Granular Meta-Ontology and Extended Allen’s logic: Some Theoretical Background and Application to Intelligent Product Lifecycle Management Systems

Valery B. Tarassov Alena V. Fedotova
CIM Department
Bauman Moscow State Technical University
Moscow, Russia
e-mail: vbulbov@yahoo.com, afedotova.bmstu@gmail.com

Rainer Stark
Faculty of Mechanical Engineering and Transport Systems
Berlin University of Technology (TU Berlin)
Berlin, Germany
e-mail: rainer.stark@tu-berlin.de

Baurzhan S. Karabekov
Institute of Informational and Computational Technologies
Ministry of Education and Science of Kazakhstan
Almaty, Kazakhstan
e-mail: bskarabekov@mail.ru

***Abstract*—A hierarchical system of ontologies is considered, where the concepts of meta-ontology and upper ontology are of primary concern. The concept of meta-ontology is discussed; the distinction between meta-ontology and upper ontology is shown. Various methodologies for constructing formal ontologies are analyzed. There is a need for a generalized approach to ontological modeling based of Maltsev’s algebraic systems is justified. Basic principles of information granulation and granular ontology construction are formulated; some formal definitions of granular meta-ontologies together with fuzzy and linguistic ontologies (based on extended linguistic variables) are introduced. An application of granular meta-ontology and upper ontology concepts to lifecycle modeling is considered in the context of building new generation product lifecycle management systems – intelligent Product Lifecycle Management (PLM) system. Circular and sequential lifecycle representations are constructed and interpreted as coarse-grained and fine-grained ontologies. The use of extended Allen’s interval logic as an integrated parallel-sequential lifecycle ontology is suggested.**

Keywords-Ontological engineering; granular meta-ontology; Allen’s logic; lifecycle ontologies; intelligent PLM.

I. INTRODUCTION

A main theoretical purpose of this paper consists in bridging the gap between two different scientific areas, named by the same term – “Ontology”. On one hand, ontology is a classical philosophical discipline that studies the nature of existence, the basic categories of being and their relations. It faces problems, such as what is a thing, why various entities exist, and how these entities may be grouped, related within a hierarchy, and subdivided according to similarities and differences [7]. On the other hand, in computer science and artificial intelligence, ontology is a description (like a formal specification of a program) of the concepts and relationships [9] that can formally exist for an agent or a community of agents.

It can be argued that in computer science «exist» means «be represented in a computer model». So, any ontology defines a set of representational primitives in order to model a domain of knowledge or discourse. These representational primitives are typically classes (or sets), attributes (or properties) and relationships (or relations among class members) [1].

From a methodological viewpoint, ontological engineering is an adoption of modern systemic, highly interdisciplinary approach in computer science. This field studies the methods and methodologies for building various ontologies: formal representations of a set of concepts within a domain and the relationships between those concepts. A large-scaled representation of the most abstract concepts, such as system, relation, time, space, action, event, etc., is the core of ontological engineering [13], [14],

Ontologies are typically specified in languages that allow abstraction away from data structures and implementation strategies. For this reason, ontologies are said to be at the "semantic" level, whereas database schema are models of data at the "logical" or "physical" level.

A practical problem to be solved consists in developing product (system’s) lifecycle models based on granular meta-ontology and extended Allen’s logic. These granular models will contribute to the development of intelligent Product Lifecycle Management (PLM)-system enabling enterprise knowledge management using lifecycle ontological engineering [29].

The paper is organized in the following way. In Section II, we consider the transition from descriptive ontological systems to formal granular ontologies. First of all, in sub-section *A* we present our modification of Guarino’s hierarchical system of ontologies that enables top-down ontology engineering. Here, the concepts of meta-ontology and upper ontology are of primary concern. To clarify the first concept, some principles of meta-logic and meta-mathematics are considered, and the distinctions between meta-ontology and upper ontology are discussed.

Sub-section *B* is devoted to a presentation of various methodologies for constructing formal ontologies. The main emphasis is made on discussing and comparing basic approaches suggested by representative of both classical philosophical-mathematical and novel computer science communities. Furthermore, the usefulness of abstract algebraic approach to ontological modeling is shown, and the perspectives of Maltsev's algebraic systems in ontological modeling are outlined.

In sub-section *C*, we consider some basic principles of information granulation, specify the concepts of granule and granularity, give the classification and interpretation of granules and mention fundamental components of ontology granulation. In Sub-section *D*, a formal granular meta-ontology together with fuzzy and linguistic ontologies (based on extended linguistic variables) are introduced.

Section 3 is devoted to applications of granular meta-ontologies to lifecycle modeling. In sub-section *A*, the crucial role of lifecycle ontological modeling and engineering for developing intelligent product lifecycle management systems. In this context, two basic time metaphors and time theories are briefly analyzed. In sub-section *B*, circular lifecycle representations as coarse-granular ontologies are given. In sub-section *C*, formal lifecycle models are proposed, and sequential lifecycle ontologies based on Allen's relations and their extensions are constructed.

II. FROM DESCRIPTIVE ONTOLOGICAL SYSTEMS TO FORMAL GRANULAR ONTOLOGIES

A. On a Hierarchical System of Ontologies

In this paper, we consider ontological modeling and engineering as a generic tool of knowledge management in multi-agent systems and intelligent organizations. Ontological modeling encompasses both cognitive and communicative sides of knowledge management. On one hand, ontological investigations deal with the problems of knowledge generation on the basis of some entities. On the other hand, their objective is to support communication processes, i.e., enable knowledge sharing and reuse.

