An overview of Conceptual Analysis and Design

Dmitry E. Borisoglebsky

d.borisoglebsky@ucl.ac.uk

University College London (Department of Civil, Environmental and Geomatic Engineering, Infrastructure Systems Institute), London, UK

Abstract

Conceptual Analysis and Design (mCAD) is an information and cognitive technology for knowledge and systems engineering. A conceptual system for a complex knowledge domain contains thousands of linked concepts, necessary in the engineering and management of big and complex systems. Naturally evolved conceptual systems usually contain conceptual gaps and have multiple logical fallacies. mCAD addresses these issues by axiomatic deduction of concepts. This article is a concise overview of Conceptual Analysis and Design, covering its foundations, technological aspects, and notable applications.

Keywords

axiomatic method, complexity, knowledge engineering, set theory, systems engineering

1 Introduction

Conceptual Analysis and Design (mCAD, where 'm' denotes mathematics) is an information and cognitive technology for analysing and designing complex objects in various knowledge domains (Kuchkarov Z. A., 2006). A knowledge domain is defined with a conceptual theory, while a complex object is designed via a conceptual model within the respective theory. It is performed through a sequence of substantial and formal acts on objects' definitions and symbols. Formal acts are performed with a mathematical derivation of symbolic structures from symbolic structures that are tightly coupled with substantial definitions. mCAD was initially developed and primarily used for analysing and designing organisational management systems for big organisations.

mCAD combines (i) semiotics for mapping objects to their symbols and definitions, (ii) set theory for symbolic manipulations on objects, including axiomatic apparatus and theory of structures by Nicolas Bourbaki (Bourbaki N. , 2004), (iii) conceptual and mathematical logic for substantial and formal manipulations on meaning, (iv) hypothetico-deductive approach common to natural sciences for theory development and synthesis of new knowledge, and (v) model-driven engineering for purposeful engineering of new objects. These foundations are supported by an extensive methodology and applied in numerous engineering projects.

The key motivation for mCAD is associated with the complexity of artificial objects and management systems established for producing and maintaining these complex objects, *e.g.*, aircrafts produced in series and operated for several decades. Repeatability and maintenance require comprehensive documentation; if printed, its weight would be several dozen of metric tonnes. According to initial estimations (Nikanorov, 2010), a complex knowledge domain is defined by a conceptual system having tens of thousands to millions of concepts, $[a \cdot 10^4, b \cdot 10^6]$ concepts. These concepts form a graph of derivations having twenty to forty levels of derivation from the foundational concepts to the most derived concepts, $[20, 40]$ maximum depth of derivation.

Assuming natural degradation of meaning if natural language is used for derivation, exponential degradation of the initial meaning can be assumed. The simplest example of a single thread of derivation of meaning (m) having multiple derivation steps (s) under an assumption of a constant degradation (k) can be modelled with an exponential decline model, $m(s) = M \cdot e^{-k \cdot s}$. Assuming the (amount of) initial meaning of the fundamental concept of 1, $M = m(0) = 1$, and degradation coefficient of 0.03, $k = 0.03$, this model is partially visualised with the following chain:

$$\boxed{m(0) = 1.000} \xrightarrow{s=1} \boxed{m(1) = 0.970} \xrightarrow{2} \cdots \xrightarrow{20} \boxed{0.549} \xrightarrow{21} \cdots \xrightarrow{40} \boxed{0.301} \xrightarrow{41} \cdots \xrightarrow{80} \boxed{0.091}.$$

Even with this very conservative degradation coefficient, twenty levels of derivation degrade almost half of the initial meaning, forty levels leave with a third, and eighty levels leave with one-tenth of the initial meaning. Derived concepts often have more than one foundational concept; thus, a faster degradation rate is expected. Utilising formal apparatus tightly coupled with semiotics, mCAD allows to maintain the original meaning through the derivation of concepts.

mCAD is analogous to complex logic (Zinov'ev, 1973), structuralist theory of science (Balzer & Moulines, 1996), applied category theory (Fong & Spivak, 2019), applied topology (Ghrist, 2014), and formal languages for software specification (Bjørner & Henson, 2008). mCAD shares a comprehensive methodology with complex logic and set theory with the structuralist theory of science. Applied category theory uses alternative fundamental mathematics to set theory of mCAD, so as applied topology. Specification languages are designed for formal model-driven engineering of software systems. While mCAD and its applications are defined in many publications (Sorokin & Shalyapina, 2008), this article is the first concise overview of mCAD and the first publication in the English language about this mature technology for complex knowledge engineering. This article contributes to knowledge sharing, comparative analysis of methodologies, and overall stimulation of formal methods for research and engineering.

One article cannot fully describe Conceptual Analysis and Design. Nevertheless, it is possible to introduce its key ideas grouped into foundations, methodology, and applications, forming respective sections. These sections are interrelated, and while the author attempted a sequential introduction to mCAD, later sections are beneficial for understanding earlier sections.

2 Foundations

2.1 Ontology

Conceptual Analysis and Design has been initially developed and is usually applied to the analysis and design of organisational management systems. The following statements are generalised versions of statements initially written within the organisational management context (Kuchkarov Z. A., 2006):

1. Objects are substantial, yet their analysis is constrained for practical reasons.
2. Knowledge domains are infinitely diverse, yet their definitions are finite; domains are interrelated; the number of domains is growing, and each domain is expanding.
3. Objects are manipulated within the defined borders of the respective domains.
4. Cognition is a combination of substantial and formal acts.
5. A domain-independent cognition utilises domain-independent concepts; thus, this cognition would benefit from an expansion of domain-independent concepts.
6. An axiomatic theory supports the axiomatic deduction of new concepts; thus, a method and methodology of the axiomatic deduction would be beneficial for domain-independent cognition.

These statements form requirements for the apparatus of conceptual analysis and design; therefore, any axiomatic apparatus that satisfies these requirements is suitable for mCAD (Kuchkarov Z. A., 2006). Mathematical, ontological, and symbolic groups of apparatuses were identified; each type has more than one apparatus that at least partially satisfies the requirements.

Mathematical apparatuses form different viewpoints on ideas. Category theory, Bourbaki’s theory of structures, and topos theory have a general language for definition, comparison, and synthesis of mathematical theories.

Ontological apparatuses fix different aspects of the world. Set theory is based on the concept of ‘quality’, while algorithm theory is based on the concept of ‘operation’.

Symbolic apparatuses formulate a symbolic description of the world. Predicate logic and theory of types define symbol manipulation.

These apparatuses fit the requirements but they are not equal (Kuchkarov Z. A., 2006). A generic and ontologically neutral apparatus for conceptual analysis and design does not exist. In the 1970s, Spartak P. Nikanorov (Nikanorov, 2010), the founder of mCAD, selected *predicate logic*, *set theory* and *theory of structures* as symbolic, ontological, and mathematical apparatuses for mCAD. This selection binds conceptual analysis and design yet, at the same time, enables research and engineering.

Systems theory is widely used for conceptual analysis and design as a domain-independent ontology because it contains domain-independent abstractions helpful in modelling complexity. Systems ontology is built on the assumption (Zinov'ev, 2006) of elementary objects; these objects are fully definable and are used to construct compound objects. Therefore, systems analysis and design assume (Nikanorov, Nikitina, & Teslinov, 2007): a systemically definable object, equality between the object and its systems model, and an ability to evaluate the validity of a systems model.

