An Innovative Approach to Manage Heterogeneous Information Using Relational Database Systems

Cosmin Sabo[1(✉)], Petrică C. Pop[1], Honoriu Vălean[1], and Daniela Dănciulescu[2]

[1] Technical University of Cluj-Napoca, Cluj-Napoca, România
cosmin_sabo@cunbm.utcluj.ro
[2] University of Craiova, Craiova, România

Abstract. In this paper, we propose a novel database design structure that can deal with all the aspects of the complexity of data that has to be managed, using the concepts of defining objects in object oriented programming (OOP). As well, we create a set of procedures in database system that allows us to manage all type of data, without knowing the structure of the database. The creation of the database structure, also the mechanism of inserting and retrieval the information is made by using a metadata set of information.

The major benefit of the proposed approach is that we can use a relational database management system (RDBMS), that can assure ACID (atomicity, consistency, isolation and durability) principles, low cost management and quick development based on metadata structure. The main advantages of our approach comparing with NoSQL database system is that we preserve ACID properties of the information and comparing with NewSQL is that the cost of the projection of database structure and management of the system is much lower.

Our proposed system is functional and can manage very large amount of data from heterogenic sources that can be managed by companies without a lot of know-how.

Keywords: Databases · Data logical models · Relational models · Object-oriented models

1 Introduction

How to manage big quantities of dataset with a high rate of changes from heterogeneous sources and preserve in the same time ACID (atomicity, consistency, isolation and durability) principles of database systems is a question that each of us has to deal with.

Most of the applications used by companies to manage their data use relational database management systems (RDMSs) and this fact will remain for the next years, even though database management systems (DMSs) are evolving very fast. On the other hand, data structure, are changing often and the number of data sources needed to be used by companies to remain competitive are rising every day. This is the reason why we create this new approach in order to retrieve and manage information datasets.

A.M. Madureira et al. (eds.), *Intelligent Systems Design and Applications*,
Advances in Intelligent Systems and Computing 557,
DOI 10.1007/978-3-319-53480-0_1

The evolution of database management system capabilities is a clear indicator of the existence of necessity of finding mechanisms for structuring information. Databases are defined by Frawley [7] as collections of data integrated into one or more files, organized to facilitate the storage, change, query and retrieval of information relevant to users needs. Frawley estimated that global information generated will double every 20 months and the size and number of databases will grow even faster, see for more information [7]. Coltri believes that the database is one of the areas with the most important developments in software engineering [19].

In the last period, librarianship has developed and implemented a suite of standards for information management, and for ensuring the sustainability of stored information. We will mention few of these standards used for resource description: BIBFRAME - Bibliographic Framework Initiative [8], EAD - Encoded Archival Description [9] EDTF - Extended Date/Time Format, MARCXML - Machine-Readable Cataloging XML [10], MODS - metadata Object Description Standard, but there are standards for digital resources, such as PREMIS - Preservation metadata or METS - metadata Encoding and Transmission Standard [11]. These standards can be extended in order to be used in other areas.

As is shown by Rokitskii [5], object oriented models are data logical in their essence but, at the same time, since object oriented models are formal, they allow one to specify formal constructions of data, formal relations between them, and formal operations on them. Object-oriented modeling has been already used at the interface level of three-level architectures of database management systems (DBMSs) and also at the conceptual level of design. In contrast to object-oriented tools occupying a rather large segment of the market of creation of application programs at the present time, the market of object-oriented database management systems has a low acceptance in small and medium business software. Based on this premises, Rokitskii [5] used relational database management systems (DBMSs) in order to create a data structure model based on object-oriented principals. For more information, see [5].

Vysniauskas and Nemuraite in [6] have presented a solution for a problem that has arisen from practical needs: namely, possibilities for storing ontological information and processing this information by user applications. For this purpose, they used relational database (RDB) considering that it is a good candidate that has proven capabilities to cope with large amounts of data. Methodologies for transforming entity relationship and object-oriented conceptual models to relational database structures are well-established and implemented in their tools.

Ontology Definition Metamodel, initiated by OMG, is seeking to define transformations between OWL, UML, ER and other modelling languages, where Simple Common Logic is chosen for definition of constraints. On the base of existing methodologies, there are some possible ways to relate ontological information described by ontological language, with relational schemas. For more information, see [4].

