An Approach to Dynamic Ontology Modification in Mediator Service-Oriented Information Systems[1]

Natalya Keberle, Vadim Ermolayev

Dept. of Mathematical Modeling and IT, Zaporozhye State Univ.,
Zhukovskogo 66, 69063, Zaporozhye, Ukraine
{kenga, eva}@zsu.zp.ua

Abstract: The paper proposes the approach to cope with the maintenance of dynamically changing resource ontologies of autonomously maintained, distributed, heterogeneous, wrapped information resources and their mappings to common mediator IS ontology. The approach intends to do the work in economical way reducing efforts to matching and aligning only modified ontology elements. Proposed is ontology model comprising both descriptive part and the set of modification primitives for each ontology structural element. The set of ontology modification invariants and the corresponding set of modification conflicts resolution rules are formulated for taxonomies. The way to provide IS services for ontology changes monitoring, matching and alignment is outlined.

1 Introduction

The problem of information (data and service) integration, sharing and reuse is important in a wide variety of application domains. Known are the deployed systems providing data retrieval from heterogeneous distributed information resources (IR) and querying them in a unified user-friendly way: e.g., OBSERVER [Sh00], InfoSleuth [Ba97]. However, the developers of these systems outline the static character of the used means for IR semantics representation. A great amount of work on modifying semantics: resource re-denotion, re-description, re-registration, ... is therefore done manually, consumes much time and labour force and unfortunately is the source of various mistakes and arising semantic conflicts.

Research on the possibilities providing for the management of dynamically changing IR semantics is thus very important primarily because of the existence of the following modification cases:

IR semantics may be extended to present an application domain more precisely

Conceptualisations of application domain itself are changed in time – good example is timely changes' maintenance in controlled medical terminologies (discussed in [Ol99])

[1] Research is run in frame of the Project financed by Ukrainian Ministry of Education and Science, Grant No. 0199Y1571

It may be thus expected that contemporary requirements to an intelligent mediator information system operating over a diversity of IR wrappers, actively providing distributed services, should contain:

Information System (IS) should be capable to operate with heterogeneous distributed IRs

IS should possess a kind of unifying semantic basis to enable the integration of the resources and services with different semantics

IS should be capable to cope with the autonomous character of IRs and services developed, maintained and managed independently, dynamically registering to the IS, leaving it and joining it back

One of the key problems for the designers of a mediator IS is the provision of the proper interoperability solutions. Interoperability problem is complex and may be viewed from various facets. An important aspect is the task of semantic interoperability, assuming that semantics changes over time.

Semantic interoperability problem is often decomposed into the following sub-problems:

First – resource semantics (re-)description, (re-)registration

The following kinds of ontologies are utilised in various solutions of the sub-problem: IR ontology, common IS ontology, domain ontology. *IR ontology* represents the partial reflection of the domain from the point of view of a certain wrapped IR and is created/updated within IR resource life cycle. IR ontologies may be extracted from existing resources [VMS94]. Otherwise, IR ontologies may be of use at the resource development stage [St97]. *Common IS ontology* provides unifying mediator specifications of domain conceptualizations obtained by mappings from IR ontologies (close to the global reference ontology, [Us00]). These mappings are provided in course of resource (re-) registration to IS. The creation of common IS ontology may also benefit from a third party domain ontology. *Domain ontology* reflects the semantics of an application domain.

Second – monitoring of resource semantics and corresponding domain semantics changes

The solutions of this subtask should provide the means for detecting the critical mass of the changes and for initiating the routines resulting from the solutions of subtasks 1 and 3.

Third – provision of the consistent modifications of resource semantics descriptions in response to the changes in the application domain view represented by the wrapped resource as well as the appropriate mapping of these modifications to the common ontology of mediator IS

In case resource structure descriptions is considered as a semantic issue the subtask will comprise the development of IS facilities enabling invocation of resource structure description mappings in response to the alert signals provided by the monitoring service (subtask 2).

The solutions of the presented subtasks may be considered as the mediator-wrapper services. An IS consumes the services from the active IR wrappers, provides its own utility services and provides some active service-initiating feedbacks to the resource wrapper side. Successful provision and execution of these services is tightly linked to the proper semantic interoperability solutions.

