



INTERNATIONAL CONFERENCE INFORMATION AND COMMUNICATION TECHNOLOGIES IN BUSINESS AND EDUCATION



CONFERENCE PROCEEDINGS



Publishing house "Science and economics"
University of Economics - Varna

International Conference
INFORMATION AND COMMUNICATION
TECHNOLOGIES IN
BUSINESS AND EDUCATION

Conference proceedings

**INFORMATION AND COMMUNICATION
TECHNOLOGIES IN
BUSINESS AND EDUCATION**

**Proceedings of the International Conference
dedicated to the
50th anniversary
of the Department of Informatics**

2019

University publishing house “Science and economics”
University of Economics – Varna

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Permission for use must always be obtained from University publishing house “Science and economics”. Please contact izdatel@ue-varna.bg

Published reports have not been edited or adjusted. The authors are solely responsible for the content, originality and errors of their own fault.

With the support of ADASTRA, 411 Marketing Business Services.

© University publishing house “Science and economics”,
University of Economics – Varna, 2019.

ISBN 978-954-21-1004-0

PROGRAMME COMMITTEE

Abdel-Badeeh M. Salem, Egypt

Aijaz A. Shaikh, Finland

Aleksandar Dimov, Bulgaria

Alexander Feldman, USA

Alexander Sergeev, Russia

Antoanela Naaji, Romania

Avram Eskenazi, Bulgaria

Dariusz Mrozek, Poland

Dimitrina Polimirova, Bulgaria

Dmitriy Berg, Russia

Elena Nechita, Romania

Emil Denchev, Bulgaria

Georgi Dimitrov, Bulgaria

Ivan Popchev, Bulgaria

Jeanne Schreurs, Belgium

Julian Vasilev, Bulgaria

Kameliya Stefanova, Bulgaria

Krasimir Shishmanov, Bulgaria

Maciej Czaplewski, Poland

Maciej Pondel, Poland

Maksim Medvedev, Russia

Maria Hristova, Bulgaria

Marian Cristescu, Romania

Marina Medvedeva, Russia

Michael Voskoglou, Greece

Nadezhda Filipova, Bulgaria

Nikola Nikolov, Ireland

Niv Ahituv, Israel

Pavel Petrov, Bulgaria

Rami Malkawi, Jordan

Sara Paiva, Portugal

Silvia Parusheva, Bulgaria

Snezhana Sulova, Bulgaria

Todorka Atanasova, Bulgaria

Valentin Kisimov, Bulgaria
Valery Trofimov, Russia
Veselin Popov, Bulgaria
Violeta Kraeva, Bulgaria
Vladimir Sulov, Bulgaria
Vladimir Zanev, USA
Zdzislaw Polkowski, Poland

ORGANIZING COMMITTEE

Chairman: Prof. Vladimir Sulov, PhD

Members:

Prof. Avram Eskenazi, PhD
Prof. Julian Vasilev, PhD
Assoc. Prof. Todorka Atanasova, PhD
Assoc. Prof. Silvia Parusheva, PhD
Assoc. Prof. Snezhana Sulova, PhD
Assoc. Prof. Pavel Petrov, PhD

Secretary: Tijen Talib

FOREWORD

These proceedings contain the papers of the International Conference “Information and Communication Technologies in Business and Education” which took place at the University of Economics – Varna, Bulgaria, 18 October 2019.

The international scientific conference is dedicated to the **50th anniversary of the Department of Informatics at the University of Economics – Varna**. The conference is also dedicated to the 100th anniversary of the University. The included papers describe recent scientific and practical developments in the field of information and communication technologies, information systems, and their applications in business and education.

The papers in the Proceedings are peer reviewed and are checked for plagiarism.

C O N T E N T S

1. Kamelia Stefanova, Dorina Kabakchieva Challenges and Perspectives of Digital Transformation	13
2. Julian Vasilev, Marian Cristescu Approaches for information sharing from manufacturing logistics with downstream supply chain partners	24
3. Abdel-Badeeh M. Salem Computational Intelligence in Smart Education and Learning	30
4. Marian Niedźwiedziński Proposal of the Multi-aspect Analysis of ICT's Needs in Management	41
5. Krasimir Shishmanov Digital Transformation in Bulgaria	52
6. Tatyana Makarchuk, Valery Trofimov, Svetlana Demchenko Modeling the life cycle of e-learning course using Moodle Cloud LMS	62
7. Vanya Lazarova Exploring the functionality of Bulgaria eGovernment site	72
8. Veselin Popov, Petya Emilova Big data and information security	83
9. Luben Boyanov Approaches for enhancing digitalization and digital transformation in supply chain management	91

