

Using ontology merging for the integration of information systems and the production capacity planning system

N Yarushkina¹, A Romanov¹, A Filippov¹, A Dolganovskaya¹ and M Grigorieva¹

¹Ulyanovsk State Technical University, Severny Venets street, 32, Ulyanovsk, Russia, 432027

e-mail: jng@ulstu.ru, romanov73@gmail.com, al.filippov@ulstu.ru, gms4295@mail.ru

Abstract. This article describes the method of integrating information systems of an aircraft factory with the production capacity planning system based on the ontology merging. The ontological representation is formed for each relational database (RDB) of integrated information systems. The ontological representation is formed in the process of analyzing the structure of the relational database of the information system (IS). Based on the ontological representations merging the integrating data model is formed. The integrating data model is a mechanism for semantic integration of data sources.

1. Introduction

As part of the work on automating the process of production capacity of the aircraft factory, it is necessary to take into account the presence of heterogeneous information systems in the aircraft factory that automates various business processes [1]. Data consistency can be realized by integrating the production capacity planning system with existing information systems of the aircraft factory. Data integration means the integration of data from different sources and the providing of data to users in a unified way. The main difficulties of data integration are:

- (i) Data models heterogeneity.
- (ii) Independence of information systems of the aircraft factory from each other.
- (iii) Data can be located in different segments of the local network of the aircraft factory and (or) on the Internet.
- (iv) Different data formats.
- (v) Different value representations.
- (vi) Loss of data relevance by one of the data sources.

Thus, the organization of the information interaction between the production capacity planning system and the existing information systems of the aircraft factory raises the need to solve the following methodological problems [2, 3, 4, 5, 6, 7, 8, 9]:

- (i) Creating an integrating data model. Integrating data model is the basis of a single user interface in the integration system.
- (ii) Development of methods for building ontological representations for specific models of various data sources.
- (iii) Development of methods for building integrating data model for specific models of various data sources.
- (iv) Solving the problem of data sources heterogeneity.
- (v) Development of mechanisms for semantic integration of data sources.

2. Ontological representation of data source

The proposed information interaction algorithm consists of the following steps:

- (i) Extracting metadata from the RDB schema for automatic generation of ontologies for the source and target RDBs.
- (ii) Ontology merging to configure correspondence between objects, attributes, and relationships of integrated ISs. Creation of metaontology.
- (iii) Using the metaontology to perform the interaction procedure on a schedule or event.

The metaontology is the settings contains correspondences between data models (tables and columns) of integrated ISs.

Ontology is a model knowledge representation of a specific problem area [10]. An ontology contains a set of classes, individuals, properties, and relations between them. An ontology is based on the dictionary of terms which reflecting the concepts of a problem area. Also, the dictionary contains a set of rules (axioms). Terms can be combined to construct a set of statements about the state of the problem area based on a set of axioms.

At the moment, a lot of researchers use the ontological approach for extracting metadata from the RDB schema:

- (i) The Relational.OWL [11] currently supporting only MySQL and DB2 database management systems (DBMS). The generated ontology contains classes: Database, Table, Column, and PrimaryKey, and properties: has, hasTable, hasColumn, isIdentifiedBy, references, scale, length. The main disadvantage of ontology generated by Relational.OWL is the presence of limited coverage of the domain, not considering, for instance, data type, foreign keys, and constraints.
- (ii) The OWL-RDBO [12, 13] currently supporting only MySQL, PostgreSQL and DB2 DBMSs. The generated ontology contains classes: DatabaseName, RelationList, Relation, AttributeList, Attribute, and properties: hasRelations, hasType, referenceAttribute, referenceRelation. The main disadvantage of ontology generated by OWL-RDBO is the presence of concepts external to the domain, such as RelationList to group a set of Relation, and AttributeList to group a set of attributes.
- (iii) Other approaches, such as [14, 15] extract the real world relations from the RDB structure, and unable to reconstruct the original schema of the RDB.

The relational data model can be represented as the following expression:

$$RDM = \langle E, H, R \rangle, \quad (1)$$

where $E = \{E_1, E_2, \dots, E_p\}$ is a set of RDB entities (tables);

$E_i = (name, Row, Col)$ is the i -th RDB entity that contains the name, set of rows Row and columns Col ;

$Col_j = (name, type, constraints)$ is the j -th column of the i -th RDB entity that contains

properties: the name, the type and set of constraints;

$H = \{H_1, H_2, \dots, H_q\}$ is a hierarchy of RDB entities in the case of using the table inheritance function:

$$H_j = E_i D(x) E_k, \quad (2)$$

where E_i and E_k are RDB entities;

$D(x)$ is a 'parent-child' relation between E_i and E_k ;

$R = \{R_1, R_2, \dots, R_r\}$ is a set of RDB relations:

$$R_l = E_i \frac{F(x)}{G(x)} E_k, \quad (3)$$

where $F(x)$ is an RDB relation between E_i and E_k ;

$G(x)$ is an RDB relation between E_k and E_i .