The paper proposes an approach to cope with the maintenance of dynamically changing resource ontologies and their mappings to a common mediator IS ontology. The ontologies are considered to be the core of resource semantics descriptions. The approach intends to do the work in economical way reducing efforts to matching and aligning only modified ontology elements.
The paper is organized as follows. Section 2 discusses related work in the field. Section 3 introduces ontology model comprising means for the description of the ontology elements as well as the set of low-level operational methods for ontology modification. Section 4 constrains the discussion of general type ontologies to taxonomies and formulates taxonomy modification invariants accompanied by the rules for modification conflicts resolution in implementation independent way. The topic of Section 5 is the outline of the ways to design ontology monitoring and modification services in mediator/wrapper IS architectures with agents orientation. Section 6 summarises the results.

2 Related Work

Mediation of semantically heterogeneous, distributed, autonomous IRs wrapped by (pro-)active software components is a relatively new research field. It originates from the growing demands arising, for instance, in VEMS, DLIB, CSCW, E-Commerce domains. The problem of interoperability among independent IRs with changing semantics is well developed by DataBase research community. Several approaches are known to maintain modifications of database semantics: modification primitives' technique, schema versioning, and schema views. Active Data Dictionary (ADD) [Er98], for instance, is the framework providing for consistent adaptation of relational database schemas to evolving application domain. [Go97] presents the approach to monitor the changes in database schemas for federated database systems.
Interesting propositions of schema changes maintenance in object-oriented databases were given, for instance, in [Za97], [Be99]. [Be99] described the framework for schema changes maintenance in which the methodologies of schema versioning and modification primitives are combined.
The task of detection of the correspondence among the schemas of arbitrary resources (schema matching) is close to the problem of modification maintenance. In both cases the criteria for reasoning on the correspondence of the semantics of several resources as well as for the reasoning on appropriate schema modification is often based on invariant paradigm. The taxonomy covering most of the existing particular approaches to schema matching and the proposition of generic matching methodology is provided by [BR00].
Modification primitives methodology for knowledge bases is exploited by OKBC [OKBC98] for accessing and modifying knowledge instances stored in OKBC-compliant (frame based) knowledge representation systems.
Research in integration of distributed IRs poses the problems of synchronous modifications, merging and alignment of ontologies. Chimaera [Chimaera] and PROMPT [NM00] provide nice examples of ontologies integration maintenance, merging and

alignment. PROMPT algorithm [NM99] is applicable for the cases when the resource ontologies modified in the course of the life cycle of underlying IRs are subjected to the aligning or merging to the common ontology of an IS.
Special attention is paid to the evolution of ontologies in On-To-Knowledge [On-to-Knowledge] project. Semi-automatic ontology integration [Om01] and ontology versioning solutions are considered by On-to-Knowledge consortium to be helpful in solving legacy and matchmaking problems.

3 Ontology Model: Descriptions and Modifications

By the natural analogy an ontology model is denoted as the structure comp rising two components: ontology descriptions and the methods for ontology manipulation.
Let *ontology model* be the structure $< O, \quad >$, where:

O is (c.f. [GK00]) a 3-tuple $< X, \quad , A >$ with X – a finite set of terms, – a finite set of term relationships, A – a finite set of assertions

- is the set of modification primitives, applicable to the elements of O.

In frame of the presented research the *basic elements* of ontology are specified as: X_1 – set of concepts, X_2 – set of properties, – set of relationships, including 'is-a', A – set of Assertions on Concepts, Properties and Relationships. $X = X_1 \quad X_2$ and $X_1 \quad X_2 = \quad .$

The ontologies of the following formal types may be described with the help of these basic elements:

A simple vocabulary of unrelated terms

$$X \quad , \quad = \quad , A = \qquad (1)$$

A lightweight ontology, e.g.: Thesaurus Ontology [Us00]

$$X \quad , \quad , A = \qquad (2)$$

A taxonomy

$$X \quad , \quad = \{is \quad a\}, A = \qquad (3)$$

A passive vocabulary

$$(X = X_1 \quad X_2) \quad , \quad = \quad , A \qquad (4)$$

A general type ontology (provides a vocabulary with the rich set of semantic relationships – elements of are "part of", "kind of", ... [St93]):

$$(X = X_1 \quad X_2) \quad , \quad , A \qquad (5)$$

The analysis of the known languages used for ontology description (KIF [GF92], Ontolingua [Ontolingua], OKBC [OKBC, 1998], OIL [Kl00], SYNTHESIS [Ka93]) shows

that they do not provide the means for *declarative* description of manipulations on ontology elements (as it was done, e.g. in SQL/DDL, SQL/DML).
In this paper ontology modification primitives are considered to be declarative means for manipulating ontology elements. The very high idea of the manipulation part of ontology model is close to that of modifying OODB schemas by means of modification primitives (see e.g.: [Za97]).
Given are the modification primitives per IR Ontology basic elements (general type ontology):

```
CONCEPT:          add, delete (retire), rename

PROPERTY:         add, delete (retire), rename,
                  change domain,
                  change constraints on property values

RELATIONSHIP:     add, delete (retire), rename

ASSERTION RULE: add, delete
```

Modification primitives' technique is often combined with versioning (e.g. [Ol99], [Be99]). It is thus reasonable to make ontology elements retired or obsolete instead of simple deletion. Ontology modification primitives are the low-level methods enabling the construction of the means for ontology manipulation. Ontology manipulation is not discussed in the paper because of the space limitations.
Simple examples of the specification of ontology elements' modification primitives (OKBC procedures) are given in Tables 1, 2. In more complex cases the algorithms for the correct propagation of changes among the related elements of the same ontology should be specified.
Suppose a fragment of a relational database (IR), presented with ER schema given on Fig.1a, is modified: entity representing concept E0 is deleted, and new entities E1 (A0, A1, A3) and E2 (A0, A2, A4) are added. They appear to become new concepts of the IR ontology. Corresponding OKBC program is presented on Fig. 1c.
The affected elements of the IR ontology later (after IS monitoring service reports the mass of ontology changes has become critical) appear to be aligned to the matching elements of the common IS ontology (Fig. 1b). See Section 5 for more details.

4 Taxonomy Modification Invariants

The implementation of ontology modification means for both wrapper and mediator services of an IS depends on the choice of ontology representation and manipulation language. Ontology modification invariants are the constraints on the possible changes of ontology elements (following [Ka83]). The invariants vary for different ontology types. They also vary depending from the formal semantics of one or another ontology representation language. These invariants may be used in various tasks:

In (re-)registration processes – mapping IR ontologies to common IS ontologies
In maintaining ontology elements' changes – matching changed IR ontology elements to common IS ontology elements

Table 1. Modifications of the element of CONCEPT category

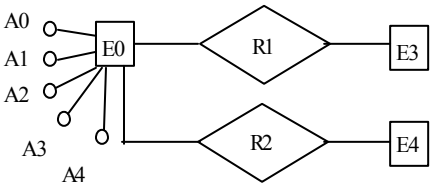
Ontology Element and corresponding Modification Primitives		**Appropriate OKBC construct**
CONCEPT – the main element of resource ontology		**FRAME** – represents a **CONCEPT** apart from hierarchy of already existing concepts
"Add concept"	Adds a concept with unique name and empty set of properties	**create-frame ()**
"Rename concept"	Renames a concept	**put-frame-name ()**
"Delete (Retire) Concept"	Deletes a concept. If there exists a property with domain that is equal to (or is the sub-domain of) the extensional of the concept examined for the deletion then forbid deletion until the definition of this property domain changes	**delete-frame ()**

