

Models of Class Specification Intersection of Object-Oriented Programming

Dmitriy Buy¹ and Serhiy Kompan¹

Taras Shevchenko National University of Kyiv, Faculty of Cybernetics,
03680 Academician Glushkov Avenue 4d, Kyiv, Ukraine

buy@unicyb.kiev.ua, skompan@mail.ru

Abstract. This paper describes the application of heterogeneous algebraic system for the construction of the formal model of object database instead of object algebra. Complete formalization of the operation of intersection of class specifications is given.

Keywords. object-oriented programming, object database, object algebra, class specification

Key terms. MathematicalModel

1 Introduction

In applications of information technologies there is a problem of construction of the so-called dependable and stable systems and infrastructures – the systems which behave stably under all, especially, critical working circumstances. Similarity of risks and increasing actuality of their decline to an acceptable level for critical applications led to the appearance of a special term “safeware”, by the analogy with the terms “hardware”, “software”, “firmware” etc., which combines two components: *safe* – secure and *ware* – a product, an item. This term was suggested and patented by the leading expert of NASA on the questions of infrastructure security, professor N. Leveson, who registered the appearance of a modern field of knowledge called safeware engineering [1]. We mention a fundamental statement both obvious, and elusive in its nature: it’s impossible to talk about stability of a working system, especially of the infrastructure, if there is no formal model of its operation which has been constructed and verified. Moreover, for the construction of a formal model, more or less complex, not “toylike”, there should exist a mathematical apparatus with the help of which software developers create a formal model and verify it according to the source demands of a customer could.

For the full confidence in the fact that informational system will work stably (will be dependable and stable), one should single out system components, describe them formally and verify. Indeed, nowadays there is nothing instead of a “divide and rule”

approach to cope with this difficulty. In fact, one of the most important components of any complex system (infrastructure) is databases. That's why there should exist an appropriate formal model. For the relational databases such a formal model has been already constructed and explored considerably. This issue is exhaustively covered in the literature, beginning from the pioneering works by E.F. Codd (see, e.g. [2], the first textbooks [3, 4] and modern textbooks [5, 6]). We mention only a collection of works done by the collaborators of Taras Shevchenko National University of Kiev on the natural generalization of classical results of the databases relational approaches [7-14].

Nowadays, there are a lot of formal models of object-oriented databases (OODB) [15-20]. Each of these models elaborates OODB to a certain extent by applying certain mathematical apparatus. The analysis of research papers dedicated to OODB has shown that authors overlook the question arising from the necessity to construct a new class specification with the two given specifications. For example, the construction of a super class from two specified classes (the operation of intersection of class specification), the construction of a subclass from two super classes (the operation of union of class specification). The intersection of class specifications is important, in our opinion, as it provides for the opportunity to construct the core of a new program with two programs which allows integrating these two programs that results in the Framework version. This paper is dedicated to the exploration of the operation intersection of class specifications and refining conditions under which the intersection of classes is possible.

2 Practical results

The authors of this paper have conducted a number of investigations in the field under research: for example, in the article [21] it has been suggested to consider an object algebraic system as a model. Formally it can be formulated like this: $\langle O, K; \Omega_{obj}; \Omega_{spec}, \leq \rangle$, where O is a set of objects' classes, K is a set of class specification, Ω_{obj} is a set of operations over objects, Ω_{spec} is a set of operations over class specifications, and a relation $\leq \subseteq K \times K$ is a partial order which formalizes inheritance. The main objective of this article is specification of the intersection operation \cap and the difference of class specifications.

Let's start with the intersection operation \cap . Let us formalize the notion of a class: by a class we mean a pair $K = \langle s, \mu \rangle$, where s is a functional binary relation which associates an attribute with its meaning (from a universal domain D), and μ is a functional binary relation, which brings to conformity a method with its signature. Therefore [21], the relations s and μ determine a class specification.

The intersection operation (of class specifications) is an operation of the form $\cap: K \times K \rightarrow K$, where: $\langle s_1, \mu_1 \rangle \cap \langle s_2, \mu_2 \rangle = \langle s_1 \cap s_2, \mu_1 \cap \mu_2 \rangle$, where \cap is a standard set-theoretical intersection.

We will demonstrate some results concerning the structure of a partially ordered set (poset) $\langle F, \subseteq \rangle$, where F is a set of all the functional binary relations (on the universal domain X), a \subseteq is an ordinary set-theoretical inclusion. These results will supplement the results of the paper [22]. All undetermined notions and designations are understood in terms of this paper.