10. Monika Tsaneva	
Challenges of GDPR compliance in consumer financing companies	103
11. Todorka Atanasova	
Main factors influencing digitization in construction companies	116
12. Silvia Parusheva	
Digitalization and digital transformation in construction – benefits and challenges	126
13. Snezhana Sulova	
The Usage of Data Lake for Business Intelligence Data Analysis	135
14. Pavel Petrov	
Web security technologies used in banks of Estonia, Latvia and Lithuania	145
15. Tanka Milkova	
ABC analysis of inventory in MS Excel	155
16. Ivan Kuyumdzhiev, Radka Nacheva	
Choosing Storage Devices with Consideration of Backup and Restore Performance in MS SQL SERVER	166
17. Yanka Aleksandrova	
Predicting Students Performance in Moodle Platforms Using Machine Learning Algorithms	177
18. Latinka Todoranova, Bonimir Penchev	
M-learning Applications	188
19. Boris Bankov	
The Impact of Social Media on Video Game Communities and the Gaming Industry	198

20. Velina Koleva, Svetoslav Ivanov	
Social media and the recruitment of IT professionals in Bulgarian IT companies	209
21. Deyan Mihaylov	
Modeling and Simulation of Some Functions of Two Independent Random Variables	225
22. Oumar Sy	
Using Association Rules as Semantics for Domain Ontologies	233
23. Ina Stanoeva	
Social media possibilities for improving arts marketing performance	247
24. Iskren Tairov, Vladislav Vasilev	
Perspectives on mobile devices adoption in healthcare sector	257
25. Angelin Lalev	
Deep Neural Networks for Detection of Credit Card Fraud	263
26. Miglena Stoyanova	
Priorities for Digital Transformation in Property Management	275
27. Mihail Radev	
Organizational Variants of IT Department at the University	284
28. Mariya Armyanova	
IoT problems and design patterns which are appropriate to solve them	291
29. Yavor Christov	
Corporate Big Data and Hybrid Integration Platforms	306

30. Ivan Belev	
Making financial data readable using Inline XBRL	314
31. Ivan Belev	
eXtensible Business Reporting Language (XBRL) – a technical overview	322
32. Anna Sobczak	
Entrepreneurship in the education of university students	331
33. Mihaela Markova	
E-commerce in the process of digital transformation: preparedness, requirements and technological tendencies	343
34. Petya Petrova	
Basic Soft Computing Methods In User Profile Modeling	353
35. Petar Dimitrov	
Methods for storing authentication data – a historical review	369
36. Svetoslav Ivanov	
Development stages of starting software company, problems and approaches for software development	382
37. Antonio Hadzhikolev	
The role of the smart city concept in the process of urban transformation	397
38. Oumar Sy	
Ontology-based materialized views for reverse engineering: Toward a set of conceptual guidelines	410

ONTOLOGY-BASED MATERIALIZED VIEWS FOR REVERSE ENGINEERING: TOWARD A SET OF CONCEPTUAL GUIDELINES

*Oumar Sy*¹

¹ *Université Gaston BERGER/U.F.R
Sciences Appliquées et de Technologie (Comp. Sc.),
Saint-Louis, Sénégal, oumar.sy@ugb.edu.sn*

Abstract

Data reuse and meta-data handling remain a tricky problem for both the designers and managers, especially when schemas reverse engineering is on demand and that some data are stored in materialized views. In this paper, we tackle such problem by using ontology-based meta-materialized views. Indeed, ontologies, which are semantics-based, ensure the stability of the underlying schemas of the data repositories and to ease the overall access and processing of the data and meta-data. Our proposal is sustained by a set of conceptual guidelines and outlined through a case study example.