In this paper we present a novel solution of structuring the information in a relational database system. Our approach is using concepts from library standards and object oriented programming, and is totally different from previous studies.

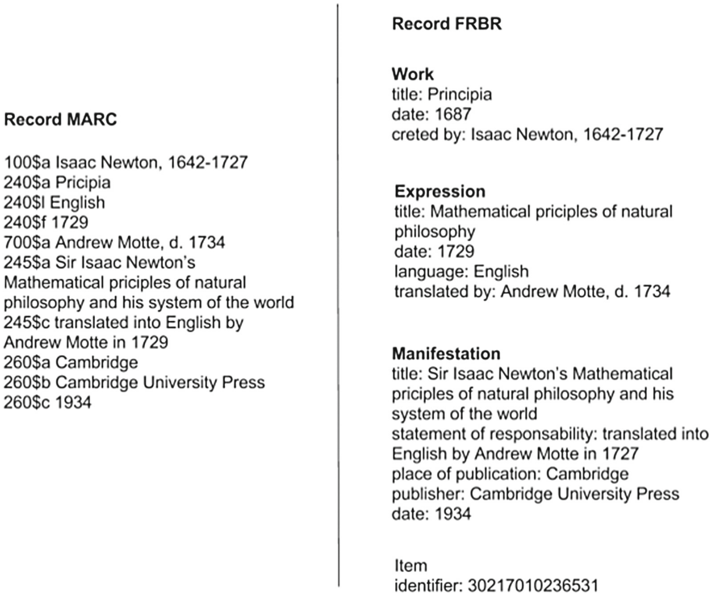
The aim of this paper is to describe an innovative approach that combines informational standards from librarianship field with object oriented programming techniques in order to manage financial and contact information of possible partners or clients.

2 Defining the Concepts Used for Solving this Problem

2.1 Library Standards for Defining Information

MARC (Machine Readable Cataloging) is currently the most widely used standard for storing and exchanging bibliographic records. Evolution of MARC standard has a history of more than forty years. MARC, is basically a concept for structuring and interchanging of information which evolved separately from concepts of management systems databases. Even if it is a standard with a high degree of use in libraries, it does not solve all problems arising from the rapid evolution of data structures and quantity of existing information.

The structure of the information in MARC format is a linear one, each type of information stored is defined by a set of metadata composed from a set of three numerical characters that defined field, an element represented by a character alphanumeric that represents subfield and field label to define the meaning of this field subfield tuple [16]. The information can be stored in field or in a subfield. Basically, if you want to specify the author of a work, the information that will identify metadata is filed 100, and the corresponding value will be written to the right of this metadata, in subfield a, as you can see in Fig. 1.



Record MARC

100$a Isaac Newton, 1642-1727
240$a Pricipia
240$l English
240$f 1729
700$a Andrew Motte, d. 1734
245$a Sir Isaac Newton's
Mathematical priciples of natural
philosophy and his system of the world
245$c translated into English by
Andrew Motte in 1729
260$a Cambridge
260$b Cambridge University Press
260$c 1934

Record FRBR

Work
title: Principia
date: 1687
creted by: Isaac Newton, 1642-1727

Expression
title: Mathematical priciples of natural
philosophy
date: 1729
language: English
translated by: Andrew Motte, d. 1734

Manifestation
title: Sir Isaac Newton's Mathematical
priciples of natural philosophy and his
system of the world
statement of responsability: translated into
English by Andrew Motte in 1727
place of publication: Cambridge
publisher: Cambridge University Press
date: 1934

Item
identifier: 30217010236531

Fig. 1. A bibliographic record represented in MARC and FRBR model

If there is repetitive information for a given metadata field, we will add a new metadata representing information and the corresponding value. If a subfield is repetitive, $ separator will be used followed by subfield name to write repetitive set of values. The maximum number of characters that can be stored in a field or subfield, of MARC standard, is 999 characters.

MARC standard has proved that are some inconvenient in information management:

- The number of fields and subfields are insufficient to encapsulate all type of information needed to be represented;
- Fields and subfields containing a large amount of information are not possible to be managed;
- Special characters can generate errors in automated process.

Based on this fact regarding MARC standards, MARCXML standard was developed. This standard representation of information eliminates previously existing limitations in storing large amounts of information in a single subfield or storing special chars.

