INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 2005 MELBOURNE, AUGUST 15 – 18, 2005

TOWARDS A FORMAL DESIGN MODEL BASED ON A GENETIC DESIGN MODEL SYSTEM

Mario Štorga, Mogens Myrup Andreasen, Dorian Marjanović

Abstract:

The research presented in this article is aimed to the investigation of the nature, building and practical role of a *Design Ontology* as a potential framework for the more efficient product development data-, information- and knowledge- description, -explanation, -understanding and -reusing. In this article, we briefly summarize our experience of converting informal definitions of the concepts from the product development domain based on the existing theoretical background into formal design model. As a main data source for extracting the content of a design ontology, Genetic Design Model System developed by N.H. Mortensen was chosen. In order to achieve the useful formalization of the GDMS structure we have followed modelling procedure on the four levels: epistemological, generic, domain and project level. As an epistemological foundation, The Suggested Upper Merged Ontology (SUMO) proposed by IEEE, provides us definitions for the most general and universal concepts, that we used in our research for derivation of terms definitions and axioms in the *Design Ontology*. The *Design Ontology* was evaluated using the OntoEdit® ontology engineering environment, on a real product example, and based on this evaluation further research steps are proposed.

Keywords: formal design model, design ontology, genetic design model system

1. Introduction

Today it is generally recognized that possessing and utilizing engineering knowledge is one of the enterprise's most important assets, decisively influencing its competitiveness. Large engineering projects involve the resources of many different clusters of cooperative subjects (human and computer) in the given situation. Each cluster makes its own contributions, and the overall success of the project depends in large measure on the degree of integration between those different clusters throughout the product development process. In addition to the dynamic and complex nature of the product development (PD) context (defined in [1] as the entire body of data, information and engineering knowledge related to design itself, that evolves throughout the product development efforts), an enormous problem in the coordination of large engineering projects is the diversity of backgrounds the various groups of engineers bring to their respective role. As a consequence, many engineers use apparently identical words with different meanings for describing situations in product development domain and utilize those descriptions in different ways. To avoid such a situation, we believe that it is necessary to define a unified vocabulary for articulating PD context instantiations in appropriate design situations that may lead to the formal design model. With this research motivation, in this article we investigate possibilities for creating a formal design language as a continuation of the previous research on the Genetic Design Model System (GDMS) [2] [3]. Based upon presented approach to the formalization of GDMS we illustrate the possibilities for creating more definite design models than is possible by using the natural languages.

2. Research aim and methodology

Any domain with a determinate subject matter has its own terminology, a distinctive vocabulary that is used when talking about characteristic concepts that compromise the domain. But the domain space is not revealed in its corresponding vocabulary alone. In order to form the logically correct statements about a situation in a domain, definition of the rules and restrictions governing the way terms in vocabulary can be combined, should be provided and clarified. Only when this additional information is available, it is possible to understand both the nature of the individual concepts that exist in the domain and the associations they bear to one another. Domain vocabulary together with set of precise definitions, or axioms, that constrain the meanings of the terms in that vocabulary sufficient to enable consistent interpretation of statements based on vocabulary, in literature is usually considered as *domain ontology* [4]. A related motivation for the researches on the domain ontology capture is the standardization of terminology in order to realize description, explanation, understanding and reusing of domain knowledge.

The ability to determine a product development domain vocabulary and its meaning in the context of use in this manner seems to be a critical task to achievement the true concurrent product development. A key to effective product development can be the accessibility to rich product development ontologies distinctive for different activities throughout the product development process. For instance, access to a manufacturing ontology that includes constraints on how a given machine part is manufactured can aid designers in their design of a complex product by giving them insight into the manufacturing implications of their concepts.

A mixed approach of existing methodologies for developing ontologies [5] together with review of the current and past research on product development related topics, have been aimed in research presented in this article to successful formalization of a *Design Ontology*. This *Design Ontology* was chosen as a start point in a research with a long term goal of defining a “general product development ontology”, because a design or a product as the result of the product development projects have been identified as common object of interest across the greater part of the product development activities.

2.1. Ontology definitions

The concept of ontologies generates a lot of controversy in discussions about it. Originally, the word ontology was taken from philosophy (metaphysic), where it means a systematic explanation of being, or the kinds of existence. In the last decade, the word ontology became a relevant word for the knowledge engineering community that has borrowed it from philosophy and has given its meaning a twist. For them the main question is not what the nature of being is, but what an artificial system has to reason about to be able to perform a useful task [6].

One of the first definitions in this new sense was given by Neches and colleagues [7] who defined ontology as follows: “Ontology defines the basic terms and relations comprising the vocabulary of a topic area as well as the rules for combining terms and relations to define extensions to the vocabulary”. This descriptive definition tells what to do in order to build an ontology, and gives us some vague guidelines: the definition identifies basic terms and relations between terms, identifies rules to combine terms, and provides the definitions of such terms and relations. In one of the newest and most used definition Studer and colleagues [8] explained ontology as follows: “Ontologies are defined as a formal specification of a shared conceptualization. Conceptualization refers to an abstract model of some phenomenon in the world by having identified the relevant concepts of that phenomenon. Explicit means that the type of concepts used, and the constraints on their use are explicitly defined. Formal refers to the fact that the ontology should be machine readable”.