Keywords: *Relational database, conceptual modeling, domain ontology, reverse engineering, materialized view, SQL.*

INTRODUCTION

The reuse of data and meta-data remain a tricky problem, especially when the reverse engineering of schemas is on demand. Yet, legacy systems such as traditional databases and the advanced databases systems, and the ontologies, must permanently be up to date. In this context, reverse engineering is useful for the understanding of existing systems. Already, in the early 90s, CASE (Computer-Aided systems-Engineering) tools were used to maintain and enhance existing systems (Chikofsky and Cross, 1990). Now, according to Gomez-Perez and Ma (Gomez-Perez and Ma, 1999, pp. 139-156), reverse engineering consists of producing a semantic model based on the code in which the data model or the knowledge model (e.g., an ontology) is implemented. On the other hand, the use of ontologies for creating more intelligent and effective enterprise information systems has increased considerably in recent years (Reynares, Caliusco, Galli, 2015). Indeed, ontologies allow the

specification of high-level formal conceptualizations (Gomez-Perez, 1999, pp. 6.1.1 - 6.1.18). However, most of the proposed approaches are not domain independent, such that the developed models are generally specific to a given domain of real world. Therefore, even for the same field of application, the reuse is not easy to achieve or not achieved at all because the common trend is to approach the problem of reverse engineering with object-oriented techniques, e.g., (Reynares, Caliusco, Galli, 2015; Gomez-Perez, 1999; Ristić, Aleksić, Celiković, Dimitrieski, Luković, 2014; Zhang, Karcher, 2012; Müller, Jahnke, Smith, Storey, Tilly and Wong, 2000; Shatnawi, Seriai, Sahraoui, Alshara, 2016; Ali, Fernandes, Paton, 2000; Brenner and Glimm, 2017; Koch, Lupei, Tannen, 2016). Conceptual modelling approaches of ontologies for data integration and reuse have been well developed in (Ristić, Aleksić, Celiković, Dimitrieski, Luković, 2014; Alberts & Franconi, 2012; Zedlitz and Luttenberger, 2014), but materialized views were not considered.

Here, we tackle the problem from the semantic models point of view and mainly when ontologies are considered. Indeed, to the best of our knowledge, only (Glimm, Kazakov, Liebig, Tran, and Vialard, 2014, pp.180-195; Volz, Staab & Motik, 2003, pp. 707-724; Volz, Staab & Motik, 2005, pp.1-34) have dealt with the problem of ontologies' materialization. However, rather than the use of materialized views with ontologies for reverse reengineering, the authors only discussed ontologies' storage, without conceptualization of materialized views in the design process. Moreover, it is not clear how ontologies' models are linked to business-data. Thus, we aim to address criticisms on the above works in the related works.

In the 1990's, Chikofsky and Cross (Chikofsky and Cross, 1990) argued that the term "reverse engineering" originate from the analysis of hardware, and defined six terms, namely (i) forward engineering, (ii) reverse engineering, (iii) redocumentation, (iv) design recovery, (v) restructuring and (vi) reengineering. The relationships between these conceptual terms have been rigorously related and defined based on a three-level's abstraction architecture, conceptually quite similar to the ANSI/X3/SPARC architecture dedicated to database schemas design. Such an architecture stands as the cornerstone in our approach, when aiming

to conciliate databases (DB) and ontologies, without redefinition of terms, concepts and methodologies (Sy, Lo, and Duarte, 2010, pp. 269-276; Sy, Duarte and Dal Bianco, 2018). In (Müller, Jahnke, Smith, Storey, Tilley & Wong, 2000, pp. 47-60) ontologies are not mentioned. Despite efforts made in the field of reverse engineering (Gómez-Pérez and Ma, 1999, pp. 139-156), ontologies' storage within DB repositories has not been sufficiently investigated. In a previous work (Sy, Duarte and Dal Bianco, 2018), the meta-model for DB and ontologies designed in a single lifecycle can efficiently ensure the reverse engineering of both DB and domain ontology by using materialized views because of its high semantic-based abstraction degree. Indeed, the better way of achieving this is to store the roles' names in the underlying database (Sy, Duarte and Dal Bianco, 2018), even though UML does not provide an "explicit marking of two arbitrary associations as inverses of each other" (Zedlitz and Luttenberger, 2014).

Running example: Let $U = \{R, E, L\}$ where, respectively, R, E, and L represent Rice Seed, Cereal Seed and Plot-land, and U is the universe of discourse of the application domain. Let VARIETY (V), STAMP (S), PURITY (P) and GERMINATION (G) be the properties of E, as shown at Figure 1.