2.2 Semiotics

The semiotic triangle is formed with relationships between an *object* and its *sense* and *symbol*. These elements were introduced in 1892 (Frege, 1948) by one of the founders of mathematical logic, Gottlob Frege, and visualised by Charles K. Ogden and Ivar A. Richardson in 1923 (Ogden & Richards, 1923), see ***Fig. 1****.i*. Conceptual logic (Voyshvillo, 1989) utilises *terms* for symbols and *concepts* for sense, allowing to form another version of a semiotic triangle, see ***Fig. 1****.ii*. Version (ii) is more concrete because it incorporates an approach to defining sense through a collection of qualities named *concept*. A quality of an object is anything that allows selecting objects and forming a group of objects (Voyshvillo, 1989). mCAD has its version of a semiotic triangle (Kuchkarov Z. A., 2006) (see ***Fig. 1****.iii*), where qualities and operations of constructs are used as foundations of concepts.

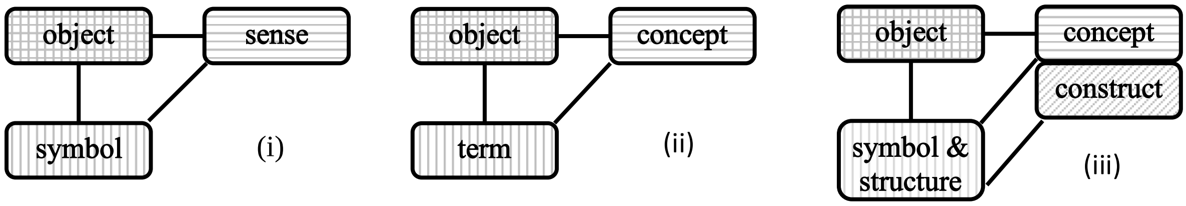


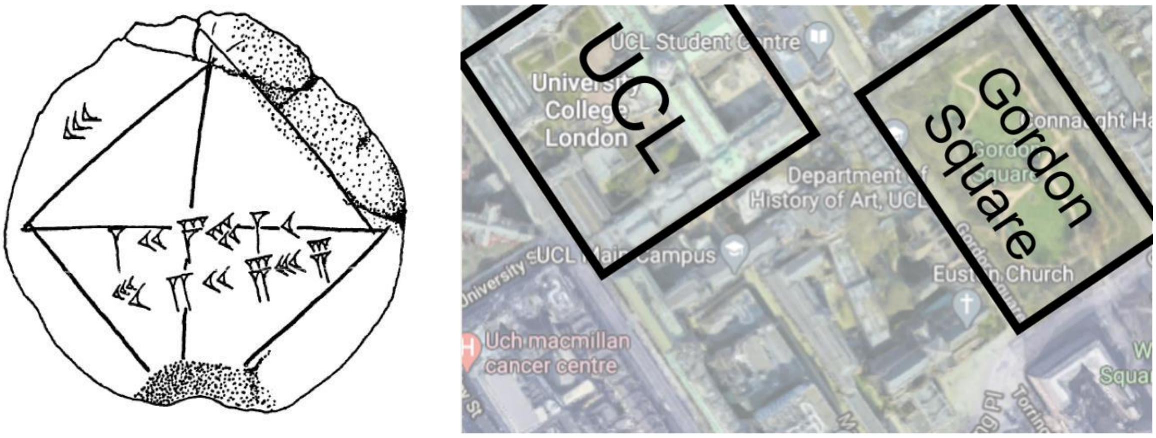
Fig. 1 *Semiotic triangle (i), with sense via concept (ii), and construct-based concepts (iii)*

Ontology is substantial and thus infinitely diverse (Kuchkarov Z. A., 2006); however, conceptual definitions are finite though growing in numbers. Some conceptual definitions share idealised universal structures that do not depend on language and culture, thus having unambiguity and allowing a complete transfer of sense through generations. These idealised universal structures are called *constructs* (Ivanov, Nikanorov, & Garayeva, 2008). Likely, numbers were the first group of constructs, followed by geometric figures. Mathematics, physics, and other disciplines have hundreds of constructs; however, constructs are yet to be defined in biology and social sciences (Ivanov, Nikanorov, & Garayeva, 2008).

The *rectangle* is an intuitively understood geometric construct (see ***Fig. 2***), which has existed and been used for four millennia. It is used in agriculture, construction, manufacturing, electronics, and other domains. The *rectangle* construct has two *length* qualities for height and width, an *angle* between *lengths*, and a derived *area* quality; the *lengths* and *area* qualities are mathematically mapped. These qualities indicate how substantial objects with a rectangular shape can be described and changed. Also, if one adds a third *length* to the *rectangle*, it is possible to get a 3-dimensional *rectangular prism*, adding *volume* and other qualities to this new construct. This example illustrates that constructs may be used to synthesise other constructs.

A construct is domain-independent; it is neither a theory nor a model (see Section 2.5) because both are domain-dependent; however, constructs' application to a domain can be a theory or a model. Constructs only define ontological universalities (Nikanorov, 2008). A construct may have many forms that can maintain and translate ideal ontological universality.

Interpretation of the semiotic triangle in mCAD (see ***Fig. 1****.iii*) supports a sufficient coverage of substance both in terms of qualities and operations, knowledge transfer from one person to another, preservation of knowledge, synthesis of new concepts, and transferability of the skill of synthesis of new concepts. These characteristics are valuable to science, engineering, and education.



(a) (b)

Fig. 2 *Construct 'rectangle' and some of its' applications: 'Theorem of Pythagoras' on a Babylonian tablet 1900 to 1600 BCE* (Yale Peabody Museum Babylonian Collection, 2022), *and a building and park on a map* (Google Maps, 2022)

2.3 Mathematics

Nicolas Bourbaki created and used the theory of structures and axiomatic apparatus for the definition and synthesis of mathematical theories. Bourbaki defined axiomatic deduction in the first chapter and mathematical structures in the fourth chapter of his book Set Theory (Bourbaki N. , 2004); the key ideas and terminology are published in an article (Bourbaki N. , 1950). While Nicolas Bourbaki used structures for axiomatic deduction of mathematical theories, mCAD uses structures for axiomatic deduction of conceptual theories of various knowledge domains.

Apparatus of structure genera is based on Bourbaki's theory of structures and axiomatic deduction (Bourbaki N. , 2004). Analysis and design of complex technical systems require exact manipulation of complex conceptual structures, which is only possible with operations on explicit axiomatic theories (Kuchkarov Z. A., 2006). The system of operations on structure genera was defined in 1972 (Nikanorov, 2010), including crucial operations of synthesis of structure genera. Later, the apparatus of structure genera was translated (Ponomarev, 2007) to Zermelo–Fraenkel set theory (see Section 3.4); currently, this dialect is the main mathematical apparatus of mCAD.

A set can be defined extensionally via an enumeration of objects (Shreyder, 1971); this covers (-object—symbol-) elements of the semiotic triangle. A set can be defined intensionally as logical qualities of objects forming this set (Shreyder, 1971); this covers (-object—concept-) elements of the semiotic triangle. Together, extensional and intensional definitions cover (-object—symbol—concept-), the elements and relationships of the complete semiotic triangle.