Today, ontologies are widely used for different purposes (natural language processing, knowledge management, e-commerce, intelligent integration information, the semantic web, etc.) and are matter of research in different research communities (i.e., knowledge engineering, databases and software engineering). To popularize it in different disciplines Uschold and Jasper [9] provided a new definition of the word ontology: “Ontology may take a variety of forms, but it will necessarily include a vocabulary of terms and some specification of their meaning. This includes definitions and an indication of how concepts are inter-related which collectively impose a structure on the domain and constrain the possible interpretations of terms.”

It may seem that there is not much difference between ontology and a data dictionary. However, a data dictionary is typically just a compendium of terms together with definitions for the individual terms stated in natural language. By contrast, the grammar and axioms of an ontology are stated in a precise formal language with a very precise syntax and a clear formal semantics. Consequently, ontologies are, in general, far more rigorous and precise in their content that a typical data dictionary. The advantage of formal definitions is that they allow a machine to perform much deeper reasoning; the disadvantage is that these definitions are much more difficult to construct. Ontologies also tend to be more complete as well. Associations between concepts and real objects in a domain and constraints on and between domain objects are made explicit rather then left implicit, thus minimizing the risk of misunderstanding logical connections within the domain. As a main conclusion of this section, we can say that domain ontologies aim to capture consensual data, information and knowledge in a generic and formal way, so that it can be reused and shared across different applications (software) and by groups of people.

2.2 Ontology building process

Every ontology development process is focused on understanding the concepts of the particular domain from multiple perspectives. Researchers from varied field such AI, philosophy, data management, mathematics, engineering, and cognitive science study ontologies using the different foundations and methods. That is the reason why the building of ontology differs from traditional information capture activities in the depth and breadth of the information captured. Ontology building process is usually a discovery process and requires extensive iterations, discussions, reviews and introspection. It requires a process that incorporates both significant expert involvement as well as the dynamics of an ontology engineering group effort. The general ontology building process consists of the following activities [4]:

- **Organizing and scoping.** This activity involves establishing the purpose, viewpoint and boundary for the ontology development project.
- **Data collection.** This activity involves acquiring the raw data needed for ontology development. Main data sources are the domain experts’ publications (scientific articles, thesis, reports, and industrial papers) relevant to the circumscribed ontology.
- **Data analysis.** This activity involves analyzing the data to facilitate an ontology extraction, by following activities: list the concepts of interest in the domain, identify concepts that are on the boundaries of the ontology, look for and individuate internal systems within the boundary of the description.

- **Initial ontology development.** This ontology involves developing a preliminary ontology from acquired data. Initial set refers tentative terms, attributes and relations that are subject to further inquiry before final change of status.
- **Ontology refinement and validation.** This activity involves refining and validating the ontology to complete development process. The ontology structures are instantiated with actual data, and the result is compared with the ontology structure. Refinements to the initial ontology are incorporated to obtain a validated ontology.

In the research presented in this paper we used previous described steps as general guidelines that were modified accordingly to the size of our research project and involved research group. The particular research steps are presented in following chapters.

3. The *Design Ontology* foundations

After defining the objectives of the research project [5], as a main data source and foundation for extracting the content of the *Design Ontology*, the Genetic Design Model System (GDMS) developed by Mortensen [2] was chosen. The main reason for such decision was that accordingly to research results, GDMS seems to be able to capture the totality of results created in product development projects, and it is a more comprehensive comparing to the other design/product model systems that can be found in literature. GDMS is able to receive and maintain the results from engineering design, to handle design genesis, to handle design rationale, to be reused, and to handle multiple views. The results of the GDMS research project are in literature presented as proposal of the genetic design language contemplated as the set of the infinite designs which are synthesized, based on a design vocabulary and syntactical rules [2]. Besides, the next step proposed for the future research is defined as formalization. The principal contents of GDMS have in literature been described by three domain languages [2], [3]: function-, organ-, and part language. Each of the languages points out the concepts of different types which can be utilized for creating the formal design models. In order to face semantic variety of the possible interactions between the different used terms in all three domains, and to guarantee the integrity and a certain robustness of the complete formal design model, it is necessary to formalize a general structure of GDMS. Mekhilef in his work [10] proposes four levels of modelling procedures that we have used (more or less) as a guideline in order to achieve the useful formalization of the GDMS structure: epistemological-, generic-, domain-, and project modelling level.