Functions $F(x)$ and $G(x)$ can take values: U is a single relation and N is multiple relations.

The ontological representation of the RDB data model is:

$$O = \langle C, P, L, R \rangle, \quad (4)$$

where $C = \{C_1, C_2, \dots, C_n\}$ – is a set of data model ontology classes;

$P = \{P_1, P_2, \dots, P_m\}$ – is a set of properties of data model ontology classes;

$L = \{L_1, L_2, \dots, L_o\}$ – is a set of data model ontology constraints;

R is a set of data model ontology relations:

$$R = \{R_C, R_P, R_L\}, \quad (5)$$

where R_C is a set of relations defining the hierarchy of data model ontology classes;

R_P is a set of relations defining the 'class-property' data model ontology ties;

R_L is a set of relations defining the 'property-constraint' data model ontology ties.

The following function is used to map the RDB structure (ex. 1) to the ontological representation (ex. 4):

$$F(RDM, O) : \{E^{RDM}, H^{RDM}, R^{RDM}\} \rightarrow \{C^O, P^O, L^O, R^O\}, \quad (6)$$

where $\{E^{RDM}, H^{RDM}, R^{RDM}\}$ is a set of RDB entities and relations between them (eq. 1);

$\{C^O, P^O, L^O, R^O\}$ is a set of ontology entities (eq. 4).

The process of mapping the RDB structure into an ontological representation contains several steps:

(i) *Formation of ontological representation classes.*

A set of ontological representation classes C is formed based on the set of RDB entities $C E_i \rightarrow C_i$. The number of classes of the ontological representation must be equal to the number of RDB entities.

(ii) *Formation of properties of ontological representation classes.*

A set of properties P of the i -th ontological representation class C_i is formed based on the set of columns Col of the i -th RDB entity $E_i Col_j \rightarrow P_j$. The number of properties of the i -th ontological representation class C_i must be equal to the number of columns of the i -th RDB entity E_i . The name of the j -th property P_j is the name of the j -th column Col_j of the RDB entity.

(iii) *Formation of ontological representation constraints.*

A set of constraints L of the properties of the i -th ontological representation class C_i is formed based on the set of columns Col of the i -th RDB entity $E_i Col_k \rightarrow \hat{L}$. The number of constraints of the i -th ontological representation class C_i must be equal to the number of constraints of the i -th RDB entity E_i . However, there are limitations to this approach due to the difficulty of mapping constraints if their presents as triggers or stored procedures.

- (iv) *Forming hierarchy of ontological representation classes.*
 It is necessary to form a set of ontology relationships R_C between all the child and parent classes corresponding to the hierarchy of RDB entities if table inheritance uses in RDB $H \rightarrow R_C$. The domain of the j -th ontological representation relationship R_{C_j} is indicated by the reference to the parent class C_{parent} . The range of the j -th ontological representation relationship R_{C_j} is indicated by the reference to the child (or a set) class C_{child} .
- (v) *Formation of relations between classes and properties of classes of ontological representation.*
 A set of ontological representation relationships R_P is formed based on the set of columns Col of the i -th RDB entity E_i and the set of RDB relations R . Two types of relationships are formed for each j -th ontological representation property P_j :
- The relationship 'class-property'. The domain of the ontological representation relationship is indicated by the reference to the i -th class C_i to which the j -th property belongs, and the range to the j -th property reference P_j .
 - The relationship 'property-data type class'. The domain of the k -th ontological representation relationship is indicated by the reference to the j -th property P_j . The range is indicated by the reference to the l -th class C_l corresponding to the l -th RDB entity E_l , or the reference to the m -th ontology class C_m corresponding to the data type of the j -th RDB column Col_j .
- (vi) *Formation of relations between properties of classes and constraints of properties of classes of ontological representation.*
 A set of relations R_L of ontological representation is formed based on the set of columns Col of the i -th RDB entity. The domain of the j -th ontological representation relationship R_{L_j} is indicated by the reference to the k -th property P_k . The range of the j -th ontological representation relationship R_{L_j} is indicated by the reference to the k -th constraint $Col \rightarrow R_L$.