This linear way of defining information within a bibliographic record cannot solve all situations generated by evolution of information needed to be stored, ensuring sustainability of the concerned information with minimal redundancy, but allows a semantic presentation of the information stored on this structure [3]. In this sense, to solve the aforementioned issues, IFLA - International Federation of Library Associations and other institutions have proposed and developed FRBR - Functional Requirements for Bibliographic Records, which adopts a description of hierarchical information in order to increase the level of granularity and to allow better information reuse.

2.2 Representing Information Structures Using FRBR Model

FRBR [2] reduces the number of descriptive elements at record level, but significantly increases the number of records that can be described in separate facilities, as is shown in Fig. 2. The linear model description of records in MARC standard is converted into a hierarchical model in FRBR model, and this model can be mapped to object oriented information, but this aspect will be treated in the next chapter.

Clearly, in FRBR representation there are few shortcomings [12], because the aim of defining FRBR was to reduce the cost of classical description of bibliographic resources, but has left aside other aspects such as:

- Prioritizing information at field level;
- Repetitive order of subfields repetitive not standardized;
- Another important aspect is related to the description of records or parts of records in a different language.

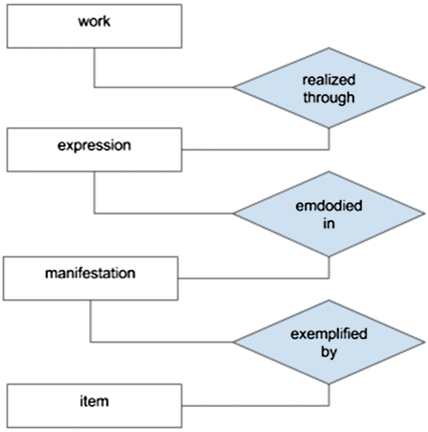


Fig. 2. Representing relations in a record FRBR

3 Problem Solution Using Metadata Definitions

Next we will define a complex informational structure using relational database management systems [18] that is easy to be managed and we will show how to combine the expertise gained by the team members in librarian bibliographic standards [1], relational database management systems and object-oriented programming [17].

Object-oriented programming (OOP) refers to a type of computer programming (software design) in which programmers define not only the data type of a data structure, but also the types of operations (functions) that can be applied to the data structure.

In this way, the data structure becomes an object that includes both data and functions. In addition, programmers can create relationships between one object and another. For example, objects can inherit characteristics from other objects [15].

OOP concepts allow the development of simple and effective solutions to many problems, enabling software programming complexity decreased, but improving the quality of the solutions obtained [14].

MARC standard with the list of fields and subfields can be associated with linear programming where we can manage, perhaps hundreds of variables and procedures, but increasing their number can generate unmanageable situations. Obviously, the continued growth of the types of records required to be described will generate situations hard to be manage in a linear structure.

Managing complex informational structures without the risk of losing information and in the same time providing tools for easy information retrieval, it is not an easy task, and this is the reason why we define a normalized information structure, using MARC definitions and concept of structuring objects specific to object oriented programming, that allow us to have an easy information management.

The MARC structure, which according to Fig. 1, is defined as linear way of defining information was mapped into a structure on three levels as we will describe in what it follows.

The first level defines the uniqueness of registration, and a set of adjacent properties. This level is equivalent in object-oriented programming with class variables defined at class level, each record in the database consist in an instance of this class.

The second level consists in the definitions MARC fields, their names having a fixed length of three characters and values between 001 and 999. Each field has clear defined meaning, and the meaning of the fields is grouped for easy identification. This level is equivalent with methods from object oriented programming.

The third level is defined by subfields, that have a fixed length that consists in one character that has values between a–z and 0–9. Most of the information is stored at subfield level. There are only a few fields that store information, the remaining fields representing the relationship between subfield and record level. This concept allows an easy way to define 1:n and n:m relations. These subfields represent in object oriented programming the variables defined inside methods.

The principle of defining the database structure for the example above is shown in Fig. 3. Since the number of tables and relations between tables is too big to be exhaustive represented here, we represent those elements that are representative for the principles shown before.