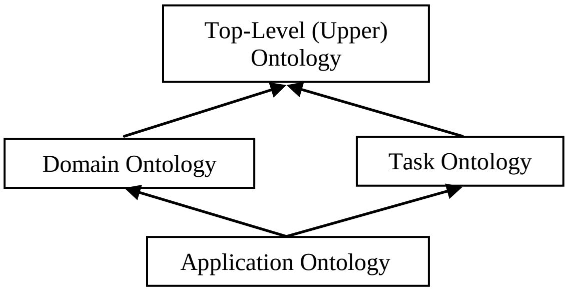


Figure 1. Guarino's Types of Ontologies

It is often difficult or even impossible to construct single, comprehensive, coherent and practically useful ontology. To simplify ontology development, integration and reuse, a modular ontological engineering technique is suitable. In these cases, it is possible to exploit the principle «divide and rule»: decompose the subject domain into sub-domains with consequent composition, introduce upper ontologies, tasks ontologies and application ontologies [1] (see Fig. 2).

Here, domain ontology (for instance, maintenance ontology) encompasses domain concepts together with basic relations between them, whereas task ontology is related to main tasks or actions. For example, in maintenance we deal with such tasks as inspection, disassembly, substitution, assembly, diagnostics. The position of application ontology may be different. On one hand, application area specifies main requirements to domain and task ontologies. On the other hand, application ontology (for instance, aircraft maintenance) describes concepts depending both on a particular domain and tasks, which are often have specializations of *both* the related ontologies. This concept often corresponds to roles, played by domain entities while performing a specific activity. All these three types of ontologies may be referred to as low level ontologies, which are domain-dependent.

An upper ontology (or a top-level ontology) [1], [2] is a model of the common objects (or concepts) that are generally applicable across a wide range of domain ontologies. Such ontologies describe very general concepts like space, time, matter, resource, event, action, etc., which are independent of a particular problem or domain. It seems quite reasonable to start enabling mutual understanding and joint work of agents by specifying basic upper ontologies.

Nevertheless, this classical Guarino's ontological system [1] as depicted in Fig. 1 corresponds to a bottom-up approach in ontological engineering.

Below we present our three-leveled hierarchy of ontologies that includes low-level ontologies, upper ontologies and meta-ontology (Fig. 2). It illustrates a top-down approach to ontological engineering. Here, the concept of Meta-Ontology is of primary concern. Unfortunately, the concepts of meta-ontology and upper ontology are often confounded.

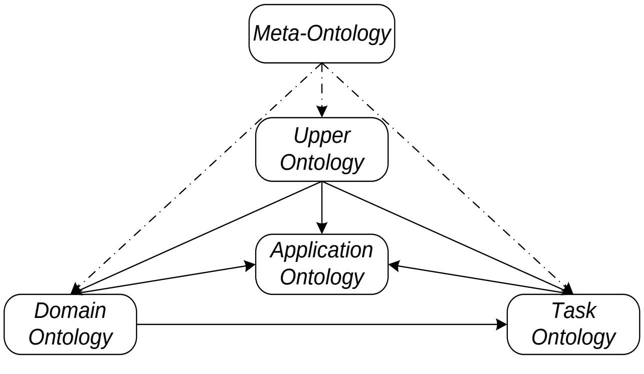


Figure 2. Representation of ontological hierarchy

To explain the concept of meta-ontology, we can use the analogy with meta-logic and meta-mathematics. About a hundred years ago, Russian scientist N.A. Vasiliev (see [3]) suggested a two-leveled logical structure, the sort of simple hierarchy of empirical logic and meta-logic. He considered human everyday logic as dual, semi-empirical, semi-rational area, and by contrast he envisaged a sort of generalized logic, called meta-logic. According to Vasiliev, we ought to make difference between two levels of knowledge: 1) an empirical level, based on real-world's events; 2) a conceptual level, depending on our thinking.

In his turn, S.C. Kleene defined meta-mathematics as the study of intrinsic capabilities of our formal models by mathematical methods.

Similarly, a meta-ontology (ontology of ontologies) studies the nature of ontological problems and specifies a family of ontologies. It provides the investigation of basic ontology properties and establishes the links between various ontologies. It means, that meta-ontology framework includes a family of mathematical theories, used for formal ontology description. In particular, it contains methods and forms of representing, integrating and merging different ontologies.

Below we shall present the concept of granular meta-ontology on the basis of Mal'tsev's algebraic system [4]. At first, we should discuss the notion of formal ontology.

B. *Formal Ontologies*

The idea of formal ontology was coined by E. Husserl who differentiated between formal logic on one hand and formal ontology on the other. In his «Logical investigations» E. Husserl has shown, that formal logic deals with the interconnections of truths (or propositional meanings in general) – with consequence relations, inference, consistency, proof, validity, whereas formal ontology considers the interconnections of things, objects and properties, parts and wholes, relations and collectives [5]. Husserl's formal ontology stands on three pillars: mereology, topology and the theory of dependence. Later on this idea was developed by St. Leśniewski – the author of three nested formal systems called respectively protothetic, ontology and mereology [6] and B. Smith [7], who suggested the concept of mereotopology as a theory of parts and boundaries.

