The Knowledge Description Language

Vykhovanets V.S.
V.A. Trapeznikov Institute of Control Sciences
of the Russian Academy of Sciences
Moscow, Russia
valery@vykhovanets.ru

Abstract – **A knowledge base with a notional model of a subject domain is considered. The notional model consists of a notional structure and a content of notions included in it. The notional structure describes each notion as a result of the generalization or association of other notions. Enumerable and solvable sets of entities define the content of the notions. The notional structure contains the knowledge inference rules and is considered a formal theory of the subject domain that preserves the truth of all the consequences deduced in it. Declarative knowledge is represented by the notions with an enumeration of the entities belonging to them. Procedural knowledge is defined using the knowledge description language, which describes the notions with the solvable sets of the entities and facts and queries to the knowledge base. The fact is a true statement with the logical connectives, parentheses, and elementary expressions: unary predicates of belonging to an entity to the notion, functors for extracting the entities from the composite entities, and relations between the entities. A query is expressed as a fact with free occurrences of an entity variable. The execution of the query creates the notion with the entities for which the fact is satisfiable. The knowledge language provides operations for creating, deleting, and changing the entities and conditional and loop operators. The use of the knowledge description language allows you to increase the expressiveness of presentation and the efficiency of processing "large" knowledge.**

Keywords – mental abstractions of notions, notional models, notional structures, knowledge representation, knowledge bases, declarative knowledge, procedural knowledge, languages for describing and manipulating knowledge, inference on knowledge, queries to the knowledge base.

I. INTRODUCTION

Unsolved problems in knowledge representation and processing create significant difficulties in deploying knowledge bases and intelligent information systems based on them.

In particular, the task of extracting knowledge has not yet been solved well [1], the forms of knowledge representation are poorly adapted to the mental and psychological characteristics of a person [2], the processing of knowledge is time-consuming and generates hidden contradictions [3].

The main reason for the problems is that when modelling human thinking, traditions borrowed from logic and philosophy are used, which do not take into account such properties of human thinking as objectivity (openness to new things), subjectivity (restructuring of thinking when solving applied problems), reflexivity (self-regulation of cognitive activity), productivity (constant generation of new knowledge).

Some of the problems were avoided with the help of notional modelling. The models are called the notional models to distinguish them from conceptual models. The conceptual models define concepts and various relationships between them. In the notional models, the relationships between the notions are the notions and are used to form other notions. In the conceptual models, the relationships between the concepts are expressed by predicates that carry the main semantic of the model and cannot be used to form other concepts. A set of notions is defined in the notional models where only their formation by generalization and association are set [4].

The primary theoretical language of the conceptual models is the language of the first-order predicate calculus, which is so expressive that it generates unsolvable problems when describing subject domains. Examples of such problems are the unsolvability of the calculus itself, the incompleteness of the elementary arithmetic, the inconsistency of some set theories [5].

For this reason, to describe the conceptual models, attempts were made to limit the expressiveness of the logical language by using the descriptive logic, which is the solvable fragment of the first-order predicate logic with a relatively low computational complexity of knowledge representation and processing [6]. The descriptive logic is based on unary predicates of an instance belonging to a concept and binary predicates of roles (relationships between instances).

For the notional models, it was possible to further reduce the expressiveness of the logical language to a unary calculus of the first-order predicates with the equality and the order relations on the set of natural numbers [7]. The calculus is solvable and consistent on the countable notional models and complete on the finite notional models.

This paper is dedicated to the knowledge description and manipulation language used in the notional analysis and notional modelling of subject domains. Due to the absence of binary predicates with subject interpretation, it was possible to improve the expressiveness of knowledge representation and increase the efficiency of knowledge processing.

II. NOTIONS AND CONCEPTS

A notion is a type of thought that correlates with a specific set of unique representations (entities) of a person's internal or external world (a subject domain). The notion is a simple idea, opinion, representation, or understanding; notional is hypothetical, imaginary [8].

978-0-7381-3302-7/21/$31.00 ©2021 IEEE

In turn, a concept is a general or an abstract idea [8]. The concept consists of instances; a set of properties characterizes each instance; properties can be defined and nondefinite [3]. Thus, in contrast to the concept, the notion is not objective but subjective by definition.

The notions are formed (defined) in abstraction by performing mental operations on the entities. There are three types of notions and three types of mental abstractions that generate them: the notion-sign (identification), the notion-generalization (generalization), and the notion-association (association).

