UDC 004.4

FORMAL APPROACH TO PERSONAL KNOWLEDGE MANAGEMENT

V. Lenko[1], **Yu. Shcherbyna**[2]

[1]*National University "Lviv Polytechnic",*
Mytropolyt Andrei str., 5, Lviv, 79013, e-mail: vs.lenko@gmail.com
[2]*Ivan Franko National University of Lviv,*
Universytetska str., 1, Lviv, 79000, e-mail: yshcherbyna@yahoo.com

The research is dedicated to the concept of knowledge and formal approaches of its management. The focus is on the logic-based models of knowledge representation and reasoning, in particular, type theories and ontologies, which are backed by formal semantics and reliable methods of deductive reasoning. The concept of knowledge is considered in the context of epistemology as justified true belief, where logic serves as a distinguished component for knowledge justification.

Key words: personal knowledge management, knowledge representation, ontology, type theory, formal reasoning.

1. INTRODUCTION

Knowledge as interpreted information is one of the key factors in the development of society. From the standpoint of pragmatism, valid knowledge helps achieve desired states and avoid undesired ones, where the states and their classification are formed evolutionally by the owner of knowledge during the whole life. The sources of knowledge consist of experience, which is formed from personal cognition and inferences, and information from other agents, obtained in symbolic, graphic, audio, and other forms of representation. It is worth noting that the same unit of information is typically mapped into knowledge structures and roles that differ between various recipients. This phenomenon is explained by the fact that information interpretation depends on the recipient's context – a set of knowledge and goals (desired and undesired states) that are inherent to a recipient during the process of interpretation.

Achievement of ambitious goals requires the use of a significant number of resources in a relatively limited period, so cooperation between stakeholders is an important factor for their implementation. The cooperation between stakeholders to achieve an ambitious goal involves the following stages:

1 set a long-term goal, with a contribution and agreement from each stakeholder;
2 identify the stepping-stones required for reaching the goal;
3 split the identified realms of work into smaller tasks;
4 create a roadmap for planning the sequence of actions;
5 distribute the tasks appropriately between stakeholders;
6 execute the tasks with active use of communication.

A necessary condition for the successful implementation of a goal in terms of cooperation is the presence of shared unambiguous interpretation of the purposes and objectives, which typically is specified explicitly. To reach a consistent understanding of the information among the stakeholders, and as a result – effective coordination, it is important to mitigate the influence of personal contexts on the interpretation of the shared information and adopt the practise of regular communication sessions.

---

2. Task analysis

Let's consider the possibility to mitigate the influence of personal context on information interpretation. The following observations suggest the existence of such a possibility. Firstly, the context for information interpretation can be specified explicitly – the definition of concepts, relationships, symbols, which reside in descriptive statements, should be accompanied by additional information. This approach provides the subject with a prioritized source for determining the context of information; moreover, context detailing reduces the number of valid interpretation models [1]. Methods and tools of ontological engineering [2] are the best fit for the practical implementation of the proposed approach. Secondly, the use of information that is difficult to interpret within an evolutionarily acquired context is possible only if the context is explicitly specified. This feature is observed in the theory of formal languages, which is used, namely, in the design and analysis of programming languages. As a result, providing an explicit context for the information and the use of formal languages for its representation mitigate the impact of the personal context on the information interpretation.

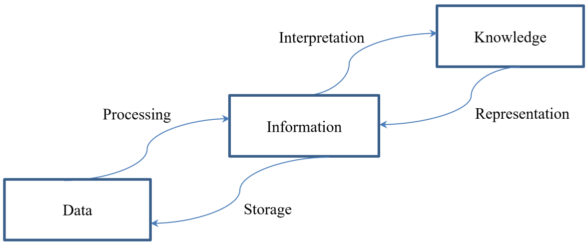


Fig. 1. Relations between the concepts of "data", "information" and "knowledge" [3]

The basis of formal language expressions construction is formal grammar – a set of rules for creating and transforming syntactic expressions of a language. The form of these rules directly affects the possibility of constructing a set of valid syntactic expressions and is characterized by the concept of the representation language "expressiveness". According to N. Chomsky [4], the hierarchy of formal grammars presupposes the existence of four types of grammars, where each type expands the previous one in the hierarchy, but none of them can be compared by "expressiveness" with the natural language. Formal languages compensate for this difference by the presence of desired computational properties, in particular, the ability to automate fully or partially the solution to some reasoning problems within the knowledge base systems.

