PAPER • **OPEN ACCESS**

The Notional Model of Knowledge Representation

To cite this article: V. S. Vykhovanets 2021 *J. Phys.: Conf. Ser.* **1864** 012058

The Notional Model of Knowledge Representation

V. S. Vykhovanets

Institute of Control Sciences of Russian Academy of Sciences, 65, Profsoyuznaya St., Moscow 117997, Russian Federation.

E-mail: valery@vykhovanets.ru

Abstract. The article describes a model of knowledge, called the notional model. A notion is a thought about a certain unique representation from the subject domain. The notional model uses notion-signs, notion-associations and notion-generalizations that are formed by primary mental abstractions: identification, generalization and association, respectively. The notional model consists of the notional structure and the contents of the notions. The notional structure describes each notion as a set of other notions united by one of the primary mental abstractions. The content of the notion is described using enumerating and resolving procedures. The notional model is semantically invariant, improves the transparency of the subject domain, and allows you to create the knowledge base with a linear estimate of a query execution time.

1. Introduction

Many unresolved problems in the knowledge representation make difficult to implement knowledge bases effectively in practice. In particular, the extraction of knowledge is not a completely solved problem [1], forms of knowledge representation are poorly adapted to the mental and psychological characteristics of a person [2], and the knowledge processing takes a long time and is a subject to hidden contradictions [3].

Some of the existing problems were avoided by using the notional analysis and modelling. The closest to the notional model is the ER-model, where abstractions of the conceptual design are classification, aggregation and generalization, which are used for classifying primitive objects and building complex classes from them [4]. For relationships between classes, the abstractions of aggregations are used.

In the notional model, the relationships between notions are notions themselves [5]. The notional model is constructed by specifying the mental abstractions that formed the notions as well as enumerating (resolving) the entities belonging to the notions. Any query to the knowledge base is reduced to searching for the entities of notions that meet the specified conditions.

2. Notions

Notions are formed during the mental abstraction by performing mental operations on entities from the subject domain. There are three types of notions: notion-signs, notion-generalizations and notion-associations. Each notion has a name and is a set of entities from the subject domain.

2.1. Identification

The notion-sign is the result of mental selection of a unique representation (elementary entity) in the subject domain and its name. During identification, a one-to-one correspondence is established between entities of the subject domain and its signs (other entities). The notion-sign is a notion that has a single entity, or the notion-sign is an elementary entity from the subject domain.

2.2. Generalization
The notion-generalization is formed when entities of generalized notions are united (the union of entities). Therefore, all entities of the generalized notions are the entities of the notion-generalization. Generalization is also used to form a notion-type, which is a union of notion-signs.

An example of the notion-type is the Color notion, which is the union of the notion-signs Red, Blue, Green, and so on. An example of the notion-generalization is the Tableware notion, which unites the entities of such notions as Spoon, Fork, and Knife.

2.3. Association
The notion-association is formed when entities of associated notions are joined, i.e. when each entity of the notion-association includes one of the entities of the associated notions (the subset of the Cartesian product of the entity sets).

An example of the notion-association is the Hourglass notion, which joins such notions as Rack, Flask, and Sand. Note that not all combinations of the Rack, Flask and Sand entities are included to the Hourglass notion. For example, there is sand that is not used in any hourglass.

3. The notional analysis
The notional analysis consists in defining the notions and the ways of forming them. Identification, generalization and association are considered as the mental operations that are necessary and sufficient for mental isolation and transformation into the notions of existing representations from the described subject domain.

3.1. Problem areas
In the notion analysis, one or more problem areas are identified. A problem area is the subject domain considered in some narrow aspect, from the point of view of some active problem. The same notion may have the different definitions in the different aspects (in the different problem areas).

For example, the Hang notion in the first aspect is the placing on a wall or in an exhibition, in the second aspect is remain static in the air, in the third aspect is a form of capital punishment, and so on.