Table 2. Modifications of the element of PROPERTY category

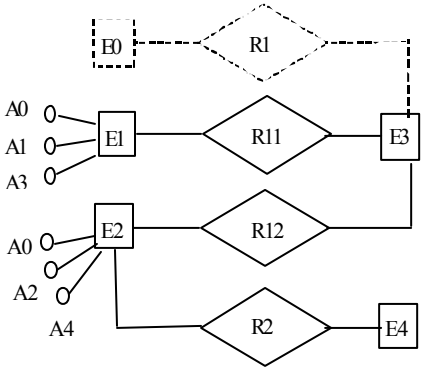
Ontology Element and corresponding Modification Primitives		**Appropriate OKBC construct**
PROPERTY – the characteristic of the CONCEPT. May represent a relationship between CONCEPTS		**SLOT** – represents a **PROPERTY**
" Add property"	Adds a property to the given concept (if exists), with given domain (if exists), property should have unique name within concept, including all parent concepts of this concept.	**create-slot ()** **attach-slot ()**
"Rename property"	Renames a property with a new unique name.	**rename-slot ()**
"Change the domain of property"	Changes the domain of property values, and of al corresponding properties in child concepts. If new domain is (sub) domain of extensional of some concepts, check if that concept is defined.	1. For single-valued domain: **replace-slot-value()** 2. For multi-valued domain: **add-slot-value()**, **put-slot-value()** 3. For domain, based on a frame: a set of OKBC-operators
"Delete (Retire) property"	Deletes a property from the concept and in all concepts which are subclasses of given, except for those where the property was defined as local.	**delete-slot()**

(a) Resource Ontology

Before modification



After modification

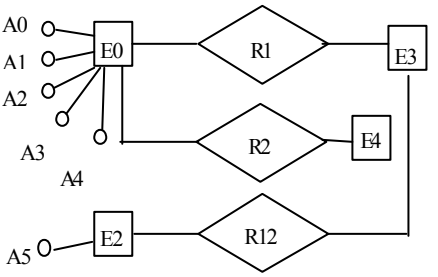


Legend: Retired elements

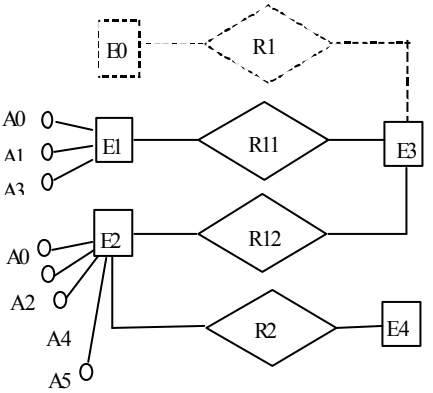
Actual elements

(b) Common IS Ontology

Before modification



After modification



(c)

```
if not frame-in-kb-p(E1 kb true)
 { create-frame (E1 :class true)
    attach-slot (E1 A0 kb (:template) true)
    attach-slot (E1 A1 kb (:template) true)
    attach-slot (E1 A3 kb (:template) true)
    create-slot (R11 kb)
    attach-slot (E1 R11 kb (:template)true)
 } else /*omitted*/
if not frame-in-kb-p(E2 kb true)
 { create-frame (E2 :class true)
    attach-slot (E2 A0 kb (:template) true)
    attach-slot (E2 A2 kb (:template) true)
    attach-slot (E2 A4 kb (:template) true)
    create-slot (R12 kb)
    attach-slot (E2 R12 kb (:template)true)
    attach-slot (E2 R2 kb (:template) true)
```

Fig. 1. (a) An example of modification of IR ontology; **(b)** corresponding modifications of the common IS ontology; **(c)** modification algorithm (OKBC)

For brevity reasons this section introduces modification invariants for a *taxonomy* (4) – a widely spread ontology type.

I1: **Concepts hierarchy invariant**. Concepts' hierarchy in a taxonomy should be preserved as a rooted and connected directed acyclic graph. The inheritance hierarchy should have only one root concept. Concepts should be uniquely named.

I2: **Distinct name invariant**. Required is that all properties and relationships linked to a concept, whether defined or inherited, have distinct names.

I3: **Single origin invariant**. All properties of a concept should have a single distinct origin.

I4: **Complete inheritance invariant**. A concept should inherit all properties from each of its parent concepts, except the cases when complete inheritance violates **I2** and **I3**. **I5**: **Property domain inheritance invariant.** If a property appears both in concept and in its child concepts, then the inherited property domain in the child concepts must be either equivalent to the one of the parent concept or constraining the one of the parent concept.

The main function of ontology modification invariants is to grant consistent ontology modifications. If these invariants are violated, modifications of ontology elements may lead to various types of conflicts (naming conflicts, inheritance conflicts etc.). Conflict resolution procedures depend on the language and the knowledge base system chosen for the implementation. There is anyway the possibility to outline these rules in the language independent manner for the given (taxonomy) type of ontology. The guideline rules based on [Za97] are given below.