Lemma 1. For the arbitrary functional binary relations f and g the following equality is true: $f \cap g = (f \cap g) \upharpoonright (dom f \cap dom g)$ \square

Proof. ■ Let us start with $X \stackrel{def}{=} dom f \cap dom g$. Let us use generally valid properties of the set-theoretical restriction operation (a binary ratio on a set) (monotony, distributivity etc.) [19].

Firstly, we have an inclusion $dom(f \cap g) \subseteq dom f \cap dom g = X$. Secondly, from this the next chain of equalities and inequalities follows:

$$f \cap g = (f \cap g) \upharpoonright dom(f \cap g) \subseteq (f \cap g) \upharpoonright X = f \upharpoonright X \cap g \upharpoonright X \subseteq f \cap g.$$

$$\text{Thus, } f \cap g = (f \cap g) \upharpoonright X = (f \cap g) \upharpoonright (dom f \cap dom g) \square$$

Below \approx is a relation of consistency: $f \approx g \Leftrightarrow f \upharpoonright X = g \upharpoonright X$, where

$X \stackrel{def}{=} dom f \cap dom g$. In [7] the main property of consistency was determined as:

$$f \approx g \Leftrightarrow f \cup g \text{ is a functional binary relation.}$$

The following lemma's corollary forms another criterion of consistency.

Corollary (the criterion of consistency of functional binary relations). Let f, g be arbitrary functional binary relations, and $X \stackrel{def}{=} dom f \cap dom g$. Then: $f \approx g \Leftrightarrow dom(f \cap g) = X$, $\neg(f \approx g) \Leftrightarrow dom(f \cap g) \subset X$. \square

Proof. ■ The proof is performed by using a Lemma 1 and inclusion $dom(f \cap g) \subseteq X$. It's important to note that the second (the first) equivalence is a formal corollary of the first one (of the second one accordingly). \square

As for the structure of the poset $\langle F, \subseteq \rangle$, there are two statements.

Statement 1. Poset $\langle F, \subseteq \rangle$ is a lower semilattice, and at the same time, $\inf\{f, g\} = f \cap g$. \square

The proof results from the fact that F is a commutative idempotent semigroup and from a well-known connection between such semigroups and lower semilattices (see, e.g. [23]).

More complete information about the poset $\langle F, \subseteq \rangle$ is given by the following statement.

Statement 2. (the structure of poset $\langle F, \subseteq \rangle$). The following statements are true:

1. The empty function f_\emptyset is the smallest element ("a bottom")

2. The largest element in poset $\langle F, \subseteq \rangle$ exists if and only if the universe D is singleton
3. The infimum exists for any nonempty set F and $\inf F = \bigcap_{f \in F} f$
4. The supremum of the set F exists if and only if in the case when the set F is restricted, and $\sup F = \bigcup_{f \in F} f$
5. The element f is an atom only when f is singleton
6. Poset $\langle F, \subseteq \rangle$ is a relatively complete poset and a complete (upper) semilattice \square

Let's proceed to the substantial interpretation of above results.

The operation \cap constructs a new class which will be basic (paternal) for classes arguments. This intersection can also be empty, in this case we will get a special empty class.

As the relation \leq on the specifications is component wise

$(\langle s, \mu \rangle \leq \langle s', \mu' \rangle \Leftrightarrow s \subseteq s' \wedge \mu \subseteq \mu')$, all properties of the relation \subseteq (statements 1, 2) can be lifted to the relation \leq . The corresponding formulations are obvious and thereby are omitted.

3 Results and conclusions

The model of intersection operation of class specifications has been examined. This operation has been specified as set-theoretical intersection. The specification $f \cap g$ has been interpreted as the largest total part of f and g , that is, the specification from which specifications-arguments can be obtained by inheritance (in other words, the result specification is the specification of a paternal class). The conditions for nonempty (equivalent, empty) intersection have been examined.

As for formal results, natural criteria of function consistency have been presented (corollary) which supplement the already known criteria; the structure of a partially ordered set of partial functions has been specified (statements 1, 2).

