Conceptual Indexing of Project Diagrams in an Electronic Archive

Alexey Namestnikov
Ulyanovsk State
Technical University
Ulyanovsk, Russian Federation
am.namestnikov@gmail.com

Valeria Avvakumova
FRPC JSC «RPA «Mars»
Ulyanovsk, Russian Federation
valeria.avvakumova73@gmail.com

***Abstract*—The objective of the research, results of which are present in the article, is the development of a method for conceptual indexing of semistructured project diagrams. The task arouses interest in development of intelligent repositories of design organizations and allows grouping the realized projects. The proposed method of conceptual indexing is based on the use of the special type of ontologies – the project ones. In essence, an ontology plays a role of a compatible model for knowledge representation in a project organization. The ontology includes semantic description of project diagram notations and design pattern used in design activity. Conceptual indexing of project diagrams are carried with the following steps: defining the project context, defining the subset of the electronic archive diagrams corresponding to the project context, defining a degree of conformity of design patterns from an ontology to the project diagram concerning each element of the obtained subset of the electronic archive diagrams. In case of conceptual indexing of project diagrams representing composite elements of information resources of the electronic archive, both a text component (comments in program code, different instructions, etc.) and elements of semistructured notations of project diagrams are taken into account. Technically, the result of conceptual indexing is represented in the form of a fuzzy hypergraph defined on sets of terms of a domain ontology and terms corresponding to design patterns. The ontology of project diagrams describing semantics of UML class diagrams and including «delegation» as a design pattern in Java.**

***Keywords*—ontology, conceptual indexing, UML, design pattern, project diagram**

I. Introduction

In recent years, researchers in the field of program engineering have been interested in intelligent systems that have their origins in ontological principle of special knowledge representation [1], [2], [3]. The ontological representation of artifacts of software development (models, source codes of program modules) allows to carry out automated analysis of program systems and software intensive systems. Realization of the ontological approach assumes constructing the ontology model taking into account aspects of semistructured modelling notations of program systems and features of architectural solutions or design patterns. The domain context can be extracted from comments texts relating to the source code. The factor of fundamental incompleteness of project diagrams and comments in the source code on the natural language implies the necessity of using corresponding mathematical models in the purpose of formalization. The article proposes to use formalism of fuzzy hypergraphs for such cases.

II. The Formal Model of an Ontology of Project Diagrams

In order to solve the problem of intelligent analysis of project diagrams included in project documentation, it's necessary to have knowledge in the field of constructing the formalized diagrams (using notations). Fig. 1 shows the structure of the fragment of the ontology of project diagrams, particularly, class diagrams in UML (Unified Modelling Language).

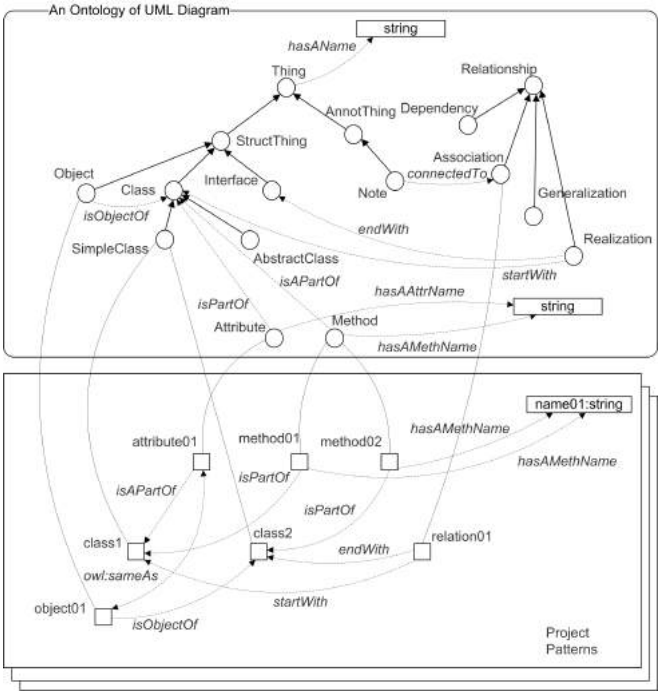

Figure 1. The structure of an ontology of project diagrams (including an example of a design pattern)

Such knowledge allows to identify design patterns used in different projects and, consequently, find projects with similar architectural solutions and approaches to realization of program subsystems of automated systems (AS). Fig. 1

cites a design pattern called «Delegation» as an example [4]. Formally, the ontology of project diagrams can be written as follows:

$$O^{prj} = \{O^{prj}_{dc}, O^{prj}_{tmp_1}, O^{prj}_{tmp_2}, \ldots, R^{prj}, F^{prj}\}, \quad (1)$$

where O^{prj}_{dc} – is an ontology of the UML diagram (a class diagram is used in the working process), $O^{prj}_{tmp_i}$ – is an ontological representation of the i-th design pattern of program systems, R^{prj} – is a relation associating a concept from O^{prj}_{dc} and an instance appertaining to $O^{prj}_{tmp_i}$, F^{prj} – is an interpretation function setting up a correspondance between instances from $O^{prj}_{tmp_i}$ and O^{prj}_{dc} classes.