Apparatus of echelons was developed to work with diversities of objects and their relationships (Nikanorov, 2010). Initial evaluations (Nikanorov, 2010) showed that this apparatus is suitable for analysing and designing conceptual systems having thousands of concepts on hundreds of levels of abstractions (echelons, see ***Fig. 6***). For example, a definition of a manufacturing organisation includes 600 basic sets (Lelyuk, 2009) (see Sections 3.4 and 4.1). Apparatus of echelons is mentioned only in this paragraph of this article.

2.4 Cognition

mCAD integrates several cognitive approaches (Kuchkarov Z. A., 2006) addressing the meaning and its symbolic representation (see Section 2.2). *Aspect*-, *attribute*-, and *normative*-based approaches address the meaning and are associated with intensional definitions of sets. Manipulation of symbolic structures is associated with the *formal* apparatus of structure genera (see Section 2.3). Additionally, *explication* covers translating meaning from less into a stricter form. Finally, the *synergy* of these approaches allows the selection of the right approach for the task and transition from one approach to another.

Aspect-based cognition utilises the concept of an aspect (or quality). The International System of Units (Taylor, 2001) provides multiple examples of aspects widely used in science and engineering, e.g., *length*, *time*, *area*, and *speed*. The set-theoretic definition (Kuchkarov Z. A., 2006) of an aspect is based on the set-theoretic concept of a power-set. Given a set $X = \{x_1, x_2, x_3, \cdots\}$, an aspect is an element of its power set $\mathfrak{P}(X)$ (see ***Table 3***), *e.g.*, $\{x_1, x_3\}$ or $\emptyset$, or $\{x_1, x_2, x_3\}$. Because of the mapping between extensional and intensional definitions of a set, each meaningful element of the power set must have at least one quality that defines the selection of the respective elements. Aspect-based cognition is useful for selection, differentiation, comparison, abstraction, and concretisation (see Section 3.2).

Attribute-based cognition (Nikanorov, 2010) utilises an integral characteristic of an object and an obligatory part of its definition. For example, the process (see Section 4.1) has three attributes: *input*, *output*, and *change.* Attribute-based cognition is useful for deduction, axiomatisation, convention, conceptualisation, and postulation (see Section 3.2).

Normative-based cognition utilises standard units of cognition. In mCAD, systems' normatives are essential, *e.g.*, *object*, *relationship*, *function*, and *process*. Systems' normatives allow rapid and accurate domain-independent cognition. Geometry's normatives support the cognition of engineers (Nikanorov, 2008); for example, a house may be treated as a *rectangular prism* (see ***Fig. 2***).

Formal cognition uses a mathematical apparatus on symbols and structures of symbols (see Sections 2.3 and 3.4).

Explicit cognition transfers definitions into a stricter form. Because of the diversity of kinds of definitions, several kinds of explicit cognition exist (Kuchkarov Z. A., 2006); for example, attributive, set-theoretic, attributive-set-theoretic, and structure genera.

Synergetic cognition allows the selection of the most suitable cognition for the task. The key characteristics of different kinds of cognition form **Table 1**

Table 1 Cognition in conceptual analysis and design

Name	Unit of cognition	Examples
aspect	characteristic used for objects' selection	$\mathrm{length}, \mathrm{colour}$
attribute	an integral characteristic of an object	$\mathrm{input}, \mathrm{change}, \mathrm{output}$ of a process
normative	normative is a standard unit of cognition	see below, *e.g.*, function
systemic	systems' normatives	$\mathrm{function}, \mathrm{process}, \mathrm{system}, \cdots$
formal	symbols and structures of symbols	$a, \alpha, (a, b), \{\alpha, \beta, \cdots\}, \cdots$
explication	(see Section 3.4)	$rectangle \xrightarrow{explication} \square$
convergent	seamless use of different kinds of cognition.	a mix of samples above

mCAD changes the cognition of the practitioner. This change is similar to the change of people after an in-depth understanding of first-order logic; however, it is more notable because of the several kinds of cognition in mCAD and their synergy. Decades of graduate and postgraduate education (MIPT, 2021; MIPT, 2022) followed by internship (CONCEPT, 2021) support the claim of the transferability of mCAD-specific cognition.

2.5 Methods

Aleksandr A. Zinov'ev developed Complex Logic to work with knowledge's logical, ontological, and epistemological aspects. It was used as the foundation for a Logical Theory of Scientific Knowledge (Zinov'ev, 1973); the latter became the foundation for Logical Physics (Zinov'ev, 1983) and Logical Sociology (Zinov'ev, 2006). He defined several (epidemiological) methods relevant to Conceptual Analysis and Design. Short introductions to these methods are given in the following paragraphs: the first paragraph in a group introduces Zinov'ev's definition, while the following paragraph discusses specifics of the respective method in mCAD.

Systemic method (Zinov'ev, 2006). A knowledge domain consists of elementary objects in a spatial-temporal space. Elementary objects cannot be divided. They exist, replicate, and interact with one another. A multiplicity of elementary objects and interactions makes elementary objects identical, thus indistinguishable. Essential and sufficient qualities for grouping are considered, and their other qualities are disregarded. Intra-group interactions could be of one or multiple types that enable the characterisation of elementary objects and their abilities to act. Compound objects are clusters of two or more elementary objects and could form groups and groups of groups having measurable ranks. Fundamentally, any quality of an elementary object is measurable.

mCAD utilises set theory and systemic method (Kuchkarov Z. A., 2006). A set consists of elements (Stoll, 1961) similar to one another (for their inclusion in a set) yet distinguishable (for they are different elements). Both undividable elements and dividable sets may form a set; this enables the codification of structures, including the codification of systemic relationships. Collections of elements and relationships enable the codification of higher-order elements composed of lower-order elements. Any conceptual theory or model is systemic, and conceptualised systems theory (see Section 4.1) is the most used theory in mCAD.

From the abstract to the concrete (Zinov'ev, 2006). Substances of concrete (real) objects and abstract (ideal) objects are different even if the same objects are reviewed from these perspectives (Zinov'ev, 2006). Mapping between the concrete and the abstract perspectives contributes to the integrity and validity of research aiming to deliver a structured understanding of an object. Such research is bidirectional, from the concrete to the abstract and from the abstract to the concrete. A researcher attempts to abstractly define a variety of concrete facts in the former case and validates definitions of various concrete facts in the latter case. Mental analysis of objects and knowledge synthesis happen during the ascent from the abstract to the concrete. The ascent in mCAD has two aspects.

Firstly, because of the duality of a set having both an extensional (via enumeration) and intentional (via qualities) definitions as well as operations of analysis, synthesis and explication of structure genera (see Sections 3.2 and 3.4 and ***Table 4***). Secondly, through a language of explications and operations for analysis and synthesis (see Sections 3.2 and 3.4), mCAD supports the synthesis of new terms (see ***Table 4***) from existing terms and the synthesis of collections of terms.

