Notional models based on primary mental abstractions

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

Abstract

Purpose – The purpose of the article is to research a semantically invariant method of modeling subject domains based on the use of primary mental abstractions of identification, generalization, and association. The modeling process does not depend on any subject domain, but is determined only by human abstraction abilities.

Design/methodology/approach – Notions are modeled, but not concepts. A notion is an abstract objective concept, and a concept is a concrete subjective notion. There are different notions belonging to the same concept. A notional model consists of a notional structure and descriptions of notions included in it. The notional structure is defined as a set of notions formed by mental abstractions of identification, generalization, and association. A description of a notion is a set of other previously defined notions.

Findings – Using the primary mental abstractions allows increasing the level of model abstraction, improving the scalability of information systems and developing an information system, which requires a small number of common algorithms with a small computational complexity. These algorithms do not depend on the subject domain as they are formulated in universal operations on notions.

Originality/value – The refusal to describe associations as relations with different semantic meanings makes the conceptual model semantically invariant (independent of subject domains). This effect is because an association between notions in notional models is a notion that can be used to define other notions.

Research/ Practical/ Social/ Environment implications – The semantic invariance of the modeling method increases the efficiency of describing complex subject domains, as well as the updatability and usability of the notional models. The use of an information system with a notional model increases the transparency of the company's business processes and reduces the risks of the ownership of the information system.

Research limitations – The main difficulty in using the notional models is the need to master the new methodology and technology of modelling, knowledge representation and knowledge inference.

Keywords: subject domain, mental abstractions, identification, association, generalization, notional structure, notional model, information system, knowledge representation, knowledge inference.

1 Introduction

Conceptual modeling is becoming an increasingly important area of the information technology. The main purpose of conceptual modeling is to formalize knowledge about a certain subject domain in the form closest to the understanding of users and developers of the information system (Kogalovsky & Kalinichenko, 2009). The conceptual approach is associated with the expectations of an effective description of complex subject domains, improving the reliability and quality of information systems, accelerating the updating of data following the change in the subject domain, ensuring the reuse of models of different fields of knowledge, etc. (Borgida, Chaudhri, & Giorgini, 2009).

However, the extraction of knowledge is still not a fully solved problem, the forms of knowledge representation are poorly adapted to the mental and psychological characteristics of people, knowledge processing takes a long time and is a subject to the emergence of hidden contradictions.

This work is devoted to the description of one of the types of conceptual models, called notional models, which based on primary mental abstractions of identification, generalization and association (Vykhovanets, 2019). Conceptual models specify concepts and the various relationships between them. In the notional model, the relationships between notions are ordinary notions, the notional model is created by specifying the abstractions used to form the notions, as well as enumerating the notions of the subject domain that belong to them. This avoided some of the existing problems in conceptual modeling.

2 Notions and concepts

A notion is a type of thought, which corresponds to a certain set of unique representations (entities) of the internal or external world of a person (subject domain). A notion is a simple idea, opinion, representation or understanding of something; a notional is hypothetical, imaginary (Thompson, 1993). A notion correlates with representations from the subject domain, i.e., entities. In turn, a concept is a general notion, an abstract idea (Thompson, 1993). Thus, unlike a concept, a notion is not objective, but subjective by definition.

Notions are formed (defined) during mental abstraction. Each notion has a name that denotes some set of entities from the subject domain. There are four types of notions: elementary, simple, single and abstract.

2.1 Elementary notions

The elementary notions (notion-signs) are the result of a mental selection (identification) of unique perceptions in a subject domain and assigning names to them. The elementary notions are formed to fix a particular state of the senses or elementary abstract ideas.

Example 1 – Examples of elementary notions may be such notions as Red, Long, One, First, Many, Often, Love, Other, etc. ♦

2.2 *Simple notions*

The simple notions (notion-types) are formed by combining elementary notions similar in some sense (it is used mental abstraction of typification). They are assigned a unique name and acceptable values range is defined, considered as a set of elementary notions.

Example 2 – An example of a simple notion is such notion as Color, which union the elementary notions of Red, Green, Blue, etc. Another example is the notion of Integer that union all representable integers. ♦

2.3 *Single notions*

The single notions (notion-entities) are formed by mental selection of unique presentations in the subject domain with several elementary notions (mental abstraction of aggregation is used).

Example 3 – The single notion of Ball may aggregate such elementary notions as Red, Small, Rubber, etc. This implies the existence of balls with different colors, sizes, and materials. ♦

2.4 *Abstract notions*

The notion-association is formed by join several notions. Aggregation is a special case of association. Unlike aggregation, not all combinations of entities can constitute the real notion-entity of the notion-association.

