attribute*)
..BranchFunction	"A &%TechnicalFunction of separation and distribution of a &%TechnicalFlow." (subclass BranchFunction TechnicalFunction)
..ChannelFunction	"A &%TechnicalFunction of transferring and guiding of a &%TechnicalFlow." (subclass ChannelFunctionFunction TechnicalFunction)
..ConnectFunction	"A &%TechnicalFunction of coupling and mixing of a &%TechnicalFlow." (subclass ConnectFunctionFunction TechnicalFunction)
..ControlFunction	"A &%TechnicalFunction of regulation and changing of a &%TechnicalFlow." (subclass ControlFunction TechnicalFunction)
..ConvertFunction	"A &%TechnicalFunction of converting of a &%TechnicalFlow." (subclass ConvertFunction TechnicalFunction)
..ProvisionFunction	"A &%TechnicalFunction of storing and supplying of a &%TechnicalFlow." (subclass ProvisionFunction TechnicalFunction)
..SignalFunction	"A &%TechnicalFunction of indicating and sensing of a &%TechnicalFlow." (subclass SignalFunction TechnicalFunction)
..SupportFunction	"A &%TechnicalFunction of stabilization and position of a &%TechnicalFlow." (subclass SupportFunction TechnicalFunction)
.TechnicalProductCharacteristic	"Any &%Attribute that is an internal characteristic of a &%TechnicalProduct." (subclass TechnicalProductCharacteristic *InternalAtrribute*)
..Form	"The spatial characteristic of &%TechnicalProduct defined by its surface area." (subclass Form TechnicalProductCharacteristic)
..Dimension	"The magnitude of &%TechnicalProduct in a particular direction." (subclass Dimension TechnicalProductCharacteristic)
..Tolerance	"A permissible difference of nominal &%Dimension of &%TechnicalProduct." (subclass Tolerance TechnicalProductCharacteristic)
..ManufacturingMethod	"A particular method applied in fabricating and assembling &%EngineeringComponent or &%EngineeringAssembly." (subclass ManufacturingMethod TechnicalProductCharacteristic)
..SurfaceTexture	"Totality of the microgeometrical incorrectness of an &%EngineeringComponent's surface." (subclass SurfaceTexture TechnicalProductCharacteristic)
..StructuralCharacteristic	"A manner of &%ProductDesigning of &%TechnicalProduct and the arrangement of its parts." (subclass StructuralCharacteristic TechnicalProductCharacteristic)
..SpatialCharacteristic	"A characteristic resulting from the arrangement of &%TechnicalProduct's parts in relation to each other and to the whole." (subclass SpatialCharacteristic TechnicalProductCharacteristic)
.DesignIssue	"Any &%Attribute that a &%TechnicalProduct has by virtue of a relationship that it bears to another &%Entity." (subclass DesignIssue *RelationalAttribute*)
..FunctionalRequirement	"Required behavior of &%TechnicalProduct under specified conditions." (subclass FunctionalRequirement DesignIssue)
..LifeCycleSystemRequirement	"Attribute of &%TechnicalProduct required by different life cycle systems." (subclass LifeCycleSystemRequirement DesignIssue)
..EnvironmentalRequirement	"Attribute of &%TechnicalProduct required by totality of surrounding conditions of its physical environment during &%ProductLifeCycleProcess." (subclass EnvironmentalRequirement DesignIssue)

viewpoint; hence, the physical product would be embedded low down in the classes.

The engineering design concepts identified have been classified depending on their nature with the purpose of resolving the complexity and uncertainty related to the engineering design process. By describing the relationships between these concepts for each category it is intended that the meaning of each concept has been clearly and consistently defined and the relationships identified. Hence, it is reasonable to suggest that MOED presented here displays ontological coherence. Further work envisaged will focus on extending and evaluating MOED for domain-specific tasks as well as for the specific applications.