3.1 Epistemological modelling level

The epistemological modelling level in general is established by defining the general set of entities and possible associations between them in order to logically correct describe the situation in a domain of discourse on the highest level of abstraction. Common sense knowledge about a domain (knowledge typical of the general population) is usually an important aspect needed for establishing this level [10]. In addition, the engineering domains require a perspective that is more structured, more based on scientifically acceptable views of reality, and less tolerant on contradiction and inconsistency, compared to a common sense. These requirements motivated us to use the Standard Upper Merged Ontology (SUMO) [11] as an epistemological foundation in developing the *Design Ontology*. SUMO is an effort by IEEE (*www.ieee.org*) collaborators from the field of engineering, philosophy and information science, aimed to creation of the framework by which disparate participants may utilize a common knowledge and from which more domain-specific ontologies (e.g. design, manufacturing, etc.) may be derived. The SUMO is intended to express and provide definitions for the most basic and universal concepts that are generic, abstract and philosophical, and therefore are general enough to address (at a high level) a broad range of different domain areas. Today, SUMO is a collection of big number well-defined and well-documented terms, interconnected into semantic network and accompanied by a number of axioms (*www.ontologyportal.org*).

Terms in SUMO are organized into a single hierarchy [Figure 1] rooted at *Entity*, representing the most general concept used for a definite descriptor that refers to: (i) all physically existent things, (ii) and all abstract, mentally represented things in the real word. At the top level of the SUMO hierarchy, the concept of *Entity* subsumes concepts of *Physical* and *Abstract,* where former category includes everything that has a position in space/time, and the latter category includes everything else. Under the concept of *Physical* the disjoint concepts of *Object* and *Process* are defined. The concept of *Object* is the most general concept that exists in space. The concept of *Process* corresponds to any sustained phenomenon or one marked by gradual changes (time-space). Returning to the highest level distinction in SUMO hierarchy, the concept of *Abstract* subsumes four disjoint concepts: *Attribute Proposition, Quantity*, and *Relation*. The concept of *Attribute* includes all qualities, properties, etc. 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 associate two or more concepts.

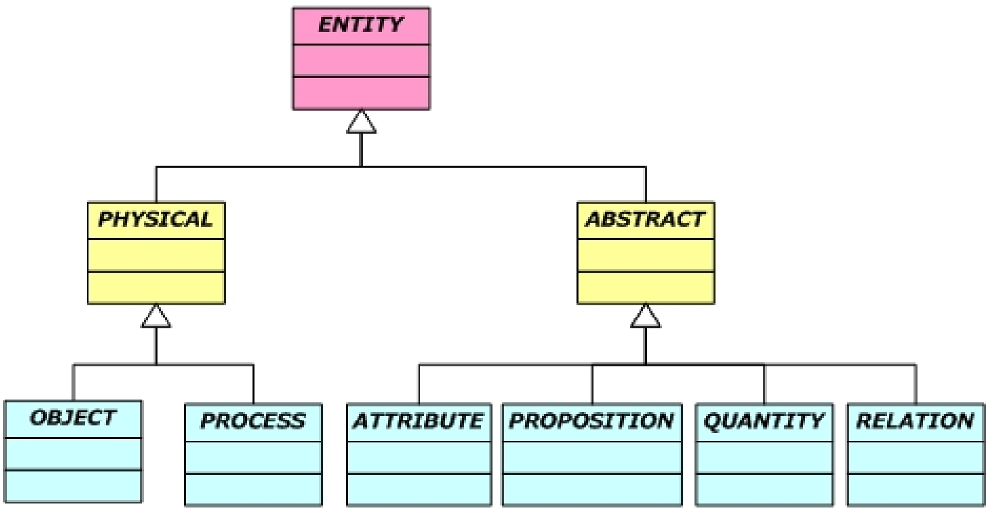

Figure 1. Top level SUMO concepts

In order to formally define concepts expressed in SUMO, the meaning of every particular SUMO term requires careful understanding of its associations to the other terms defined in SUMO. Definitions of all terms in SUMO are formalized in the form of axioms with the purpose to constrain interpretation of terms, and to provide guidelines for automated reasoning systems. An example of such an axiom is:

> "?*PHYSICAL* is an *Instance* of *Physical* if and only if there *Exists* ?*LOCATION*, ?*TIME* so that ?*PHYSICAL* is *Located* at ?*LOCATION* and ?*PHYSICAL Exists* during ?*TIME*".

The axiom coded in formal logical language looks like follows:

(<=>
(*Instance* ?*PHYSICAL Physical*)

(*Exists*
    (?*LOCATION* ?*TIME*)
    (And
        (*Located* ?*PHYSICAL* ?*LOCATION*)
        (*Time* ?*PHYSICAL* ?*TIME*))))

There are some distinct advantages of SUMO. First, the SUMO is the working effort sponsored by open-source community. This means that potentially users of upper ontology can be more confident that this upper ontology will eventually be embraced by a large class of users. Second, the SUMO was constructed with reference to very pragmatic principles. Any distinctions of strictly philosophical interest have been removed from proposed upper ontology. And third, SUMO is the only formal upper ontology that has been mapped to the entire WordNet® lexicon (*wordnet.princeton.edu*). That mapping provides a link between formal content expressed in SUMO and natural language, paraphrasing on such way the hard-to-read logical inscription of axioms into natural language. All together makes SUMO easy and simple to use as the epistemological foundation for building specific domain ontologies.