References

1. Kharchenko, V. S.: Safety of Critical Infrastructures: Mathematical and Engineering Methods of Analysis and Ensuring. N.E. Zhukovsky National Aerospace University (2011) (in Russian)
2. Codd, E. F.: A Relational Model of Data for Large Shared Data Banks. Comm. ACM, 13 (1970)
3. Maier, D.: The Theory of Relational Databases. Computer Science Press (1983)
4. Ullman, J., Garcia-Molina, H., Widom, J.: Database Systems: The Complete Book. Prentice Hall Inc., Stanford (2002)
5. Kroenke, D. M.: Database Processing: Fundamentals, Design, and Implementation. Prentice Hall (2011)
6. Date, C. J.: An Introduction to Database Systems. In: Addison-Wesley, (2000)

7. Buy, D. B., Kahuta, N. D.: Full Image, Restriction, Projection, Relationship Compatibility. Theoretical and Applied Aspects of Program Systems Development: International Conference, December 8-10, pp. 244-260 (2009) (in Ukrainian)
8. Buy, D. B., Bogatiryova, J. A.: The Theory of Multisets: Bibliography, Use the Table in Databases. Radio Electronic and Computer Systems, 7(48), 56–62 (2010) (in Ukrainian)
9. Buy, D. B., Polyakov, S. A.: Compositional Semantics of Recursive Queries in SQL-like Languages. Bulletin of Kyiv University. Series. Phys.-Math. Science, 1, 45–56 (2010) (in Ukrainian)
10. Buy, D. B., Glushko, I. M.: Generalized Table Algebra, Generalized Tuple Calculus, Generalized Domain Calculus and Theirs Equivalence. In: Bulletin of Kyiv University. Series. Phys.-Math. Science. 1, 86–95 (2011) (in Ukrainian)
11. Buy, D. B., Puzikova, A. V.: Completeness of Armstrong Axioms. In: Bulletin of Kyiv University. Series. Phys.-Math. Science, 3, 103–108, (2011) (in Ukrainian)
12. Redko, V. N., Brona, J. Y., Buy, D. B., Polyakov, S. A.: Relational Databases: Tabular Algebra and SQL-like Language. AcademPeriodika (2001) (in Ukrainian)
13. Buy, D., Silveystruk, L.: Formalization of Structural Constraints of Relationships in «Entity-Relationship» Model. In: Electronic Computers and Informatics 2006: International Scientific Conference, September 20-22, pp. 96-101, Kosice, Slovakia (2006)
14. Buy, D., Glushko, I.: Equivalence of Table Algebras of Finite (Infinite) Tables and Corresponding Relational Calculi. In: Proceedings of the Eleventh International Conference on Informatics INFORMATICS'2011, November 16-18, pp. 56-60. Rožňava, Slovakia, (2011)
15. Piskunov, A. G.: The Formalization of the Object-Oriented Programming Paradigm, <http://www.realcoding.net/dn/docs/machine.pdf> (in Russian)
16. Piskunov, A. G.: The Formalization of the OOP: Types, Sets, Classes, <http://agp1.hx0.ru/articles/typeSetsClasses.pdf> (in Russian)
17. Chaplanova, E. B.: Operating Specification of Object-Relational Data Model. Radioelektronika, Informatika, Upravlinnya, 12, 75–79 (2011) (in Russian)
18. Richta, K., Toth, D.: Formal Models of Object-Oriented Databases. In: Objekty 2008. Žilina: Žilinská univerzita v Žiline, Fakulta Riadenia a Informatiky, pp. 204-217, <http://www.ksi.mff.cuni.cz/~richta/publications/richta-toth-Objekty2008.pdf> (2008)
19. Sarkar, M., Reiss, S.: A Data Model and a Query Language for Object-Oriented Database. In: Island, Department of Computer Science Brown University Providence, Rhode, CS-92-57, <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.34.4531&rep=rep1&type=pdf> (1992)
20. Gail, M., Shaw, S.: Zdonik A Query Algebra for Object-Oriented Databases. Island, Department of Computer Science Brown University Providence, Rhode, CS-89-19 <http://trac.common-lisp.net/elephant/raw-attachment/wiki/RelationalAlgebra/shaw89query.2.pdf> (1989)
21. Buy, D. B., Kompan, S. V.: Union and Intersection Operations of Classes Specifications in Heterogen Algebraic System for Object-Oriented Programming. In: Proc. SWorld. Int. Sci-Pract. Conf. Modern Problems and Solutions in Science, Transportation, Manufacturing and Education. KUPRIENKO, Odessa, vol. 4, pp. 45–49 (2012) (in Russian)
22. Buy, D. B., Kahuta, N. D.: Properties Related Confinality and Order a Set of Partial Functions. Bulletin of Kyiv University. Series. Phys.-Math. Science, 2, 125–135, (2006) (in Ukrainian)
23. Skornyakov, L. A.: Elements of the Theory of Structures. Nauka, Moskow (1982) (in Russian)