First group of rules regulates the resolution of conflicts caused by multiple inheritance and the redefinition of properties in child concepts.

R1: If a property is locally defined in a concept and its name corresponds to the name of an inherited property of one of its parent concepts, the locally defined property overrides the inherited one.

R2: If two or more parents of a concept have properties with the same name but distinct origin, both properties are inherited and renamed. Assigned are new default names to the inherited properties. Alternatively, new names may be assigned manually.

R3: If two or more parents of a concept *X* have property with the same name and with the same origin, the property is inherited only once. If one of these properties has been locally redefined in these parents, then the property must be redefined in concept *X*.

Second group of rules concerns the propagation of modifications to child concepts.

R4: A modification to a property of a concept is always propagated to all its child concepts, except to those in which the property has been locally redefined.

R5: A creation of a property in a concept requires that no locally defined property with the same name is present in the concept. Further, it requires the presence of not more than one child concept containing a property with the same name.

R6: Only locally defined properties can be deleted (retired) from a concept.

R7: Modification of the name of a property in a concept is not propagated to its child concepts.

Third group of rules concerns the aggregation and deletion (retirement) of inheritance relationships between concepts and the creation and removal (retirement) of inheritance relationships.

R8: If a concept *Y* is added to the list of parent concepts of a concept *X*, any conflict of inheritance is resolved by **R1**-**R3**.
R9: The deletion (retirement) of a concept *Y* from the list of parent concepts of a class *X* removes the inherited properties in *X* and its child concepts. If *Y* is the last parent concept in the list, removing of concept *Y* makes *X* a direct child of the root concept.
R10: Only leaf concepts can be removed from the ontology.
R11: New concepts can be created only as leaf concepts in the taxonomy. If a new concept *X* is created without any indication of which concept it should inherit from, *X* becomes the child of the root concept.

5 Ontology Monitoring and Modification Services

One of the characteristic tendencies observed now in the IS and IT domains is the movement from operating the collections of data to the provision of the bulks of intelligent services distributed over the Net [WS00].
This means that today's generation of mediator IS require and the IRs provide information services for the IS to acquire the information and to aggregate the results of the cluster of queries rooted to matching resource wrappers. Analogous architectural solutions may be found in InfoSleuth, OBSERVER, TSIMMIS [Ga95], MOMIS [Be98]. Most of these ISs are agent-based. Various services like resource wrapping, information matchmaking query formulation, decomposition, rooting, resource monitoring are executed by collaborative teams or coalitions of intelligent software agents.
The introduction of the agents' layer to these architectures brings up one more aspect of semantic interoperability – agents' interoperability. One of the evident solutions of the problem of agents' semantic interoperation is the introduction of an ontology agent providing common ontologies for the team of co-operating agents.
Ontology agent, or a dedicated Monitoring agent (InfoSleuth) may be as well assigned to perform the tasks mentioned in Section 1: monitoring of the changes in IR semantics in order to detect the critical mass of these changes and to initiate the routines of IR (re-)registration and (re-)mapping of the IR ontology to the common IS ontology.
The hints leading to the design of the monitoring behaviour are as follows. The changes of an IR ontology may not very critical or really critical:

Renaming of Concepts, Properties, Relationships are *not very critical* modifications to an IR ontology and they do not require re-registration of IR at mediator IS. These modifications do not affect hierarchy of categories and, therefore, only re-mapping of resource ontology elements to the changed terms of underlying resource is necessary.
Deletion (retirement) of Properties or Concepts used in definition of other IR ontology elements, Insertion of new Concepts and Properties to the hierarchy of concepts are *really critical* to an IR ontology and they require re-registration and consequently, re-mapping of the ontology elements.

The monitoring agent of a mediator IS should therefore be tolerant to the modifications of the first type until their quantity has reached a threshold value. The occurrence of the changes of the 2-nd type should cause immediate initiation of the re-registration service.