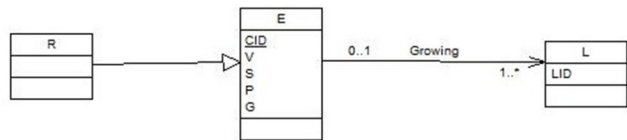


Figure 1. UML-based semantic model of the universe U

Knowing the value of S and P, the database manager may update column G of the underlying table E. using the following instruction:

```
UPDATE E SET G=93 WHERE (S='J') AND (P='A');
```

Moreover, knowing that rice seed is a cereal seed, one would like to propagate those update to all data related with the updated table. This is feasible thanks to the capabilities of relational databases' management systems (RDBMS) through triggers, based on keys and foreign keys. A general and simplified syntax of such triggers is as follows:

```

CREATE TRIGGER <trigger_name>
BEFORE | AFTER INSERT | UPDATE | DELETE
ON <table_name> FOR EACH ROW
<trigger_body>

```

Therefore, let R_i ($K_i, A_i, \dots, B_i$) and R_j ($K_j, B_j, \dots, B_j, \#K_i$) be two n -ary relations that represent the concepts C_i and C_j of a given UML-based semantic model, with $C_i \leq C_j$ meaning that C_j inherits of all the properties and features of C_i . Clearly, $T_j \subseteq T_i$, where T_i and T_j are respectively instances of R_i and R_j .

Now, let denote by M_v (CID, V, G, LID) a materialized view where LID represents the plot-land identifier. Accordingly, a state of M_v contains the instances of all cereal seeds and plot-lands where they have been seeded.

The posed problem is how reverse engineering – rebuilding – the UML-based semantic model of the universe U . Without loss of generality, the most reliable solution resides on the use of ontologies; with the assumption that there exists a well-designed meta-model that link the DB to the ontologies. Accordingly, we propose here a set of conceptual guidelines for schemas' reverse engineering by using ontology-based materialized views (OBMMV4RE).

The rest of the paper is organized as follows. In Sections 1, 2 and 3, we give a theoretical background comprising conceptual foundations of the notions of view and of materialized view, the notion of ontology, and their use in schemas reverse engineering process and reengineering, as well. Section 4 presents the related work. In Section 5, we discuss the OBMMV4RE approach based on a set of conceptual guidelines. A case study is presented in section 6. Finally, in Section 7, we conclude and present the direction of our future work.

1. DATA VIEW VS. MATERIALIZED VIEW

A typical data view is a simple named SPJ query to be executed by the RDBMS. Such named query, consisting of a set of n -ary tuples, is each time accessed by a user, and recomputed. For example, the code (under PostgreSQL) below returns, if any exists, the name N , the germination G and the cultivated area of all rice seeds and the plot of lands where they are growing.

```
SELECT "N" As "Name", "LID" AS "Plot land", "G" As Germination, "Cultivated_area" FROM "Seed", "Cultures"
WHERE "Seed"."CID"="Cultures"."CID";
```

The relational data view (Codd, 1970, 377-387) is shown at Figure 2.

Data Output				
	Name text	Plot land integer	germination integer	Cultivated_area numeric (6,2)
1	Oryza sativa	3	92	7.00
2	Oryza sativa	1	95	3.50
3	Oryza sativa	2	95	2.00
4	spanish	4	93	1.00

Figure 2. Data view dvGrowingSeeds

Now, let assume that a new tuple of an existing seed variety, e.g. corn, has been sowed in the plot of land number 5. Thus, whenever these data are added, to know the current state of the cultivated cereals and the related data (e.g., plot of lands and cultivated areas), the user must re-run the query dvGrowingSeeds above. Yet, according to rule 6, ‘all views that are theoretically updatable must be updated by the system’ (Codd, October 14, 1985; Codd, October 21, 1985; Rick, 2014). Thus, materialized views can be useful by caching the result of queries and allow refreshing them, once a user needs to access their content.

As example, the dvGrowingSeeds data view, theoretically updatable, and that we call mvGrowingSeeds, is materialized as follows:

```
CREATE VIEW public."mvGrowingSeeds"
WITH (check_option=local) AS
SELECT "N" As "Name", "LID" AS "Plot land", "G" As Germination, "Cultivated_area" FROM "Seed", "Cultures"
WHERE "Seed"."CID"="Cultures"."CID";
```

The above view is easily maintainable up to date based on foreign keys and triggers. However, difficulties can occur for the practical up-datability with respect to the data and meta-data. Moreover, when facing with more complex queries, which are frequent in the advanced and sophisticated information systems such as the data warehouses and or Business Intelligent environments, the answer may take more time. Thus, novelty in the methodologies, rather than in the concepts, is neces-

sary for efficient solutions. Accordingly, our proposal is the use of ontology-based materialized views not for overcoming the complex queries but especially for ensuring reuse and reverse engineering for recovering or improving conceptual schemas.