Hypothetico-deductive method (Zinov'ev, 2006). A researcher makes some hypotheses about the objects of research. These hypotheses are abstractions formed by including or excluding objects' qualities. Hypotheses might be empirically indeterminate or unverifiable yet support valuable deductions. The researcher's intentions determine hypotheses selections. Intentions are unrelated to truth-values and cannot be confirmed nor denied using logic.

mCAD is a hypothetico-deductive method used in scientific research of theories and engineering design of models (Kuchkarov Z. A., 2006). Intentions of the conceptual analyst and designer and hypotheses selection are determined by the project objective and knowledge domain. Explication (see Section 3.2 and ***Table 4***) highlights the capabilities of mCAD for the mathematical deduction of terms and concepts. An application of this method for concept design is shown in Section 4.1 and ***Table 5***.

Theory (Zinov'ev, 1973; Zinov'ev, 2006). The theory is a method for the acquisition of new knowledge. If a knowledge domain and a collection of statements frequently support the deduction of true statements, then this collection is a theory. Statements (and terminology) could be initial (primitive) or derivative. Primitives are taken as given, while derivatives are obtained from primitives.

mCAD uses the hypothetico-deductive method for theory design (Kuchkarov Z. A., 2006) via an apparatus of mathematical deduction. Initial sets (and terms and concepts, see ***Table 5***) and initial relationships among sets taken as given and form the theory core. Explication and other operations are used to deduce derivative terms and relationships that form the theory body. The core is finite, while the body potentially has a countable infinity of terms. An example of a conceptual theory is shown in Section 4.1.

If conceptual theories are categorised on a scale, several classes are formed and characterised as follows (Kuchkarov Z. A., 2006): (a) micro-theory having 3—6 relationships and 2—4 levels of concretisation; (b) meso-theory having 5—20 levels of concretisation; (c) hyper-theory having more than 20 levels of concretisation. For example, a morphological relationship (see ***Table 5*** and respective text) is a meso-theory, while the theory of socio-economic systems (see Section 4.4) is a hyper-theory. Currently, mCAD is a technology for the analysis and design of micro- and meso-theories.

Model (Zinov'ev, 1973; Zinov'ev, 2006). A model is an object of a different class to the original object, each belonging to a different class of objects. A model is designed so findings could be valid for the respective original. Usually, modelling is used if an original object is yet to exist or is unfeasible for research purposes for practical reasons, *e.g.*, monetary, temporal, or availability.

mCAD is used to design and apply conceptual models (Kuchkarov Z. A., 2006) usually based on system-theoretic constructs (see Section 4.1.). A knowledge domain is defined by a conceptual theory later used for the conceptual modelling objects in this domain. A construct maps the respective object-model to the object-original (see ***Fig. 1***, ***Fig. 3***, Section 3.3). An abstract or a concrete object can be used for scientific research or engineering design.

3 Technology

3.1 Substance

mCAD is used in projects for the analysis and design of knowledge domains. Objects are described using several definition forms: set-theoretic (*element, set, universe*) for their symbolic representation, conceptual logic (*concept, theory*) for definitions, and construct-based (*construct, a graph of constructs*) for construct-based analysis and design. These forms are shown in ***Fig. 3***, where vertical lines indicate affiliation and horizontal lines indicate the mapping between forms of objects.

A set can be defined extensively with an enumeration of objects (Shreyder, 1971). Extensional definitions show the mapping between objects and the universe of all objects bounded by the knowledge domain and project objectives. Objects of the knowledge domain (box fill) and set-theoretic (vertical stripes) columns in ***Fig. 3*** map objects with their set-theoretic symbolic representation, *e.g.,* an object is related to a set-theoretic element, and a group of objects is related to a set (see the semiotic triangle in Section 2.2). mCAD utilises three types of semiotic pairs: object ↔ element, group of objects ↔ set, knowledge domain ↔ universe.

A set can be defined intensively through qualities (Shreyder, 1971) that allow selections of groups of objects from the universe. Intensional definitions show the mapping between ideas. Constructs (diagonal stripes) and concepts (horizontal stripes) columns are associated with intensional definitions. mCAD utilises three levels of abstraction: concepts as a collection of qualities or aspects, concepts as a collection of attributes, and concepts as normatives in a normative system (See Section 2.4). Attributes and normatives of systems theory are critical to conceptual analysis and design.

The concept "construct" was developed in logic (Nikanorov, 2008) and represented a kind of idealised meaning that could have symbolic and structural representation. Examples of constructs are natural numbers, geometric figures, algebraic operators and operands, function, and process. Constructs in mCAD have the following functions: (1) consolidation of a refined knowledge, (2) definite translation of the meaning in communication, and (3) definition of qualities and operations on objects.

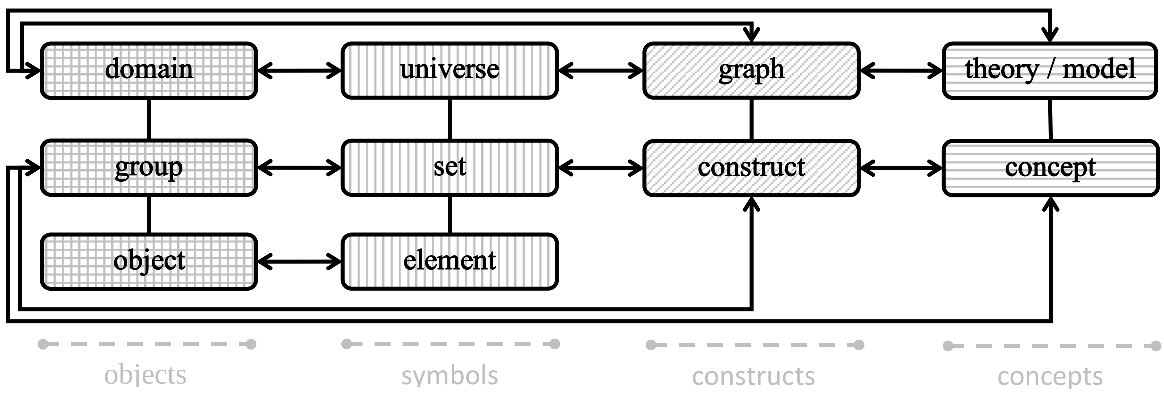


Fig. 3 *System of substantial and formal elements of mCAD*

Conceptual analysis and design is a technology that utilises axiomatic apparatus on constructs for information and knowledge engineering.

The concept is a term reserved by logic for the substantial meaning of an associated term (see ***Fig. 1***.*ii*). A *conceptual system* can be assembled from concepts as elements because of substantial relationships between concepts. A *conceptual model* is a conceptual system that is used instead of the respective original object. A *conceptual theory* is a conceptual system that defines the respective knowledge domain and can be used to deduce new domain knowledge (see Section 2.5). In mCAD, concepts are based on constructs; most of these constructs are axiomatically deduced. If a conceptual system is sound and addresses the diversity of objects of the respective project, then this conceptual system is fit for theoretical studies and model-driven engineering.

3.2 Operations

mCAD has been mostly used for the analyse and design management systems in big organisations. This shaped mCAD operations that use elements from ***Fig. 3*** as operands. ***Table 2*** introduces 21 operations; this list is incomplete as some operations while maintaining the same meaning, have operand-specific versions excluded from this paper. Further development of mCAD, *e.g.*, via redesign of ***Table 2*** for ***Fig. 3***, may change the system of mCAD operations.