Another classical approach rises to Tarski's concept of logical notion [8]. To define it, Tarski drew on Klein's Erlangen Programme that classified the various types of geometry (Euclidean geometry, affine geometry, etc.) by the type of one-one transformation of space onto itself that left the objects of this geometrical theory invariant. Generalizing, he specified the concept of logical notion by considering all possible one-to-one transformations (automorphisms) of a domain onto itself.

In the context of ontological engineering a basic proposal is to interpret model-theoretic languages (abstract logics) as formal ontologies [9]. Here, an abstract logic consists of: (1) a collection of structures closed under isomorphism, (2) a collection of formal expressions, and (3) a relation of satisfaction between the two [10].

In modern computer science and artificial intelligence the term «ontology» stands for clear and formal specification of problem domain structure. According to T.R. Gruber, it is often referred to as an explicit specification of shared conceptualization [11]. In other words, ontology is a conceptualization of a domain into a human understandable, but machine-readable format. A formal definition of conceptualization has been given in [12]. It is a structure $CON_1 = \langle D, R\rangle$, where D is a domain and R is a set or relevant relations on D.

In his turn, N. Guarino [13] defines ontology as a logical theory that specifies some conceptualization; it includes some basic terms forming taxonomy, their definitions and attributes, related axioms and inference rules. In other words, ontology is seen both as a formal view to semantics and a thesaurus used by logical theory. In essence, any ontology expresses some convention about shared methods of constructing and using conceptual models. It can play the role of knowledge representation and reuse method, knowledge management tool, learning technique, etc.

He also focuses on the need of transition from extensional to intensional relations. A standard way to represent intensions (and therefore conceptual relations) is to see them as functions from possible worlds into sets. While ordinary relations are defined on a certain domain, conceptual relations are defined on a domain space – a structure $\langle D, W\rangle$, where D is a domain and W is a set of possible worlds. Then a conceptual relation ρ^n of arity n on $\langle D, W\rangle$ is given as a function $\rho^n: W \rightarrow 2^{D^n}$ from W into the set of all n-ary (ordinary) relations on D.

Thus, a conceptualization for D can be now defined as an ordered triple $CON_2 = \langle D, W, \mathrm{P} \rangle$ [13], where P is a set of conceptual relations on the domain space $\langle D, W\rangle$.

Nowadays, some generalized approaches to building formal ontologies based on universal algebras and category theory are worth noticing [14]. Moreover, a new ontological framework Basic Formal Ontology (BFO) [15] that consists in a series of sub-ontologies at different levels of granularity is of special concern. Here, the basic concepts are subdivided into Continuants (e.g., Objects and Functions) and Occurents (e.g. Processes and Events).

To construct granular ontology (and meta-ontology) both, new formal ontological models and fundamentals of information granulation theory, ought to be developed.

C. *Information Granulation and Granular Ontologies*

Let us consider some basic principles of information granulation in the context of creating granular meta-ontology. The main concepts of granulation theory are granules, granularities, hierarchies, levels, granular structures and theirs mappings.

Information granulation is a basic capacity of cognitive agent that supposes processing information on such level of abstraction, which is consistent with the allowable level of imprecision. The term “granule” is originated from Latin word “granum”, that means grain, to denote a small particle in the real or imaginary world. Typical interpretations of granules are: part of the whole, sub-problem of the problem, cluster, variable constraint, uncertainty area, etc.

According to L. Zadeh, granule is seen as a collection of objects which are drawn together by indistinguishability equivalence, similarity or functionality [16]. Information granules are complex dynamic information entities which are formed to achieve some goal. The arrival of information granulation means the transition from ordinary machine-centric to human-centric approach in information gathering and knowledge discovery [17]. By selecting different granulation levels one can construct heterogeneous ontological models with modifiable abstraction degrees.

It is easy to clarify the sense of the term “granular” by comparing it with the antonym “singular”. For example, one of the founders of multi-valued logics J. Lukasiewicz specified the basic concepts of Truth and Falsity as singular objects, whereas Zadeh’s consideration of linguistic variable Truth with such linguistic hedges as «more or less true», «rather true than false» and so on supposes the shift from singular to granular truth values.

The same idea underlies granular ontology: the transition from singular (pointwise) representation primitives to interval and regional representation primitives is the essence of ontological granulation. For example, the transition from fine-grained low-level ontology, given by an ordinary graph, to coarse-grained upper ontology, given by a hyper-graph, may be fulfilled.

Some classical approaches to ontological granulation are presented in [18, 19].

Now let us discuss some basic components of ontology granulation theory. These are: 1) ontology granulation principles and criteria; 2) interpretation and classification of granules; 3) approaches and methods of granulation; 4) formal models of ontological granules; 5) Ontological granular structures; 6) mappings of granular structures; transitions from fine-grained to coarse-grained ontologies and vice versa; 7) quantitative indices for granular ontologies and granulation process itself.

D. Formal Granular Meta-Ontologies.

From the systemic viewpoint, meta-ontology makes appeal to the most universal domain-independent categories, such as *concepts*, *relations*, *changes*. The timely adaptation for changes and the management of these changes characterizes a dynamic meta-ontology.

A natural mathematical basis for specifying meta-ontologies is Maltsev’s theory of algebraic systems [4]. Below we shall recall the concept of algebraic system and extend it to take into account granularity and fuzziness.