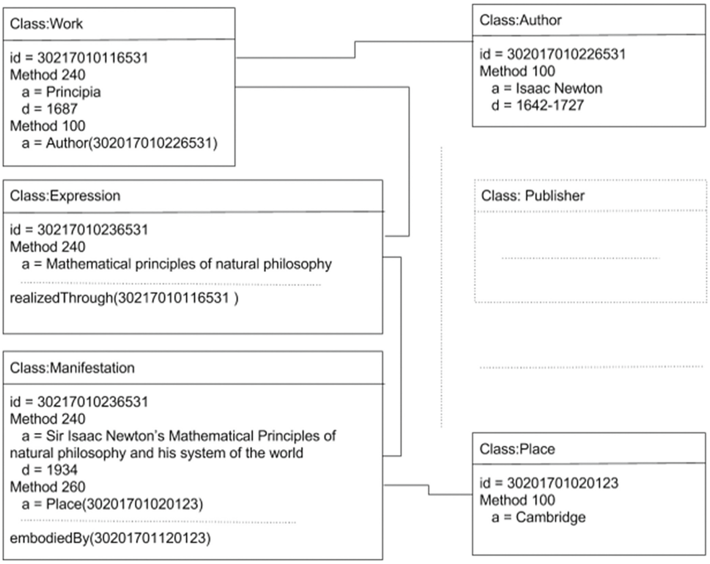


Fig. 3. OO information mapping using MARC fields

This concept has as starting point librarianship and information science, and concepts specific to object-oriented programming, which were used to define a metadata structure, which automatically generate data structures necessary through stored procedures. The result is that any user can manage all this data structure and information stored in without knowledge of database management or knowledge about standard query language, the entire process of defining and extending the database structure is based on the metadata definition.

Basically, we defined a set of tables in the database that store all metadata we need to generate: structures of tables, fields, data types, restrictions and relations between them. Based on triggers, when changes are made on these tables the appropriate operation for adding, modifying or deleting tables and relations between them are generated.

Similarly, when we interrogate the database we shall use a single stored procedure with the list of required parameters to be entered, such as the list of fields needed, the list of conditions to be fulfilled and other optional parameters.

It does not matter if you want to take information about a person, a task list, employment or any other information, you can always call the same procedure, without needing to know the database structure.

This model is implemented using MySQL database system, a system that allows storing sufficient quantities of information at the level of TB or PB, with the possibility of clustering. The concepts defined in this material may be used for other database systems. The reason for choosing this database system is the ability to achieve rapid transition from SQL to NoSQL [13] using schemas. For further studies we intend to compare our results with other NoSQL database systems and also, to map our structure to some web ontology language (OWL) [6].

4 Preliminary Results

We took in consideration datasets provided by http://data.gov.ro, a governmental website, that offers information under licensed distribution OGL-ROU-1.0, which allows the collection, processing and distribution of this information without any warranty. Now the website contains 669 sets of structured data in 16.770 files structured in various formats and informational structure.

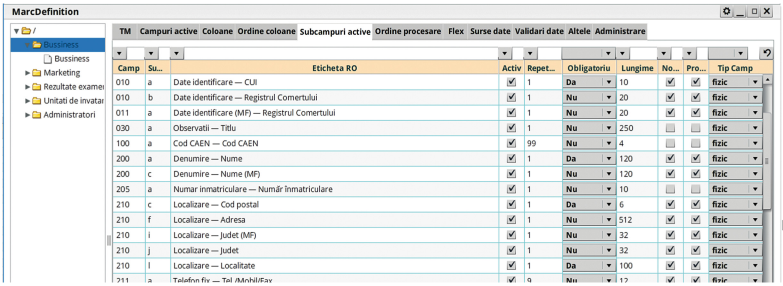
In order to highlight the innovative concept presented in this article, we took information about Romanian companies in a set of years (2016, 2015, 2014 and 2013). Each set of data has source files consist of six separate documents structured as CSV (comma separated values) format, the content of this files is different.

Also, in order to check the entire functionality of the mentioned concepts we have taken information about companies in Romania, from other public governmental sites, to ensure the heterogeneity of information. Data source in this case was in html format, retrieving information was performed using XPath.