Table 2 *Operations of Conceptual Analysis and Design* (Kuchkarov Z. A., 2006)

No	Name	Meaning
1	Differentiation	select object's qualities present in one object and absent in another
2	Comparison	map qualities present in both objects, naming qualities that are absent in either of the objects
3	Abstraction	define a new concept by the selection or exclusion of qualities
4	Concretion	opposite to abstraction, the inclusion of qualities
5	Deduction	form a derivation/deduction graph of concepts
6	Axiomatisation	select or define the theory's conceptual core (Theory in Section 2.5))
7	Convention	an agreement that a concept is intuitively defined, or an agreement on the conceptual systems' boundaries
8	Schematisation	nominal conceptualisation that does not cover the substance of a knowledge domain
9	Conceptualisation	substantial conceptualisation using domain-specific constructs
10	Postulation	define the respective knowledge domain using normatives
11	Reconstruction	definition of the respective knowledge domain with existing constructs
12	Formalisation	symbolise each concept and relationship in a formal language
13	Explication	re-write definition in a stricter language
14	Interpretation	map constructs to objects of the knowledge domain
15	Synthesis	generate a new conceptual system from existing systems using the conceptual apparatus of mCAD
16	Expansion	derive new concepts from existing concepts using the mathematical apparatus of mCAD
17	Operationalisation	translate a conceptual system into theoretical-model relationships
18	Normalisation	translate the respective conceptual system into: * conceptual theory for a management object * conceptual theory of a goal-oriented system * other system-theoretic classes, *e.g.*, adaptive system, growing system, developing system
19	Functionalisation	translate the respective conceptual system into a functional model
20	Methods' selection and design	define or select methods for the functional model using function-methods relationships
21	Documentation	document project using a selected symbolic system

Section 4.1 illustrates applications of several operations from ***Table 2*** on elements from ***Fig. 3***. An algorithm for the application of these operations does not exist. Conceptual analysis and design is an iterative process. For example, if a functional model is sufficient for its project, then an analyst would utilise system-theoretic constructs for functional modelling, while other projects would require the development of constructs for a new domain. Conceptual Analysis and Design has several functions (Nikanorov, 2008):

1. Define the project substance and constraints.
2. Define the situation of the project.
3. Define the respective knowledge domain.
4. Postulate a conceptual theory or model that covers objects of this project domain.
5. Formulate a conceptual model for this project domain.
6. Normalise the conceptual model.
7. Operationalise the normalised model.
8. Define metrics for the operationalised model.
9. Select and design methods.
10. Define a meta-theoretic model description of the solution.
11. Evaluate the capabilities of end-uses.
12. Document.

3.3 Modelling

Modelling is an integral part of Conceptual Analysis and Design. Conceptual models define mental models of objects in the respective knowledge domain. Conceptual models could be based on different constructs, *e.g.*, morphological models (see Section 4.1), function-based models, and process-based models, which are static deterministic models. An mCAD model may have two kinds of applications. Imitational models are used in studies of the as-is objects these models imitate, and projective models are used in designing the to-be original objects. Any study incorporates an element of projection, and any projection incorporates an element of study (Kuchkarov Z. A., 2006). Multiple models could be created and used in one mCAD project (see ***Fig. 4***).

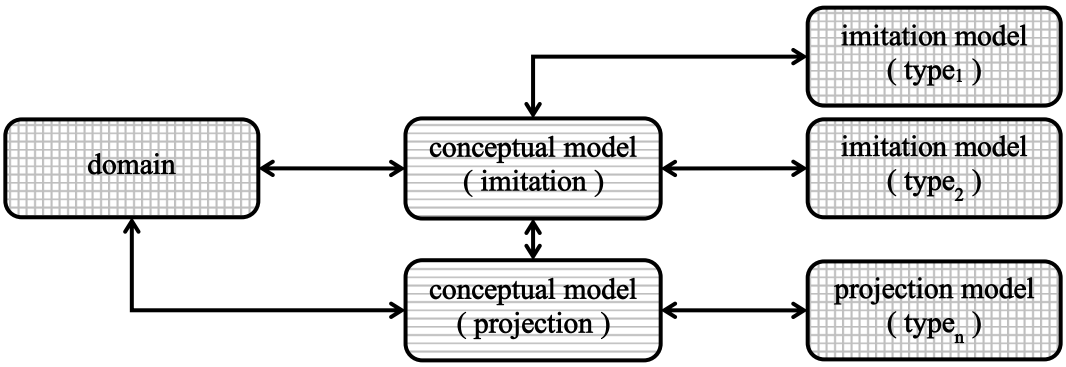


Fig. 4 *Imitational and projective models in mCAD*

3.4 Explication

The system of operations on set-theoretic structures was defined in 1972 (Nikanorov, 2010). Later, a language for explication of species' structures was translated (Ponomarev, 2007) to Zermelo–Fraenkel set theory. This language uses alphabets of predicate logic, set theory, and mCAD-specific operations. Currently, it is the primary mathematical language of mCAD.

Predicate logic uses (Bergmann, Moor, & Nelson, 2014) logical operators and quantifies predicate processing. Predicates are symbolic statements that include individual variables and constants. The alphabet of predicate logic consists of sentences ($A, B, \dots, Z, A_1, \dots, Z_n$), constants ($a, \dots, v, a_1, \dots, v_n$), variables ($w, x, y, z,\ w_1, \dots, z_n$), predicates ($Aa, Awx, \dots, Z_n z_2 b_1$), connectives (¬, ∧, ∨, ⇒, ⇔), quantifiers (∀, ∃), and punctuation (*"(", ")"*).

Set theory (Halmos, 2017; Enderton, 1977) defines a set as a collection of objects in X that could be written in figure brackets, $X = \{x_1, x_2, \dots\}$, or as qualities present in all elements of a set, $X = \{objects\ with\ a\ property\}$, extensional and intentional definitions, respectively. Set A is in B (or $A \subseteq B$) if all elements of A are also elements of B. Other concepts of set theory are introduced in ***Table 3***. Predicate logic is utilised in set theory for intensional definitions of sets.

Out of 29 elements forming the language of explication of species structures, ***Table 4*** contains a subset of elements necessary for reading statements on morphological relationship introduced in ***Table 5***.

Table 3 *Some concepts of set theory; given sets A and B* (Halmos, 2017; Enderton, 1977)

Name	Symbol	Use
Empty set	$\emptyset$	The empty or null set contains no element; it is a subset of any set.
Union	$A \cup B$	$A \cup B = \{x \mid x \in A \vee x \in B\}$, read as the union of sets *A* and *B* is a set of *x* such that *x* in *A* or *x* in *B*.
Intersection	$A \cap B$	$A \cap B = \{x \mid x \in A \wedge x \in B\}$, read as the intersection of sets *A* and *B* is a set of *x* such that *x* in *A* and *x* in *B*.
Difference	$A \setminus B$	$A \setminus B = \{x \mid x \in A \wedge x \notin B\}$, read as the difference of sets A and B is a set of x such that x in A and x not in B.
Complement	A'	All elements not in *A*
Power	$\mathfrak{P}(A)$	$\mathfrak{P}(A) = \{X \mid X \subseteq A\}$, power of set *A* is a set of all subsets of *A*, including itself and the null set.
Product	$A \times B$	$A \times B = \{(x, y) \mid x \in A \wedge y \in B\}$, a Cartesian product is a set of ordered pairs $\langle x, y \rangle$ such that *x* in *A* and *y* in *B*.
Function	$F: A \to B$	A function is a relation *F* such that for each *a* in domain *F*, there is only one b such that aFb.