3.2. Notional structures
The purpose of the notional analysis is to obtain the notional structure of the subject domain. The notional structure defines the names and aspects of the notions, their mental operations, and the sets of the notions that these notions are defined.

Any notion has a common attribute Title – the notion name, a common attribute Icon – the notion image, and a common attribute Aspect – the problem area of the notion, as well as a common attribute Abstraction, which is a notion-type that has the following entities: Sign, Type, Generalization and Association. Private attributes of the notion are notions that are necessary for their formation and are determined during the notional analysis.

3.3. Schemas of notions
The notional structure consists of schemas of notions. The schema of the notion is an ordered set of the notional attributes. The first three elements of this set are Title, Aspect and Abstraction, which are the common attributes of notions. Then follow the private attributes.

For example, let there be a schema (Project, , Managerial, Association, Start, End, Resource, …), where parenthesis denote an ordered set. Then Project is the entity of the Title; is the entity of the Icon; Managerial is the entity of the Aspect; Association is the entity of the Abstraction; Start, End, Resource, and so on are the private attributes of the Project.

3.4. Notional diagrams
The notional diagram is a graphical form of the notional structure and is created during the notional analysis. The notional diagrams show notions, their full names and abstractions. The full name of the notion is unique and consists of two parts divided by @ sign: the first part is the notion name; the second part is the aspect of the notion. Abstractions are shown as shapes of the notions: circle (Sign), oval (Type), rectangle (Generalization) and trapezoid (Association). The private attributes of the notions are shown as arrows.

Figure 1 shows a fragment of a notional diagram for a certain subject domain in the empty aspect. Here are such notions as notion-signs, which are shown as black circles 1, 2, 3; notion-types, which are shown as green ovals Profession, Name, Class and Team; notion-generalization, which is shown as a blue rectangle Employee; notion-associations, which are shown as red trapezoids Worker, Engineer, Manage and Work.

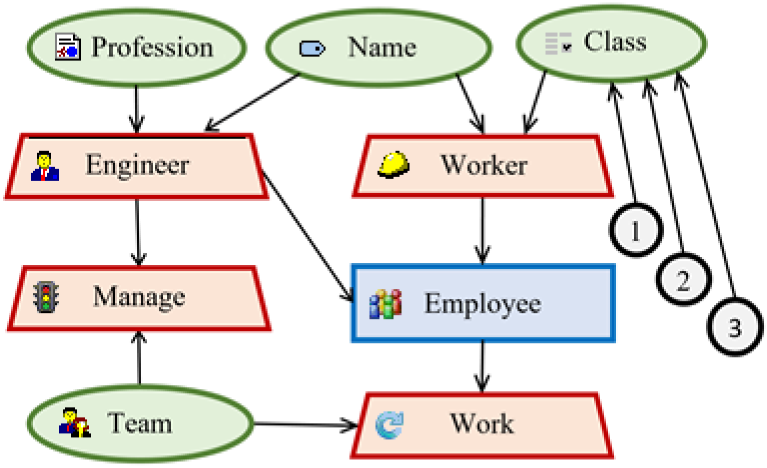

Figure 1**.** The notional diagram

4. The notional model
The notional model consists of the notional structure and the entities of the notions, which are described using the enumerating and resolving procedures. The enumerating procedure allows you to extract an entity from the notion by its number. The resolution procedure lets you know whether an entity belongs to the notion or not.

4.1. Entities of notions
Any entity in the notional model is a record in a database table with a unique global identifier or is a value of a simple data type. The simple data types are Bit, Integer, Float, Date, Time, Character, String, Binary, Image, and so on.

The values of the simple data types are used as the notion-signs. The simple data types are the notion-types. The notion-types also can be created as the simple data types with restrictions on their values, which are set by the resolving procedures. The enumerated notion-types are created as the tables of the values.