A. Identification

The notion-signs result from a mental selection of the unique representations from the subject domain and assigning them names (signs). The notion-signs are formed to fix a specific state of the sensory organs or elementary representations. When forming the notion-sign, the entity is mentally replaced by a name (sign), resulting in a one-to-one correspondence between the entities and the names (signs). The opposite abstraction for identification is interpretation.

Example 1. The examples of the notion-signs are such signs (words) as Red, Little, Last, One, Many, Love, etc. In turn, the sign of Red is interpreted as the entity of red colour.

B. Generalization

The notion-generalization is formed when merging the entities of the generalized notions (merging the sets of the entities). The generalization abstraction is also used to form a particular type of the notion-generalization – the notion-type, which is a generalization (typing) of the notion-signs. Generalization (typing) has an inverse abstraction called specialization (concretization).

Example 2. The notion-generalization is the notion of Duck, which merges ducks of various species: Ancona, Campbell, Duclair, Magpie, Mallard, Orpington, Rouen, etc. In turn, the notion of Mallard is a specialization of the notion-generalization of Duck.

Example 3. The notion-type of Taste can be formed by typing such notion-signs as Salty, Sour, Sweet, Bitter, Tart, Metallic, Chalky, Burning, etc., and the notion-sign Salty is a concretization of the notion-type of Taste.

C. Association

The notion-association is formed when joining (enlarging) the entities of the associated notions when each entity of the notion-association includes one entity of the associated notions (a subset of the Cartesian product of the sets of the entities).

Not all combinations of the entities of the associated notions can be the entities of the notion-association. Association has an inverse abstraction called individualization.

Example 4. The notion-association of Air joins such notions as Temperature, Humidity, Pressure, etc. The entity of the notion of Air can be ”68°C, 45 %, 101,325 Pa,” etc.

III. THE NOTIONAL ANALYSIS

The notional analysis identifies the notions and abstractions necessary and sufficient to describe a given subject domain. The abstractions of identification, association, and generalization used in the notional analysis are considered mental operations necessary and sufficient to isolate and transform the existing representations from the described subject domain into the notions. The main goal of the notional analysis is the multivariate structuring of the subject domain in the form of its notional structure.

A. Problem areas

Problems of unambiguous and consistent descriptions of the concepts related to the variability of the set of the properties of instances have led to the emergence of alternative theories of conceptual modelling [3]. Some of these theories are based on a probabilistic approach. The set of properties of instances of the concept is expanded, and their appearance has a probabilistic nature. Other theories divide the properties of the instances into typical, inherent in all instances, and atypical, which can appear only in some instances.

In notional modelling, the problems that have arisen in the description of the concepts are solved based on identifying the problem areas in the subject domain [7]. Since the concept assumes a subjective reflection of the subject domain, it provides for the possibility of several alternative descriptions of the same notion in each problem area in its way.

The problem area is a subject domain considered in some narrow sense (in some aspect), from the point of view of some functional problem, which structures the analyzed subject domain differently. The same notion in the different aspects (in the different problem areas) can have different definitions and descriptions. The generalization of the same notions from the different problem areas can be considered a concept of the subject domain.

B. The notional structure

The result of the notional analysis is the notional structure of the subject domain. The notional structure lists the notions and their aspects and the ways of their formation – their abstractions.

The notions used to form others are called attributes, and the notion itself is considered an association of its attributes. The common attributes of the notions include the notion-type of Name and the notion-type of Aspect, as well as the notion-type of Abstraction. The private attributes of the notion are the notions used for its formation.

The notional structure consists of schemas of the notions. A notional schema is an ordered set of the attributes, the first three elements of which are the common attributes, and the rest are the private attributes of the notion.

Example 5. Fig. 1 shows a fragment of a notional diagram of a specific subject domain. The notional diagram consists of geometric shapes, inside which the names of the notions and their aspects are indicated. Circles are used to define the notion-signs, ovals – the notion-types, rectangles – the notion-associations, and trapezoids – the notion-generalizations.

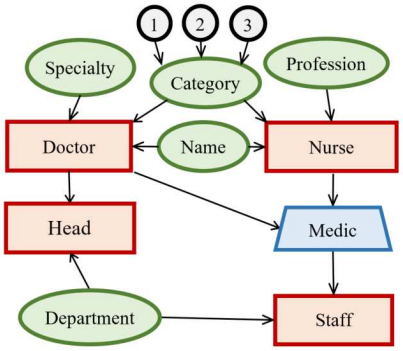

Fig. 1. A fragment of the notional diagram.