Table 4 *Some elements of the language for explication of species of structures* (Ponomarev, 2007)

Element	Syntax, example	Use
Global identifier	X_1 – basis set, S_1 – species of structure, D_1 – term, A_1 – axiom, T_1 – theorem, F_1 – term-function	Reference to global ID's value
Declaration of species structure	$S_1 ::= echelon(X_i, S_i, D_i)$	Declaration of structure (see ***Fig. 6***)
Declaration of a global statement	$A_1 ::= P(X_i, S_i, D_i)$	Declaration of a global statement (axiom) or a hypothesis (theorem)
Declaration of a term	$D_1 ::= S_1 \cap S_2$	Typified global value or a property that is deduced using a formal definition
Small projection	$pr_1\big((\xi_1, \xi_2)\big) = (\xi_1)$, $pr_{2,1}\big((\xi_1, \xi_2, \xi_3)\big) = (\xi_2, \xi_1)$.	Get a tuple of selected elements from a tuple
Big projection	$Pr_1(S_1) ::= \{\xi_1 \mid \exists (\alpha_1, \alpha_2) \in S_1\ \xi_1 = \alpha_1\}$.	Get a set of modified tuples from a set of tuples
Cardinality	$card(X_1)$	Get the number of elements in a set
Term-function	$F_1 ::= [\alpha \in \mathfrak{P}(X_1), \beta \in \mathfrak{P}(X_1)]\ \alpha \cap \beta$	Define a function that returns the respective set-theoretic statement
Term-function call	$F_1[S_1, S_2]$.	Computation of the function for given arguments
Recursive expression	$R\{\xi := S_1 \mid \xi \cup F_1[\xi]\}$,	Iterative computation of the expression

4 Applications

Conceptual Analysis and Design is a domain-independent technology for information and knowledge engineering. mCAD can be used for (Nikanorov, Nikitina, & Teslinov, 2007): definition of the knowledge domain, design of complex systems, analysis of alternatives, strategic planning, and other applications. Four notable examples are introduced in this section.

4.1 Systems

Systems theory is a research and engineering discipline created in the 1950s (Nikanorov, 2010). Systems theory is designed for and used as a generic language of various knowledge domains (Boulding, 1956), *e.g.*, mechanics, biology, sociology, and organisational management. It is assumed (Zinov'ev, 2006) that a knowledge domain consists of elementary objects in a spatial-temporal space; elementary objects cannot be divided; they exist, replicate, and interact with one another. However, systems theory development slowed down at the end of 1980 because of the absence of mathematical apparatus suitable for the complexities in systemic knowledge domains (Kuchkarov Z. A., 2006). Conceptual Analysis and Design allows terminology design of systems theory and uses these terms as normatives of systems thinking (see Section 2.4), thus supporting the analysis and design of big and diverse systems.

In mCAD, a system is a structured collection of objects. These objects and their relationships are observed from a viewpoint relevant to the respective project (Nikanorov, Nikitina, & Teslinov, 2007). The following characteristics are present in a system: (1) an object can be represented as a system; (2) research position allows systemic representation of this object; (3) equivalence of this object and its systemic aspect; (4) objects' holism.

System-theoretic constructs do not form a systems theory; these constructs are tools created for systems definitions (Nikanorov, 2008). These constructs' role is different to theory and model because constructs form an idealised and universal standard for cognition. System-theoretic constructs form two classes of constructs: *object constructs, e.g.*, object, statical systems, process systems); *subject constructs*, *e.g.*, subject, subject-object system, subject-subject system. Constructs are organised by their morphology (consists of/belongs to, see ***Table 4***). Some mCAD operations (see Section 3.2, ***Fig. 3***) use constructs as operands.

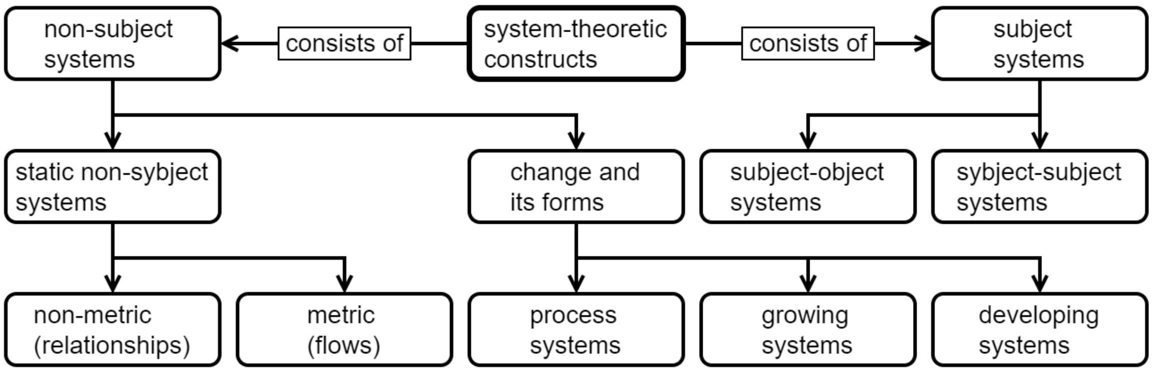


Fig. 5 *System-theoretic constructs*

Object is a thing whose symbol can be an element of a set (see *extensional definition* in Section 2.2; here and below, the primary literature are (Kuchkarov Z. A., 2006; Nikanorov, Nikitina, & Teslinov, 2007; Nikanorov, 2008)).

Aspect. If X is a collection of elements, $X = \{x_1, \ldots, x_n\}$, then an aspect is a subset A of X, $A \in \mathfrak{P}(X)$, (see intensional definition in Section 2.2, and its relationship to extensional definition).

Object-aspect relationship. An object of the world can be overviewed from many viewpoints; thus, an object may have many aspects relevant to different R&D objectives. Object-aspect relationships are defined for a project.

Subject is a pair of *subjectivity* and its *holder (person* or *organisation)*. $Subjectivity$ is a compound concept having $\langle choice, interest, capability, interest - capability\ relationship \rangle$ attributes.

Morphological relationship (MR). Represents the idea *consists of* or *belongs to*, *e.g.*, an object *consists of* other objects, or an object *belongs to* another object. MR has two axioms, (1) an object *consists of* two or more objects; (2) an object cannot *consist of* itself. Three major types of decomposed objects can be deduced, (a) a *final object* does not belong to another object, (b) an *intermediate object belongs to* another object and *consists of* other objects, and (c) an *elemental object* does not *consist of* other objects. This textual description is mathematically defined in ***Table 5*** and ***Fig. 6***. ***Table 5*** contains 11 concepts of the conceptual theory of *morphological relationship*. The first two terms identified by X_1 and S_1 form the core of this theory; other terms form the body of this theory, including final (D_3), intermediate (D_1), and elemental (D_4) objects.

