Towards the Conceptual Normalisation

Martin Molhanec[1]

[1] Czech Technical University in Prague,
Faculty of Electrical Engineering, K13113
Technicka 2, Prague 6, Czech Republic
molhanec@fel.cvut.cz

Abstract. This article deal with some considerations of conceptual normalisation and very briefly how can be derived the relational and object normalisation from it. This contribution is only an introduction to this very interesting and serious issue, but infrequently discussed. Nor it is a comprehensive paper covering all particular problems and questions and offering precise mathematical proofs of authors theses presented herein. The idea of this contribution is to be a proper starting point for the discussion of this issue in the frame of international expert community engaging in conceptual, object and data modelling.

Keywords: data normalisation, conceptual, object and relational normal forms, conceptual modelling

1 Introduction

This article deals with an issue of conceptual normalisation. The author engaged in this issue some years ago but the results from his work have been published only at local Czech conferences and seminars [8, 9, 10, 11, 12, 13, 14 and 15] till now. So this article is first attempt to disseminate the author's opinions in wider scope of international expert community engaging in conceptual, object and data modelling. The author hopes that his contribution can be a starting point of wider discussion about these very interesting and serious problems.

2 Motivation and Problem Statement

The paradigm of relational normalisation (1st to 3rd normal form and other not such common as BNCF, etc.) is a very common practise in university courses in the field of database theory and design. No one have a doubt about usefulness and advisability of it. Regrettably, there are many opinions about nonsensicality of the normalisation principle in the field of object oriented paradigm by the reason that object oriented databases do not use the concept of primary and foreign keys. It is a false opinion according with my view.

J. Barjis, M.M. Narasipuram, G. Rabadi, J. Ralyté, and P. Plebani (Eds.):
CAiSE 2010 Workshop EOMAS'10, Hammamet, Tunisia, pp. 133-141, 2010.

It is clear that we do not need the primary and foreign keys for object oriented databases, because the normalization is particularly a process – how we can remove off redundancy from our data model and the definitions of normal forms for relation database with the aid of keys are purely the very proper formulation of this fact in the relational area. This approach is supported by many experts [1, 5, 16, 18 and 19]. But still, there exist some problems:

- There are not any definitively and generally accepted definitions of object normal forms.
- Most of the authors have a problem how to replace the concept of relational keys with another proper concept in the definition of object normalisation.
- It is not clear if there exist in the object area the same count of normal forms as well as in the relational area.

3 Approach

My approach rise from the following conjecture:

> ***Conjecture:*** *Both the relational and object models are specialisations of more common conceptual model; hence both the relational and object normalisations are specialisations of conceptual normalisation as well.*

I would like to mention herein one misguided argument with which I very frequently meet in the course of conversation about the necessity of object normalisation and differences between the object and relational approaches.

> ***False conjecture:*** *The object approach has not and do not need any primary and foreign keys, therefore it is better than the relational approach and its normalisation is thus nonsense.*

This argument along with the idea about normalisation as a something what relates with the primary and foreign keys implicates an idea, that normalisation in the area of object oriented paradigm is nonsense. However, as I mentioned before, the normalisation paradigm is not about the keys, that are in the relational area only proper devices how to define the normal forms, but the normalisation is essentially about a redundancy which rises from absence of redundancy in the real world.

In addition if we reformulate the definitions of normal forms by means of functional dependencies or by means of internal identifiers in the case of the object area we eventually get more or less the same conclusion.

In subsequent text we can show how it is possible to define the conceptual normalisation and only very briefly how to deduce the relational and object normalisation from it as well. But we have to overcome one basic problem – we have not any keys, the same way as in the object area, with the help of them we would very easily realised the definitions of normal forms in the conceptual area.

3.1 Conceptual Model

The first thing we must remember is that the conceptual model has nothing common with computer technologies, programming and databases as well. This is, as mentioned above, a model of concepts occurred in the real world around us. So the conceptual model is an outcome of the ontological observation of the real world and rises from the ontological model of the universe. Consequently I will use only an ordinary conceptual model without a need to deal with ontology too much. In this article I will use only the following concepts.