The implementation process has required the following steps:

- Defining the structure based on information provided by the documents published on the website data.gov.ro and information retrieved from other websites;
- Identifying other sites that provide information about public companies;
- Extension metadata structure to embed this new information;
- Taking information from files provided by data.gov.ro;
- Generate the automation to extract relevant data from HTML (Hyper Text Markup Language) files;
- Insert datasets into database structure using stored procedures specialized for this operation.

The metadata information defines 41 fields and a total of 89 subfields. The next figure shows on the left side the list of fields defined in this metadata structure and the type of information that can be stored at field level, with length limitation if it is needed and on the right side, the list of subfields defined for the filed selected (Fig. 4).



Camp	Su...	Eticheta RO	Activ	Repet...	Obligatoriu	Lungime	No...	Pro...	Tip Camp
010	a	Date identificare — CUI	☑	1	Da	10	☑	☑	fizic
010	b	Date identificare — Registrul Comertului	☑	1	Nu	20	☑	☑	fizic
011	a	Date identificare (MF) — Registrul Comertului	☑	1	Nu	20	☑	☑	fizic
030	a	Observatii — Titlu	☑	1	Nu	250	□	□	fizic
100	a	Cod CAEN — Cod CAEN	☑	99	Nu	4	□	□	fizic
200	a	Denumire — Nume	☑	1	Da	120	☑	☑	fizic
200	c	Denumire — Nume (MF)	☑	1	Nu	120	☑	☑	fizic
205	a	Numar inmatriculare — Număr înmatriculare	☑	1	Nu	10	□	□	fizic
210	c	Localizare — Cod postal	☑	1	Da	6	☑	☑	fizic
210	f	Localizare — Adresa	☑	1	Nu	512	☑	☑	fizic
210	i	Localizare — Judet (MF)	☑	1	Nu	32	☑	☑	fizic
210	j	Localizare — Judet	☑	1	Nu	32	☑	☑	fizic
210	l	Localizare — Localitate	☑	1	Da	100	☑	☑	fizic
211	a	Telefon fix — Tel./Mobil/Fax	☑	9	Nu	12	☑	☑	fizic

Fig. 4. Field and subfield metadata definition

Metadata set allow to define if a field or a subfield is active or not, based on this information, stored procedures will allow or not to insert information in field or subfield. We also define if a subfield is repetitive or not, how many times it can be repeated, maximum length of information stored in it, if the value stored in a field subfield tuple is entered by user, or is generated based on some automatic procedures specific to each data structure.

This metadata structure defines if a field or subfield value should be validated by a regular expression and if this validation does not pass what will be the results returned to user. Validation rule can send to user an info note that this information does not respect the rules defined, a warning, or can generate an error message and reject the insert or update operation.

Each field or subfield can have different data source information. For example, subfield CAEN code (activities classification) has as source a dictionary of values, based on fact that CAEN code is a predefined list. The owner of the company represents an authority information, which means that behind that name is a complex structure that respects the same principals. The owner company name is an authority structure because we can add another information about the owner, like contact information. The complexity of this information can be extended based on new data source information that we find. Link data source represents a connection to another data set from the same structure. For example, if a company is connected to another, we can represent this by using the link data source.

After the insertion process in database based on metadata that define the entire structure the result is 1.738.053 companies from Romania with contact information and main financial indicators from the last eight years.

To obtain the list of companies that exists in database we use the next stored procedure:

```
marc_select_advanced_fast('b2b', 'rou', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '',
'', '', '', '', '', '', '', '', '', '', '', '200a,010a*,210j,210l,220b,220a,212a*,217a*', '', '27',
'539973', '', '', '');
```

The parameters set in this example show object type referred, in this case ‘b2b’; also if a record it is stored in more languages it allows us to select the preferred language, values from list of field subfield tuples that will be returned, data offset is 539.973 and the number of results is 27. Parameter ‘*’ after field subfield tuple means it will return all values if a field subfield tuple is repetitive.

If we want to find all the companies from Bucharest that have a mobile phone and email address in database, we will use:

```
    CALL marc_select_advanced_fast_count('b2b', 'rou', ", ", ", ", ", ", ", ", ", ", ", ", ", ",
", '210j', 'b', 'București', ", '212a', 'b', '07', ", '217a', 'c', '@', ", ", ", ", ", ", ", ", ", ", ", ", ",
", ");
```

5 Conclusions

In this paper, we consider the problem of defining a complex database structure, based on premises that the input data are provided by heterogeneous sources and sources are changing the information provided often.