The notion-types are implemented as the built-in data types or as the database tables with one Title field. The notion-associations are implemented as the tables with fields corresponding to the associated notions, and the notion-generalizations are implemented as the views that select the entities from the tables (views) of the generalized notions.

4.2. The notion of a notion
Any notion is a notion, and the same means are used to describe it as for other notions. The kernel diagram of the notional model is shown in Figure 2, where presents five notion-tables: Abstraction, Aspect, Notion Icon and Attribute. Each record of these tables is the entity that has the following

common fields: Entity – the global unique identifier of the entity, Title – the name of the entity, Icon – the global unique identifier of the image.

The global unique identifier consists of two parts. The first part is the record number of the entity in the current table. The second part is the record number of the current notion in the Notion table.

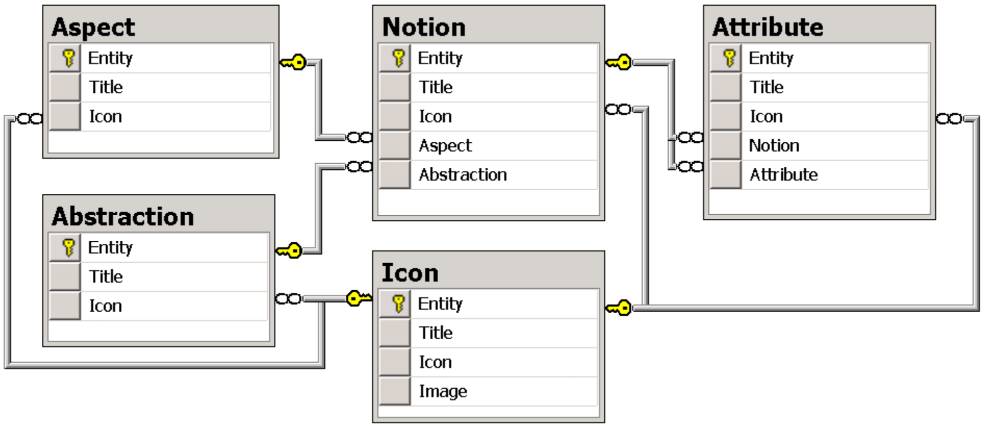

Figure 2. The kernel diagram of the notional model

The Aspect and Abstraction tables implement the enumerated notion-types and have no private attributes. The Icon table has the private attribute Image, which is the build-in data type. The Notion table has the private attributes Aspect and Abstraction, which are the global unique identifiers of the entities from the Aspect and Abstraction tables respectively. The Attribute table has two private attributes Notion and Attribute, and enumerates the private attributes of the notions.

5. The knowledge base

Any knowledge base is a database that contains the notional model, as well as an inference mechanism that allows you to perform the queries and acquire new knowledge. Facts are used to represent knowledge, and inference rules are used for reasoning, which allow making the inferences based on existing facts and obtaining new facts.

5.1. Facts

Facts are true expressions with logical connectives AND ($\wedge$), OR ($\vee$), NOT ($\neg$), parentheses, predicates of the forms $N(E)$ and relations of the form $N[E] \bullet V$, where N is a notion, E and V are entities, • is a relation allowed between the entities.

The predicate $N(E)$ is true if and only if the entity E belongs to the notion N. The functor $N[E]$ returns an entity of the attribute N of the entity E, or the notion-sign ε if the notion of the entity E has no attribute N. The notion-sign ε is reserved for the non-existent value (empty value), and ε is false in the logical context. A universal relation is the equality of the entities, which is defined recursively: two entities E_1 and E_2 are equal if and only if the entities of the same private attributes are equal,

$$E_1 = E_2 \leftrightarrow \forall N(N[E_1] = N[E_2]), \quad (1)$$

where N is an attribute of the entities E_1 or E_2. Since the notion table does not contain two rows with the same values of the private attributes, the comparison of the attribute values in the formula (1) is reduced to a comparison of the simple data types or the global unique identifiers at the recursion depth equal to one. Other relations can be inherited from the notion-types used in the notional model.