The hypothesis of knowledge representation that was formulated in the dissertation of the philosopher B. Smith [5] in 1982 is a fundamental paradigm in the design of knowledge bases:

Any mechanically embodied intelligent process will be comprised of structural ingredients that

bullet *we as external observers naturally take to represent a propositional account of the knowledge that the overall process exhibits, and*

bullet *independent of such external semantic attribution, play a formal but causal and essential role in engendering the behavior that manifests that knowledge.*

By accepting this hypothesis, the knowledge-based systems are the ones that satisfy it by design. According to the hypothesis, the structures in knowledge-based systems have to satisfy two major requirements [6]:

1) the ability to interpret them as statements, which in their totality constitute all knowledge of a system (structures must be the expressions of language that is equipped with a corresponding truth theory, while syntactic requirements to their form are not established, i.e., sentence);

2) symbolic structures should play a causal role in the behaviour of a system (i.e., comments in programming languages do not meet this requirement).

The statement of the knowledge representation hypothesis reflects the approach to knowledge representation and its important role in shaping the behaviour of an intelligent agent. Models of knowledge representation and methods for knowledge management, which aim to reach the effective behaviour of an intelligent agent, have been studied from the time of Socrates. It is worth noting that the results of the conducted research on these matters are thesis, principles, theories, hypotheses, models, methods, and observations that belong to various fields of science, including philosophy, artificial intelligence, mathematical logic, game theory, economics, psychology.

3. Knowledge structure

The system analysis of the "knowledge" concept in the context of epistemology reveals the main components of its structure, but there is no final answer to this question, namely because of Gettier cases. Epistemology is one of the four major branches of philosophy that researches the concept of "knowledge" in terms of its nature, structure, sources, limitations, and connections, including the notion of "truth".

Modern philosophers distinguish three main types of knowledge, but in general, there exist multiple classifications [7]:

1) Declarative knowledge ("know-what"): descriptive statements – statements (propositions) that describe a particular aspect of reality. Declarative knowledge is also called "descriptive" knowledge;

2) Procedural knowledge ("know-how"): knowledge as an ability to do something, a skill;

3) Knowledge by acquaintance: this knowledge is formed from personal experience or by direct interaction with an object of knowledge.

Much of modern and ancient epistemology focuses on the study of declarative knowledge due to the ease of its spread between intelligent agents. Moreover, the knowledge representation hypothesis assumes the presence of declarative knowledge, so it is reasonable to analyse its properties in more detail.

Declarative knowledge is conveniently represented with the construction "S knows that p", where S denotes a subject that knows, and p is a statement that is known [8]. The question arises, what conditions are necessary and sufficient for S to know that p? According to the traditional approach to answering this question, which dates to the time of Plato, to know that p is a "justified true belief". Therefore, "S knows that p" if the following conditions are met:

1 the statement p is true;
2 S believes that p;
3 S belief that p is justified.

Together these conditions are considered necessary and sufficient for the definition of knowledge. If they are satisfied, namely p is true, and S believes that p and this belief is justified, then "S knows that p". Moreover, if "S knows that p", then S has a justified true belief that p. Thus, it is argued that declarative knowledge is equivalent to a justified true belief.