- *Object*
- *Class*
- *Inheritance*
- *Attribute*
- *Composition*
- *Relationship*

Apart from the fact that we can see herein proposed conceptual model identical with any general object model, we cannot anticipate about this conceptual model any characteristic related to particular real implementation in computer hardware or object oriented programme language.

> ***Conjecture:*** *There is not any unique internal identifier of the object in contrast to any usual object oriented systems.*

This is an elementary assertion. We cannot predicate that any natural world objects have any real internal identifiers. Any unique identification of object such as serial number is an artificial property created by human being just for intent of unique identification. But majority of natural world objects have not any such identification at all. Of course we do not take, for example, into consideration the unique genetic identification. Finally, simple to say, in the real world modelled by means of conceptual model do not exist any simple natural object identifiers.

> ***Lemma:*** *There are not any identical classes in light of its intension.*

We need a brief explanation now. The real world classes are created by different way in contrast to the object oriented programming classes. In usual object oriented programme languages we can created two classes with the same intension, i. e. they will be defined by the same set of attributes and methods and nevertheless they will be comprehend as two different classes by the computer system despite the fact that their behaviour and properties will be identical. In the real world the class as a matter of fact do not exist at all (regardless of the Plato's concept of ideas). The real world class is a concept arising from the human being capability to comprehend and perceive the fact that specific sets of objects are similar to each other. Therefore the conceptual model class is defined solely by its extension; the intension is not exactly obtainable. We can only assign different names to the class, but it is just the same class all the time. We have only different synonyms or other names for it.

3.2 The Base for Conceptual Normalisation

All subsequent considerations used by myself in this article arise from very simple and basic assumption.

Axiom: *The real world has not any redundancy.*

Let appreciate following facts. All objects in the real world exist only in one occurrence. There is only one "Martin Molhanec" (it's me), there is only one occurrence of EOMAS 2010 conference etc. In other words, in the real world are not any two identical instances of the same object at the same time and space. The informational systems, that we used, hold by means of included data a model of the real world, so it is clear that as the real world exist without any redundancy, thus the model of that world has not any redundancy at all.

Of course, it is not the case of warehouses that use redundancy for purpose of achievement certain specific features. Similarly, in database practise we break above mentioned principle to achieve an increasing throughput of database system.

3.3 Conceptual Object and its Features

Herein we introduce certain features of conceptual objects. Let commence with very natural lemma.

Definition: *Property is certain characteristic of object, which has its value.*

An example of such property can be possibly: colour or age. Generally, the property can be set as well. Actually, a relationship to other objects can be a property too. Thus, a property is certain abstraction of object characterisation, constituent of its intension and distinguishable or perceivable by a human being.

Lemma: *A set of object properties is unique in relation to him.*

Really, a car has only one colour property. Of course the colour of car body is another property than the colour of car chassis. These two colours are two different properties. A person has only one age property, denominate "age" etc. A proof can be done with respect to the fact, that property is a concept as well, thus it is abstract object and so the aforementioned lemma is true for them too. In other words, any real car has not two colours or any person has not two ages at the same time.

The following theorems proposed by us relate to object properties as well.

Theorem: *Object property is not dividable, in other words, object property is atomic. If we need to work with a part of it, the part becomes to be an object of one's own, often abstract, with its own properties.*

Theorem: *If we need to work with a group of object properties as if it was one concept in one's, the group becomes to be an object of one's own, often abstract, with its own properties.*

Theorem: *Prospective atomicity of any property depends on domain oriented point of view.*

Herein we explain above mentioned theorems with the aid of very simple example relates to the name of person. If we work, in the domain of our special-interest, always with the person name just like a single property including both first and last person name, we can comprehend this property as atomic and unique in the frame of the concept of person.

But, if we need to work, in the domain of our special-interest, with the first and last name separately, then the person name becomes an abstract concept by itself with two properties (the first name property and the last name property). On the contrary, if the concept of person has two properties, a first and last name, and we need to work with this pair of properties any time jointly, the pair becomes a single concept by itself with its own denomination, in others words, the pair is a new named concept.

Finally, we may not overlook the fact that in the conceptual world it is principally possible to think about any property always by a dual manner.

Lemma: *A property of any object can be perceived simultaneously as another object related to just aforementioned object.*

From this duality and above mentioned lemmas result finally an explication why it is possible to model the real world by many different ways and all that ways can be right.