As example, let us assume that seed of wheat (W) have to be added (See Figure 3, dashed line) and, by the same time, a new tuple of sowed wheat is added. Given the sub-schema that describes the context, mvGrowingSeeds is still updatable, and from it is possible to rebuild the semantic model of Figure 1.

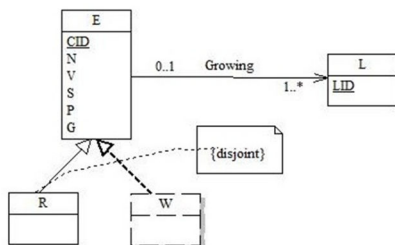


Figure 3. UML-based semantic model of the universe U (updated)

2. REVERSE ENGINEERING

According to Rick (Rick, 2014), there is no universally accepted definition of software re-engineering. However, there is a confusion between reengineering and ‘reverse engineering’ (Chikofsky and Cross, 1990). Indeed, ‘reverse engineering’ originates from the analysis of hardware. Precisely, it was defined as the ‘‘process of developing a set of specifications for a complex hardware system by an orderly examination of specimens of that system’’ (Chikofsky and Cross, 1990). In the field of software systems, several formulations have been proposed for defining the notion of reverse engineering. As examples, in (Gómez-Pérez and Ma, 1999, pp. 139-156), reverse engineering is defined as ‘‘a process of analysing a subject system to identify the system’s components and their interrelationships and to create representations of the system at a higher level of abstraction’’. In (Chikofsky and Cross, 1990), it is defined as ‘‘the process of analysing an existing system in order to identify its components and their interaction, illustrating it in a different

or more abstracted form''. According to (Reynares, Caliusco and Galli, 2015), a reverse engineering approach includes the extraction of information from source codes, and documentations; the abstraction of the extracted information; and the representation of the above abstraction.

In this work, we consider reverse engineering as the process, which aims to produce a conceptual model based on the code in which the ontology has been implemented. Moreover, we emphasize that reverse engineering is the opposite process of forward engineering that is traditional process for schemas design starting from the requirements analysis to the internal level. For a better understanding, borrowing the schema given in (Chikofsky and Cross, 1990), these two processes are illustrate at Figure 4.

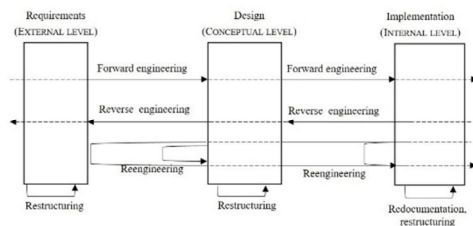


Figure 4. Reverse engineering and related processes between or within abstraction levels

Accordingly, reengineering is a process between or within abstraction levels while reverse engineering is a process that covers the entire three-level architecture, starting from the internal level up to the views' level.

3. ONTOLOGY-BASED MATERIALIZED VIEW

In the Web Semantic community literature, ontology is defined as a "formal, explicit specification of a shared conceptualization" (Studer, Benjamins and Fensel, 1998, pp. 161-197). However, there is no consensually accepted definition of the notion of ontology. Thus, in this paper, an ontology is understood as "a formalization of the universe of the discourse as an organized structure, and constrained by a set of axioms, according to the knowledge domain" (Sy, Duarte and Dal Bianco, 2018). More precisely, such formalization is the set $O = \{\Sigma, \tau \leq, A, \Omega, \rho, \phi\}$,

where: Σ is a set of concepts $\{C_1, \dots, C_n\}$ assigned with a taxonomic partial order relation $\tau \leq C_i, C_j$; Aa finite set of attributes $\{A_i, \dots, A_m\}$ describing the concepts $C_i \ 1 \leq i \leq n$; Ω is a set of relations; ρ a relation over Ω assigning to each $R_i \in \Omega$ a domain $\rho \text{ dom}: R_i \rightarrow \Sigma \times \Sigma$ and a range $\rho \text{ range}: R_i \rightarrow \Sigma \times \Omega$; and ϕ a set of axioms which hold with the concepts in Σ and the relations over Ω .