3. Integrating data model

It is necessary to form an integrating data model based on the ontological representations that obtained after mapping the RDB structure of each of the integrated information systems into the ontological representation. The definition of an ontological system is used as a formal representation of an integrating data model:

$$\sum^O = \langle O^{META}, O^{IS}, M \rangle, \quad (7)$$

where O^{META} is the integrating data model ontology (metaontology);
 $O^{IS} = \{O_1^{IS}, O_2^{IS}, \dots, O_g^{IS}\}$ is a set of ontological representations of information systems that must be integrated;
 M is a model of reasoner.

The following steps are necessary to form an integrating data model based on the set of ontological representations of the information systems that must be integrated:

- Formation of the universal concept dictionary for the current domain.*
 The process of forming an integrating data model O^{META} is based on the presence of common terminology. Ontological representations of all information systems that must be integrated O^{IS} should be built from a single concept dictionary. The concept dictionary is formed by the expert based on the analysis of the obtained ontological representations.
- Formation an integrating data model O^{META} .*
 At this step, the set of top-level classes C^{META} are added to the integrating data model O^{META} . The set of top-level classes C^{META} describes systems that must be integrated and is used as the basis for ontology merging.

(iii) *Formation of class hierarchy of integrating data model O^{META} .*

At this step, the integrating data model establishes a correspondence between the class hierarchies $C^{O^{IS}}$ of ontological representations O^{IS} of information systems that must be integrated.

(iv) *Formation of class properties of the integrating data model O^{META} .*

At this step, the integrating data model establishes a correspondence between the properties $P^{O^{IS}}$ of ontological representations O^{IS} of information systems that must be integrated. The expert decides which class properties of ontological representations O^{IS} should be included in the integrating data model O^{META} .

(v) *Formation of axioms of classes and properties, checking the integrating data model O^{META} for consistency.*

At this step, constraints $L^{O^{IS}}$ are applied to the properties $P^{O^{IS}}$ and classes $C^{O^{IS}}$ of the integrating data model O^{META} based on the constraints presents in the ontological representations O^{IS} . After that, the resulting integrating data model O^{META} should be checked for internal consistency using the reasoner M . However, the development of methods for checking the conditions of constraints is required, since the existing reasoners do not support working with such objects.

The proposed method is allowed to configure the correspondence between tables and fields of two RDBs. The main problem is the need for ontology merging. However, that problem can be solved due to the use of specialized tools to automate the ontology merging process. Also, specialized tools allow dividing the developer and domain expert roles. The main advantage of the proposed method is the ability to dynamically generate the necessary SQL queries for select and insert data from/to the RBD based on metaontology.

4. Example of creation the ontological representation of data source

Let see the following example of the ontological representation formation.

Table 1 shows the structure of the "Equipment and Tools" table of the aircraft factory IS.

Thus, the ontological representation of the "Equipment and Tools" entity (tab. 1) can be represented as:

$$\begin{aligned}
 O &= \langle \\
 C &= \{ \text{Equipment and Tools (E\&T), CHAR, NUMBER, BLOB, DATE} \}, \\
 P &= \{ t2_ob, t2_ng, t2_nn, t2_r1, t2_r2, t2_r3, t2_p1, t2_z1, t2_p2, t2_z2, \\
 &\quad t2_p3, t2_z3, t2_gm, t2_p3, t2_z3, t2_gm, up_dt, up_us, t2_dc, t2_vid, t2_doc, \\
 &\quad t2_prim, t2_yyyy \} \\
 L &= \{ \text{nullable, } \langle \text{length, 2} \rangle, \langle \text{length, 4} \rangle, \langle \text{length, 8} \rangle, \langle \text{length, 32} \rangle, \\
 &\quad \langle \text{length, 100} \rangle, \langle \text{length, 200} \rangle, \langle \text{length, 255} \rangle, \langle \text{precision, 5} \rangle, \\
 &\quad \langle \text{precision, 6} \rangle \} \\
 R_P &= \{ \langle \text{E\&T, t2_ob, CHAR} \rangle, \langle \text{E\&T, t2_ng, NUMBER} \rangle, \\
 &\quad \langle \text{E\&T, t2_nn, NUMBER} \rangle, \langle \text{E\&T, t2_r1, CHAR} \rangle, \\
 &\quad \langle \text{E\&T, t2_r2, CHAR} \rangle, \langle \text{E\&T, t2_r3, CHAR} \rangle, \\
 &\quad \langle \text{E\&T, t2_p1, CHAR} \rangle, \langle \text{E\&T, t2_z1, CHAR} \rangle, \\
 &\quad \langle \text{E\&T, t2_p2, CHAR} \rangle, \langle \text{E\&T, t2_z2, CHAR} \rangle, \\
 &\quad \langle \text{E\&T, t2_p3, CHAR} \rangle, \langle \text{E\&T, t2_z3, CHAR} \rangle, \\
 &\quad \langle \text{E\&T, t2_gm, CHAR} \rangle, \langle \text{E\&T, up_dt, DATE} \rangle, \\
 &\quad \langle \text{E\&T, up_us, CHAR} \rangle, \langle \text{E\&T, t2_dc, BLOB} \rangle, \\
 &\quad \langle \text{E\&T, t2_vid, CHAR} \rangle, \langle \text{E\&T, t2_doc, CHAR} \rangle, \\
 &\quad \langle \text{E\&T, t2_prim, CHAR} \rangle, \langle \text{E\&T, t2_yyyy, CHAR} \rangle \} \\
 R_L &= \{ \langle \text{E\&T, t2_ob, } \langle \text{length, 200} \rangle \rangle, \langle \text{E\&T, t2_ng, } \langle \text{precision, 5} \rangle \rangle, \\
 &\quad \langle \text{E\&T, t2_nn, } \langle \text{precision, 6} \rangle \rangle, \langle \text{E\&T, t2_p1, } \langle \text{length, 2} \rangle \rangle,
 \end{aligned}$$

```

    < E&T, t2_p1, nullable >, < E&T, t2_z1, < length, 8 > >,
    < E&T, t2_z1, nullable >, < E&T, t2_p2, < length, 2 > >,
    < E&T, t2_p2, nullable >, < E&T, t2_z2, < length, 8 > >,
    < E&T, t2_z2, nullable >, < E&T, t2_p3, < length, 2 > >,
    < E&T, t2_p3, nullable >, < E&T, t2_z3, < length, 8 > >,
    < E&T, t2_z3, nullable >, < E&T, up_us, < length, 32 > >,
    < E&T, t2_vid, < length, 4 > >, < E&T, t2_doc, < length, 100 > >,
    < E&T, t2_prim, < length, 100 > >, < E&T, t2_doc, nullable >,
    < E&T, t2_yyyy, < length, 4 > > }
    ).
    
```

Table 1. The "Equipment and Tools" table of the aircraft factory IS.

Column	Data type	Description
t2_ob	CHAR(200)	Name
t2_ng	NUMBER(5)	Group
t2_nn	NUMBER(6)	Position
t2_r1	CHAR	Type #1: 0 — equipment; 1 — tool; 2 — material; 6 — special tool.
t2_r2	CHAR	Type #2: 0 — standard; 1 — special.
t2_r3	CHAR	Type #3: 20 — no; 21 — design; 30 — model; 31 — design and model.
t2_p1	CHAR(2) nullable	Parameter #1
t2_z1	CHAR(8) nullable	Parameter #1 value
t2_p2	CHAR(2) nullable	Parameter #2
t2_z2	CHAR(8) nullable	Parameter #2 value
t2_p3	CHAR(2) nullable	Parameter #3
t2_z3	CHAR(8) nullable	Parameter #3 value
t2_gm	BLOB	Geometric model
up_dt	DATE	Date of last update
up_us	CHAR(32)	User
t2_dc	BLOB	Attachment
t2_vid	CHAR(4)	Tooling type
t2_doc	CHAR(100)	Document name
t2_prim	CHAR(100) nullable	Notes
t2_yyyy	CHAR(4)	Production date

As you can see from this example, the resulting ontology representation O has some sets of

objects:

- (i) A set of classes C contains the "Equipment and Tools" table and some data types: CHAR, NUMBER, BLOB, DATE. The OWL representation of ontology O uses Class signature to represent the table.
- (ii) A set of properties P contains all columns of the "Equipment and Tools" table. The OWL representation of ontology O uses built-in data types to represent RDB data types (*xsd:string*, *xsd:double*, *xsd:dateTime*, *xsd:base64Binary*), and Class signature to represent RDB relationships.
- (iii) A set of constraints L contains all variants of restrictions for columns of the "Equipment and Tools" table. This set is not translated to OWL representation directly.
- (iv) A set of relations between classes and properties R_P contains ties between table and columns that belong to this table. The OWL representation of ontology O uses ObjectProperties and DataProperties signatures to represent a set of relations R_P . ObjectProperties signatures are used to represent foreign keys. DataProperties signatures are used to represent columns that contain a value.
- (v) A set of relations between properties and constraints R_L contains a tie between column and constraints of this column. OWL datatype restrictions are used for constraints specification. For example:
DatatypeRestriction(xsd:integer xsd:minInclusive "5" ^ ^xsd:integer xsd:maxExclusive "10" ^ ^xsd:integer).