Let us consider the main components of the UML diagram ontology:

$$O^{prj}_{dc} = \langle C^{prj}, R^{prj}, F^{prj} \rangle,$$

where $C^{prj} = \{c^{prj}_1, \ldots, c^{prj}_l\}$ – is a set of terms defining the main elements of UML class diagram (e.g. «Thing», «Class», «Object», «Interface», «Relationship», etc.); R^{prj} – is a set of relations allowing to construct ontological representations of project diagrams in compliance with the appropriate notations (e.g. relations: defining the name of the diagram element of string type «hasAName», inheritance relation «isA», relation connecting a class with an object of this class «isObjectOf», etc.); F^{prj} – is a set of interpretation functions defining on relations R^{prj}.

The ontological representation of the i-th design pattern of program systems is defined as follows:

$$O^{prj}_{tmp_i} = \{instance(c^{prj}_1), \ldots, instance(r^{prj}_1), \ldots, r_{sameAs}\}.$$

Actually, the ontological representation of an individual design pattern represents a set of terms and relations from the ontology of project diagrams with addition of relation r_{sameAs}, representing the «owl:sameAs» relation embedded in OWL. If the specified relation associates two individuals of an ontological representation of a design pattern, individuals are considered as the same object.

Fig. 1 shows attribute01 and object01 associated by the «owl:sameAs» relation. This means that the specified object of a class (class2) is an attribute of another class (class1).

III. The Method of Conceptual Indexing of Project Diagrams

The objective of the conceptual indexing of project diagrams is calculating an index of electronic archive. In order to solve the task, it's necessary to define instances of ontology classes of project diagrams and calculate the degree of conformity of design patterns to a project diagram for the indexable diagram.

Let us consider the following data as source ones:

- $\{\langle cs_1, dc_1\rangle, \langle cs_2, dc_2\rangle, \ldots, \langle cs_n, dc_n\rangle\}$ – is a set of analyzed projects of an electronic archive, each of them includes the source code cs_i and a class diagram dc_i, i – is the project number;
- the ontology $O^{prj} = \{\langle C^{prj}, R^{prj}\rangle, \{tmp_1, tmp_2, \ldots, tmp_m\}\}$ including a set of concepts of a notation of project diagrams C^{prj} (UML class diagrams as elements), a set of relations between classes R^{prj} and a set of design patterns $\{tmp_1, tmp_2, \ldots, tmp_m\}$;
- $Tz^p = \{\langle t^p_1, f^p_1\rangle, \langle t^p_2, f^p_2\rangle, \ldots, \langle t^p_l, f^p_l\rangle\}$ – is a technical assignment on a new project of an automated system p preprocessed and defined as a number of terms $t^p_1, \ldots, t^p_l$ with corresponding frequencies $f^p_1, \ldots, f^p_l$;
- dc^p – is a project diagram as a part of a new project of the AS (the project diagram corresponds the technical assignment) Tz^p.

According to the scheme of calculating the conceptual index of a design organization, the conceptual indexing of project diagrams in carried as follows:

1) Defining the project context.
2) Defining a subset of the electronic archive diagrams corresponding to the project context.
3) Defining the degree of conformity of design patterns from the ontology O^{prj} to the project diagram dc^p with relation to each element of the obtained subset of diagrams of an electronic archive.

Defining the context of the project is carried out on the basis of the method of conceptual indexing of textual information resources [5], [6]:

$$oV_{tz} = F_{oV}(Tz, O^{dom}, O^{tz}).$$

The input for the function of the conceptual indexing of the textual information F_{oV} is the technical assignment Tz preprocessed, the domain ontology O^{dom} and the thesaurus O^{tz}.