Change is a relationship between two states of an object (Nikanorov, 2008).

Process is a method having the following structure $\langle inputs, change, outputs\rangle$, where every $input$ is $output$, every $output$ is an $input$, and $output$ cannot be an $input$ to the same $process$. Similarly to *morphological relationship*, these core concepts are used in the deduction of new concepts, a *hierarchical process* (if the *process* is used as an element of *morphological relationship*), *e.g.*, *preceding process* and *following process*, *pre*- and *post-shadowing processes* in a *process network.*

System is formed from a $process\ network$, while $input$, $output$, and $change$ are $systems\ objects$. $Problem$ is the difference between the $existing\ state$ and $desired\ state$. $Desired\ system$ is a $process\ network$ that provides the $desired\ output$. $Solution$ to the problem is implementing a system that provides the desired state, for example, by $changing$ the current system. A collection of all processes outside the system is called an external environment.

Table 5 *Morphological relationship as an example of mathematical deduction of meaning in mCAD.*

ID	Term	Species	Construct	Interpretation
X_1		$\mathcal{B}(X_1)$	object	
S_1	$\mathcal{B}(X_1 \times X_1)$	$\mathcal{B}(X_1 \times X_1)$	morphological relationship	an object belongs to an object
D_1	$Pr_1(S_1)$	$\mathcal{B}(X_1)$	intermediate object	objects that belong to other objects
D_2	$Pr_2(S_1)$	$\mathcal{B}(X_1)$	compound object	objects that consist of other objects
A_1	$card(S_1) = card(D_1)$	logical		any intermediate object belongs to one and only one object
D_3	$D_2 \setminus D_1$	$\mathcal{B}(X_1)$	final object	a final object does not belong to another object
D_4	$D_1 \setminus D_2$	$\mathcal{B}(X_1)$	elemental object	an elemental object does not contain other objects
F_1	$[\alpha \in \mathcal{B}(X_1)]$ $Pr_1(Fi_2[\alpha](S_1))$	$\mathcal{B}(X_1)$	direct composition of objects in a set	objects that belong to one of the objects in a set of objects
F_2	$[\alpha \in X_1]\ F_1[\{\alpha\}]$	$\mathcal{B}(X_1)$	direct composition	all objects that belong to an object
F_3	$[\alpha \in X_1]$ $R\{\xi := F_2[\alpha] \mid \xi \cup F_1[\xi]\}$	$\mathcal{B}(X_1)$	full composition	direct composition of an object and all objects that belong to direct composition of the full composition
A_2	$\forall \xi \in X_1\ \xi \notin F_3[\xi]$	logical		an object does not belong to its full composition

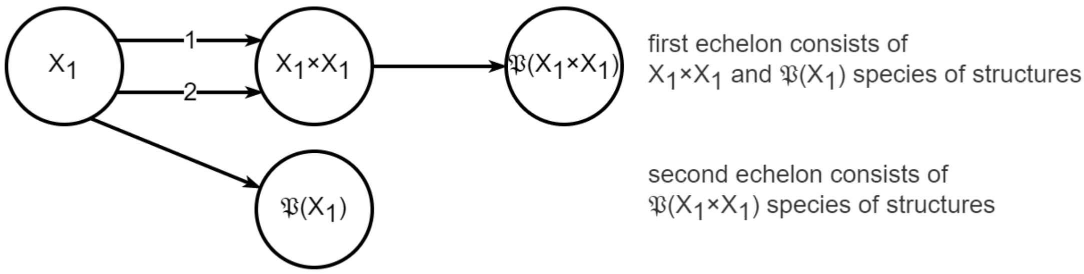


Fig. 6** M-graph for 'morphological relationship', referring to elements from **Table 5

Similarly to the $morphological\ relationship$ with ***Fig. 6*** and ***Table 5***, $process$, $system$, and other system-theoretic constructs have explicit mathematical definitions. A collection of system-theoretic constructs is suitable for conceptual modelling of complex systems.

4.2 Organisations

mCAD is mainly used for analysis and design of organisational management systems (CONCEPT, 2022). The conceptualisation of systems theory (see Section 4.1), design of operations and R&D methodology (see Sections 3.1 and 3.2), an establishment of graduate and postgraduate education (MIPT, 2021), and published journal papers and books (Sorokin & Shalyapina, 2008) cover aspects of management systems of big organisations.

An organisation's management system (Kuchkarov Z. A., 2006) is a collection of decision-making procedures on an object of management. Conceptual design of a management system is a kind of design centred around conceptual models (see Sections 2.5, 3.3, and 4.1) for sound and valuable decision-making on the holistic representation of objects of a project domain. Usually, the following conceptual models are created (Kuchkarov Z. A., 2006): organisational goals, functional model, process model, organisational structure, and methods model based on the functional model. Conceptual theories for these and other models are introduced in (Kuchkarov Z. A., 2006; Nikanorov, Nikitina, & Teslinov, 2007; Kuchkarov Z. A., 2004).

4.3 Legislation

Legislation and laws can be conceptually analysed and designed (Kuchkarov Z. A., 2006), which was proven by designing an industrial ecology code (Kuchkarov & al., 2015). At the publication date (Kuchkarov & al., 2015), industrial ecology was defined by more than 800 legal documents forming seven hierarchy levels. Analysis of these documents showed multiple logical absurds of several types: poorly defined terminology, absence of terminology, poorly defined procedures, contradictions, and absence of direct actions.

The legislation affects the economy. For example, transitioning from the current legal documents to the industrial ecology code could reduce a new business registration from 18 months to 8 months. This evaluation uses the synergy between the current and the proposed processes model. The latter means the mapping between legal documents and simulated entities of the respective knowledge domain, addresses to these legal documents.

This project showed: an ability to map legal documents to domain-specific entities; the presence of legal absurdities; and the ability to design concise legal code, which improves the efficiency of the regulations of the knowledge domain. Adaption of this practice would strengthen and simplify governance.

The industrial ecology code is a meso-theory (see Theory in Section 2.5) as it includes 101 conceptual models, 12 levels of deduction and more than 900 concepts. This project (Kuchkarov & al., 2015) allowed to define a mechanism for fixing the individual logical absurd, design a concise and logically sound ecology code, and create a legal documents design procedure.

4.4 Economy

The theory of socio-economics systems is a conceptual hyper-theory (see Theory in Section 2.5) and is the only theory with a published methodology of conceptual analysis and design of hyper-theories (Kuchkarov & Nikanorov, 2007). This theory illustrates mCAD capabilities for analysis and design of hyper-theories; however, hyper-theory design requires a different set of operations; also, the logical fallacy of circular definition is not absolute in the hyper-theory design of concepts. This theory includes concepts suitable for complex socio-technical systems modelling and integrating models developed in different socio-economic theories.

5 Concluding remarks

Concept Analysis and Design is a mature technology for knowledge engineering and complex systems analysis and design. mCAD converges several disciplines: semiotics, logic, mathematics, systems theory, and hypothetico-deductive reasoning. This convergence allows axiomatic deduction of conceptual systems of complex knowledge domains.