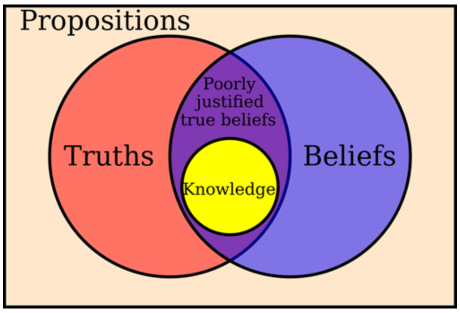


Fig. 2. Knowledge structure presented as Venn diagram [9]

Let's consider the role of each condition from the perspective of the difference between knowledge and belief. The first condition states that the statement p that S knows is true, thus not false. Of course, a subject may think that it knows p, and then it turns out that p is wrong; however, in this case, p is not knowledge, but a misbelief – the subject was wrong when it thought it knew p. The second condition states: S must believe that p to know that p. Although some philosophers set that knowledge and beliefs are very different mental states, the described case with a misbelief suggests the existence of a belief that can be mistaken for knowledge. Thus, their similarity allows assuming that knowledge is a kind of belief.

The third condition requires a belief to be justified – it allows distinguishing knowledge from true belief, which exists due to a successful coincidence. According to the traditional approach, the belief of S that "p is true" is not a successful coincidence when it is reasonable or rational to accept p as true from the standpoint of S. In the theory of evidentialism, the possession of evidence makes a belief justified. The main idea is that a belief is justified to the extent that it satisfies the evidence from S. In a non-traditional approach, the justification ensures that a belief of S is true with high objective probability and therefore is not a successful coincidence. One of the key ideas is that a high objective probability of truth can be achieved only when belief arises from reliable cognitive processes and sources. This approach is known as reliabilism.

It is worth noting that both approaches to the justification of true belief are not sufficient for knowledge since they are affected by Gettier problems. The essence of these problems is the existence of examples of justified true beliefs, where truth is a successful coincidence, thus three necessary conditions of knowledge are not sufficient. Gettier problems clearly show that the connection between what makes a belief true and what instantiates its truth needs better detail. The recent paper "Intuitionistic Epistemic Logic" [10] proposed an intuitionistic approach to the interpretation of the concept of "knowledge", which, according to the authors, is not affected by Gettier problems. It allows considering the truth of knowledge in the context of Brouwer-Heyting-Kolmogorov interpretation and creating constructive proofs of its truth using Curry-Howard isomorphism along with the expressive formal systems of typed lambda calculus.

4. ONTOLOGIES

There are several kinds of models in knowledge representation and reasoning, where the most popular are logic-based models, semantic networks, frames, and production rules. All of them can be characterized by expressiveness, which is the ability to represent arbitrary knowledge, by well-developed formal apparatus for reasoning that allows deriving new knowledge consistent to the existing one, and by the ease of use for the user. The diversity of knowledge representation models can be explained by the varying set of advantages inherent to each model. At the same time, the design of large and hyper-large knowledge bases requires a flexible and reliable knowledge representation model, which is diverse and effectively meets the users' needs. According to the professional community, an ontological model is one of the most suitable for this task.

An ontology is a formal, explicit specification of a shared conceptualization that is characterized by high semantic expressiveness required for increased complexity [11]. In general, an ontology is represented as a tuple $O =< C, R, F >$, where C is a finite set of domain concepts, $R : C \to C$ denotes a finite set of relations between concepts, and F corresponds to a finite set of interpretation functions (constraints, axioms) [12]. The main advantage of ontologies is that they combine the benefits of multiple knowledge representation models within a united structure, which allows their application to knowledge of varying nature. Concepts and relations can be depicted as a semantic network, the internal structure of a concept resembles a frame model, constraints are represented by production rules, while formal logic is used for domain axiomatization.