Accordingly, an ontology-based materialized view (OBMV), can intuitively be seen as a view built on the basis of O such that it contains semantic-based knowledge, e.g. rice seed is a cereal seed, or rule-based knowledge, e.g. a plot of land is a “rice crop”, a “farm land” or a “research station”. Now, since views are built upon concept-based relations over Ω which involve attributes of A , let F_c be the FROM clause of the SPJ query. We define an OBMV as follow.

Definition – A view V materialized with the CREATE VIEW instruction is an OBMV if and only if F_c satisfies the following condition:

$F_c \supset R_i \in \Omega \wedge C_j \in \Sigma \mid \tau \leq C_i, C_j \vee \tau \leq C_j, C_i$. i.e. F_c contains at least one concept-based relation and it exist a concept C_j such that C_j subsumes C_i or conversely.

Proposition 1 – Based on the properties of the subsumption relation $\tau \leq$, an OBMV is an updatable view.

Corollary 1- A well-defined OBMV is an OBMV such that its F_c contains two concept-based relations $R_i \ R_j \wedge C_i, C_j \in \Sigma \mid \tau \leq C_i, C_j \vee \tau \leq C_j, C_i$.

Yet, the semantic model M of an ontology contains at least one taxonomic relation.

Corollary 2 – The semantic external model MOV of an OBMV can be rebuilt using reverse engineering based on properties of the subsumption relation $\tau \leq$.

Corollary 3 – The ontology-based model MO of a given universe Σ can be reconstructed by reverse reengineering according to a unique well-defined OBMV and MOV.

Indeed, on the basis of $R_i (K_i, A_i, \dots, A_k)$ and $R_j (K_j, B_i, \dots, B_l, \#K_i)$ that represent C_i and C_j , with $C_i \leq C_j$ (or conversely), the underlying UML-based sub-model can be redesigned by using reverse engineering. Moreover, such inheritance deriving from the taxonomic relation $\tau \leq$

intrinsically implements the well-known functional dependencies (FD) in RDBs. Yet, any FD graph constitutes a theoretic access structure to the DB content, including the met-data of the RDBMS. Accordingly, the relational schema can be rebuilt.

4. RELATED WORK

As previously emphasized, only (Glimm, Kazakov, Liebig, Tran, and Vialard, 2014, pp.180-195; Volz, Staab & Motik, 2003, pp. 707-724; Volz, Staab & Motik, 2005, pp.1-34) tackled ontologies' materialization. However, materialized views were not used for reverse reengineering. In (Glimm, Kazakov, Liebig, Tran, and Vialard, 2014, pp.180-195), reasoning over "large ABoxes" by "abstraction refinement" was the focus. However, the content – state – of an ontology ABox, obviously, is larger than the number of axioms in its TBox, which stands as conceptual schema. Besides, the comparison between "the number of different concept names" and "the number of different individuals" is fuzzy.

Incremental maintenance of materialized ontologies was discussed in (Volz, Staab & Motik, 2003, pp. 707-724; Volz, Staab & Motik, 2005, pp.1-34). The work in (Volz, Staab & Motik, 2003, pp. 707-724) also focused on reasoning over ontologies through materialized views. However, the work is especially declined for the Web Semantic ontologies, namely RDF/RDFS and the well-known Ontology Web Language (OWL). In (Volz, Staab & Motik, 2005, pp.1-34), the authors are mainly interested on ontology-based materialization views, but the technical approach is based upon logic databases. Nevertheless, the ontology-based materialized views are not used or related to schema reverse engineering, which is the main goal of this paper. Accordingly, we propose a set of conceptual guidelines for OBMMV4RE with the aim to create the missing link between the use of materialized views, ontologies, and reverse engineering.

5. OBMMV4RE: TOWARD A SET OF CONCEPTUAL GUIDELINES

In the aim to develop an efficient methodology for reverse engineering using ontology-based materialized views, we propose a first set of useful conceptual guidelines as follows:

- Guideline 1: The DB catalog contains meta-data related to the ontology meta-model, which completes and enriches the Db schema.

Such ontology meta-model called KBSM (Knowledge-based Semantics Data Model) was proposed in (Sy, Duarte and Dal Bianco, 2018).The KBSM (See Figure 5) is a common meta-schema establishing a logical link between RDBs and ontologies.

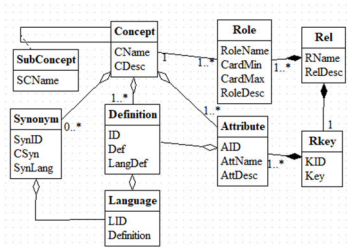


Figure 5. Knowledge-based Semantics Data Model (KBSM) of Domain ontologies

- Guideline 3: Based on Corollary 2, building of a semantic external model of M^{OV} , by reverse engineering.
- Guideline 4: According to Corollary 3, and based on the meta-data of the DB catalog enriched by the KBSM whose stored instances contain, re-build the global model M^O .