Thus, the ontological approach is commonly used to solve the methodological problem of building an integrating data model of information systems.

5. Conclusion

This article presents the implementation of the method of integrating the information systems of the aircraft factory with the production capacity planning system. The principles of ontological engineering allows mapping database structure of each information system that must be integrated into ontological representation. From the proposed methodology, an integrated data model is formed based on the obtained ontological representations for each information systems that must be integrated.

The proposed method allows organizing information interaction without the participation of developers in contrast to the traditional approach of consolidation, based on the method of direct data exchange. The only requirement of the proposed method is the presence of metaontology. The disadvantages of the proposed method implementation currently are:

- (i) The need for implementation of the data type casting algorithms in case of their mismatch for each DBMS.
- (ii) The need for adapting the proposed method implementation to the SQL dialect of DBMS involved in the exchange process. Random DBMS cannot be supported by this implementation.

6. References

- [1] Yarushkina N, Romanov A, Filippov A, Guskov G, Grigorieva M and Dolganovskaya A 2019 The building of the production capacity planning system for the aircraft factory *Research Papers Collection OpenSemantic Technologies for Intelligent Systems* 3 123-128
- [2] Clark T, Barn B S and Oussena S 2012 A method for enterprise architecture alignment *Practice-Driven Research on Enterprise Transformation* 48-76

- [3] Rouhani D B 2015 A systematic literature review on Enterprise Architecture Implementation Methodologies *Information and Software Technology* 1-20
- [4] Medini K and Bourey J P 2012 SCOR-based enterprise architecture methodology *Int. J. Comput. Integrat. Manuf.*
- [5] Poduval A 2011 *Do more with SOA Integration: Best of Packt*
- [6] Caselli V, Binildas C and Barai M 2008 *The Mantra of SOA. Service Oriented Architecture with Java* (Birmingham. UK)
- [7] Berna-Martinez V J, Zamora C, Ivette C, Perez M, Paz F, Paz L and Ramon C 2018 *Method for the Integration of Applications Based on Enterprise Service Bus Technologies*
- [8] Evsutin O O, Kokurina A S and Meshcheryakov R V 2019 A review of the methods of embedding information in digital objects for security in the Internet of things *Computer Optics* **43(1)** 137-154 DOI: 10.18287/2412-6179-2019-43-1-137-154
- [9] Rycarev I A, Kirsh D V and Kupriyanov A V 2018 Clustering of media content from social networks using bigdata technology *Computer Optics* **42(5)** 921-927 DOI: 10.18287/2412-6179-2018-42-5-921-927
- [10] Gruber T 2019 *Ontology* URL: <http://tomgruber.org/writing/ontology-in-encyclopedia-of-dbs.pdf>
- [11] de Laborda C P and Conrad S 2005 Relational. owl: a data and schema representation format based on owl *Proceedings of the 2nd Asia-Pacific conference on Conceptual modeling* 43 89-96
- [12] Trinh Q, Barker K and Alhadj R 2006 Rdb2ont: A tool for generating owl ontologies from relational database systems *Telecommunications International Conference on Internet and Web Applications and Services/Advanced* 170
- [13] Trinh Q, Barker K and Alhadj R 2007 Semantic interoperability between relational database systems *Database Engineering and Applications Symposium* 208-215
- [14] Barrett T, Jones D, Yuan J, Sawaya J, Uschold M, Adams T and Folger D 2002 Rdf representation of metadata for semantic integration of corporate information resources *International Workshop Real World and Semantic Web Applications*
- [15] Bizer C 2003 D2R MAP – A Database to RDF Mapping Language *Proc. of the 12th International World Wide Web Conference – Posters*

Acknowledgments

The study was supported by:

- the Ministry of Education and Science of the Russian Federation in the framework of the project No. 2.1182.2017/4.6. Development of methods and means for automation of production and technological preparation of aggregate-assembly aircraft production in the conditions of a multi-product production program;
- the Russian Foundation for Basic Research (Projects No. 18-47-732016, 18-47-730022, 17-07-00973, No. 18-47-730019).