Among the most useful traits of ontologies are spreading a common understanding about the structure of a domain, facilitation of domain knowledge reuse, explicit specification of domain context, separation of the domain and operational knowledge. The breadth of ontology applications correlates with the generality of the domain it represents. As a result, a foundational ontology contains a taxonomy of general-purpose concepts. In contrast, task-oriented ontologies usually contain only the concepts that are important for the description or execution of a concrete task. Domain-oriented ontologies have a broader practical application and are based on expert knowledge about the specific domain.

The methods and tools for ontology engineering are described in [11, 13]. The process of ontology engineering comprises of the following steps:

1 Specification of the domain and ontology scope;
2 Investigation of the possibility to reuse the existing ontologies;
3 Identification of significant concepts in the domain;
4 Definition of classes and their hierarchy;
5 Definition of slots and properties of classes;
6 Definition of facets;
7 Creation of the instances of classes.

Steps 3-4 correspond to the process of domain conceptualization that creates a taxonomy of significant concepts. Steps 5-6 detail the properties of the concepts to ensure the possibility of logical reasoning.

The logical consistency of stored knowledge is a cornerstone property in knowledge bases and ontologies. The insecurity of this property leads to the derivation of absurd statements, which are the source of logical fallacy. In modern ontologies, consistency is ensured by formal systems, which provide syntax and semantics for knowledge representation and a deduction system for logical reasoning. When choosing between the formal systems, it is necessary to analyse their key properties, namely: language expressiveness, algorithmic decidability and computational complexity of the core reasoning problems. Nowadays, many ontology engineers use OWL 2 DL technology, which is based on the description logic formalism. Since description logic is the subset of first-order logic, its expressiveness is limited. The recent research on the application of higher-order logic and type theory to ontology engineering brings a higher level of expressiveness into the field of knowledge representation and reasoning [14].

5. TYPE THEORIES

The main feature of higher-order logics is an organization of propositional statements, individuals, functions, and other elements into a hierarchy of types, where the entities can operate only with the elements that reside lower in a hierarchy. Moreover, in higher-order logics, it is possible to describe the relationships between universals and individuals clearly and conveniently, without the need for additional constructions [15]. While first-order logic possesses good meta-properties of compactness and Lowenheim-Skolem, its limited expressiveness is an obstacle for multilevel reasoning.

The discovery of the Curry-Howard-Lambek isomorphism established the connection between the formal systems of proof theory, type theory, and category theory. According to isomorphism, the elements of first-order logic can be represented by the elements of type theory with dependent types (Tab. 1). Moreover, it states that logical proving can be implemented within the systems of typed λ-calculus. This led to the emergence of new concepts, namely "propositions as types", "proofs as programs" and "simplification of proofs as evaluation of programs". These concepts are embodied in the software called proof assistants, in particular Coq [16], Lean [17], Nuprl [18] and others, which facilitate the construction of mathematical theories and logical theorems proving using an interactive semi-automatic mode.

Table 1

Relationships between the first-order logic and type theory

First-order logic	**Type theory**	**Description**
proposition	A	type A
proof	$a : A$	term a of type A
$\perp$, Top	0, 1	bottom type, top type
$A \vee B$	$A + B$	co-product type
$A \wedge B$	$A \times B$	product type
$A \Rightarrow B$	$A \to B$	functional type
$\neg A$	$A \to 0$	functional type
$\exists_{(x:A)} B(x)$	$\sum_{(x:A)} B(x)$	dependent sum type
$\forall_{(x:A)} B(x)$	$\prod_{(x:A)} B(x)$	dependent product type

Representation of the ontology elements by the means of type theory is presented in papers [14, 15]. The extensive use of Coq proof assistant for the representation of ontology elements is not a coincidence, since it provides an implementation of a very expressive type theory called "Calculus of Inductive Constructions", facilitates the reasoning process with a set of handy tactics, ensures the termination of reasoning problems and supplies an interactive IDE for knowledge representation and reasoning. The most common elements of an ontology can be encoded in Coq via the following correspondence:

Table 2

Representation of an ontology structure in Coq [19]