Spartak P. Nikanorov, the founder of mCAD, in his letter to Nicolas Bourbaki (French Academy of Sciences) (Nikanorov, 2010), wrote the following:

> *We managed to understand why the theory of echelons stays in the background of your work. It is a consequence of the fact that all modern mathematics can be represented in the first three echelons. Meanwhile, the mathematics of organisational management is determined by the echelons from the 10th to the 45th and above… Modern mathematics, coming from mechanics, cannot theorise psychology, sociology, economics, and history, the difference between which consists in a vast diversity and dynamics of concepts.*

mCAD can be developed further. For example, the algebraization of operations and substantial elements would be a significant step forward. Additionally, it could be beneficial to use category theory parallel to Bourbaki's structures and axiomatic apparatus. Set theory provides an 'internal' perspective to mathematical structures, while category theory provides an 'external' perspective; thus, the transition from one perspective to another might result in valuable insights into systems and knowledge engineering.

6 References

Balzer, W., & Moulines, C. U. (Eds.). (1996). *Structuralist theory of science: focal issues, new results.* Berlin: Walter de Gruyter. doi:https://doi.org/10.1023/A:1005518631916

Bergmann, M., Moor, J., & Nelson, J. (2014). *The Logic Book* (6th ed.). New York: McGraw-Hill.

Bjørner, D., & Henson, M. C. (Eds.). (2008). *Logics of Specification Languages.* Verlag Berlin Heidelberg: Springer. doi:https://doi.org/10.1007/978-3-540-74107-7

Boulding, K. E. (1956). General systems theory—the skeleton of science. *Management science, 2*(3), 197-208. doi:https://doi.org/10.1287/mnsc.2.3.197

Bourbaki, N. (1950). The architecture of mathematics. *The American Mathematical Monthly, 57*(4), 221-232. doi:https://doi.org/10.1080/00029890.1950.11999523

Bourbaki, N. (2004). Structures. In *Theory of sets.* Berlin: Springer. doi:https://doi.org/10.1007/978-3-642-59309-3_5

CONCEPT. (2021). *O centre*. (Non-Profit-Making Partnership 'Company for innovations and high technologies "CONCEPT"') Retrieved 12 15, 2021, from https://www.acconcept.ru/o-centre/

CONCEPT. (2022). *Proekty*. (Non-Profit-Making Partnership 'Company for innovations and high technologies "CONCEPT"') Retrieved 06 10, 2022, from https://www.acconcept.ru/proekty/

Enderton, H. B. (1977). *Elements of set theory.* Academic press. doi:https://doi.org/10.2307/2273270

Fong, B., & Spivak, D. (2019). *An invitation to applied category theory: seven sketches in compositionality.* Cambridge University Press. doi:https://doi.org/10.1017/9781108668804

Frege, G. (1948). Sense and reference. *The philosophical review, 57*(3), 209-230. doi:https://doi.org/10.2307/2181485

Ghrist, R. W. (2014). *Elementary applied topology.* Seattle, WA: Createspace.

Google Maps. (2022). *UCL and Gordon Square.* (Google) Retrieved 03 21, 2022

Halmos, P. R. (2017). *Naïve set theory.* Courier Dover Publications. doi:https://doi.org/10.1007/978-1-4757-1645-0

Ivanov, A. Y., Nikanorov, S. P., & Garayeva, Y. R. (2008). *Spravochnik po teoretiko-sistemnym konstruktam.* Moskva: Kontsept.

Kuchkarov, Z. A. (2004). *Strategicheskoye planirovaniye i upravleniye organizatsiyey.* Moskva: Kontsept.

Kuchkarov, Z. A. (2006). *Metody kontseptual'nogo analiza i sinteza v teoreticheskom issledovanii i proyektirovanii sotsial'no-ekonomicheskikh sistem, 2-e izdanie.* Moskva: Kontsept.

Kuchkarov, Z. A., & al., e. (2015). *Ekologicheskiy kodeks Rossiyskoy Federatsii (v chasti promyshlennoy ekologii) (proyekt).* Moskva: Kontsept.

Kuchkarov, Z. A., & Nikanorov, S. P. (2007). *Issledovaniye sotsial'no-ekonomicheskikh sistem. Metodologiya. Teoriya. Sledstviya.* Moskva: Kontsept.

Lelyuk, V. A. (2009). *Opyt polaganiya predmetnykh oblastey, prinadlezhashchikh proizvodstvennym organizatsiyam, v apparate stupeney.* Moskva: Kontsept.

MIPT. (2021). *Kafedra kontseptual'nogo analiza i proyektirovaniya*. (Moscow Institute of Physics and Technology) Retrieved 12 15, 2021, from https://mipt.ru/education/chairs/concept_analyse_design/

MIPT. (2022). *Chitayemyye kursy*. (Moscow Institute of Physics and Technology) Retrieved 01 06, 2022, from https://mipt.ru/education/chairs/concept_analyse_design/

Nikanorov, S. P. (2008). *Teoretiko-sistemnyye konstrukty dlya kontseptual'nogo analiza i proyektirovaniya* (2-e ed.). Moskva: Kontsept.

Nikanorov, S. P. (2010). *Vvedeniye v apparat stupeney mnozhestv.* Moskva: Kontsept.

Nikanorov, S. P., Nikitina, N. K., & Teslinov, A. G. (2007). *Vvedeniye v kontseptual'noye proyektirovaniye ASU: analiz i sintez struktur.* Moskva: Kontsept.

Ogden, C. K., & Richards, I. A. (1923). *The meaning of meaning: A study of the influence of thought and of the science of symbolism.* New York: A Harvest Book. doi:https://doi.org/10.1038/111566b0

Ponomarev, I. N. (2007). *Vvedeniye v matematicheskuyu logiku i rody struktur: Uchebnoye posobiye.* Moskva: MFTI. Retrieved from https://inponomarev.ru/textbook

Shreyder, Y. A. (1971). *Ravenstvo, skhodstvo, poryadok.* Moskva: Nauka.

Sorokin, A. A., & Shalyapina, S. K. (2008). *Bibliograficheskiy ukazatel' publikatsiy po kontseptual'nomu analizu i proyektirovaniyu, 1967-2008* (2nd ed.). (S. P. Nikanorov, Ed.) Moscow: Kontsept.

Stoll, R. R. (1961). *Sets, Logic, and Axiomatic Theory.* Freeman.

Taylor, B. N. (2001). *The international system of units (SI).* US Department of Commerce, Technology Administration, National Institute of Standards and Technology. doi:https://doi.org/10.6028/nist.sp.330e2001

Voyshvillo, Y. K. (1989). *Ponyatiye kak forma myshleniya: logiko-gnoseologicheskiy analiz.* Moskva: MGU.

Yale Peabody Museum Babylonian Collection. (2022). *An approximation of the square root of 2, YPM BC 021354.* (Yale Peabody Museum) Retrieved 05 23, 2022, from https://www.history-of-mathematics.org/artifacts/babylonian-square-root-of-2-tablet

Zinov'ev, A. A. (1973). *Foundations of the Logical Theory of Scientific Knowledge (Complex Logic). Vol. 9.* Springer Netherlands. doi:https://doi.org/10.1007/978-94-010-2501-0

Zinov'ev, A. A. (1983). *Logical physics.* Dordrecht: D. Reidel Publishing Company.

Zinov'ev, A. A. (2006). *Faktor ponimaniya.* Moskva: Eksmo.