The following set is the result of conceptual indexing

$$oV_{tz} = \{\mu(c^{tz}_1)/c^{tz}_1, \mu(c^{tz}_2)/c^{tz}_2, \ldots, \mu(c^{tz}_k)/c^{tz}_k\} = \mu_{oV}(c^{tz}).$$

The set includes the concept $c^{tz}_i \in C^{dom}$ with the corresponding value of the function of membership of the i-th concept $\mu_i(c^{tz}_i)$ to the technical assignment Tz (the degree of manifestation of the concept in the text of the technical assignment). Let us consider the obtained set oV_{tz} as the realized project concept.

The conceptual indexing is carried analogically for the set of projects of the electronic archive $\{\langle Sc, Dc\rangle\}$. The comment text is extracted for each text of the source code $sc_i \in Sc$:

$$\forall i : tc_i = F_{extcomm}(sc_i),$$

where tc_i is the preprocessed textual representation of the program module sc_i:

$$tc_i = \{\langle t^{sc_i}_1, f^{sc_i}_1\rangle, \langle t^{sc_i}_2, f^{sc_i}_2\rangle, \ldots, \langle t^{sc_i}_s, f^{sc_i}_s\rangle\}.$$

The result of performing a function of the conceptual indexing is the ontological representation of comments of the source code for each program module:

$$oV_{sc_i} = F_{oV}(tc_i, C^{dom}, C^{tz}),$$
$$oV_{sc_i} = \{\mu(c^{sc_i}_1)/c^{sc_i}_1, \mu(c^{sc_i}_2)/c^{sc_i}_2, \ldots, \mu(c^{sc_i}_l)/c^{sc_i}_l\}.$$

Let us define a set of project diagrams of an electronic archive of a project organization compliant with the context oV_{tz} as follows:

$$Dc|_{oV_{tz}} = \{dc_i : \&(\mu_{oV}(c^{sc_i}) \leftrightarrow \mu_{oV}(c^{tz})) \geq 0.5\},$$

where «↔» is the equality operator and «&» is the conjunction operator for all $c^{sc_i}, c^{tz} \in C^{dom}$.

In other words, the specified set includes project diagrams for which the condition of fuzzy equality of ontological representations on source codes of programs and a technical assignment is fulfilled.

Let us consider the process of defining the degree of manifestation of patterns of an ontology of project diagrams. That allows to calculate the conceptual index of a project organization according to UML project diagrams presented in the electronic archive.

Let us denote the degree of membership $\mu_{tmp_j}(dc_i)$ of a project diagram dc_i to a pattern tmp_j. $\mu_{tmp_j}(dc_i)$ is defined analytically as follows:

$$\mu_{tmp_j}(dc_i) = \frac{N(ABox^{prj}_{dc_i})}{N(ABox^{prj}_{tmp_j})},$$

where $N(ABox^{prj}_{dc_i})$ is a number of facts that are true if terminology $TBox^{prj}$ is true, and corresponds to the base of facts $ABox^{prj}_{tmp_j}$; $N(ABox^{prj}_{tmp_j})$ is the number of facts of the tmp_j.

The number of facts $N(ABox^{prj}_{tmp_j})$ of a pattern tmp_j is quite simple to define (by summation of the number of facts of the j-th design pattern), but in order to define $N(ABox^{prj}_{dc_i})$, the following algorithm should be used.

Step 1. Transformation of a project diagram of an electronic archive dc_i to a number of facts $ABox^{prj}_{dc_i}$ of the type shown below:

$$elem^{dc_i}_k : Concept$$
$$\langle elem^{dc_i}_k, elem^{dc_i}_s \rangle : Role,$$

where $Concept$ is the concept defined in $TBox^{prj}$ and $Role$ is the role defined in $TBox^{prj}$; $elem^{dc_i}_k, elem^{dc_i}_s$ are terms instances extracted from a project diagram dc_i.

Step 2. Defining the set of basic classes from $ABox^{prj}_{dc_i}$ regarding to a pattern tmp_j.

The basic class is such an instance $elem^{dc_i}_k$ of a term «Class» (or its subsidiary term «Subclass» from $ABox^{prj}_{dc_i}$, that corresponds to an instance $cls^{tmp_j}_l \in Class$ from $ABox^{prj}_{tmp_j}$ and in case of which a pattern tmp_j includes the maximum number of fact of the following type:

$$elem^{dc_i}_k : Concept$$
$$\langle elem^{dc_i}_k, * \rangle : Role, \quad \langle *, elem^{dc_i}_k \rangle : Role.$$

The obtained set of basic classes of a project diagram dc_i regarding to a pattern tmp_j can be denoted as follows:

$$\{\langle elem^{dc_i}_1, cls^{tmp_j}_1 \rangle, \langle elem^{dc_i}_2, cls^{tmp_j}_2 \rangle, \dots\}$$

where a sequence $\langle elem^{dc_i}_k, cls^{tmp_j}_k \rangle$ means that a term instance of a project diagram $elem^{dc_i}_k$ is equal to an instance of a class of a project pattern $cls^{tmp_j}_k$.