Element of ontology	**Coq representation**
Concept	Class C: Type.
Instance	Instance X: C.
Properties, Inference rules	Class C (att1:nat) (att2:bool): Type := { f1: attr1 -> bool; }.
Binary relation	Parameter R: C -> C -> Prop.
Concept inheritance	Parameter D1: SubClass_G -> C. Coercion D1: SubClass_G > -> C.
"Part-whole" relation	Definition PartOf (x y: C) := R x y. Axiom A1: Reflexive PartOf. Axiom A2: Asymmetric PartOf. Axiom A3: Transitive PartOf.
Quantifiers ∃, ∀	exists X: C, forall X: C.

Example. Let's consider a fragment of the ontology $O =< C, R, F >$ that represents a domain of an Enterprise System Architecture, where

$$C = \{Network, Request, Service\};$$

$$R = \{Participate_In : Request > Service > Prop,$$

$$Are_Connected : Service > Service > Prop\};$$

$$F = \{Request_Part_Participate_In,$$

$$Service_Part_Participate_In,$$

$$intro_of_are_connected,$$

$$elim_of_are_connected\}.$$

Then the representation of the ontology in Coq and the reasoning on it are encoded as follows:

1. Import used Coq modules:
   Require Import Coq.Classes.RelationClasses.
2. Define the root concept in a hierarchy and the relation concept:
   *Definition **Kind**:= Type.*
   *Parameter **Relation**: Kind -> Kind -> Prop.*
3. Define the taxonomy of concepts:
   *Class **Network**: Type.*
   Parameter A1: Network -> Kind.
   Coercion A1: Network > -> Kind.

   *Class **Request**: Type.*

```
Parameter A2: Request -> Kind.
Coercion A2: Request > -> Kind.

Class Service: Type.
Parameter A3: Service -> Kind.
Coercion A3: Service > -> Kind.
```

4. Define relations of the ontology and their properties:

```
Definition Participates_ In (r:Request)(s:Service) := Relation r s.
Definition Are_ Connected (s1 s2:Service) := Relation s1 s2.
Axiom s_ of_ are_ connected: Symmetric Are_ Connected.
Axiom t_ of_ are_ connected: Transitive Are_ Connected.

Definition Part_ of (x y:Kind) := Relation x y.
Axiom r_of_part_ of: Reflexive Part_ of.
Axiom a_ of_part_ of: Asymmetric Part_ of.
Axiom t_ of_part_ of: Transitive Part_ of.
```

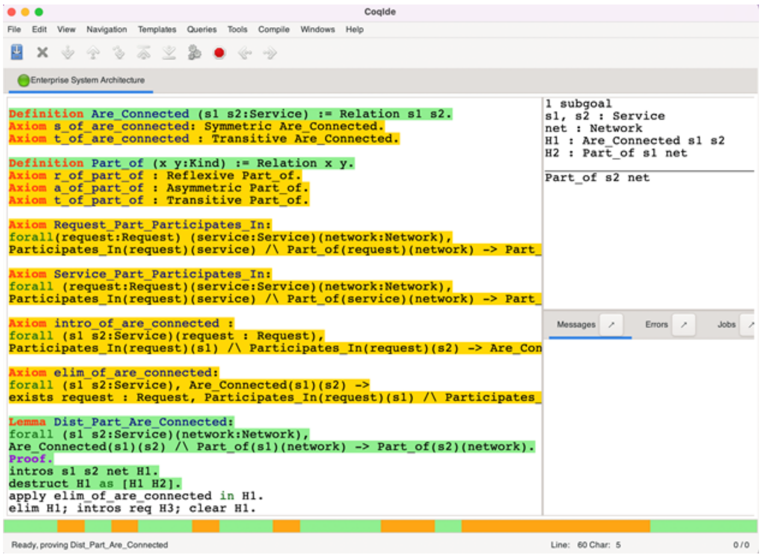
5. Define axioms of the ontology:

```
Axiom Request_ Part_ Participates_ In:
forall(request:Request) (service:Service)(network:Network),
Participates_In(request)(service) / Part_ of(request)(network) -> Part_ of(service) (network).

Axiom Service_ Part_ Participates_ In:
forall (request:Request)(service:Service)(network:Network),
Participates_In(request)(service) / Part_ of(service)(network) -> Part_ of(request) (network).

Axiom intro_ of_ are_ connected:
forall (s1 s2:Service)(request: Request),
Participates_In(request) (s1) / Participates_In(request) (s2) -> Are_ Connected (s1) (s2).

Axiom elim_ of_ are_ connected:
forall (s1 s2:Service), Are_ Connected(s1)(s2) ->
exists request: Request, Participates_In(request)(s1) / Participates_In(request) (s2).
```

