Large-Scale Information Systems based on Conceptual Models

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

Abstract— Information systems using conceptual models of the subject domain are considered. The conceptual model consists of the conceptual structure and description of the content of the concepts included in it. The conceptual structure is defined as a set of concepts on which generalization and association map are given. It is shown that due to the semantic invariance of the formalism used, the characteristics of information systems, in particular, their scalability, are improved.

Keywords—information system, subject domain, mental abstractions, identification, association, generalization, conceptual structure, conceptual model, knowledge presentation, knowledge inference.

I. Introduction

Conceptual modeling is becoming increasingly important as a direction in 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 [1].

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. [2].

This work is devoted to the description of one of the types of conceptual models based on fundamental mental abstractions: identification, generalization and association [3].

This allows increasing the level of model abstraction and improving the scalability of information systems from conceptual models.

II. Concepts

Concepts are formed (defined) during mental abstraction. There are four types of concepts: elementary, simple, single and abstract.

Elementary concepts (concept-values, concept-signs) are the result of a mental selection (mental abstraction of identification) of unique perceptions in a subject domain and assigning names to them. Elementary concepts are formed to fix a particular state of the senses or elementary abstract ideas.

Example 1. Examples of elementary concepts are such concepts as Green, Sour, One, Many, Love, Concept, etc.

Simple concepts (concept-features, concept-types) are formed by combining elementary concepts 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 concepts.

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

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

Example 3. Single concept of Ball may aggregate such elementary concepts as Red, Small, Rubber, etc.

Formation (determination) of abstract concepts includes using forms of abstraction based on the establishment of relations of differentiation (for generalization) and integration (for associtation) between the concepts.

Concepts-associations are formed by join several concepts, called concept-attributes (mental abstraction of association). Similarly, the connecting of simple concepts forms a single concept. Therefore, aggregation is a special case of association. Unlike aggregation, not all combinations of entities of attributes can constitute the real entity of the concept-association.

Example 4. The concept of Application may consist of such concepts as User (author of the application), Date (date of creation of the application), Subject (description of the application), etc.

Concepts-generalizations are formed by union several concepts, also called concept-attributes, when new concepts join all or part of the common attributes of the original concepts (mental abstraction of generalization). In the same way, union of elementary concepts forms a simple concept. Therefore, typification is a special case of generalization.

Example 5. An example of the concept of generalization is the concept of Fruit, which is the result of union the entities of such concepts as Apple, Pear, Peach, Apricot, etc. The concept of Fruit has common features of all generalized concepts: Edible, Presence of seeds, etc.

III. Conceptual structure

A conceptual structure is designed to reflect the results of conceptual analysis of the subject domain, expresses displaying some concepts into others. The abstractions of identification, association and generalization are considered as mental operations that are necessary and sufficient for the isolation and transformation into separate concepts of presentations that are accumulated relative to a subject domain.

Let K is predefined set of concept kinds, $K = \{c, t, a, g\}$, where c denotes elementary concept, t denotes simple concept, a denotes concept-association and g denotes concept-generalization.

Let N is finite set of concept names (signs), $N = \{n_i \mid i = 1, 2, \ldots\}$.

Elementary concept $N_i^C \in N^C$ is described by its schema $H_i^C = (c, n)$, where ordered sets are given in parentheses, $c \in K$, $n \in N$, N^C is set of elementary concepts.

Simple concept $N_i^T \in N^T$ has schema $H_i^T = (t, n, \{N_i \in N \mid i = 1, 2, \ldots\})$, where, $t \in K$, $n \in N$, N^T is set of simple concepts.

Concept-association $N_i^A \in N^A$ has schema $H_i^A = (a, n, \{N_i \in N \mid i = 1, 2, \ldots\})$, where $a \in K$, $n \in N$, N^A is set of concepts-associations.

Concept-generalization $N_i^G \in N^G$ has schema $H_i^G = (g, n, \{N_i \in N \mid i = 1, 2, \ldots\})$, where $g \in K$, $n \in N$, N^G is set of concepts-generalizations, $N = N^C \cup N^T \cup N^A \cup N^G$, where $\cup$ is union.