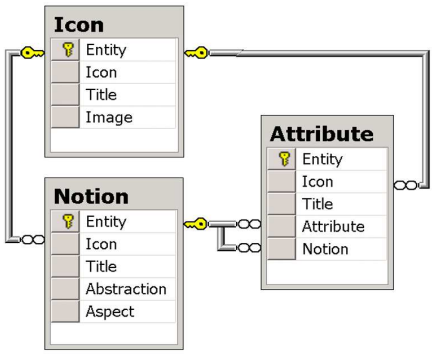

Fig. 2. A fragment of the database diagram.

Arrows on the diagram connect the notion-attributes with the notions they form. Schemas of the notions can be obtained from the notional diagram: Category: Typification, 1, 2, 3; Doctor: Association, Specialty, Category, Name; Nurse: Association, Profession, Category, Name; Head: Association, Doctor, Department; Medic: Generalization, Doctor, Nurse; Staff: Association, Department, Medic, where the name of the notion is separated from other attributes by a colon. The aspects of the notions are not indicated.

IV. Notional Modelling

The notional model consists of the notional structure and descriptions of the entities of its notions, performed using enumeration and resolving of sets.

A. The Notional Model

In modern information systems, the notional model of the subject domain can be implemented using a relational database management system [9].

The values of the simple data types are used as the notion-signs, and the simple data types themselves are considered the built-in notion-types, for example, Bit, Integer, Float, etc.

The notion-associations are implemented in the database tables that join the entities of the associated notions in one record. The notion-generalizations are represented as queries that union the records from the tables of the generalized notions.

B. The Database

A fragment of a database diagram with the notional model is shown in Fig. 2, which shows the tables of three notions: Notion, Attribute, and Icon, as well as the key fields that store the primary keys of records and relationships between the tables implemented in the form of the fields with the foreign keys [9].

Each record in the tables of the notional model is an entity that has the following standard fields (the common attributes): Entity – the unique identifier of the entity, Tittle – the name of the entity (the decorative name), Icon – the small image of the entity (the unique identifier of the image in the Icon table). The Title and Icon fields are used to display the entities in the user interface.

C. The Knowledge base

The notional knowledge base is the database containing the notional model and an inference mechanism for processing subject knowledge. Knowledge processing implies the presence of the forms of knowledge representation and the methods of manipulating them to simulate human reasoning.

In the notional knowledge base, the facts are true propositions with logical connectives AND ($\wedge$), OR ($\vee$) and NOT ($\neg$), parentheses and two types of elementary expressions: unary predicates $N(X)$ belonging the entity X to the notion N; relations $X[A] \div Y$, where $X[A]$ is the functor that returns the entity of the attribute A of the entity X, $\div$ is the relationships permitted between entities $X[A]$ and Y (equals, greater than, etc.). The functor $X[A]$ returns the unique entity ε (the empty notion) if the entity X has no attribute A.

Example 6. For the notional model, the diagram of which is shown in Fig. 1, the following facts can be expressed: Category(2) – 2 is the entity of Category, Nurse(X) – X is the entity of Nurse, Y[Category] = 3 – Y has Category equal to 3.

The scheme of the notion-generalization generates the following inference rule: an entity belongs to the notion N if and only if this entity belongs to at least one notion-attribute of the notion N. The scheme of the notion-association generates the following inference rule: if an entity belongs to the notion N, then entity's attributes are the attributes of the notion N.

Example 7. The notional structure in Fig. 1 contains the following inference rules (but not all):

$$\forall X\ \mathrm{Medic}(X) \leftrightarrow \mathrm{Doctor}(X) \vee \mathrm{Nurse}(X);$$
$$\forall X\ \mathrm{Staff}(X) \rightarrow \mathrm{Medic}\{X\} \wedge \mathrm{Department}\{X\},$$

where $\forall$ – the universal quantifier; $\rightarrow$ ($\leftrightarrow$) – implication (bidirectional implication); $A\{X\}$ is the predicate of the existence of the attribute A in the entity X, $A\{X\} \leftrightarrow \neg\, X[A] = \varepsilon$, ε – the empty notion.

The inference rules following from the notion-generalization define the notions to be processed. The inference rules following from the notion-association allow you to determine the presence of the attributes for the entities of this notion and extract the attribute entities (values), if necessary.

Example 8. Let's consider the knowledge base with the notional structure shown in Fig. 1. The figure shows that the following predicates and functors are defined in the notional model: Doctor(X), X[Name], X[Specialty], X[Name]; Staff(Y), Y[Name], Y[Profession], Y[Category]; Head(Z), Z[Doctor], Z[Department]; Medic(V), V[Doctor], V[Nurse]; Staff(W), W[Medic], W[Department], where Name, Specialty, Profession, Category and Department are notion-types defined by an enumeration of the entities belonging to them.