5.2. Inference rules

The schemas of the notions set the inference rules, and the notional structure is a formal theory that preserves the truth of all the consequences deduced in it.

From the scheme of the notion-generalization follows the inference rule, according to which the entity *E* belongs to the notion *N* if and only if this entity belongs to at least one private attribute of the notion *N*,

$$N(E) \leftrightarrow \bigvee\nolimits_{\forall N_i \in N} N_i(E). \quad (2)$$

From the schema of the notion-association follows the inference rule, according to which if the entity *E* belongs to the notion *N* then the private attributes of the entity *E* are the private attributes of the notion *N*,

$$N(E) \rightarrow \bigwedge\nolimits_{\forall N_i \in N} N_i[E]. \quad (3)$$

5.3. *Queries*

The result of any query to the knowledge base is a search for the notion entities that meet a given condition, expressed as a fact. In reasoning, the rules (2) define the notions to be processed, and the rules (3) allow you to extract the private attributes from the entities to verify the condition.

Figure 3 shows a query form for searching for the notion entities that meet the conditions set using the template on the right panel. The template is displayed after the pointer is set on the search node (the node with the binoculars icon). The template consists of a list of the notion attributes with its searched entities. Note that these entities, separated by the # symbol, are the results of the nested queries; see Executor (Исполнитель) in Figure 3.

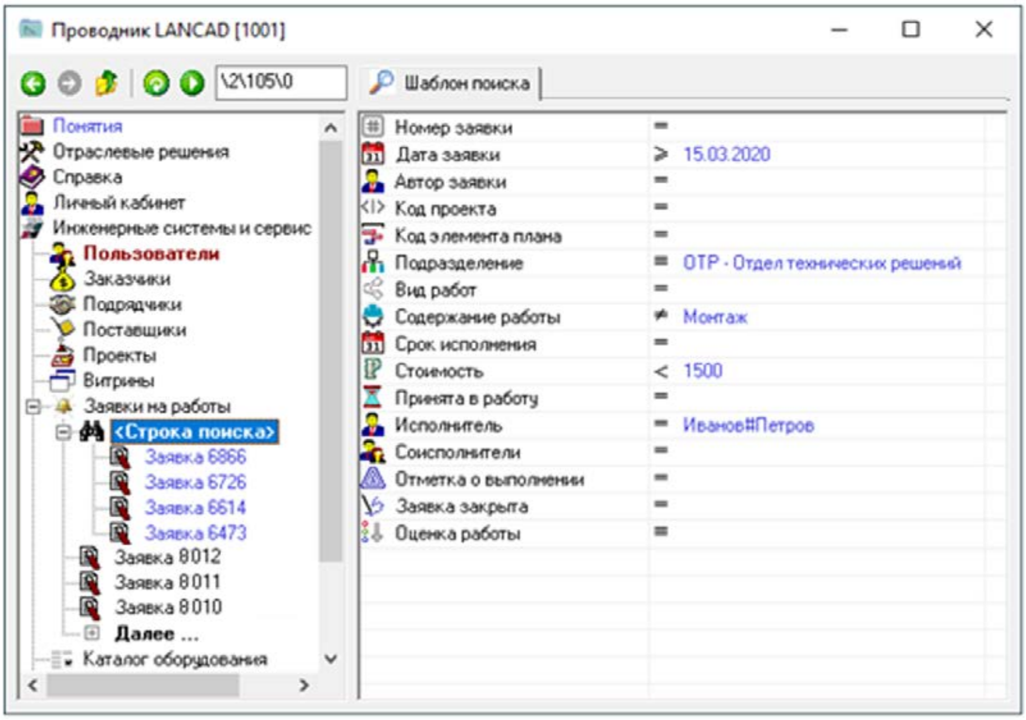

Figure 3. The query form of the knowledge base (it is published with the permission of LANIT)

After filling in the template, the knowledge base searches for the entities that meet the specified conditions and displays the found entities as the child nodes on the left panel (is marked in blue). The simplified entities search is performed after entering a string as the name of the search node. Note that the query execution time has a polynomial estimate, since the time to search for an entity depends asymptotically on the number *n* of records in the database table as *n*.