6. Define and prove a lemma about relations of the ontology using Coq tactics language:

```
Lemma Dist_ Part_ Are_ Connected:
forall (s1 s2:Service)(network:Network),
Are_ Connected(s1)(s2) / Part_ of(s1)(network) -> Part_ of(s2)(network).

Proof.
intros s1 s2 net H1.
destruct H1 as [H1 H2].
apply elim_ of_ are_ connected in H1.
elim H1; intros req H3; clear H1.
destruct H3 as [H3 H4].
assert (H5:Participates_In(req)(s1) / Part_ of(s1)(net)).
```

split; assumption.
apply Service_Part_Participates_In in H5.
assert (H6: Participates_In(req)(s2) / Part_of(req)(net)).
split; assumption.
apply Request_Part_Participates_In in H6; assumption.
Qed.



```
Definition Are_Connected (s1 s2:Service) := Relation s1 s2.
Axiom s_of_are_connected: Symmetric Are_Connected.
Axiom t_of_are_connected : Transitive Are_Connected.

Definition Part_of (x y:Kind) := Relation x y.
Axiom r_of_part_of : Reflexive Part_of.
Axiom a_of_part_of : Asymmetric Part_of.
Axiom t_of_part_of : Transitive Part_of.

Axiom Request_Part_Participates_In:
forall(request:Request) (service:Service)(network:Network),
Participates_In(request)(service) /\ Part_of(request)(network) -> Part_

Axiom Service_Part_Participates_In:
forall (request:Request)(service:Service)(network:Network),
Participates_In(request)(service) /\ Part_of(service)(network) -> Part_

Axiom intro_of_are_connected :
forall (s1 s2:Service)(request : Request),
Participates_In(request)(s1) /\ Participates_In(request)(s2) -> Are_Con

Axiom elim_of_are_connected:
forall (s1 s2:Service), Are_Connected(s1)(s2) ->
exists request : Request, Participates_In(request)(s1) /\ Participates_

Lemma Dist_Part_Are_Connected:
forall (s1 s2:Service)(network:Network),
Are_Connected(s1)(s2) /\ Part_of(s1)(network) -> Part_of(s2)(network).
Proof.
intros s1 s2 net H1.
destruct H1 as [H1 H2].
apply elim_of_are_connected in H1.
elim H1; intros req H3; clear H1.
```

```
1 subgoal
s1, s2 : Service
net : Network
H1 : Are_Connected s1 s2
H2 : Part_of s1 net
______________________________________
Part_of s2 net
```

```
Ready, proving Dist_Part_Are_Connected                Line:  60 Char:  5     0 / 0
```

Fig. 3. Ontology representation and reasoning in Coq IDE

6. CONCLUSIONS

Knowledge plays a significant role in the development of individual perception and affects the progress of society in general. Ambitious goals require time and resources, and related false beliefs are usually the main obstacle to success. The conducted analysis of knowledge structure in the context of epistemology as justified true belief emphasized the role of justification for increasing the objective probability of its truth. Logic-based models of knowledge representation and reasoning come with several exceptional advantages, namely: 1) formal syntax and grammar that abstract ontology engineers from the overlapping associations; 2) formal semantics that enriches syntax constructions and provide an unambiguous way of language expressions interpretation; 3) a deductive system that derives only valid expressions from valid arguments; 4) explicit context specification that is regularly validated and detailed during the process of proof construction.