6. CONCLUSION

The comparison of the two separate ontologies, DO, with a theoretical foundation, and EDIT, with an empirical foundation, has been undertaken. The process of comparing the ontologies required a deep understanding of the concepts, classes, and relations contained within both ontologies. The comparison process of both approaches enabled the researchers to

Table 7. *Taxonomy of the MOED abstract-relation domain*

Term	Definition
.CompositionalRelation	"A relation that capture semantics of part/whole concept between two &%Entities." (instance CompositionalRelation *InheritableRelation*) (instance CompositionalRelation *PartialOrderingRelation*) (subclass CompositionalRelation *Relation*)
..partOf (Physical, Physical)	"A relation that capture semantics of part/whole concept between two &%Physical entities." (subclass partOf CompositionalRelation)
..elementOf (Abstract, Abstract)	"A relation that capture semantics of part/whole concept between two &%Abstract entities." (subclass elementOf CompositionalRelation)
TopologicalRelation	"A relation that capture semantics of the spatial arrangement between two &%Objects." (instance TopologicalRelation *InheritableRelation*) (instance TopologicalRelation *AsymmetricRelation*) (subclass TopologicalRelation *Relation*)
..contacts (Object, Object)	"A relation that capture semantics of the physical contact between two &%Objects." (subclass contacts TopologicalRelation)
..contains (Object, Object)	"A relation that capture semantics of the fact that an &%Object has a space which is at least partially filled by other &%Object." (subclass contains TopologicalRelation)
.LifeCycleProcessRoleRelation	"A relation that capture semantics of the distinguished roles of the different &%Objects in the &%ProductLifeCycleProcess." (subclass LifeCycleProcessRoleRelation *CaseRole*)
..operandOf (Object, Process)	"A relation that capture semantics of the operand roles of the different &%Objects in the &%ProductLifeCycleProcess." (subclass operandOf LifeCycleProcessRoleRelation)
..operatorOf (Object, Process)	"A relation that capture semantics of the operator roles of the different &%Objects in the &%ProductLifeCycleProcess." (subclass operatorOf LifeCycleProcessRoleRelation)
..resourceOf (Object, Process)	"A relation that capture semantics of the resource roles of the different &%Objects in the &%ProductLifeCycleProcess." (subclass resourceOf LifeCycleProcessRoleRelation)
..instrumentOf (Object, Process)	"A relation that capture semantics of the instrument roles of the different &%Objects in the &%ProductLifeCycleProcess." (subclass instrumentOf LifeCycleProcessRoleRelation)
.DependencyRelation	"A relation that capture semantics of the fact that an &%Entity depends existentially on another &%Entity." (instance DependencyRelation *InheritableRelation*) (instance DependencyRelation *AsymmetricRelation*) (instance DependencyRelation *TransitiveRelation*) (subclass DependencyRelation *Relation*)
..isPurpose (Attribute, Attribute)	"A relation that capture semantics of the fact that an &%Attribute is purpose of another &%Attribute." (subclass isPurpose DependencyRelation)
..dependsOn (Attribute, Object)	"A relation that capture semantics of the fact that an &%Attribute depends on &%Object." (subclass dependsOn DependencyRelation)
..isConsequence (Attribute, Process)	"A relation that capture semantics of the fact that an &%Attribute is consequence of &%Process." (subclass isConsequence DependencyRelation)
.InfluenceRelation	"A relation that capture semantics of the fact that an &%Entity somehow influences another &%Entity." (instance InfluenceRelation *InheritableRelation*) (instance InfluenceRelation *AsymmetricRelation*) (instance InfluenceRelation *TransitiveRelation*) (subclass InfluenceRelation *Relation*)
..influences (Process, Attribute)	"A relation that capture semantics of the fact that a &%Process influences an &%Attribute." (subclass influences InfluenceRelation)
..constrains (Attribute, Attribute)	"A relation that capture semantics of the fact that an &%Attribute constrains another &%Attribute." (subclass constraints InfluenceRelation)
.TemporalRelation (Process, Process)	"A relation that includes notions of (temporal) topology of intervals between two &%Processes." (instance TemporalRelation *InheritableRelation*) (subclass TemporalRelation *Relation*)
.GeneralRelation	"A relation that capture semantics of the most general relations between two &%Entities." (instance GeneralRelation *InheritableRelation*) (instance GeneralRelation *AsymmetricRelation*) (subclass GeneralRelation *Relation*)
..describes (Attribute, Physical)	"A relation that capture semantics of the fact that an &%Attribute describes &%Physical entity." (subclass describes GeneralRelation)
..realizes (Process, Object)	"A relation that capture semantics of the fact that a &%Process realizes &%Object." (subclass realizes GeneralRelation)