Using the names of entities, their variables, predicates, and functors listed above, you can formulate the queries to the knowledge base, for example, "Who from the staff of the Department of Surgery is a nurse without the third category?" In the language of the formal logic, this query will have the expression

$$\text{Staff}(X) \wedge X[\text{Department}] = \text{Surgery} \wedge$$
$$\wedge \forall Y\,(X[\text{Medic}] = Y \wedge \text{Nurse}(Y) \wedge \neg\, Y[\text{Category}] = 3),$$

where X is the free variable and Y is the bound variable.

V. Procedural Knowledge

Declarative knowledge in the notional knowledge base is represented by the notions and the entities belonging to them. Procedures and functions are provided to represent procedural knowledge, which, like everything in the notional model, are entities of corresponding notions.

The operations of creating, modifying, and deleting the entities are used to express procedural knowledge. An assignment operation is provided, as well as conditional operators and loop operators to raise the level of the procedural language to universal programming languages. The conditions for these operators are expressed as the facts. An internal variable can be assigned the result of a query for further processing.

A. Built-in data types

Let's consider the procedural language of the notional knowledge base. The built-in data types of this language are bytes, characters, integer numbers, floating-point numbers with built-in C-style operations.

The built-in Date type has been added to the language, as well as String and Binary types with the concatenation operation ` (grave accent) and the assignment with concatenation `=.

The built-in Program type has the entities that are the sequences of the commands of some virtual machine generated by the Java-style lambda expressions (X) –> Y, where X is the arguments (may be missing), and Y is the operator of the body of the defined function.

The only built-in notional constant is the empty notion ″ (double apostrophe) with the operation of coalescing with the empty notion ?? and the assignment and coalescing with the empty notion ??= in the C#-style.

B. Operations

The logical part of the language is standard: ==, != – the equality and inequality relations; >, <, >=, <= – the order relations; !, &&, || – the logical connectives. To parallelize calculations, sequential and parallel calculation operations are used: X, Y, ..., Z and $X :: Y :: ... :: Z$.

The number of the ternary operations has been increased: $X\,?\,Y : Z$ – conditional calculation (if X, then Y, otherwise Z); $X \sim Y : Z$ – cyclic calculation (while X, calculate Y, then Z); $X\,!\,Y : Z$ – blocking X while Y is calculated, then Z; X ` $Y : Z$ – searching for occurrences of Y in X and replacing Z.

The abstraction operations $[N1 : X1, ..., ND : XD]$ and $\{N1 : X1, ..., ND : XD\}$ are creation of the notion-association and the notion-generalization with the attributes Xj and names Nj (j = 1, ..., D).

The intentional operations: $X[]$ – the number of the attributes of the notion X, $X[Y]$ – access to the attribute Y of the entity X, $[X]Y$ – access to the attribute X of the notion Y, $[]X$ – the notion of the entity X.

The extensional operations: $X\{\}$ – the number of the entities of the notion X, $X\{Y\}$ – access to the entity Y of the notion X, $\{X\}Y$ – creating the entity X of the concept Y, $\{\}X$ – deleting the entity X.

The procedural operations: $X.Y$ – the function (procedure) Y of the notion X, $X()$ – execution of the procedure X, $X(Y)$ – execution of the function X with arguments Y, $(X)Y$ – definition of the function with the arguments X and the body Y, $()X$ – definition of the procedure with the body X.

The queries to the knowledge base: $<X>Y$ – the query Y with the free variable X, $< >X$ – update the entity X.

C. Operators

The language operators are standard: the assignment operation with a semicolon, the compound operator (the operators in curly brackets), goto, switch, if-then-else, while, do, for, try-catch, and throw operators. Also, it is required the transaction operator begin-end-undo [9].

Example 9. Below is the procedure that increases the categories of the nurses of the Surgery department (see Example 8).

```
// The definition of the category boost function
Boost = (Category) Category < 3 ? 1 : 0;
// Opening the transaction named Categories
begin Categories;
// The query for the entities of the local notion of Nurse
Nurses = <X>(Staff(X) != ″
        && X[Department] == Surgery
        && Nurse(X[Medic]) != ″;
// Creating the index variable and initializing it
Index = 0;
// Cycle through the entities of the notion of Nurse
while (Index < Nurses{})
{
        // Creating the local entity of Nurse
        Nurse = Nurses{Index};
```

```
        // Increasing the Category attribute of the Nurse entity
        Nurse[Category] += Boost(Nurse[Category]);
        // Increment of the index of the entity of Nurse
        Index = Index + 1;
    }
    // Updating the changed entities
    < > Nurses;
    // Successful completion of the transaction
    end Categories;
```

D. Methods

Functions of the language are used to calculate the virtual attributes of the notions. So, for the notion-type of Date, such virtual attributes are Year, Month, Day, Time, Hour, etc. The notion-type of String (Binary) has the virtual attribute of Length.