Definition 1 [4]. An algebraic system is a triple

$$AS = \langle X, O, \Pi \rangle, \qquad (1)$$

where X is a non-empty set of objects, called the underlying set (or basis) of the algebraic system, *O is an operation set, i.e., the set of finitary operations on* X *and Π is predicate set. Here,* $O=\{o_i^j\}$, $i = 1,\ldots, m$, $j = 0,\ldots, n$, $o_i: X \to X$, $o_i^2: X \times X \to X, \ldots, o_i^n: X^n \to X$. Constants are also included into O as 0-ary functions. $\Pi=\{\pi_k^l\}$, $k=1,\ldots, p$, $l=1,\ldots,q$, $\pi_k: X\to \{0,1\}$, $\pi_k^2: X \times X \to \{0,1\}, \ldots, \pi_k^q: X^q \to \{0,1\}$.

The union of operation set and predicate set $O\cup\Pi$ is called a signature (type) of algebraic system. The algebraic systems with coinciding signatures have the same type.

The algebraic system can have multiple basis, for instance, $X = (X_1,\ldots, X_n)$. When underlying set, provided with the structure of topological space and operations, are continuous, we obtain topological algebraic system. Various topological spaces may be represented as sets equipped with closure operation.

If the operation set may include partial operations, then algebraic system is called partial algebraic system.

Two special cases of algebraic system are: a relational system (model) for $O=\varnothing$ and a universal algebra for $\Pi=\varnothing$.

Remark 1. In case of fuzzy relations, the concepts of relation and predicate coincide.

Definition 1*. A Meta-Ontology is given by an algebraic system

$$MONT = \langle C, R, O \rangle, \qquad (1^*)$$

where C is a non-empty set of concepts, R is a set of relations on C and O is a set of operations over concepts and/or relations.

Remark 2. It is worth noticing**,** that formal specification of meta-ontology (ontology) by relational system is not sufficient for ontology integration and investigation. The intersection of various concept sets allows us to specify the kernel of ontology (degree of sharing), their union gives its range and difference helps to compare ontologies. Moreover, specific operations of generalization and specialization provide the changes of ontology granularity.

Definition 2. Let C be a non-empty set of concepts. We call a conceptual granule any subset $g\in 2^C$, where 2^C is a power set of C.

Definition 3. For any two conceptual granules g, $g'\in 2^C$, if $g\subseteq g'$, then g is called a sub-granule of g', and, in its turn, g' is a super-granule for g.

Definition 4. Let us denote $G\in 2^C$ a non-empty set of conceptual granules. Then a pair $GS = \langle G, \subseteq\rangle$ is called a granular structure if $\subseteq$ is set inclusion.

Now, let us give an unfolded definition of granular meta-ontology [20].

Definition 5. A granular meta-ontology is a quadruple

$$GMONT = \langle C, G^C, R_G, O_G \rangle, \qquad (2)$$

where C is a non-empty set of concepts, G^C is a basis of ontological granulation, R_G is a set of granular relations on C_G, O_G is a set of operations over C_G. and/ or R_G. Among main bases of ontological granulation we take the ordinary power set 2^C, the set of fuzzy subsets $[0,1]^C$, the set of sub-lattices L^C of a lattice L.

Typical ways of specifying granular sets of concepts are the following: 1) a set of concepts C together with a quotient set C/E, denoted by $C_{G1}=(C, C/E)$,where E is an equivalence relation; 2) a set of concepts C with a family of nested sets $F = \{A_0, \ldots, A_n\}$, $C_{G2} = (C, F)$, $F = \{A_0, \ldots, A_n\}$, where $A_i \subseteq C$, $i=0, \ldots$, n, $A_0 = X$, $A_0 \supseteq A_1 \supseteq \ldots \supseteq A_n$ or more generally as a set of α-cuts defined on the lattice L, $A_\alpha: L \rightarrow 2^X$, $\alpha \in L$; 3) a set C with a family of fuzzy subsets $[0,1]^C$, $C_{G3} = (C, [0,1]^C)$. 4) a universal set C together with a rough set, given by lower and upper approximation.

A special granular computing view of an ontology, based on rough set methodology, is developed in [21].

A good example of granular ontology is fuzzy ontology, where fuzzy concepts and/or fuzzy relations and/or fuzzy attributes are considered. Two definitions of lightweight and heavyweight fuzzy ontologies are given below

Definition 6. A fuzzy ontology is a quadruple

$$FONT = \langle I, C_F, H, R_F \rangle, \qquad (3)$$

where I is the set of individuals (instances of concepts), C_F is the set of fuzzy concepts, H is the hierarchy, $\boldsymbol{R_F}$ is the family of fuzzy relations sets.

Definition 7. A completely fuzzy ontology is a quintuple

$$FONT = \langle I, C_F, R_F^k, O_F^j, AX \rangle, \qquad (4)$$

where I is the set of individuals (instances of concepts), C_F is the set of fuzzy concepts, $\boldsymbol{R_F^k}$ is the family of fuzzy relations sets, k = 1,2,…, s; O_F^j is the set of finite operations over fuzzy concepts and/or fuzzy relations, $j=0,..,n$, AX is the set of axioms.

Fuzzy ontologies were already extensively studied (see, for instance, [22]) On the contrary, the specification of ontologies on the basis of Zadeh's linguistic variable remains a rather rare case. Below a fuzzy linguistic ontology is introduced on the basis of extended linguistic variable.