6. OBMMV4RE: A CASE STUDY EXAMPLE

An overview of our application domain (certified cereal seeds) is given at Figure 6. In this context, e.g., given a variety V_i of certified rice seed, one would like to know its breeder (maintainer) and the plot of land where it is growing or where it has been produced.

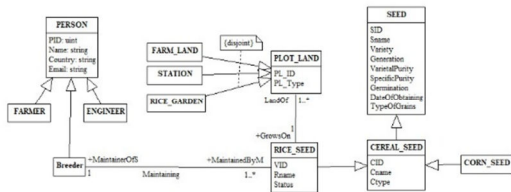


Figure 6. A case study: an overview

Accordingly, let mvBreeders be the materialized view that stores all farmers (name, country, e-mail) who are rice seed breeders (rice name). The SQL code is the following:

```
CREATE VIEW public."mvBreeders"  
WITH (check_option=local) AS  
SELECT "Name", "Country", "Email", "RName" FROM "Breeder"  
As "B"  
WHERE EXISTS (SELECT 1 FROM "Rice_Seed" AS "R"  
WHERE "R"."PID"="B"."PID" AND "B"."TYPE"='F');
```

As the relation Breeder in the FROM clause Fc of the mvBreeders view is concept-based and it is subsumed by the concept Person, mvBreeders is an OBMV. Thus, mvBreeders is updatable. Furthermore, based on the KBSM's DB (See Figure 5) and corollaries 2 and 3, the semantic model of Fig. 6 can be rebuilt.

CONCLUSION

In this paper, we tackled a tricky problem in information systems and software systems maintenance, namely data and meta-data reuse, especially schemas reverse engineering based on ontology-based materialized views.

Hence, we first have shown that most of the works, which have dealt with these topics, were inclusive, i.e. either ontologies or materialized views were left out when reverse engineering was on demand. Moreover, we clarified one again the use and the signification of the term 'reverse engineering' often confused with reengineering.

Finally, in our aim to improve the process of reverse engineering, we proposed a set of conceptual guidelines as useful methodology for overcoming the existing drawbacks.

However, actually these guidelines are not implemented. Accordingly, the next step in our future work is the experimentation of their theoretical foundations based on CASE tools that allow reverse engineering of advanced conceptual schemas as ontology-based ones.

REFERENCES

1. Alberts, R., & Franconi, E. (2012). "An Integrated Method using Conceptual Modelling to generate an Ontology-based Querying Mechanism". In OWL: Experiences and directions workshop.
2. Anas Shatnawi, Abdelhak-Djamel Seriai, Houari Sahraoui (2016). "Recovering software product line architecture of a family of object-oriented product variants", In The Journal of Systems and Software (2016), <http://dx.doi.org/10.1016/j.jss.2016.07.039>.
3. Anas Shatnawi, Abdelhak-Djamel Seriai, Houari Sahraoui, Zakarea Alshara (2016). "Reverse engineering reusable software components from object-oriented APIs", In The Journal of Systems and Software (2016), <http://dx.doi.org/10.1016/j.jss.2016.06.101>.
4. Asunción Gómez-Pérez, Ma Dolores Rojas-Amaya (1999). Ontological Reengineering for Reuse, D. Fensel and R. Studer (Eds.): EKAW'99, LNAI 1621, pp. 139-156, 1999, ©Springer-Verlag Berlin Heidelberg 1999.
5. Christoph Koch, Daniel Lupei, Val Tannen (2016). "Incremental View Maintenance For Collection Programming", PODS'16, June 26-July 01, 2016, San Francisco, CA, USA 2016 ACM. ISBN 978-1-4503-4191-2/16/06, DOI: <http://dx.doi.org/10.1145/2902251.2902286>.
6. Codd, E. F. (1970). A relational model of data for large shared data banks. Communications of the ACM, 13(6), 377-387.
7. Codd, E.F. (1985). "Is Your DBMS Really Relational?". Computerworld, October 14, 1985.
8. E. Chikofsky and J. Cross (1990). "Reverse engineering and design recovery: A taxonomy.", IEEE Software, 7(1):13-17, January 1990.
9. E. Reynares, M.L. Caliusco, M.R. Galli (2015). "A set of ontology design patterns for reengineering SBVR statements into OWL/SWRL ontologies", Expert Systems with Applications 42 (2015) 2680-2690.
10. Edgar Frank Codd, "Does Your DBMS Run By the Rules", ComputerWorld, 21 octobre 1985.
11. Glimm B., Kazakov Y., Liebig T., Tran TK., Vialard V. (2014). Abstraction Refinement for Ontology Materialization. In: Mika P. et