Notions functions and procedures are defined in a style of class methods of object-oriented programming languages. For example, for the notions of String sand Binary – the indexers *X*.Char(*Y*) and *X*.Byte(*Y*) that return, respectively, a character and a byte with the index *Y*.

VI. CONCLUTIONS

The fundamental difference between the considered approach to the analysis and modelling of the subject domain, the representation, and processing of knowledge is the use of another semantic invariant in addition to the formal logic – the formal theory of notions. The formal logic is based on three logical connectives (NOT, AND, OR), which are used to construct the formal propositions, and the formal theory of the notions is based on three mental abstractions (identification, association, generalization), which are used to construct the formal notions.

The semantic methods of the formal theory of the notions that make it invariant to subject domains are: naming the entities of the subject domain during identification and formation of primary semantic categories, combining in some sense the entities of the subject domain during generalization, joining the entities of the subject domain during association so that the meaning of such a combination is to create the composite (enlarged) entities, enumeration of the entities of the subject domain during generalization and association in the sense of limiting the existence of such entities.

Analysis and modelling are called notional analysis and modelling to distinguish them from conceptual analysis and modelling. The notion is different from the concept. A concept is an abstract objective notion, and a notion is a concrete subjective concept. For this reason, many formal notions have the same name but different notional structures and content in different aspects (problem areas). In this case, the concept is defined as generalizing the same name notions in various aspects.

An enumeration of the entities in the database tables represents declarative knowledge in the notional knowledge base. An essential property of the notional knowledge base is the storage of procedural knowledge as well as declarative. It allows you to abandon complex software tools that run on the user's side and use the knowledge base servers as the primary program storage.

The notional models' main difficulty is using a new methodology for analyzing and modelling subject areas and new technology for representing and processing knowledge. However, the high efficiency and expressiveness of the implemented knowledge models, their easy updating and adaptation allow us to hope for the widespread introduction of notional modelling into the practice of creating and using modern intelligent information systems.

The next step in studying the formal theory of notions is the optimization of algorithms for executing the queries to the notional knowledge base embedded in the knowledge description language and obtaining estimates of their computational complexity. However, we can already say that the query execution time will have an excellent polynomial estimate. A search for an entity in a notion table with the number of records n is carried out in time with an asymptotic estimate of n, and in the case of using an index, then in time with an asymptotic estimate of log n, where a log is a logarithmic function [9].

Another direction in the development of the notional analysis and notional modelling is the development of the methods for teaching notional models based on the perception of the surrounding world in the form of an incoming stream of unstructured data in the knowledge description language. The notional model allows you to structure this stream, highlighting the entities of known and unknown notions in it. The unknown entities can change the notional model by replenishing the entities of the existing notions or by forming new ones. A large class of image recognition tasks also belongs in this direction.

REFERENCES

[1] Y. Zao, C. Zhang, L. Cao, "Post-mining of association rules: techniques for effective knowledge extraction." Hershey, NY: Information science reference, 2009, pp. 81-149.

[2] R. J. Brachman, H. J. Levesque, "Knowledge representation and reasoning." San Francisco, CA: Morgan Kaufmann, 2004, pp. 31-45.

[3] A. Olive, "Conceptual modelling of information systems." New York, NY: Springer, 2007, 471 p.

[4] V. S. Vykhovanets, "Large-scale information systems based on conceptual models," Management of large-scale system development. Moscow, Russia: V.A. Trapeznikov Institute of control sciences, 2019. URL: https://ieeexplore.ieee.org/document/8911106.

[5] E. Mendelson, "Introduction to mathematical logic." New York, NY: CRC Press, 2010, 496 p.

[6] "The description logic handbook: theory, implementation and applications," eds. F. Baader, D. Calvanese, D. L. McGuinness. New York: Cambridge University Press, 2003, 573 p.

[7] V. S. Vykhovanets, "The notional model of knowledge representation," Journal of physics: conference series, vol. 1864, No 012058, 2021. URL: https://iopscience.iop.org/article/10.1088/1742-6596/1864/1/012058.

[8] "The Oxford dictionary of current english", ed. D. Thompson. New York, NY: Oxford University Press, 1993, 1091 p.

[9] R. Elmasri, S. B. Navathe, "Fundamentals of Database Systems." Hoboken, NJ: Pearson, 2016, 413 p.