3.4 Property, Object and Class

Herein we define a conceptual class as follows. (I strongly note that we deal with conceptual classes and not with programmer classes.)

Definition: *The conceptual class is a named set of properties belonged to each other similar object.*

We must be aware that in the real world the class is only an abstract concept denominating a set of abstract and real objects which are similar to each other. Thus the class named "car" is only a notation of the set of objects with such similar properties that we have a need to have a common notation for them. This notation is the name of the class. We notate, the conceptual class do not exist as a real world object at all.

Herein we submit another two definitions:

Definition: *An intension of the class is a set of all possible properties of this class.*

Definition: *An extension of the class is a set of all possible objects of this class.*

In contrast to the world of programmers where is possible very easily to define the intension of certain class and the extension of this class results from this definition. In the real world exist only real natural objects which constitute the extension of similar objects and the intension is constructed by us on the basis of investigation of their similarities.

3.5 Object, Property, Class and Relationship

Up to now we have not deal with relationships yet. In agreement with an ontological approach [3] we recognize in conceptual model proposed by us four different types of relationships.

- *Relationship between the class and its property.*
- *Relationship of inheritance between classes.*
- *Relationship which represents a composition of classes.*
- *General relationship between objects.*

The above mentioned relationships are generally used at different modelling techniques with exception of the first which is implicitly supposed. It is worth mentioning that although we named all above mentioned relationships with one common lexical label, namely "relationship", they are essentially different. Actually, all that relationships present unrelated ontological concepts and common denomination of them can be very misguided.

4 Results – Definitions of Conceptual Normal Forms (CNF)

Herein we present our definitions of conceptual normal forms understood by us as the rules of redundancy prohibition already introduce above and here altogether mentioned again.

0. CNF: The real world has not any redundancy.

1. CNF: A set of object properties is unique in relation to him.

2. CNF: Object property is not dividable, in other words, object property is atomic. If we need to work with a part of it, the part becomes to be an object of one's own, often abstract, with its own properties.

3. CNF: If we need to work with a group of object properties as if it was one concept in one's, the group becomes to be an object of one's own, often abstract, with its own properties.

The grounds for these conceptual normal forms have been introduced hereinbefore. I believe that the relational and object forms can be derived from these more common conceptual forms as can be very briefly shown in hereinafter text.

4.1 Relational Normal Forms (RNF)

The author of this article suggests that 1. RNF can be substantiating by our 0. to 2. CNF. The basis for this suggestion laid in the fact that 1. RNF deals with atomicity of data attributes, prohibition of multi-attributes and necessity of primary key existence. Evidently the issue of atomicity relates to herein proposed 2. CNF, the prohibition of multi-attributes results from the 1. CNF and issue of necessity of primary key existence relate to very principal 0. CNF.

According to author's humble opinion the 2. RNF is a specialisation of more common 3. RFN, but more detailed discussion of this issue is outside this article for now. That means that both the 2. and 3. RFN follow from the above suggested 3. CNF. The evidence can be done on the basis of consideration that transitive dependency between relational keys at the level of relation data paradigm is simply an implication of incorrect recognition of conceptual object at higher level of comprehensibility. It is worth to note that the concept of relational keys in relational database systems by self presents only the programmer implementation of the concept of functional dependency by implication of mutual relationships among conceptual objects. Thus, the incorrect recognition of object at conceptual level leads to transitive dependency between relational keys in the relational level.

4.2 Object Normal Forms (ONF)

At this time there is not any standard and commonly accepted concept of object normal forms, notwithstanding there are many scientists engaged in this issue. The brief summary of them is included in [7]. This work contains definitions of object normal forms based on an approach introduce originally by Ambler in [1] and further elaborated in aforementioned work by Merunka and Molhanec.

Despite the fact that the authors deal with the object oriented paradigm, the definitions of object normal forms are simply based on analogical relational forms, i. e., 1. ONF is based on 1. RNF and so on. The fundamental difference lays in the fact, that the definitions of all ONF are constructed without the use of the relational keys of course. Thus, the presented definition of ONF at aforementioned work is very similar to the definitions of CNF proposed by us herein and we can apply the same or very similar argumentation for their reasoning.