3.2 Generic modelling level

The generic modelling level is established by set of formal informational structures that describe a situation in particular domain. This level should be generic in a given domain (product development domain in this case) and constrained by the content of the epistemological foundation (SUMO in this case). According to the results of previous GDMS related research [2] knowledge about the product as the result of the development process is centred around three different conceptual viewpoints: design, life phase systems, and meetings. At this stage of research we applied many competency questions on those tree viewpoints to find out more about reasoning, synthesis, selection, documentation, business aspects, organizational responsibilities, etc. related to this three viewpoints [Figure 2]. In questioning we followed the basic idea, that the physical product cannot be designed without articulating the designing and fit to product life aspects. That procedure provided us a base for the extraction of the main *Design Ontology* terms and associations between them.

The basic terms were defined first and based on them the related terms were defined as deep as possible. At this point of research many terms were discarded and duplicates were removed. The terms have been grouped based on SUMO top level concepts [Figure 1.], so that terms closely related by nature to each other appear close together. As the result of this stage, the initial *Design Ontology* have provided the intended semantics of the vocabulary and laid the foundations for the specification of terms' definitions in formal language. The initial *Design Ontology* is in more details presented in chapter 4 of this article.

3.3 Domain and project modelling levels

The domain modelling level should be an reuse and extension of a generic level and should be specific for an application domain (for example configuration design, sustainable design, design for assembling etc.). Terms at this level are organized in a way characterizing specialization of common features specific for implementation of a generic model in particular use case. In the next step of the research, our plan is to extend the proposed *Design Ontology* with terms and axioms needed for achievement of the traceability among the PD context during a specific design episode in product development [12].

The project level in addition extends an application domain modelling level to include information about additional relevant concepts found within specific real implementation project, depending on the situation and requirements of the concrete product development scenario. These additional concepts could arise for instance from the specific synonyms for the general defined terms that are used in particular company, customer specification of design policies, company internal procedures (organizational, safety and confidential tasks, quality standards), company procedures related to implemented PDM or ERP systems, etc.

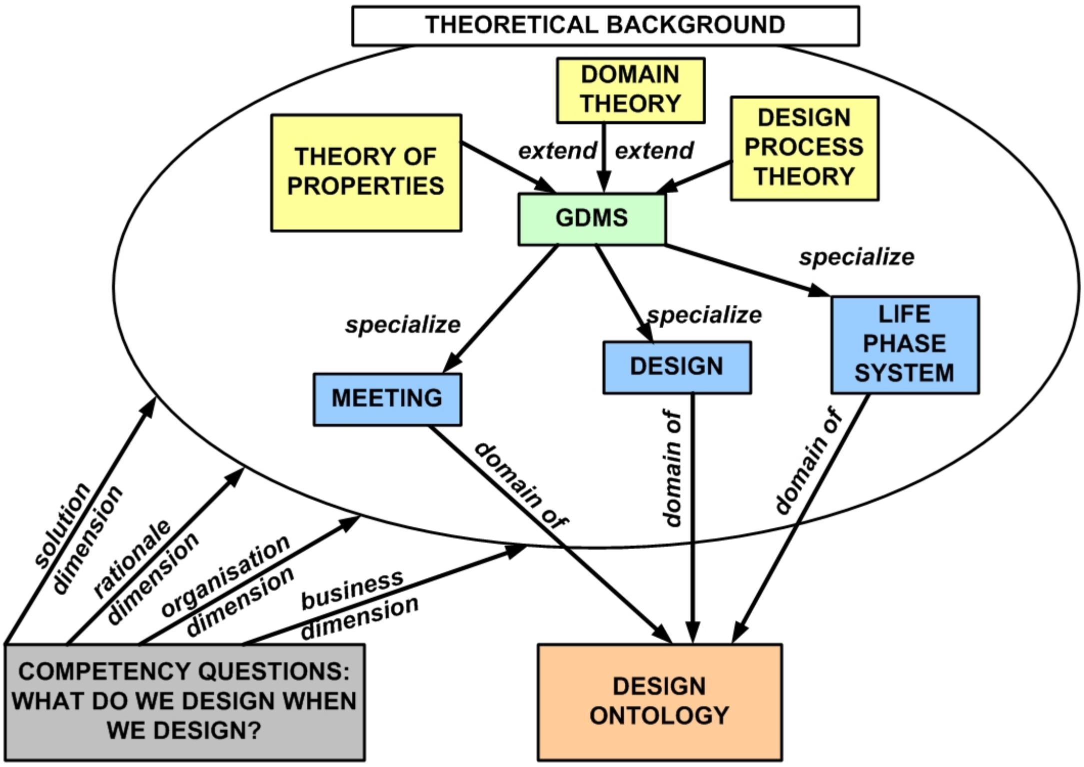

Figure 2. The *Design Ontology* content theoretical background

4. Initial *Design Ontology*

The first proposal for the *Design Ontology*, after extracting and analyzing the core concepts had an informal form, consisting of terms and definitions expressed in natural language. All the extracted terms were classified and their definitions were derived accordingly to the SUMO structure. The terms have been chosen as far as possible to match the natural use of English language. As the result of the previously described analyze, a core of about more then 150 different terms of wide variety was extracted and definitions in natural language were provided. For space reasons we can only show a few examples of definitions proposed as a part of initial *Design Ontology*:

> ***Device*** - a *Physical Entity* that has *Purpose* to serve as an *Instrument* in a specific type of *Task*. It is subclass of the *Artefact*…
>
> ***Engineering_Component*** - a *Physical Entity* that is *Constituent* of a *Device* and is a physically whole *Object* (such as one might see listed as standard parts in a catalogue). An *Engineering_Component* is not an arbitrary sub region, but a *Part of* a *System* with a stable *Identity*. It is subclass of the *Device*…

Structure – an *Abstract Entity* in the manner of *Construction* of something and the *Disposition* of its *Constituents*; the complex *Composition* of *Elements* and *Relations*. It is the subclass of *Internal Attribute*…

Operand - (*Operand* ?*Process* ?*Object*); an *Abstract Entity* expressing the *Object* of *Transformation* in a *Process* that may be changed, experienced, etc. The *Operand* of a *Process* may or may not undergo structural change as a *Result* of the *Process*. It is a subclass of *Case Role* relation…

Operator - (*Operator* ?*Process* ?*Agent*); an *Abstract Entity* expressing the *Agent* that is active determinant of the *Process*, with or without its voluntary intention. It is a subclass of *Case Role* relation…

It should be noted here that was really hard task to give generally definitions for all concepts especially for the ones that can be in the same time considered as abstract and as physical (as a *Design* for example that can be consider as a specification, something planned to be realized in a future time, or something which is composed and represented in drawings, computer models, etc. and has an intention/is intended to satisfy or solve a problem). Even that is almost unacceptable from the viewpoint of rigorous defined terms definitions, it was necessary to support multiple inheritance property for the some concepts to capture the nature of the human engineered systems. It the following sections the overview of domain terms classification into the main groups accordingly to the SUMO foundation is presented.

4.1 Processes

By its nature, *Processes* typically involve two sorts of change: change in kind and change in state. In a combustion process, for example, there is a transformation of some quantity of fuel into kinetic energy and exhaust gas; the fuel is destroyed and quantity of kinetic energy and exhausted gas result. By contrast, a process in which ice is melted simply involves a change state of a given quantity of water from frozen to liquid; the water itself is not destroyed, but only altered. *Processes* also represent a sustained phenomenon or one marked the way on which things are gradually changed. Besides, *Processes* can have a specific purpose for the *Agent* who performs it. For better understanding of product development, processes of especial interest are *Intentional processes* like planning, classifying, learning, reasoning, selecting, comparing, predicting etc. The definition of *Process* accordingly to SUMO is: the class of *Things* that happen and have temporal parts or *Stages*; or more generic: anything that lasts for *Time* but is not an *Object*. The *Process* is whole of the participants 'inside' it which have *Case Roles* in a *Process,* and a space/time dimensions. The main terms extracted from GDMS and classified as a *Process* are (definitions of terms are not provided here because of limited space and are planed to be published as an extended article):

Action process, Activity, Changing, Creating, Decision making, Designing, Flow, Life cycle, Making, Meeting, Development, Operation, Project, Reaction, Technical process, Transformation

4.2. Objects

The concept of *Object* corresponds roughly to the class of ordinary objects. Objects can be classified further as: *Agent* that can act on its own and produce changes in *Environment;*, *Corpuscular Objects* whose parts have *Properties* that are not shared as a whole; *Content*

Bearing Objects that expresses *Information*; and *Collection* which members has position in space-time and can be added or subtracted without thereby changing the identity of *Object*. Accordingly to SUMO, definition of *Object* is: something whose spatiotemporal extent is thought of as dividing into spatial parts roughly parallel to the time axis. The main terms extracted from GDMS and classified as a *Object* are:

AGENT: Human, Person

CORPUSCULAR OBJECT: Artifact, Assembly, Device, Engineering Component, Engineering Connection, Equipment, Feature, Interface, Machine, Material, Matter, Mechanism, Organ, Product, Surface, Skeleton, Technical plant, Transformational organism

CONTENT BEARING OBJECT: Description, Document, Signal, Specification, Statement, Symbol

COLLECTION: Assortment, Family, Organization, Set

4.3 Attributes and Design Attributes

The definition of *Attribute* accordingly to SUMO is: the qualities which we cannot or choose not to regard into subclasses of *Object*. To harmonize the background theory and upper ontology proposal, the subclass of a *Design Attribute* was specialised. The *Design attributes* can be further classified following the background theories as: *Internal (Design characteristics)* and *External (Design properties)*. *Internal Attribute* is any *Design* a*ttribute* that describes constitution of the *Design*, e.g. its shape, dimension, surface, structure etc. *External Attribute* is any *Attribute* that a *Design* has by virtue of *Internal Attributes* and influence from *Environment*. Some of the *External Attributes* are relational (describe behaviour of the meetings between a design and life phase system), and other are inherent (describe behaviours of a design in a certain environment). The main terms extracted from GDMS and classified as *Design* a*ttribute* are:

INTERNAL: Dimension, Form, Structure, Surface quality, Tolerance, Position, Orientation