Definition 1. A conceptual structure $S = (N, H)$ is a finite set of concepts names N with finite sets of schemas H, $H = H^C \cup H^T \cup H^A \cup H^G$, where H^C is schemas of elementary concepts, H^T is schemas of simple concepts, H^A is schemas of concepts-associations and H^G is schemas of concepts-generalizations.

Example 6. Let N = {User, File, Program, Principal, Password, Login, String, a, b, …, aa, ab, …}.

Let elementary concepts have the following schemas: (c, a), (c, b), …, (c, aa), (c, ab), …

Simple concept of String has the schema (t, String, {a, b, …, aa, ab, …}).

Concepts-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}).

Concept-generalization of Principal has schema (g, Principal, {User, Program}).

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

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

When you combine multiple subject domains, the concept name is divided into two parts: the concept name itself and the interpretation area name (aspect). To identify any concept requires its name and possibly aspect. The name of concept without aspect defines the concept-generalization uniting all concretizations of this concept in subject domains.

IV. Conceptual model

Definition 2. A conceptual model M of the subject domain is its conceptual structure S that is supplemented by a description of contents D of all concepts in it, $M = (S, D)$, where the content of the concept is the set of concept-entities belonging to this concept.

Signs represent entities of elementary concepts, and entities of simple concepts are described as enumerable or recursive sets of elementary concepts.

In information systems, elementary concepts are encoded by the values of simple data types, and the simple data types themselves are built-in simple concepts.

Entities of abstract concepts are represented in the form of ordered sets of concepts-attributes, the sequence of enumeration of which is given by the sequence of concepts-attributes in the concept scheme.

In information systems, computable functions to define recursive sets are usually used to describe simple concepts.

Abstract concepts-entities are typically enumerated in tables whose columns correspond to attributes, whose rows correspond to the entities of the defined concepts, and whose row fields correspond to the entities of the attribute concepts.

To represent a concept-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 conceptual model consists only of concepts: concepts-values, concepts-features, concepts-entities, concepts-generalizations and concepts-associations.

V. Operations on concepts

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 concepts: concept creation, concept changing and concept removal.

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

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

The operation of concept removal occurs in the case of reengineering of the conceptual model and consists in changing the description of all concepts, the definitions of which include the concept to be removed.

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

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

Moreover, the information system itself is a subject domain and has a built-in and developed conceptual model. This model may include such concepts as module loaded into the client application, event and its handler for creating, deleting, or changing concepts, data input and display form; user interaction scenario, etc.

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

VI. KNOWLEDGE PRESENTATION AND INFERENCE

Knowledge processing requires forms of knowledge presentation and methods of manipulating them to simulate human reasoning.

Usually facts (true propositions) about the subject domain are used to present knowledge. For the knowledge processing, inference rules are used that allow, based on available facts, to receive new facts about existing or newly derived facts.

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

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

The equality of the two concept-entities is defined recursively: two concepts-entities E_1 and E_2 are equal if and only if the attributes N_i and its values of these concepts 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 schema of E_1 (E_2), $\leftrightarrow$ is a logic consequence.

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

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

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

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

In the conceptual model, the inference rules are given in the conceptual structure, and the conceptual structure itself is considered as a formal theory that 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 concept-generalization (a concept-association), E is a concept-entity, $\rightarrow$ is a logic consequence, N_i is attribute of concept, H_j^G (H_j^A) is schema of N_j^G (N_j^A).

Example 7. Consider the conceptual model that describes the staffing structure of a company. In this model, there can be such concepts as Trainee, Employee, Position, Division, Vacancy (concepts-generalizations of Trainee and Employee) and Staff (concepts-associations of Division, Position, Vacancy). In the staff world, there are the following propositions:

- "E (an entity) is a Trainee (an Employee, Position, Department, Vacancy, Staff)",

and the following conclusions:

- "if $E = (A, B, C)$ is the entity of Staff, then A is Division and 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 a conceptual model implements an 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 queries for extracting facts and inferencing meaningful propositions about the modeled domain. The query language and an inference engine uses the inference rules above.

VII. KNOWLEDGE REPRESENTATION

In addition to knowledge presentation, extraction and actualization, another important task must be solved – knowledge representation. It consists in changing the form of knowledge presentation. It is based on construction of conceptual submodels with their subsequent visualization by special programs.