5.4. *Reasoning*

Trees on the left panel in Figure 3 are a result of reasoning with some root nodes. The reasoning procedure follows:

1) let there are the root nodes, which are the entity nodes of some initial notion or the notion nodes of some initial entity;

2) if the parent node is the entity node, then the child nodes are the notion nodes, which have the notion of the parent entity as a private attribute;

3) if the parent node is the notion node, then the child nodes are the entity nodes, which have the entity of the parent notion as a value of the private attribute;

4) the procedure is repeated from step 2 for each parent node, which is clicked by the user.

For example, the root nodes in Figure 3 are Notions (Понятия), Industry solutions (Отраслевые решения), Help (Справка), User profile (Кабинет пользователя), Engineering systems and services (Инженерные системы и сервис), and so on.

Let the user click on the Engineering systems and services. This is the entity node of the Company notion. Then child nodes of this node are the notion nodes, which have the Company notion as a private attribute. In Figure 3, these notion nodes are Users (Пользователи), Customers (Заказчики), Contractors (Подрядчики), Suppliers (Поставщики), Projects (Проекты), Shows (Витрины), Job requests (Заявки на работы), Equipment catalog (Каталог оборудования), and so on.

If the user clicks on the Job requests node, then entity nodes are displayed, which have the Engineering systems and services as the value of the Company attribute. The first child node of the notion node is the search node (<Строка поиска>).

6. Conclusion

The fundamental difference between the considered approach for the knowledge representation and processing is the use of another semantic invariant in addition to formal logic – the formal theory of the notions. The formal theory of the notions is based on three primary mental abstractions to construct notions as well as the formal logic is based on three logical connectives to construct propositions.

The analysis and modelling are called notional to distinguish them from the conceptual analysis and modelling. The notion is different from the concept. One can say that a concept is an abstract objective notion, while a notion is a concrete subjective concept. For this reason, there are many formal notions that have the same name, but are described differently in the various aspects. In this case, a concept can be presented as a set of eponymous notions in various aspects.

The query to the knowledge base requires linear execution time and consists only of the unary predicates of belonging the entity to the notion. It allows us to abandon the binary and other predicates, as well as the second and higher-order logic in the modelling of the subject domain. This ultimately makes the knowledge bases simpler and more efficient.

The experience of using the notional model has increased the transparency of the business processes, reduced the risks of the ownership of the knowledge base and increased the investment attractiveness of the company.

References

[1] Yanchang Z, Chengqi Z, Longbing C 2009 *Post-Mining of Association Rules: Techniques for Effective Knowledge Extraction* (Hershey, New York: Information science reference) section III pp 81-149

[2] Brachman R J, Levesque H J 2004 *Knowledge Representation and Reasoning* (San Francisco, CA: Morgan Kaufmann) chapter 3 pp 31-45

[3] Talens G, Boulanger D, Séguran M 2007 Domain Ontologies Evolutions to Solve Semantic Conflicts *Ontologies-Based Databases and Information Systems* Collard M (Berlin, Heidelberg: Springer) pp 51-67.

[4] Batini C, Ceri S, Navathe S B 1992 *Conceptual Database Design: An Entity-Relationship Approach* (Redwood City, CA: The Benjamin/Cummings Publishing Company) p 42

[5] Vykhovanets V S 2019 Large-Scale Information Systems based on Conceptual Models *Proc. Int. Conf. on Management of Large-Scale System Development* (Moscow: V.A. Trapeznikov Institute of Control Sciences) (IEEE Xplore)