Definition 8 [20]. An extended linguistic variable is given by a tuple

$$LV_{ex} = \langle L, T, U, G, M, R_T, R_U,, O_g, TR_U \rangle, \qquad (5)$$

where L is the name of linguistic variable, T is its term set, U is the universal set (numerical scale), G is the set of syntactic rules (grammar), M is the set of semantic rules, R_T is the set of relations on T, R_U is the set of relations on U, O_g is the set of granulation operations, TR_U is the set of universe transformations.

Definition 9. A fuzzy linguistic ontology based on extended linguistic variable is a tuple

$$LVONT = \langle I, C_A, C_F, R, U, [0,1]^U, R_F \rangle, \qquad (6)$$

where I is the set of individuals (instances of concept), $C_A = \{c_A\}$ is an abstract concept (singleton) that corresponds to the name of linguistic variable, C_F is the set of fuzzy concepts (the term set of linguistic variable), $R = \{r \mid r \subseteq C_F \times C_F\}$ is the set of binary relations between fuzzy concepts. Let us note, that the strict order relation $<$ is of special concern. The pair $\langle C_F, < \rangle$ generates an ordered structure. Here, U is the universal set, $[0,1]^U$ is the set of fuzzy subsets on U, R_F is the set of fuzzy relations on $[0,1]^{\boldsymbol{U}}$.

III. ON THE USE OF GRANULAR META-ONTOLOGY: LIFECYCLE UPPER ONTOLOGIES

A. The Role of Lifecycle Modeling for Intelligent PLM

At present, the concept of ontology development lifecycle is thoroughly studied (see, for instance, [23], [24], [29]), but the problems of system's lifecycle ontology and lifecycle ontological modeling still remain open.

The aim of cyclic product definition is to realize both products and processes, and economic solutions that are better and more intelligent by integrating lifecycle philosophy into technology and economy.

The lifecycle concept may be analyzed from various viewpoints; different variants of specifying its phases and activities were suggested. In marketing theory products follow such stages as introduction, growth, maturity, and decline. In industry, all products or systems have a particular life span considered as a sequence of stages, which is called product lifecycle (or complex system lifecycle).

The term «system's lifecycle» expresses the idea of a circulation of produced artifacts between the fields of design, production and usage (consumption). One of fundamental resources for lifecycle management is time. Any cycle as a whole is characterized by the presence of finite and repetitive parts on some temporal intervals; here, key parameters are durations.

Nowadays, Product Lifecycle Management is viewed as a basic manufacturing strategy for XXIst century [25]. It is deployed as a process of managing the entire lifecycle of a product (system) from its conception, through design and manufacture, to service, disposal and dismantling. An implementation of PLM-system means integration of data, processes, personnel and organizations to provide a product information backbone for modern computer integrated (in particular, virtual and extended) enterprises.

We point out, that PLM initiative considers both questions: «how an enterprise works» and «what is being created». An effective PLM improves the ability of manufacturing enterprise to make better and faster product-related decisions. It enables the formation of a consistent set of concerted industrial solutions that support the collaborative creation, management, dissemination and use of product-definition information [26], [31].

It seems quite reasonable, that an advanced PLM-system has to support various lifecycle representations. Nevertheless, even the most popular industrial PLM-systems like Teamcenter Enterprise or Agile 9 lack this capacity and are in fact product data management systems. In our opinion, this situation is mainly explained by the absence of lifecycle ontological subsystem. The creation of intelligent PLM-system supposes the development of lifecycle engineering methods based on lifecycle ontologies [30].

B. Time Metaphors and Theories for Lifecycles

Basic time theories should be envisaged in the context of lifecycle modeling: substantial and relational, static and dynamic, pointwise and interval time. Two well-known time metaphors – «time wheel» and «time arrow» – bring about lifecycle's circular and sequential models respectively. On one hand, sequential linear models express such time properties as course, ordering facility, irreversibility. On the other hand, circular time models make emphasis on alternations, reiterations, rhythms, self-sustaining processes. We shall support such a pluralism of lifecycle ontologies by constructing and analyzing both circular and sequential representations.

A specific lifecycle feature is its heterochronous character, i.e., irregularity related to the different vision of temporal criteria and constraints on various stages [27]. In fact, we try both to accelerate design and manufacturing time and slow down usage time. For instance, during the design stage a basic criterion is to decrease design time, e.g. by using concurrent design strategies. Contrarily, on the usage stage we tend to keep or increase reglamentary period, for example, by improving maintenance system.

C. Circular Lifecycle Representations: Coarse-Grained Ontologies

In case of system's lifecycle, two basic granule types are lifecycle stages and phases. Lifecycle stages are usually divided into lifecycle phases, where each phase corresponds to a specific system's state. So, the stage is viewed as a coarse-grained lifecycle part, whereas the phase is a fine-grained part.

At first, we shall represent lifecycle stages in the framework of set-theoretic approach as granules obtained by partition. Let us introduce natural denotations for systems's lifecycle stages: D – design; M – manufacturing; U – use; R – recycling. Then we have

$$LC_1 = D \cup M \cup U,\ D \cap M = \varnothing,\ M \cap U = \varnothing,\ U \cap D = \varnothing \qquad (7)$$

or

$$LC_2 = M \cup U \cup R,\ M \cap U = \varnothing,\ U \cap R = \varnothing,\ R \cap M = \varnothing \qquad (7^*)$$

Here, the structure of LC_2 (7*) expresses the «ecological imperative» of modern manufacturing being tightly related to Kimura's lifecycle inversion concept [27]. The first lifecycle partition LC_1 (6) may be depicted by sectors of the circle (Fig. 3).