The amount of information tends to grow with time, so it is logical to assume that the size of knowledge bases will also keep growing. To incorporate the knowledge of various nature it is advisable to select expressive and structured means of knowledge representation and reasoning models, in particular, ontological model and type theories with dependent types. The presented method of ontology elements representation in Coq proof assistant enables the construction of knowledge bases for personal and shared usage, depending on the process of context definition. In personal knowledge bases, the context is defined by a single person, while in shared knowledge bases the context is defined by the community or experts, which tend to discuss and define the context based on the consensus.

REFERENCES

1. Baader F. An introduction to description logic / F. Baader, I. Horrocks, C. Lutz, U. Sattler. – Cambridge University Press, 2017.
2. Basiuk T. Ontological engineering / T. Basiuk, D. Dosyn, V. Lytvyn. – Lviv: Publishing House of Lviv Polytechnic National University, 2017.
3. Liew A. Understanding Data, Information, Knowledge and Their Inter-Relationships / A. Liew. // Journal of Knowledge Management Practice. – 2007. – Vol. 8 (2), P. 1–10.
4. Chomsky N. Three models for the description of language / N. Chomsky // IRE Transactions on Information Theory. – 1956. – Vol. 2. – P. 113–124.
5. Smith B. Procedural reflection in programming languages / B. Smith // Ph.D. Thesis. – Cambridge: MIT, 1982.
6. Levesque H. Expressiveness and tractability in knowledge representation and reasoning / H. Levesque, R. Brachman // Computational Intelligence. – 1987. – Vol. 3 (1). – P. 78–93.
7. Epistemology // Stanford Encyclopedia of Philosophy. – Retrieved from URL: https://plato.stanford.edu/entries/epistemology
8. The Analysis of Knowledge // Stanford Encyclopedia of Philosophy. – Retrieved from URL: https://plato.stanford.edu/entries/knowledge-analysis
9. Classical definition of Knowledge // Wikiquote. – Retrieved from URL: https://en.m.wikiquote.org/wiki/File:Classical_definition_of_Kno.svg
10. Artemov S. Intuitionistic epistemic logic / S. Artemov, T. Protopopescu // The review of symbolic logic. – 2016. – Vol. 9 (2). – P. 266–298.
11. Feilmayr C. An analysis of ontologies and their success factors for application to business. / C. Feilmayr, W. Wöß// Data & Knowledge Engineering. – 2016. – Vol. 101. – P. 1–23.
12. Lytvyn V. Pidkhid do pobudovy intelektualnykh system pidtrymky pryiniattia rishen na osnovi ontolohii. / V. Lytvyn // Problemy prohramuvannia. – 2013. – Vol. 4. – P. 43–52.
13. Noy N. Ontology Development 101: A Guide to Creating Your First Ontology / N. Noy, D. McGuinness. – 2001. – Retrieved from URL: http://protege.stanford.edu/publications/ontology_development/ontology101-noy-mcguinness.html
14. Dapoigny R. Towards a Conceptual Structure based on Type theory. / R. Dapoigny, P. Barlatier. – 2008. – Retrieved from URL: http://ceur-ws.org/Vol-354/p63.pdf
15. Hafsi M. Toward a Type-Theoretical Approach for an Ontologically-Based Detection of Underground Networks / M. Hafsi, R. Dapoigny, P. Bolon // Knowledge Science. Engineering and Management Lecture Notes in Computer Science. – 2015. – P. 90–101.
16. The Coq Proof Assistant // French Institute for Research in Computer Science and Automation. – Retrieved from URL: https://coq.inria.fr.
17. The Lean Theorem Prover // Microsoft Research. – Retrieved from URL: https://leanprover.github.io
18. Nuprl Proof Development System // Cornell University. – Retrieved from URL: http://www.nuprl.org
19. Lenko V. Knowledge representation and formal reasoning in ontologies with Coq. / V. Lenko, V. Pasichnyk, N. Kunanets, Y. Shcherbyna // Advances in Computer Science for Engineering and Education. – 2018. – Vol. 754. – P. 759–770. https://doi.org/10.1007/978-3-319-91008-6