5 Conclusion

The author of this article suggests that relational as object normalisation arises from the same source i.e. from the conceptual normalisation. Surprisingly, this principal and serious subject matter is not widely discussed in expert community at all. In addition, the need of object normalisation that directly arises from the conceptual normalisation is often disputed.

The author believes that his contribution in this very interesting and serious subject matter can be a good starting point for the discussion of this issue in the frame of international expert community engaging in conceptual, object and data modelling.

References

1. Ambler Scott: Building Object Applications That Work, Your Step-By-Step Handbook for Developing Robust Systems Using Object Technology, Cambridge University Press/SIGS Books, 1997, ISBN 0521-64826-2
2. Edgar F. Codd, Wikipedia, online: [http://en.wikipedia.org/wiki/Edgar_F._Codd]
3. Heller, B., Herre, H. 2004. „Ontological Categories in GOL.“, Axiomathes 14(1):57-76 Kluwer Academic Publishers.
4. Christopher J. Date, Wikipedia, online: [http://en.wikipedia.org/wiki/Christopher_J._Date]
5. Khodorkovsky V. V.: On Normalization of Relations in Databases, Programming and Computer Software 28 (1), 41-52, January 2002, Nauka Interperiodica.
6. Merunka, Vojtěch, „Datové modelování“, ALFA Publishing, Praha 2006. ISBN 80-86851-54-0.
7. Molhanec, M., Merunka, V., Object normalization as the contribution to the area of formal methods of object-oriented database design, In: Advances in Computer and Information Sciences and Engineering. Berlin: Springer Science+Business Media , 2007, p. 100-104. ISBN 978-1-4020-8740-0.
8. Molhanec, M., Ontologické základy konceptuální normalizace, In: Objekty 2006. Ostrava: Technická universita Ostrava - Vysoká škola báňská, 2006, s. 81-90.
9. Molhanec, M., Konceptuální modelování, formální základy a ontologie, In: Tvorba softwaru 2006. Ostrava - Poruba: VŠB - Technická univerzita Ostrava, 2006, s. 121-127. ISBN 80-248-1082-4.
10. Molhanec, Martin. „Konceptuální modelování“, In Objekty 2005, Ostrava: VŠB, Technická Universita, 2005. ISBN 80-248-0595-2.
11. Molhanec, Martin. „Zásady konceptuálního totálně objektově orientovaného modelování“ In: Tvorba softwaru 2005. Ostrava: VŠB, 2005, s. 153-158. ISBN 80-86840-14-X.
12. Molhanec, Martin. „Několik poznámek k porozumění objektového paradigmatu“. Sborník konference Objekty 2004, Praha. ČZU PEF 2004, s. 189-197. ISBN 80-248-0672-X.
13. Molhanec, Martin. „Kritika některých výkladů objektově orientovaného paradigmatu“. Sborník konference Tvorba software 2004. Ostrava. Tanger, 2004, s. 163-172. ISBN 80-85988-96-8.
14. Molhanec, Martin. „Objektové metodologie – jejich užití a výklad“. Sborník konference Tvorba software 2003. Ostrava. Tanger, 2003, s. 111-115. ISBN 80-85988-83-6.
15. Molhanec, Martin. „UML – několik kritických poznámek“. Sborník konference Tvorba software 2002. Ostrava. Tanger, 2002, s. 150-159. ISBN 80-85988-74-7.
16. Nootenboom Henk Jan: Nuts - a online column about software design. http://www.sum-it.nl/en200239.html

17. Opdahl, A. L., Henderson-Sellers, B., Barbier, F., "Ontological analysis of whole–part relationships in OO-models", Information and Software Technology, Volume 43, Issue 6, 1 May 2001, Pages 387-399
18. Tari Zahir, Stokes John, Spaccapietra Stefano: Object Normal Forms and Dependency Constraints for Object-Oriented Schemata, ACM Transactions on Database Systems 513-569, Vol 22 Issue 4, December 1997.
19. Yonghui Wu, Zhou Aoying: Research on Normalization Design for Complex Object Schemes, Info-Tech and Info-Net, vol 5. 101-106, Proceedings of ICII 2001, Beijing.