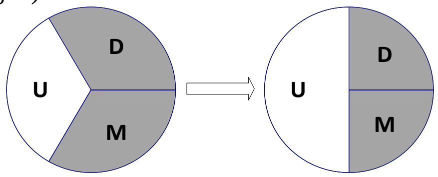


Figure 3. A Circular Representation of System's Lifecycle: an Illustration of Reducing Lead (Design and Manufacturing) Time and Increasing Period of Usage

It is worth noticing that the representation of lifecycle by ordinary partition is rather simplistic and does not express many existing interrelations and co-operation links between partially overlapping stages. Moreover, this simultaneous work enables very important functions. For example, on the crossroad of usage and design system's specification is made, production technologies ought to be discussed on the edge of design and manufacturing, whereas maintenance requires the collaboration of users and manufacturers. With taking into consideration such factors, we obtain the circular lifecycle model with fuzzy boundaries. For these cases lifecycle granulation is based on covering (Fig. 4). Here,

$$LC_1 = D \cup M \cup U, \text{ but } D \cap M \neq \varnothing, M \cap U \neq \varnothing, U \cap D \neq \varnothing \quad (4)$$

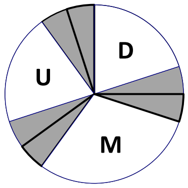


Figure 4. A Circular Lifecycle Representation on the Basis of Covering: the Presence of Collaborative Works at All Stages

D. Sequential and Parallel- Sequential Lifecycle Represenation : Fine-Grained Ontologies Based on Extended Allen's Relations

Generally our approach is based on relational time model and interval time primitives. We use a fuzzy extension of well-known Allen's temporal logic [28] to model the links between lifecycle phases (or lifecycle stages and phases). These are mainly two types of relations: consequence and overlapping relations (see Table I).

Let us recall, that fuzzy quantity is defined as a fuzzy set of the real line. Fuzzy quantities are more suitable to describe flexible requirements on lifecycle parts duration.

We introduce a formal model of lifecycle ontologies ONT_{LC} as a quadruple

$$ONT_{LC} = \langle C_{LC}, R_{LC}, \Omega_{LC}, T_{LC} \rangle, \qquad (8)$$

where C_{LC} is the set of concepts related to lifecycle, R_{LC} is the set of relations between these concepts, Ω_{LC} is the set of operations over concepts and/or relations, T_{LC} is the set of temporal characteristics for lifecycle.

Basic concepts for lifecycle are its phases and stages; therefore, the triple below can be taken as lifecycle systemic kernel

$$\mathrm{ONTS} = \langle \mathrm{S}, \mathrm{Rs}, \mathrm{Os} \rangle, \qquad (9)$$

where S is the set of lifecycle stages (phases), R_s is the set of relations between these stages (phases), O_s is the set of operations used on these stages (phases).

It is worth noticing that each lifecycle phase may be seen as an interval primitive $s=[a^-, a^+]$, where a^- is the starting point and a^+ is the end point of the interval. A fuzzy interval extending the concept of an interval is a special kind of fuzzy quantity that is represented by a convex fuzzy subset of a real line. As a special case, we have

$$ONT_{S1} = \langle S, <_f, \approx_f \rangle, \qquad (10)$$

where $<_f$ is a fuzzy strict linear order relation that is non-reflexive, asymmetric, transitive and linear, $\approx_f$ is a fuzzy simultaneity relation, i.e., fuzzy reflexive, symmetric relation.

More generally, we can use the linguistic variable «Time» with a linguistically ordered term set such as {almost simultaneously, a bit later, later, much later, very much later}.

A general representation of lifecycle ontology can be depicted by a mind map (Fig. 5). Here, such ontology characteristics as its goal, role, language, representation form and basic relations are of special concern.

IV. CONCLUSION

In this paper, some links between two different scientific areas called "Ontology" have been established through formal granular ontologies. The concept of granular meta-ontology has been discussed, formal models of granular, fuzzy and linguistic ontologies have been developed. An application of granular meta-ontology and extended Allen's logic to system's lifecycle ontological engineering has been considered. Our future work will be focused on specifying basic indices for granular ontologies and developing an ontological sub-system for intelligent PLM-system.

ACKNOWLEDGMENT

This work has been supported by Russian Fund for Basic Research (Project No 14-07-00846), Ministry of Education and Science of Kazakhstan (Project 0115 RK 00532) and DAAD and Russian Ministry of Education (Project HM3754).

REFERENCES

[1] N. Guarino, "Formal ontology and information systems", Proceedings of the 1st International Conference on Formal Ontologies in Information Systems, FOIS'98, Trento, Italy, June 6-8, 1998, Ed. by: N. Guarino, Amsterdam, IOS Press, 1998, pp. 3-15.

[2] J.F. Sowa, "Top-level ontological categories", International Journal of Human-Computer Studies, vol.43, No.5-6, 1995, pp. 669-685.

[3] G. Priest, "N.A. Vasiliev and imaginary logic", History and Philosophy of Logic, vol.21, No.1, 2000, pp. 135-146.