Step 3. Calculating the number of true facts by pairwise replacement of instances of classes of the j-th pattern tmp_j and the i-th project diagram dc_i:

$$\forall k : cls^{tmp_j}_k \leftrightarrow elem^{dc_i}_k. \tag{2}$$

The fact of a design pattern is true regarding to a project diagram if the relation to it can be found in the set of facts of a project diagram taking into account the fact that different term names of instances denotes different individuals. Different names of terms belong to the same ontology instance only when the instances are associated through the relation «owl:sameAs».

The stated steps of the algorithm of conceptual indexing of a project diagram dc_i are carried out for each design pattern of an ontology of project diagrams. As a result, ontological representation of a project diagram of an electronic archive is defined as follows:

$$oV_{dc_i} = \{\mu_{tmp_1}(dc_i)/tmp_1, \mu_{tmp_2}(dc_i)/tmp_2, \dots, \mu_{tmp_s}(dc_i)/tmp_s\}. \tag{3}$$

Practically, the expression (3) represents a fuzzy set based on a set of design patterns of an ontology of project diagrams, where $\mu_{tmp_j}(dc_i)$ is a degree of membership of a project diagram dc_i to a design pattern tmp_j.

IV. The Model of a Conceptual Index of Project Diagrams

The formal structure of a conceptual index of project diagrams is more difficult that the one of textual technical documents. The reason lies in the fact that, in conceptual indexing of project diagrams which are the composite elements of information resources of an electronic archive, both textual component (comments in the source code, different instructions, etc.) and elements of semiformalized notations of a representation language for project diagrams are taken into account.

As in case of document resources, $C = \{c_i\}$, $i \in I = \{1, 2, 3, ..., n\}$ is a finite set of the domain terms fixed in the ontology. A set of patterns of project diagrams in the ontology is denoted as $T = \{tmp_k\}$, $k \in K = \{1, 2, 3, ..., l\}$. A set of project diagrams is denoted as $Dc = \{\tilde{dc_j}\}$, $j \in J = \{1, 2, 3, ..., m\}$ is a family of fuzzy subsets in $C \cup T$. $\widetilde{CI}_{prj} = (C, T, Dc)$ is called a fuzzy undirected hypergraph if $\tilde{dc_j} \neq \varnothing, j \in J$ and $\bigcup\limits_{j \in J} \tilde{dc_j} = C \cup T$; herewith $c_1, c_2, ..., c_n \in C$ and $tmp_1, tmp_2, ..., tmp_l \in T$ are vertices of a hypergraph, a set Dc consisting of $\tilde{dc_1}, \tilde{dc_2}, ..., \tilde{dc_m}$ is a set of fuzzy edges of a hypergraph.

A project diagram has an ontological representation as a result of conceptual indexing. Hence, let us denote a set $Dc = \{\tilde{dc_j}\}$ as a set of project diagrams in the conceptual index, $\tilde{dc_j}$ is an individual ontological representation of the

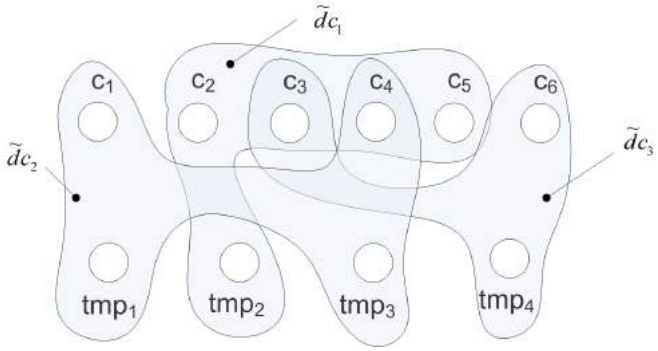

Figure 2. An example of a part of conceptual index

j-th diagram of an electronic archive. The following fuzzy undirected hypergraph

$$\widetilde{CI}_{prj} = (C, T, Dc) \tag{4}$$

defines the conceptual index of a base of projects (Fig. 2) practically.