Example 4 – An example of a notion-association is the notion Weather, which combines such notions as Place, Date, Temperature, Humidity, Wind, Cloudiness, etc. The notion-entity of the notion Weather can be "Moscow, August 15, 2021, 28 ° C, 45%, 3 m/c, Cloudy, ...". ♦

The notion-generalization is formed by union several notions. In the same way, the union of elementary notions forms a simple notion. Thus, typification is a special case of generalization.

Example 5 – An example of the notion-generalization is the notion of Tree, which is the union of the notion-entities of such notions as Birch, Beech, Fir, Pine, Poplar, etc. ♦

2.5 *Aspects*

When you use the complex subject domains, the notion name is divided into two parts: the notion name itself and the interpretation area name (aspect). To identify any notion requires its name and aspect. The name of a notion without aspect defines a concept with the same name, which is the notion-generalization uniting all concretizations of these notions in the subject domain.

Example 6 – The concept Opposite is a generalization of the notions Opposite in the aspects Light, Beauty, etc., which, in turn, are notion-associations, connecting the notions, the entities of which are opposite in some sense (association by contrast). For the notion of Opposite in the aspect of Light, these are the notion-entities "Bright, Dim", "Bright, Dull", "Dull, Strong", "Visible, Invisible", etc. For the notion of Opposite in the Beauty aspect, these are the notion-entities "Beautiful, Ugly", "Beautiful, Terrible", etc. ♦

3 Notional structures

A notional structure is the result of the notional analysis of a subject domain and expresses the mapping of some notions into other notions. The abstractions of identification, association and generalization are considered as mental abstractions that are necessary and sufficient for the isolation and transformation into separate notions the presentations accumulated in a subject domain.

Definition 1 – Let K be a predefined set of kinds of notions, $K = \{c, t, a, g\}$, where c denotes notion-sign, t denotes a notion-type, a denotes a notion-association and g denotes a notion-generalization. ♦

Definition 2 – Let N be a finite set of names of notions, $N = \{n_i \mid i = 1, 2, \ldots\}$. A notion-sign $N_i^C \in N^C$ is described by its schema $H_i^C = (c, n)$, $c \in K$, $n \in N$, N^C is the set of notion-signs. A notion-type $N_i^T \in N^T$ has schema $H_i^T = (t, n, \{N_j \in N \mid j = 1, 2, \ldots\})$, where $t \in K$, $n \in N$, N^T is the set of notion-types. A notion-association $N_i^A \in N^A$ has schema $H_i^A = (a, n, \{N_j \in N \mid j = 1, 2, \ldots\})$, where $a \in K$, $n \in N$, N^A is the set of notions-associations. A notion-generalization $N_i^G \in N^G$ has schema $H_i^G = (g, n, \{N_j \in N \mid j = 1, 2, \ldots\})$, where $g \in K$, $n \in N$, N^G is the set of notions-generalizations, $N = N^C \cup N^T \cup N^A \cup N^G$, where $\cup$ is the union relation of sets that are given in parentheses. ♦

Definition 3 – A notional structure $S = (N, H)$ is a finite set of names N and a finite set of schemas H, $H = H^C \cup H^T \cup H^A \cup H^G$, where H^C is schemas of elementary notions, H^T is schemas of simple notions, H^A is schemas of notions-associations and H^G is schemas of notions- generalizations. ♦

Example 7 – Let N = {User, File, Program, Principal, Password, Login, String, a, b, …, aa, ab, …}. Let the elementary notions have the following schemas: (c, a), (c, b), …, (c, aa), (c, ab), and so on. The simple notion of String has the schema (t, String, {a, b, …, aa, ab, …}). The notion-associations Login, Password, User, File and Program have the following schemas: (a, Login, {String}), (a, Password, {String}), (a, User, {Login, Password}), (a, File, {String}), (a, Program, {File, Password}). The notion-generalization of Principal has schema (g, Principal, {User, Program}). ♦

The fundamental differences between the notional structure and other forms of ontologies are as follows:

- there is no division of terms into signs, values, features, types, entities, links, relationships, roles; there is only a notion;
- it is possible to represent associations as independent notions, which allows, for example, to express generalization of associations;
- there is a semantic invariance of the formalism, which does not require the involvement of subject knowledge for the interpretation of the notional structure.