[4] A.I. Mal'tsev, "Algebraic Systems", Berlin: Springer, 1973.

[5] E. Husserl, "Logical Investigations", London: Routledge and Kegan Paul, 1970.

[6] St. Leśniewski, "Collected Works", ed. by S.J. Surma, J.T.J. Srzednicki, D.I. Barnett, Dordrecht: PWN - Kluwer, vol.1 and 2, 1992.

[7] B. Smith, "Mereotopology: a theory of parts and boundaries," Data and Knowledge Engineering, vol.20, 1996, pp. 287-303.

[8] A. Tarski, "What are logical notions?", History and Philosophy of Logic, vol.7, 1986, pp. 143-154.

[9] E. Dragalina-Chernaya, "Abstract Logics as Formal Ontologies of Relations", L'ontologie de la relation, Caen: l'Université de Caen, 2012.

[10] J. Barwise, "Model-theoretic logic: background and aims", Model-Theoretic Logic, ed. by J. Barwise. and S.Feferman, New York, 1985, pp. 3-23.

[11] T.R. Gruber, "A Translation approach to portable ontologies", Knowledge Acquisition, vol.5, No.2, 1993, pp. 199-220.

[12] M.R. Genesereth and N.J. Nilsson, "Logical Foundation of Artificial Intelligence", Los Altos, Ca: Morgan Kaufmann, 1987.

[13] N. Guarino, "Formal Ontology, Conceptual Analysis and Knowledge Representation", International Journal of Human-Computer Studies, vol.43, No.5-6, 1995, pp. 625-640.

[14] M. Johnson and R. Rosenbrugh, "Ontology engineering, universal algebra and category theory", Theory and Applications of Ontology: Computer Applications, ed. by R.Poli et al., Heidelberg: Springer, 2010, pp. 565-576.

[15] R. Arp, B. Smith and A.D. Spear, "Building Ontologies with Basic Formal Ontology", Cambridge, MA: MIT Press, 2015.

[16] L. Zadeh, "Toward a Theory of Fuzzy Information Granulation and its Centrality in Human Reasoning and Fuzzy Logic", Fuzzy Sets and Systems, vol.90, 1997, pp. 111-127.

[17] A. Bargiela and W. Pedrycz, "Granular Computing: an Introduction", Dordrecht: Kluwer Academic Publishers, 2003.

[18] T. Bittner and B. Smith, "Vagueness and granular partition", Formal Ontology in Information Systems, ed. by C.Welty, B.Smith, New York: Sheridan Press, 2001.

[19] T. Bittner and B. Smith, "Granular spatio-temporal ontologies", Proceedings of the AAAI Spring Symposium on Foundations and Applications of Spatio-Temporal Reasoning (FASTR), Menlo Park, CA: AAAI Press, 2003, pp. 12-17.

[20] V.B. Tarassov, A.P. Kalutskaya and M.N. Svyatkina, "Granular, fuzzy and linguistic ontologies to enable mutual understanding between cognitive agents", Proceedings of the 2nd International Conference on Open Semantic Technologies for Intelligent Systems, OSTIS-2012, Minsk, Belarus, February 16-18, 2012, Minsk: BSUIR Press, 2012, pp. 267-278.

[21] S. Calegari and D. Ciucci, "Granular computing applied to ontologies", International Journal of Approximate Reasoning, vol. 51, 2010, pp. 391-409.

[22] S. Calegari and D. Ciucci, "Towards a fuzzy ontology definition and a fuzzy extension of an ontology editor", Enterprise Information Systems. Lecture Notes in Business Information Peocessing, vol.3, Heidelberg: Springer, 2008, pp. 147-158.

[23] A. Gomez-Perez, M. Fernandez-Lopez and O. Corcho, "Ontological Engineering", Heidelberg: Springer, 2005.

[24] A. Borisov, G. Kuleshova and T. Zmanovska, "Introduction to Ontology Engineering", Riga: RTU Press, 2014.

[25] J. Stark, "Product Lifecycle Management: 21st Century Paradigm for Product Realization", 2nd ed., London: Springer, 2011.

[26] A. Smirnov and N. Shilov, "Ontology Matching in Collaborative Recommendation System for PLM", International Journal of Product Lifecycle Management, vol.6, No.4, 2013, pp. 322-338.

[27] A. Fedotova and V. Tarassov, "Development and interpretation of spiral lifecycle's model: a granular computing approach, part 1. Lifecycle granulation and spiral representation", Proc. of the 7th International Conference on Soft Computing, Computing with Words and Perceptions in System Analysis, Decision and Control, ICSCCW'2013, Izmir, Turkey), Kaufering: b-Quadrat Verlag, 2013, pp. 432-440.

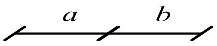
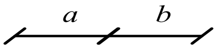
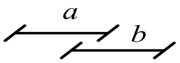
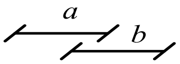
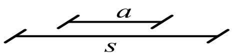
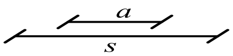
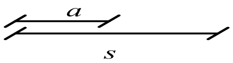
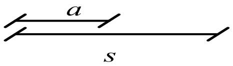
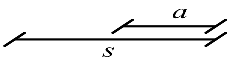
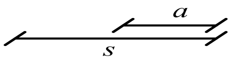
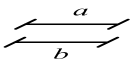
[28] J.F. Allen, "Maintaining knowledge about temporal intervals", Communications of the ACM, vol.26, 1983, pp. 832-843.