Article: received 15.09.2021
revised 03.11.2021
printing adoption 24.11.2021

ФОРМАЛЬНИЙ ПІДХІД ДО УПРАВЛІННЯ ПЕРСОНАЛЬНИМИ ЗНАННЯМИ

В. Ленько[1], **Ю. Щербина**[2]

[1]*Національний університет "Львівська політехніка",*
вул. Митрополита Андрея, 5, Львів, 79013, e-mail: vs.lenko@gmail.com
[2]*Львівський національний університет імені Івана Франка,*
вул. Університеська, 1, Львів, 79000, e-mail: yshcherbyna@yahoo.com

Дослідження присвячено поняттю "знання" і формальним підходам до управління ним. Фокус зосереджено на логічно-орієнтовані моделі подання знання та міркування, зокрема, на теорії типів та онтології, які підкріплені наявністю формальної семантики та надійними методами дедуктивного міркування. Поняття "знання" розглянуто в контексті епістемології як істинне обґрунтоване переконання, де логіка є визначальним компонентом обґрунтування знання.

Знання відіграють фундаментальну роль у розвитку індивідуального сприйняття та впливають на прогрес суспільства загалом. Амбітні цілі потребують часу та ресурсів, а пов'язані з ними помилкові переконання є зазвичай основною перешкодою на шляху до успіху. Аналіз структури поняття "знання" в контексті гносеології як обґрунтованого істинного переконання закцентував роль обґрунтування у підвищенні об'єктивної ймовірності його істинності. Логічні моделі подання знання та міркування над ним мають багато суттєвих переваг, а саме: 1) формальний синтаксис і граматику, які абстрагують розробників онтологій від природно набутих асоціацій; 2) формальну семантику, яка збагачує різноманіття синтаксичних конструкцій і забезпечує однозначний спосіб інтерпретації виразів формальної мови; 3) дедуктивну систему, яка уможливлює виведення виключно істинних тверджень, за умови коректності початкових аргументів; 4) явну специфікацію контексту, яка регулярно перевіряється і деталізується під час доведення логічних виразів, властивостей математичних теорій і програмних специфікацій.

Обсяг інформації має тенденцію зростати з часом, тому логічно припустити, що розміри баз знань також будуть зростати. Для охоплення знань різної природи доцільно вибрати виразні та структуровані засоби подання знання та міркування над ним, зокрема, онтологічні моделі та теорії типів із залежними типами. Запропонований метод подання елементів онтології в інтерактивному асистенті доведення теорем Coq дає змогу проєктувати бази знань для особистого та спільного використання, які здебільшого відрізняються процесом означення контексту. У персональних базах знань контекст визначається однією особою, тоді як у спільних базах знань контекст визначається спільнотою чи експертними групами, які схильні обговорювати пропозиції й означити контекст за наявності консенсусу.

Використання асистента доведення теорем Coq для подання елементів онтології не є випадковістю, адже він надає реалізації дуже виразної теорії типів "Числення індуктивних конструкцій", полегшує процес міркування за допомогою набору тактик доведення, забезпечує розв'язність основних задач міркування і надає інтеґроване середовище розробки Coq IDE для подання знання та міркування над ним.

Ключові слова: управління персональними знаннями, подання знання, онтологія, теорія типів, формальне міркування.