We defined a set of metadata that generates entire data structure using relational database management system and the set of stored procedures that allow us to manage the information stored in this data structure without needing to know the internal structure.

Some important features of this concept:

- Capability of generating complex data structures only using a metadata definition
- Data management using only one stored procedure to insert any type of data
- Data search and retrieve using one stored procedures
- Structure management based on metadata definition

Implementation of this concept was made using a large amount of data from heterogeneous sources.

References

1. Francu, V., Sabo, C.: Implementation of a UDC-based multilingual thesaurus in a library catalogue: the case of bibliophil. Knowl. Organ. **37**(3), 209–215 (2010)
2. Carlyle, A.: Understanding FRBR as a conceptual model: FRBR and the bibliographic universe. Libr. Resour. Tech. Serv. **50**(4), 264–273 (2006)
3. Giannopoulou, E., Mitrou, N., Chimos, K., Karvounidis, T., Douligeris, C.: A semantic web approach in the implementation of a linked data portal using a CMS. In: Proceedings of 10th International Conference on Signal-Image Technology and Internet-Based Systems, SITIS, p. 164 (2014)
4. Agosti, M., Crivellari, F., Di Nunzio, G.M., Gabrielli, S.: Understanding user requirements and preferences for a digital library web portal. Intl. J. Digital Libr. **11**(4), 225–238 (2010)
5. Rokitskii, R.B.: Object-oriented databases with relational DBMSs. Cybern. Syst. Anal. **36**(6), 813–822 (2000)
6. Vysniauskas, E., Nemuraite, L.: Transforming ontology representation from OWL to relational database. Inf. Technol. Control **35**(3A), 333–343 (2006)

7. Frawley, W.J., Piatetsky-Shapiro, G., Matheus, C.J.: Knowledge discovery in databases: an overview. AI Mag. **13**(3), 57–70 (1992)
8. http://www.loc.gov/bibframe/pdf/marcld-report-11-21-2012.pdf
9. Gartner, R.: An XML schema for enhancing the semantic interoperability of archival description. Arch. Sci. **15**(3), 295–313 (2015)
10. Gardner, J.R.: Information architecture planning with XML. Libr. Hi Tech **19**(3), 231–241 (2001)
11. Cheslow, S.: METS for the cultural heritage community: a literature review. Libr. Philos. Pract. **2014**(1) (2014)
12. Pacheco, K.L., Ortega, C.D.: Model FRBR in origin. Biblios **60**, 63–75 (2015)
13. Lee, C., Zheng, Y.: Automatic SQL-to-NoSQL schema transformation over the MySQL and HBase databases. In: IEEE International Conference on Consumer Electronics - Taiwan, ICCE-TW, pp. 426–427 (2015)
14. Cavaiani, T.P.: Object-oriented programming principles and the Java class library. J. Inf. Syst. Educ. **17**(4), 365 (2006)
15. SAP AG: Patent issued for systems and methods for generating a common data model for relational and object oriented databases. J. Eng. 3005 (2013)
16. Holler, J., Tsiatsis, V., Mulligan, C., Avesand, S., Karnouskos, S., Boyle, D.: From Machine-To-Machine to the Internet of Things, pp. 1–331. Academic Press, Cambridge (2014)
17. Pop, P.C., Matei, O., Sabo, C.: A new approach for solving the generalized traveling salesman problem. In: Blesa, María J., Blum, C., Raidl, G., Roli, A., Sampels, M. (eds.) HM 2010. LNCS, vol. 6373, pp. 62–72. Springer, Heidelberg (2010). doi:10.1007/978-3-642-16054-7_5
18. Cola, C., Valean, H.: Ambient recognition using BigData model and Arduino controller. In: Proceedings of 2016 IEEE International Conference on Automation, Quality and Testing, Robotics THETA 20th edition, 19–21 May, Cluj-Napoca, Romania (2016). ISBN: 978-1-4673-8691-13
19. Coltri, A.: Databases in health care. In: Lehman, H.P., Abbott, P.A., Roderer, N.K., et al. (eds.) Aspects of Electronic Health Record Systems, 2nd edn, pp. 225–251. Springer, New York (2006). (Chap. 11)