[29] R. Stark and A. Pförtner, "Integrating ontology into PLM-tools to improve sustainable product development", CIRP Annals - Manufacturing Technology, vol. 64, Issue 1, 2015, pp. 157-160.

[30] R. Stark, H. Grosser, B. Beckmann-Dobrev and S. Kind, "Advanced Technologies in Life Cycle Engineering", Proc. Of the 3rd International Conference in Through-life Engineering Services. Cranfield, UK: Elsevier, 2014 pp. 3-14.

[31] R. Stark, F.-L. Krause, C. Kind, U. Rothenburg, P. Müller, H. Hayka and H. Stöckert, "Competing in Engineering Design – the Role of Virtual Product Creation", Journal of Manufacturing Science and Technology Manuscript, Elsevier Verlag, 2010 pp. 157-184.

TABLE I MAIN TEMPORAL RELATIONS BETWEEN LIFECYCLE PHASES AND STAGES: A CRISP MODEL

Notation	Relations and Their Inversion	Illustrations	Examples
r_1	Phase *a* is performed before (precedes) phase *b*		Detailed Design phase precedes Maintenance phase
r_2	Phase *b* is performed later (follows) phase *a*		Maintenance phase follows Detailed Design phase
r_3	Phase *a* immediately precedes (is adjacent to) phase *b*		Preliminary Design phase immediately precedes Basic Design phase
r_4	Phase *b* immediately follows phase *a*		Basic Design phase immediately follows Preliminary Design phase
r_5	Phase *a* partially overlaps with phase *b*		Detailed Design phase partly overlaps with Production Planning phase
r_6	Phase *b* partially overlaps with phase *a*		Production Planning phase partly overlaps with Detailed Design phase
r_7	Phase *a* lies inside stage *s*		Maintenance phase lies inside Usage stage
r_8	Stage *s* comprises phase *a*		Usage stage comprises Maintenance phase
r_9	Phase *a* lies inside stage *s*, so that their starting points coincide		Production Specification phase lies inside Development stage, so that their starting points coincide
r_{10}	Stage *s* comprises phase *a*, so that their starting points coincide		Development stage comprises Production Specification phase, so that their starting points coincide
r_{11}	Phase *a* lies inside stage *s*, so that their endpoints coincide		Removal from Usage phase lies inside Usage stage, so that their endpoints coincide
r_{12}	Stage *s* comprises phase *a*, so that their endpoints coincide		Usage stage comprises Removal from Usage phase, so that their endpoints points coincide
r13	Phase *a* coincides with phase *b*		Detailed Design stage coincides with Basic Design and Work phase

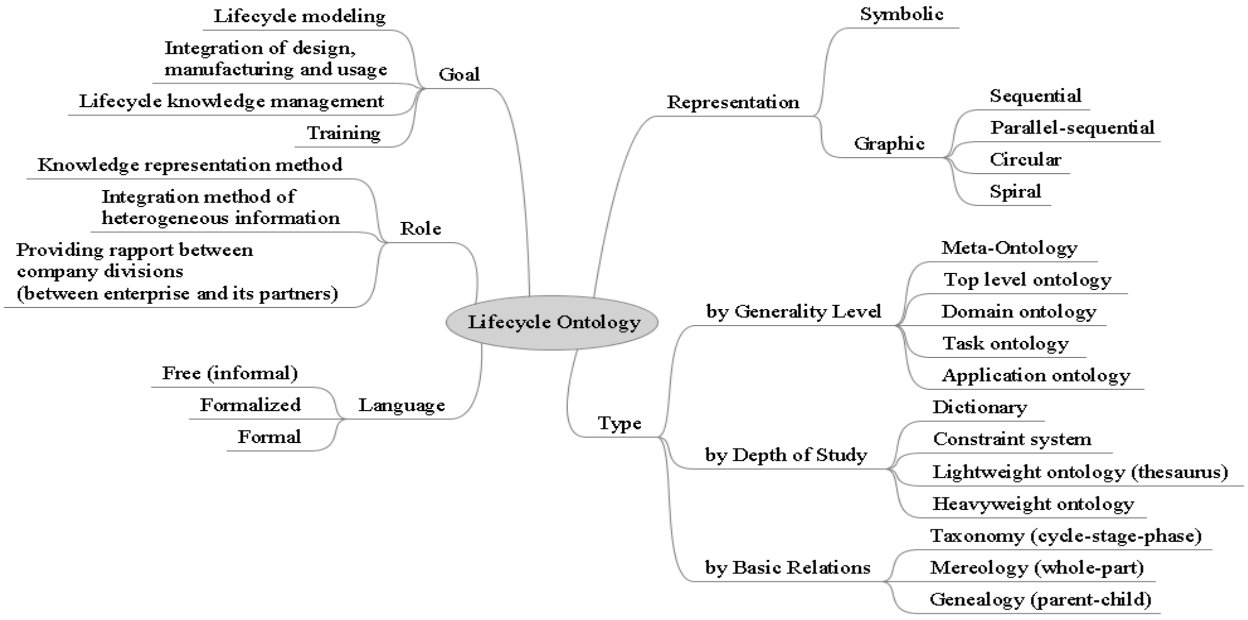


Figure 5. General Lifecycle Representation by a Mind Map