Table 7 (*cont.*)

Term	Definition
..satisfies (Object, Attribute)	"A relation that capture semantics of the fact that an &%Object satisfies &%Attribute." (subclass satisfies GeneralRelation)
..appliesTo (TechnicalFunction, TechnicalFlow)	"A relation that capture semantics of the fact that a &%TechnicalFunction applies to &%TechnicalFlow." (subclass appliesTo GeneralRelation)

gain a deep understanding of an alternative research approach to their own, and to validate the two ontologies. The findings are promising in that despite the different approaches employed, the vast majority of concepts and classes were common to both ontologies. All of the top levels contained in EDIT could be found in DO; however, the taxonomies (e.g., function, and issues) may be fragmented and placed in different locations. That is some of the concepts could be found in more than one place, for example, nouns that are physical, but when used in combination with verbs to describe a function become an abstract concept. The comparison process resulted in an evaluation of both ontologies, with a proposal of changes resulting from this.

It was found that it is difficult to set the boundaries of a theoretical ontology; by confronting these with an applied ontology EDIT some of the boundaries became apparent. Without testing a theoretical ontology it is difficult to assess the validity, in terms of usefulness for the particular application. Similarly, an ontology that is based empirically with a particular purpose in mind, such as EDIT, which is primarily focused on indexing of engineering design knowledge, may be presented from the viewpoint of the user in that particular application; hence, concepts may be placed differently from how they would placed in ontologies based on theory.

These conclusions points to the need for tailoring general domain ontologies for a specific application. These conclusions are based upon the ontology studied within the engineering design domain; however, it is not clear that these are specific to the engineering design domain. The approach adopted in the presented research was to create MOED as a template, which is theoretically consistent, but aligned with practitioners' view on the engineering process and product description. It is expected that MOED can contribute to the development of effective and efficient engineering knowledge management methods and tools in the future.

ACKNOWLEDGMENTS

This work was partially funded by the National Foundation for Science, Higher Education and Technological Development of the Republic of Croatia. The materials, ideas, and results reflect the views of the authors and do not necessary reflect the views of the National Foundation for Science. The authors acknowledge the support of EPSRC, Rolls–Royce, and BAE Systems through the UTP for Design for the research leading to the development of EDIT. In particular, we thank Dr. Michael Moss and all of the participating designers for giving their time.

REFERENCES

Ahmed, S. (2005). Encouraging reuse of design knowledge: a method to index knowledge. *Design Studies 26(6)*, 565–592.

Ahmed, S. (2006*a*). Prompting designers to design. *Proc. Design 2006*, Dubrovnik, Croatia.

Ahmed, S. (2006*b*). An approach to assist designers with their queries and design. *Proc. ASME 2006 Design, Theory and Methodology Conf.*, Pittsburgh, PA.

Ahmed, S., Kim, S., & Wallace, K.M. (2006). A methodology for creating ontologies in engineering design. *ASME Computing in Information Engineering 7(2)*, 132–140.

Ahmed, S., Kim, S., & Wallace, K.M. (2007). A methodology for creating ontologies in engineering design. *Transactions of the ASME Journal of Computing in Information Science in Engineering 7*, 132–140.