- al. (eds) The Semantic Web – ISWC 2014. ISWC 2014. Lecture Notes in Computer Science, vol 8797. Springer, Cham.
12. Gómez-Pérez A (1999). Evaluation of Taxonomic Knowledge in Ontologies and Knowledge Bases, Banff Knowledge Acquisition for Knowledge-Based Systems (KAW'99), Proceedings of the Banff Knowledge Acquisition for Knowledge-Based Systems, Workshop. KAW'99 Volumen: II, Editor: Gaines, B.R.; Musen, M., Editorial: University of Calgary, Alberta, Canadá, 16-21 de octubre de 1999., Banff, Alberta, Canadá., Pages: 6.1.1 - 6.1.18.
 13. Jesper Zedlitz and Norbert Luttenberger (2014). "Conceptual Modelling in UML and OWL-2", International Journal on Advances in Software, vol 7 no 1 & 2, year 2014, <http://www.iariajournals.org/software/>.
 14. Markus Brenner and Birte Glimm (2017). "Incremental Materialization Update via Abstraction Renement", In the Proceedings of the 30th International Workshop on Description Logics, Montpellier, France, July 18-21, 2017.
 15. Müller, H. A., Jahnke, J. H., Smith, D. B., Storey, M. A., Tilley, S. R., & Wong, K. (2000, May). "Reverse engineering: A roadmap.", In Proceedings of the Conference on the Future of Software Engineering (pp. 47-60). ACM.
 16. Oumar Sy, Denio Duarte, Guilherme Dal Bianco (2018). "Ontologies and Relational Databases Meta-Schema Design: A Three-Level Unified Lifecycle", 5th International Conference on Control, Decision and Information Technologies, CoDIT'2018 10-13 April, Thessaloniki - Greece.
 17. Oumar Sy, Moussa Lo, Denio Duarte (2010). "Integrating Ontologies in Database Scheme: Ontology-Based Views Conceptual Modeling", In the Proceedings of the 6th International Conference on Signal-Image Technology and Internet-Based Systems, SITIS, Kuala Lumpur, Malaysia, December 15-18, 2010, pp. 269-276.
 18. Prabhot Kaur Chahal, Amritpal Singh (2014). "Software Reuse and Reengineering: With A Case Study", In International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-3, Issue-6, January 2014.

19. Qi Zhang, Andreas Karcher (2012). "System Reverse Engineering to Requirements and Tests", The Seventh International Conference on Systems ICONS 2012, Copyright (c) IARIA, 2012. ISBN: 978-1-61208-184-7 35.
20. Rick F. van der Lans (2014). "Is Actian PSQl a Relational Database Server?", A Technical Whitepaper, March 2014, Copyright © 2014 R20/Consultancy, all rights reserved, http://www.pervasive.com/Portals/55/documents/WP_IsPSQLaRelationalDatabaseServer_R20_March2014.pdf
21. Rudi Studer, V. Richard Benjamins, Dieter Fensel (1998). Knowledge Engineering: Principles and methods, Data & Knowledge Engineering, 25 (1998) 161-197, © 1998 Elsevier.
22. Sonja Risti, Slavica Aleksi, Milan Celikovi, Vladimir Dimitrieski, Ivan Lukovi (2014). "Database reverse engineering based on meta-models", Cent. Eur. J. Comp. Sci. • 4(3) • 2014 • 150-159, DOI: 10.2478/s13537-014-0218-1.
23. Volz, R., Staab, S., & Motik, B. (2003, November). Incremental maintenance of materialized ontologies. In OTM Confederated International Conferences" On the Move to Meaningful Internet Systems" (pp. 707-724). Springer, Berlin, Heidelberg.
24. Volz, R., Staab, S., & Motik, B. (2005). "Incrementally maintaining materializations of ontologies stored in logic databases". In Journal on Data Semantics II (pp. 1-34). Springer, Berlin, Heidelberg.