4 Notional models

Definition 4 – A notional model *M* of the subject domain is its notional structure *S* that is supplemented by a description of contents *D* of all notions in it, *M* = (*S*, *D*), where the content of the notion is the set of notion-entities belonging to this notion. ♦

Signs represent entities of elementary notions, and entities of simple notions are described as enumerable or solvable sets of elementary notions. In information systems, elementary notions are encoded by the values of simple data types, and the simple data types themselves are built-in simple notions.

Entities of abstract notions are represented in the form of ordered sets of notions-attributes from its schemas. In information systems, computable functions to define solvable sets are usually used to describe the simple notions.

The notion-entities which are belonged to a notion are typically enumerated in tables whose columns correspond to notion-attributes, whose rows correspond to the entities of the defined notion, and whose fields correspond to the entities of the attribute notions.

To represent a notion-generalization, you can use a virtual table that is generated by a query that selects rows from the others tables with a specified list of common attributes.

It should be noted that the notional model consists only of the notions: the notion-signs, the notion-types, the notion-entities, the notion-generalizations and the notion-associations.

5 Operations on notions

If we abstract from a specific content of actions and procedures in the algorithms for solving applied problems, we can conclude that all such actions might be reduced to three elementary operations on the notions: creation, changing and removal.

The operation of creating a notion occurs when the notional model is complicated and consists in specifying the name of the new notion, the way it is abstracted and the list of notion-attributes.

The operation of changing the notion is used to fill the notion with a specific subject content. In this case, there are three possible actions: editing an existing notion-entity, deleting an existing notion-entity, and adding a new notion-entity to the notion.

The operation of removal occurs in the case of reengineering of the notional model and consists in changing the description of all notions, the definitions of which include the notion to be removed.

To implement the operations of creating, changing or removing in an information system, there might be special procedures that ensure integrity and consistency of the notional model of the subject domain.

It should be noted that any notion itself is a notion and the above operations are applicable to it. However, the notion of a notion is predetermined and only the operation of change is applicable to it. The notion of abstractions (the set of kinds of notions) is also predetermined, but unchanged.

Moreover, the information system itself is a subject domain and has a notional model. This model may include such notions as modules loaded into the client application, events and its handlers, display forms, user interaction scenarios, etc.

To restriction of the operations on the notions, as well as to form the individual notional models, the information system implements a mechanism for determining and inheriting rights.

6 Knowledge representation and inference

Knowledge processing requires forms of knowledge representation and methods of manipulating them to simulate human reasoning and mental operations.

6.1 Declarative knowledge

Usually, facts about the subject domain are used to present knowledge. For the knowledge processing, inference rules are used, which allow, based on the available facts, to infer the new facts about the existing or newly derived facts.

The facts are true propositions with the logical connectives AND ($\wedge$), OR ($\vee$), NOT ($\neg$) and with two types of predicates:

- the one-place predicates of belonging of the notion-entity E to the notion N, $N(E)$;
- the relations $N[E] \circ V$, where $N[E]$ is a functor that returns the value of the attribute N of the notion-entity E, $\circ$ is a relation (=, >, <, etc.), V is some notion-entity.

The equality of the two notion-entities is defined recursively: two notions-entities E_1 and E_2 are equal if and only if the attributes N_i and its values of these notions are equal,

$$E_1 = E_2 \leftrightarrow \forall N_i \in H_1 (\exists N_j \in H_2 \wedge N_i[E_1] = N_j[E_2]),$$

where H_1 (H_2) is the schema of E_1 (E_2), $\leftrightarrow$ is the logic consequence.

The notion-entity belongs to the notion if and only if the set of attribute values of this entity belongs to the notion content,

$$N(E) \leftrightarrow \exists E' \forall N_i \in H \; (N_i[E] = N_j[E']),$$

where H is the schema of N, N_i is an attribute of N.

Any inference can be defined as a transition from one or more facts to a new fact – the consequence of the inference. Rules for constructing inferences are based on the rules generating true proposition under all possible premises.

In the notional model, the inference rules are given in the notional structure, and the notional structure itself is considered as a formal theory, which preserves the truth of all the consequences deduced in it. In this case, we have

$$N_j^G(E) \leftrightarrow \bigvee_{\forall N_i \in H_j^G} N_i(E),\ \ N_j^A(E) \rightarrow \bigwedge_{\forall N_i \in H_j^A} N_i(N_i[E]),$$

where N_j^G (N_j^A) is a notion-generalization (a notion-association), E is a notion-entity, $\rightarrow$ is the logic consequence, N_i is the attribute of the notions, H_j^G (H_j^A) is the schema of N_j^G (N_j^A).