EXTERNAL: Appearance, Cost, Duration, Energy consumption, Environment condition, Function, Maintenance cycle, Manufacturer, Patent, Packaging principle, Performance, Price, Reliability, Safety, Standard, Storage, Task, Time limit

ORGANIZATIONAL: Identity, State, Status, Phase, Type, Time-stamp

4.4. Propositions

The *Propositions* are not restricted to the content expressed by individual sentences of a language. They may encompass the content expressed by theories, books, and even whole libraries. It is important to distinguish *Propositions* from the *Content Bearing Objects* that expresses them. A *Proposition* is a piece of *Information* but a *Content Bearing Object* is an *Object* that represents this *Information*. A *Proposition* is an abstraction that may have multiple representations like: strings, symbols, sounds, drawings, etc. The SUMO definition of *Proposition is:* the *Abstract* entities that express a complete thought or a set of such thoughts. The main terms extracted from GDMS and classified as a *Proposition* are:

Assumption, Behavior, Concept, Composition, Constitution, Criteria, Design, Element, Information, Need, Plan, Principle, Problem, Project, Requirement, Solution, System, Technology, Whole, Wish

4.5. Quantities

The *Quantities* are any specification of how many or how much of something there is. There are two subclasses of *Quantity* defined in SUMO: *Number* (how many) and *Physical Quantity* (how much). A *Physical Quantity* is a measure of some quantifiable aspect of the modelled world, such as 'the shaft's diameter' (a constant length) and 'the stress in a loaded deformable solid' (a measure of stress, which is a function of three spatial coordinates). Although the name and definition of *Physical Quantity* is borrowed from physics, *Physical Quantities* need not be material. Aside from the dimensions of length, time, velocity, etc., non-physical dimensions such as currency are also possible. The main terms extracted from GDMS and classified as a *Quantity* are:

Energy, Effect, Space, Time, Time interval, Time point, Volume

4.6. Relations

The *Relations* are in SUMO defined as general associations which can be shared by distinct pairs (triples, etc.) of individuals. The relations are generally binary, but there is no theoretical bound on the number of arguments of a relation. From the analysis of the GDMS foundation, we concluded that the necessary domain axioms can be specified based on the different associations between the terms e.g. cause, connects, follows, is subclass of etc. The research identified a huge diversity of relations that can be described in the design domain and for the most of them there does not exists complete explanation of their meaning in the background theories. Most of them are characterized in different design models as causal, only to denote their existence, without further explanation of their nature. The huge number of unclassified and undefined relations that creates the complex semantic network between extracted terms in *Design Ontology* may be highlighted as the one of the biggest obstacle in fully formalization of GDMS structure.

In order to formalize the meaning of the different relations, the first step was a classification of the different association by their nature and characterization of commonly used relations that exists between terms in design domain. The standards and literature provide little guidance on what different kinds of semantic relation appear in design models [13], [14]. In order to make characterization of the numerous relations that exists between terms in design domain, associations were grouped and defined by additional axioms considering logical symmetry, reflectivity and transitivity of the specific group.

- **CASE-ROLE RELATIONS:** The class of *Relations* relating the spatially distinguished parts of a *Process*. The relation is *Antisymmetric* and *Irreflexive* by definition. *Case-role* includes, for example, the agent, patient or destination of a transformation. The main terms extracted from GDMS and classified as a *Case-role relation* are:

  Instrument, Operand, Operator, Resource, Input, Output

Example: (*Operand* ?*Process* ?*Entity*) means that ?*Entity* is a participant in ?*Process* that may be moved, changed, experienced, etc.

- **CAUSAL RELATIONS:** The class of *Relations* that capture semantic of the fact that one concept has some effect or impact on another concept. The relation is *Antisymmetric, Irreflexive* and *Transitive*. The main terms extracted from GDMS and classified as a *Causal relation* are:

Aim, Causes, Consequence, Factor, Influence, Opposing, Purpose, Reason, Response, Result, Stimulus, Supporting

Example: (*Causes* ?*Process1* ?*Process2*) means that the ?*Process1* brings about the ?*Process2*.

- **CLASSIFICATION RELATIONS:** The class of *Relations* that capture semantic of kinds, classes and types. The relation is *Antisymmetric*, *Reflexive* and *Transitive*. The main terms extracted from GDMS and classified as a *Classification relation* are:

Is a, Instance of, Sub-kind of

Example: (*Subkind_of* ?*Machine* ?*Device*) means that the ?*Machine* is sub-kind of *Devices* (that that have a well-defined resource and result and that automatically convert the resource into the result).

- **GENERAL RELATIONS:** The class of *Relations* that capture semantic of very general predicates. The main terms extracted from GDMS and classified as a *General relations* are:

Argument, Base for, Describes, Has attribute, Inhibits, Possesses, Precondition, Depends on, Represents, Realises

Example: (*Represents* ?*Object* ?*Entity*) means that ?*Object* in some way indicates, expresses, connotes, pictures, describes, etc. ?*Entity*.