Two project diagrams $\tilde{dc_\gamma}$ and $\tilde{dc_\delta}$ are called unfuzzy adjacent if $\tilde{dc_\gamma} \cap \tilde{dc_\delta} \neq \varnothing$. The following value

$$\mu(\tilde{dc_\gamma}, \tilde{dc_\delta}) = \bigvee_{c \in (dc_\gamma \cap dc_\delta)} \mu_{dc_\gamma \cap dc_\delta}(c) \quad \& \bigvee_{tmp \in (dc_\gamma \cap dc_\delta)} \mu_{dc_\gamma \cap dc_\delta}(tmp) \tag{5}$$

is called a degree of adjacency of project diagrams $\tilde{dc_\gamma}$ and $\tilde{dc_\delta}$. The value $1 - \mu(\tilde{dc_\gamma}, \tilde{dc_\delta})$ describes the distance between the project diagrams in an information base on the basis of contexts of projects and the degree of membership of a project diagram to design patterns from an ontology.

V. Conclusion

The paper presents the formal model of a project diagram ontology allowing to describe diagram notations at the semantic level and design patterns used in the organization. The developed method of conceptual indexing of semiformalized UML project diagrams allows to reduce the task of information resource analysis to operations with hypergraphs. The fuzzy metrics of distance between ontological representations of project diagrams is used of program subsystems for structuring the content of the electronic archive of FRPC JSC «RPA «Mars» (Ulyanovsk, Russia).

Acknowledgment

The authors acknowledge that this paper was supported by the project no. 16-47-730742 and 16-47-732033 of the Russian Foundation for Basic Research.

References

[1] Wongthongtham P., Pakdeetrakulwong U.,Marzooq S. Ontology annotation for software engineering project management in multisite distributed software development environments. Springer International Publishing, Cham, 2017, pp. 315–343.

[2] Emdad A. Use of ontologies in software engineering SEDE (Hisham Al-Mubaid and Rym Zalila-Wenkstern, eds.), ISCA, pp. 145–150 (2008).

[3] Dillon T., Chang E., Wongthongtham P. Ontology-based software engineering- software engineering 2.0. Australian Software Engineering Conference, IEEE Computer Society, pp. 13–23 (2008).

[4] Mark Grand. Java enterprise design patterns: Patterns in java, John Wiley & Sons, 2002.

[5] Guskov G., Namestnikov A. Ontological mapping for conceptual models of software system: Seventh Conference, OSTIS 2017, Minsk, Belarus, February 18-20, 2017, Proceedings, pp. 111-117 (2017).

[6] Namestnikov A., Filippov A., Avvakumova V. An ontology based model of technical documentation fuzzy structuring. CEUR Workshop Proceedings. SCAKD 2016. Moscow. Russian Federation. Volume 1687, pp. 63-74 (2016).

КОНЦЕПТУАЛЬНОЕ ИНДЕКСИРОВАНИЕ ПРОЕКТНЫХ ДИАГРАММ В ЭЛЕКТРОННЫХ АРХИВАХ

Наместников А.М., УлГТУ
Аввакумова В.С., ФНПЦ АО «НПО «Марс»

Целью исследования, результаты которого представлены в данной работе, является разработка метода концептуального индексирования слабоструктурированных проектных диаграмм. Данная задача представляет интерес при разработке интеллектуальных репозиториев проектных организаций и позволяет на семантическом уровне производить группировку реализованных проектов. Предложенный метод концептуального индексирования основан на использовании специального вида онтологий – онтологии проектных диаграмм. Фактически, онтология выполняет роль согласованной модели представления знаний в проектной организации. Данная онтология включает в себя семантическое описание нотаций проектных диаграмм и применяемых в проектной деятельности шаблонов проектирования.

Концептуальное индексирование проектных диаграмм выполняется посредством следующих шагов: определение контекста проекта, определение подмножества диаграмм электронного архива, соответствующих контексту проекта, определение степени соответствия шаблонов проектирования из онтологии проектной диаграмме относительно каждого элемента найденного подмножества диаграмм электронного архива. При концептуальном индексировании проектных диаграмм, являющихся составными элементами информационных ресурсов электронного архива, учитывается как текстовая составляющая (комментарии в программном коде, различные инструкции и т.д.), так и элементы слабоформализованных нотаций представления проектных диаграмм. Формально результат концептуального индексирования представляется в виде нечеткого гиперграфа, определенного на множествах понятий онтологии предметной области и понятий, соответствующих шаблонам проектирования.

Разработана онтология проектных диаграмм, описывающей семантику диаграмм классов языка UML и включающая представление шаблона проектирования на языке Java «делегирование».