Example 8 – Consider the notional model that describes the staffing structure of a company. In this model, there can be such notions as Trainee, Employee, Position, Division, Vacancy (the notions-generalizations of Trainee and Employee) and Staff (the notions-associations of Division, Position, Vacancy). In the staff world, there are the following propositions:

- *E* (a notion-entity) is Trainee (Employee, Position, Department, Vacancy, Staff),

and the following conclusions:

- if *E* = (*A*, *B*, *C*) is the entity of Staff, then *A* is Division, *B* is Position and *C* is Vacancy;
- if *E* is a Vacancy, then *E* is Trainee or *E* is Employee. ♦

It should be noted that an information system with the notional model implements the open world model, as in the process of inference the monotony is violated.

To turn an information system into a knowledge base, it is necessary to implement the queries for extracting the facts and inferencing the meaningful propositions about the modeled subject domain. The query language and the inference engine of the knowledge base use the inference rules above.

6.2 Procedural knowledge

Declarative knowledge in a 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 models, are the entities of the corresponding notions.

Without limiting generality, the operations through which procedural knowledge is expressed are those of creating, changing, and deleting entities. To represent procedural knowledge, the built-in data type Program is used, whose entities are sequences of commands of some virtual machine generated by Java-style lambda expressions (*X*) -> *Y*, where *X* are arguments (may not exist) and *Y* is the body operator of the function (procedure).

A knowledge language is a universal programming language that has special notional operations to create, modify, and delete entities. A feature of the language namespace is the interpretation of all variables as notions. Any variable is an array of entities. If there is one element in the array, such a variable is an entity, but it is also a notion with one entity belonging to it. Any entity is a structure whose definition is given by a schema of a notion. To represent abstract notions in the structures, key fields from the notional tables in the database are used.

7 Conclusions

Well-known formalisms of knowledge representation use relationships of different nature between concepts. Unlike them, here there is another formalism – a notional structure that is defined by a set of notions the only purpose of which is to show ways of the notion's formation, ways of its abstracting.

The refusal to describe associations in the form of the relationships with different semantic markup makes the notional structure semantically invariant, independence of the subject domains. This effect is because associations between notions in the notional model are notions, and the notional model is built based on the primary mental abstractions of identification, association and generalization that used to form (to define) all necessary notions.

The fundamental difference of the considered approach to the analysis and modeling of subject domains, representation and processing of knowledge is the use of another semantic invariant besides formal logic - the formal theory of notions. It is obvious that the process of abstraction does not depend on any subject domain, but is determined only by the abilities of the cognizing person.

Consequently, the formalization of the ways of formation and expression of the notions can be considered as a theory that claims, as well as the calculus of predicates, to semantic invariance in all "conceivable worlds". Subject semantics is completely defined by the notional model, and the notional structure determines the structure of the content of the notions.

The notional model, like the description logic, is decidable, complete and consistent because it is equivalent to the pure monadic predicate calculus (Mendelson, 2010). However, unlike the description logic, the notional model does not contain inclusions and roles, because all rules required for inference are directly contained in the notional structure.

The advantages of information systems with notional modeling are their semantic invariance, which allows you to scale the information system to different subject domains, while creating large-scale information systems. This allows increasing the level of abstraction and developing an information system, which requires a small number of common algorithms with a small computational complexity. These algorithms do not depend on the subject domain as they are formulated in the universal operations on notions.

The next step in the research of the formal theory of notions is to develop efficient algorithms for executing queries to the knowledge base and to obtain estimates of their computational complexity. However, we can already say that the query execution time will have a good polynomial estimate, because searching for an entity in a notion table with the number of entries n takes time with the asymptotic estimate n, and in case of using an index it takes time with the asymptotic estimate $\log n$.

References

Kogalovsky, M., Kalinichenko, L. (2009) Conceptual and ontological modeling in information systems. *Programming and Computer Software*, 35(5), 241-256.

Borgida, A.T., Chaudhri, V.K., & Giorgini, P. (2009) *Conceptual Modeling: Foundations and Applications*. Berlin: Springer-Verlag.

Vykhovanets, V.S. (2019) Large-Scale Information Systems based on Conceptual Models. *Twelfth International Conference "Management of large-scale system development" (MLSD)*, 2019, 1–4. doi: 10.1109/MLSD.2019.8911106.

Thompson, D. (Ed.). (1993) *The Oxford Dictionary of Current English*. New York, NY: Oxford University Press.

Mendelson, E. (2010) *Introduction to Mathematical Logic*. New York: CRC Press.