Definition 3. A submodel M' of the conceptual model $M = (S, D)$, is a conceptual model $M' = (S', D')$ with the following relations: $S' \subseteq S$, $D' \subseteq D$, where S' is some substructure (fragment) of the conceptual structure S, D' is a description of concept contents in the conceptual structure S'.

A conceptual submodel is constructed according to the following procedure [3]. First, a number of basic concepts is identified, which must be included in the submodel according to the conditions of the problem being solved. Then the conceptual substructure is iteratively constructed. It includes all concepts having links with the initial and then with the current set of them. Iterations finish when a current set of concepts ceases to replenish itself. At the end of the procedure, a description of contents of the concepts from the conceptual substructure is created.

The construction of conceptual submodels is necessary to create data that is required for visualizing subject domain fragments by third-party programs. To display such submodel, the forms implemented by the corresponding application programs might be used: Gantt charts, resource lists, resource scheduling, figures, slides, videos, etc. For this purpose, an information system includes a module, which performs their visualization based on their conceptual models.

Submodels might be also used to automatically create various kinds of documents (files). In this case, the submodel has expression (representation) rules of concepts in a document body. The expressive means for such representation will depend on the required display form (text, graphics, sound, animation, etc.).

To display a submodel in text form, representation rules might be made in the form of a document template. Template creation involves using a special markup language that allows specifying the forms of concept expression in the text.

The template text consists of arbitrary strings and extraction, calculation, installation, selection, and iteration operators. The extraction operator allows retrieving and inserting into its location a formatted value retrieved from the conceptual model according to the specified path. The calculation operator is used to represent a formatted value of some calculated expression in the text form. The syntax and semantics of expressions are like those of high-level languages. The setting operator is used to change values in the conceptual model and can be used to create temporary simple concepts (variables). The selection operator is required for text branching and the iteration operator – for the representation of composite concepts. Operators can be nested within each other.

Submodels of other stable fragments of subject domains and their corresponding visualization forms have a similar way of creation. For example, charts and diagrams; infographic (graphical representation of charts, maps, figures, formulas, etc.); technical graphics (graphical representation of schemes, drawings, axonometry); dynamic business process models in various notations (graphical representation of processes and their current states).

VIII. CONCLUTIONS

Well-known formalisms define many relationships of different nature on concepts. Unlike them, here there is another formalism – a conceptual structure that is defined by a set of concepts the only purpose of which is to show ways of concept formation, ways of abstracting.

The refusal to describe associations in the form of links with different semantic markup makes the conceptual structure semantically invariant (independence from subject domains).

This effect is because the associations between concepts in conceptual models are concepts, and the model is built based on the identification, association and generalization that used to form (to define) all concepts.

The conceptual model, like description logic, is decidable, complete and consistent because it is equivalent to the calculus of predicates with one argument. However, unlike description logic, the conceptual model does not contain inclusions and roles, because all rules required for inference are directly contained in the conceptual framework.

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 universal operations on concepts.

Thus, the fundamental difference of the considered approach is the use of semantic invariant – the methods of formation and expression of concepts in addition to logic.

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 human.

Consequently, the formalization of the ways of formation and expression of concepts 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 conceptual model, and the conceptual structure determines the structure of the descriptions of the content of concepts.

The advantages of information systems with conceptual modeling are their semantic invariance, which allows you to scale the information system to different subject domains, while creating large-scale information systems.

The use of the information system with conceptual model creates prerequisites for improving the transparency of business processes of the enterprise and reduces the risks of ownership of the information system.

The main difficulty of using conceptual models is the need to master the new methodology and technology of knowledge presentation and knowledge inference.

REFERENCES

[1] M. Kogalovsky, L. Kalinichenko, “Conceptual and ontological modeling in information systems” // Programming and Computer Software. 2009. vol. 35(5). pp. 241-256.

[2] Conceptual Modeling: Foundations and Applications / Borgida A.T., Chaudhri V.K., Giorgini P., et al. Springer-Verlag, 2009. 527 p.

[3] V. Vykhovanets “Information system using conceptual domain model” // Large-Scale Systems Control, 2017, vol. 66, pp. 25–67.