Alberts, L.K., & Dikker, F. (1994). Integrating standards and synthesis knowledge using the YMIR ontology. *Proc. Artificial Intelligence in Design '94* (Gero, J.S., Ed.). Boston: Kluwer Academic.

Anderl, R., Grabowski, H., & Pratt, M. (1991). *Advanced Modelling for CAD/CAM Systems*, Research Report ESPRIT Project 322, Vol. 7. New York: Springer–Verlag.

Andreasen, M.M. (1980). *Machine design methods based on systematic approach—contribution to the design theory.* PhD Dissertation. Lind Institute of Technology, Sweden.

Bakemann, R., & Gottam, J.M. (1997). *Observing Interaction: An Introduction to Sequential Analysis*. Cambridge: Cambridge University Press.

Beghtol, C. (1986). Semantic validity: concepts of warrant in bibliographic classification systems. *Library Resources & Technical Services 30(2)*, 109–125.

Blessing, L. (1994). *A process based approach to computer supported engineering design.* PhD Dissertation. University of Twente.

Brewster, C., & O'Hara, K. (2007). Knowledge representation with ontologies: present challenges—future possibilities. *International Journal of Human–Computer Studies 65(7)*, 563–568.

Darlington, M., & Culley, S. (2008). Investigating ontology development for engineering design support. *Advanced Engineering Informatics 22(1)*, 112–134.

Davis, R., Shrobe, H., & Szolovits, P. (1993). What is a knowledge representation. *AI Magazine 14(1)*, 17–33.

Fernandez-Breis, J., & Martinez-Bejar, R. (2002). A cooperative framework for integrating ontologies. *International Journal of Human–Computer Studies 5*, 665–720.

Gruber, T. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition 5(2)*, 199–220.

Gunendran, G., Cutting-Decelle, A.F., Young, R.I.M., & Bourey, J.P. (2007). Information organisation in design: an application of formal ontologies. *Proc. 16th Int. Conf. Engineering Design, ICED '07*, Paris.

Hirtz, J.M., Stone, R.B., Szykman, S., McAdams, D.A., & Wood, K.L. (2001). Evolving a functional basis for engineering design. *Proc. ASME Design Engineering Technical Conf. DETC 2001*, Report No. DTM-21688, Pittsburgh, PA.

Hjorland, B. (2007). Lifeboat for knowledge organization. Accessed at http://www.db.dk

Hubka, V. (1976). *Theorie der Konstruktionsprozesse*. Berlin: Springer–Verlag.

Hubka, V., & Eder, W.E. (1988). *Theory of Technical Systems—A Total Concept Theory for Engineering Design*. Berlin: Springer–Verlag.

ISO. (1997). *ISO 10303: External Representation of Product Data Model*, ISO Report TC184/SC4. Geneva: ISO.

Kalfoglou, Y., & Schorlemmer, M. (2003). Ontology mapping: the state of the art. *Knowledge Engineering Review 18(1)*, 1–31.

Kitamura, Y., & Mitzoguchi, R. (2004). Ontology-based systematization of functional knowledge. *Journal of Engineering Design 15(4)*, 327–352.

Krause, F.L. (1988). Knowledge integrated product modelling for design and manufacture. *Proc. 2nd Toyota Conf.*, Aichi, Japan.

Kumar, P., & Mocko, G. (2007). Modeling and analysis of an ontology of engineering design activities using the design structure matrix DETC2007-35634. *Proc. ASME Int. Design Technical Conf. and Computers and Information in Engineering Conf.*, Las Vegas, NV.

Mehkilef, M., Bourey, J.P., & Bigand, M. (2003). An UML modelling of an architecture for knowledge documentation. *Proc. 14th Int. Conf. Engineering Design, ICED '03*, Stockholm, Sweden.

Merkamm, H. (1995). Product modelling: a prerequisite for effective product development. *Proc. Produktmodeller95*, Linkoping University of Technology.