- **INTENTIONAL RELATIONS:** The class of *Relations* between an *Agent* and one or more *Entities*, where the *Relation* requires that the *Agent* have awareness of the *Entity*. The relation is *Antisymmetric* and *Irreflexive*. The main terms extracted from GDMS and classified as a *Intentional relation* are:

Affect, Decision, Dislikes, Need, Wants

Example: (*Needs* ?*Agent* ?*Object*) means that ?*Object* is physically required for the continued existence of ?*Agent*.

- **COMPOSITIONAL RELATIONS:** The class of *Relations* that capture semantic of whole/part concept. The relation is *Antisymmetric, Irreflexive and Transitive*. The main terms extracted from GDMS and classified as a *Meronymic relation* are:

Component of, Element of, Material of, Member of, Portion, of

Example: (*Part* ?*EngineeringComponent* ?*Assembly*) simply means that the *Object* ?*EngineeringComponent* is physical part of the *Object* ?*Assembly*.

- **SPATIAL RELATIONS:** The class of *Relations* that capture semantic of the geometric, physical and other form of connections, contacts or interactions. The relation is *Reflexive* and *Symmetric*. The main terms extracted from GDMS and classified as a *Spatial relation* are:

  Connected, Contains, Encloses, Located, Meets spatially, Overlaps spatially

  Example: (*Overlaps spatially* ?*Object1* ? *Object2*) means that the *Objects* ? *Object1* and ? *Object*2 have some parts in common.

- **TEMPORAL RELATIONS:** The class of *Relations* that capture semantic of the time depend relations. The relation is *Antisymmetric*, *Irreflexive* and *Transitive*. The main terms extracted from GDMS and classified as a *Temporal relation* are:

  After, Before, Co-occur, During, End, Meets temporally, Overlaps temporally, Proceeds, Relative time, Start, Temporally between, Time position

  Example: (*Co-occur* ?*Process1* ?*Process2*) means that the *Process* ?*Process1* occurs at the same time as, together with, or jointly with the *Process* ?*Process2*.

In ontology acquisition it is often possible to distinguish broad natural viewpoints or base categories within the field. In our application, previously presented distinctions refer to the group of concepts' properties that are seen as naturally belonging together following the epistemological foundation. Distinguishing and separating such basic groups appears to be an important structuring principle in *Design Ontology* building process: giving a rise to the strong internal coherence of the ontology proposal.

5. Formalization of the *Design Ontology*

After structuring the initial *Design Ontology*, the next step in our research was developing the formal representation of it, by adding the properties and domain axioms extracted from the GDMS background. The main problem of this phase was that some of the terms have weakly specified semantics, with no related axioms in background theories. What we have found is that the number of possible domain axioms is huge, and we decided in this research phase to include only the simple one. Complex axioms together with completeness theorems that specify necessary conditions for formally rigour ontology will be objective of the future research efforts. An example of simple axiom that was extracted and formally defined is shown in Table 1. The definition provided by GDMS, was in the first step interpreted utilizing the terms included in the *Design Ontology*, and then formalized as an ontology axiom.

In order to do refinement and evaluation, the proposed *Design Ontology* structure was instantiated with actual data in order to articulate the data information and knowledge evolved throughout development of the real product. As an example, the coffee maker machine was chosen following the previous research on GDMS structure [2] [Figure 3]. The software tool that has been used in this phase of the *Design Ontology* building process is OntoEdit® developed by Ontoprise GmbH, Karsruhe, Germany [15]. OntoEdit® is an ontology engineering environment supporting the development and maintenance of ontologies by using graphical means. The paradigm of OntoEdit® supports developing of the concept hierarchy, axioms, and instantiations as much as possible independent of a concrete representation language. The OntoEdit® includes inferencing mechanism and knowledge base that can be used to test an ontology and its axioms. As the result of the inferencing sequence, the new knowledge is inferred based on the existing statements and axioms in the ontology. There is also a possibility for enabling and disabling specific axioms for testing purpose.

Table 1. Example of the *Design Ontology* axiomatization

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 life-cycle ?*MEETING*, then there exists ?*FUNCTION* so that ?*MEETING Results* with purposeful ?*EFFECT*.
Formal ontology axiom:	(=> (*and* (*Instance* ?*MACHINE Machine*) (*Instrument* ?*MEETING* ?*MACHINE*)) (*Exists* (?*FUNCTION*) (*Result* ?*MEETING* ?*EFFECT*)))

Figure 3. Visualization of the coffee maker instantiation of the *Design Ontology*

For example if (*OverlapsTemporally* ?*INTERVAL1* ?*INTERVAL2*) means that the *TimeIntervals* ?*INTERVAL1* and ?*INTERVAL2* have a *TimeInterval* as a common part, then we can make semantic query on the existing statements in the knowledge base asking the system to find by inference which instances of the processes are overlapping in a time (activities like electric energy supplying and water boiling in coffee making process) without definition of particular time association of co-occurring between those instances.

5. Implications

In the product development domain, an ontology is needed to solve many heterogeneity problems. Using formal structures of ontology has advantages over the standardized approach (e.g. STEP schemes), because standardized approach needs a pre-agreement about everything, and in an ontology approach we need just to agree about common terminology. The contribution of this paper can be summarize in merging existing methodologies of building ontologies with our experiment of building ontologies in a product development domain. The major findings we encounter in building the *Design Ontology* are as follows:

1. Formalization of the ontology depends mainly upon background theories. The many statements that we are using in for describing situation in domain of discourse are not understandable without recognizing and respecting the background theories where they are originated and which brings concepts together.
2. Formalization of the *Design Ontology* requires much more detailed specification and explanation of the concepts and associations between them than is provided by current theoretical models in order to provide the framework for useful reasoning about design/product domain.
3. It is necessary that the *Design Ontology* exists both in the form of a comprehensive, carefully prepared natural language and in a formal language in order to be accessible and understandable to the all subjects in product development process.

Aside to mentioned problems, it should be clear that the *Design Ontology* is a working research. Since presented work was built upon predefined theoretical background, our outgoing work is to define complex rules composed of two or more simple rules that will enable us to enforce more constrains on defined structures. On that base we believe will be possible to reach our final goal: develop more knowledgeable information systems that provide intelligent support to the end users that are from related but different communities, thus facilitating knowledge transfer between different communities.

6. Conclusion

In this article we have given a description of the *De*sign *Ontology* research project that is aimed to the achievement of the formal description of the Genetic Design Model System structure. This paper has established the problem being studied, laid out the methods being used, and indicated the possible problems, and benefits that may be achieved. From the research we have learned how existing general upper ontologies can be used to derive, organize and classify terms and their definitions in specific domain ontology in order to gradually develop it in a structured fashion. We believe that such collected experience can be generalized and utilized for the building future ontologies in product development domain.

Acknowledgements: This research is part of funded project “Models and methods of improving the computer aided product development” supported by the Ministry of Science and Technology of the Republic of Croatia. The presented paper is result of the research performed in co-operation with the research staff of Section of Engineering Design and Product Development, Technical University of Denmark.

References

[1] Štorga, M., Andreasen, M.M.: “Complexity of the product development context”; 4th International Seminar and Workshop EDIProD 2004; Poland, 2004.

[2] Mortensen, N.H.: "Design modelling in a Designer’s Workbench – contribution to design language"; Dissertation; IKS, Technical University of Denmark, Denmark, 1999.

[3] Hansen, C. T., Andreasen, M. M.: "Two approaches to synthesis based on the domain theory", in A. Chakrabarti: “Engineering Design Synthesis”; Springer-Verlag; ISBN 1-85233-492-4, Ch. 6, pp. 93-108; 2002.

[4] Benjamin, P.C., Menzel, C:P., Mayer, R.J., Fillon, F., Futrell, M.T., deWitte, P.S., Lingineni, M.: “IDEF5 Method Report”; Knowledge Based System Inc., CN F33615-C-90-0012, Texas, USA; Armstrong Laboratory, Wright-Patterson Air Force Base, Ohio, USA; 1994.

[5] Štorga, M., Marjanović, D., Bojčetić, N.: "Consideration on IT systems interoperability in product development", Proceedings of the TMCE 2004, Millpress, Rotterdam, ISBN, 2004.

[6] Borst, W.N.: "Construction of engineering ontologies for knowledge sharing and reuse"; PhD thesis, Centre for Telematics and Information Theory; Enschelde; Netherlands; 1997.

[7] Neches, R., Fikes, R.E., Finin, T., Gruber, T.R., Senator, T., Swartout, W.R.: “Enabling technology for knowledge sharing”; AI Magazine 12 (3); pp. 36-56; 1991.

[8] Studler, R., Benjamins, V.R., Fensel, D.: “Knowledge engineering: principle and methods”; Data and Knowledge Engineering 25; pp. 161-197; 1998.

[9] Uschold, M., Jasper, R.: “A framework for understanding and classifying Ontology Application”; Proc. IJCAI99 Workshop on Ontologies and Problem-Solving Methods; Stockholm; 1999.

[10] Mehkilef, M., Bourey, J.P., Bigand, M.: “An UML modelling of an architecture for knowledge documentation”; 14th International Conference on Engineering Design, ICED 03, Stockholm, Sweden, 2003.

[11] Niles, I., Pease, A.: “Origins of the IEEE Standard Upper Ontology”; Working Notes of the IJCAI-2001 Workshop on the IEEE Standard Upper Ontology; Seattle; pp. 37-42; 2001.

[12] Štorga, M.: “Traceability in product development”; 8th International Design Conference, DESIGN 2004; Dubrovnik, Croatia, 2004.

[13] Knowledge Base Systems: “IDEF 05 Method Report”; 1994.

[14] Salustri, F.A., Lockledge, J.C.: “Towards a formal theory of products including mereology”; Proceedings of the International Conference on Engineering Design ICED 99, Munich, Germany, 1999.

[15] “How to work with OntoEdit Version 2.6”; Ontoprise GmbH, Karsruhe, Germany, www.ontoprise.de; 2002.

M. Sc. Mario Štorga, Dipl. Ing. ME
PhD student, University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture
Ivana Lucica 5, 10000 Zagreb, Croatia
Tel. +385 1 6168 119, Fax. +385 1 6168 284
Email: mario.storga@fsb.hr; URL: www.cadlab.fsb.hr