Mortensen, N.H. (1999). *Design modelling in a designer's workbench—contribution to a design language*. PhD Thesis. Delft University of Technology.

Nanda, J., Simpson, T.W., Kumara, S.R.T., & 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)*, 103–113.

Niles, I., & Pease, A. (2001). Origins of the IEEE standard upper ontology. *Working Notes of the IJCAI-2001 Workshop on the IEEE Standard Upper Ontology*, pp. 37–42, Seattle, WA.

National Information Standards Organisation. (1994). *ANSI/NISO Z39.19-1993: Guidelines for the Construction, Format and Management of Monolingual Thesauri*. Bethesda, MD: NISO Press.

Pahl, G., & Beitz, W. (1988). *Engineering Design*. London: Design Council.

Olsen, J. (1992). *Concurrent development in manufaturing—based on dispositional mechanisms.* PhD Thesis. University of Denmark, Department of Control and Engineering Design.

Rezgui, Y. (2006). Ontology-centered knowledge management using information retrieval techniques. *Journal of Computing in Civil Engineering 20(44)*, 261–269.

Rosenman, M.A., & Gero, J.S. (1998). Purpose and function in design: from the socio-cultural and the techno-physical. *Design Studies 19(2)*, 161–186.

Salminen, V., & Verho, A. (1991). Experiences of using meta modelling in systematic design of mechatronics. *Proc. ICED '91*. Zurich: Heurista.

Sim, S.K., & Duffy, A.H.B. (2003). Towards an ontology of generic engineering design activities. *Research in Engineering Design 14(4)*, 200–223.

Štorga, M., Andreasen, M.M., & Marjanović, D. (2005). Towards a formal design model based on a genetic design model system. *Proc. 15th Int. Conf. Engineering Design ICED '05* (Samuel, A., & Lewis, W., Eds.). Melbourne: Engineers Institution of Engineers, Australia, II National Circuit.

Štorga, M., Andreasen, M.M., & Marjanović, D. (in press). The design ontology—contribution to the design knowledge exchange and management. *Journal of Engineering Design*.

Štorga, M., Marjanović, D., & Andreasen, M.M. (2007). Relationships between the concepts in the design ontology. *Proc. 16th Int. Conf. Engineering Design, ICED '07*, pp. 531–532, Ecole Centrale Paris and The Design Society, Paris.

Tomiyama, T., Kiriyama, T., Takeda, H., Xue, D., & Yoshikawa, H. (1989). Metamodel: a key to intelligent CAD systems. *Research in Engineering Design 1(1)*, 19–34.

Uschold, M., & Gruninger, M. (1996). Ontologies: principles, methods and applications. *Knowledge Engineering Review 11(2)*, 93–136.

Wand, Y., & Weber, R. (1993). On the ontological expressiveness of information systems analysis and design grammars. *Journal of Information Systems 3*, 217–237.

Saeema Ahmed is an Associate Professor in the Department of Management Engineering at the Technical University of Denmark. She achieved her PhD at the Engineering Design Centre Cambridge, where she was also a Research Associate and a Fellow of New Hall. Dr. Ahmed has published over 40 journal and conference papers and book contributions and has served on the scientific boards of three journals and the scientific committees of numerous international conferences. Her research interests include design thinking, management of design knowledge and information, ontology, and product and process modeling.

Mario Štorga is an Associate Professor in the Department of Design in the Faculty of Mechanical Engineering and Naval Architecture at the University of Zagreb, Croatia. He has been a Visiting Researcher at the Technical University of Denmark, Ecole Centrale de Paris, and University of Bath UK. He received his PhD at the University of Zagreb. Dr. Štorga has published over 30 journal and conference papers and has served on the scientific boards of two journals and numerous international conferences. His research interests include product life-cycle management, product and process formal modeling, and design knowledge and information management.

https://doi.org/10.1017/S0890060409000146 Published online by Cambridge University Press