There are two possible ways of changes' propagation: from resource ontology to the common IS ontology and vice versa.
Assume that resource wrappers are the initiators of the changes within IS. Common IS ontology service (CIOS) provides means for ontology modifications and for matching/aligning IR ontology elements to common IS ontology. Changes' propagation algorithms for the example from Section 3 may then be formulated as follows:
Let: Sch_i – schema of the *i*-th IR; O_i - *i*-th IR ontology; CO – common IS ontology; $C_A, C_D, C_R,$ – sets of added, retired, renamed concepts, $A_A, A_D, A_R, A_{CD}, A_{CC}$ – sets of added, retired, renamed attributes, attributes with changed domain, attributes with changed constraint; R_A, R_D, R_R – sets of added, retired, renamed relationships, AR_A, AR_D – sets of added, retired assertion rules.

Wrapper side:

1. Wrapper ontology service (WOS) monitors structure of the resource and detects the following changes: concept E0 – retired, concepts E1 and E2 – added. Relationship R1 – retired, relationships R11, R12 – added. Attributes A0, A1, A2, A3, A4 – retired from concept E0, then attributes A0, A1, A3 – added to the concept E1, attributes A0, A2, A4 – added to the concept E2.
2. WOS modifies the resource ontology in a semiautomatic way, preserving the invariants, and if necessary, modifies the mapping of ontology elements to the correspondent resource schema elements.
3. WOS generates messages about changes.

Mediator side:

1. CIOS service accepts messages about changes.
2. CIOS matches added and renamed elements. If the matching result for the added element is negative, adds this new element to the common IS ontology.
3. For retired IR ontology elements CIOS checks if there exist other resource ontologies containing similar elements. In case similarities were not found CIOS makes the element retired.

6 Summary

Normally the process of ontology modifications is not delegated to the software because of the possible consequences affecting at least the whole IS. For the testbeds characterised by the presence of both the bulk of autonomously maintained IR ontologies and the common IS ontology Uschold [Us00] have proposed to solve the problem from organisational point of view: to assign a dedicated group of human experts who are responsible for the control (monitoring) of ontology changes and for synchronisation (re-mapping) of resource ontologies. However, a kind of intelligent assistance is helpful to reconcile conflicts emerging within the process. The paper presented the approach to organise the sort of such an intelligent assistance and corresponding automated ontology monitoring and modification services. The approach

intends to do the work in economical way reducing efforts to matching and aligning only modified ontology elements.
The ontologies are classified by the expressive power of underlying formal theories – (2)-(5) and by their function within a mediator IS – IR ontologies, common IS ontology and third party domain ontologies. Proposed was ontology model $< O, \quad >$ comprising both descriptive part $O =< X_1, X_2, R, A >$ and the set of ontology elements' modification primitives . Given is the structural representation of the general type ontology (5). Modification primitives for each ontology structural element are provided. The set of ontology modification invariants and the corresponding set of modification conflicts resolution rules are formulated for taxonomies. The way to provide IS services for ontology changes monitoring, matching and alignment was outlined as well.
One of the important inferences we may conclude the summary with is that well known ideas and approaches from conceptual modeling domain based on the modification primitives and invariants that have been around for years may be used in pretty standard way for ontology manipulation, monitoring, versioning, mapping and matching and alignment.

References

[Ba97] Bayardo Jr. R. J., et al.: InfoSleuth: Agent-Based Semantic Integration of Information in Open and Dynamic Environments (Experience Paper). In (Peckham, J. Ed.): Proc. ACM SIGMOD Int. Conf. on Management of Data, Tucson, 1997. ACM Press, 1997; pp.195-206.

[Be99] Benatallah, B.: A Unified Framework for Supporting Dynamic Schema Evolution in Object Databases. In (Akoka, J.; Bouzeghoub, M.; Comyn-Wattiau, I; MÈtais, E. Eds.): Proc. of the 18th Int. Conf. on Conceptual Modeling (ER 1999), Paris, 1999. Springer, 1999; pp.16-30.

[Be98] Bergamaschi, S.; Castano, S.; De Capitani di Vimercati, S.; Montanari, S.; Vincini, M.: An Intelligent Approach to Information Integration. In (Guarino, N. Ed.): Proc. of the 1st Int. Conf. on Formal Ontologies in Information Systems (FOIS'98), Trento, 1998. IOS Press/ Ohmsha, 1998; pp. 253-268.

[BR00] Bernstein, P.A.; Rahm, E.: Data Warehouse Scenarios for Model Management. In (Laender, A.H.F.; Liddle, S.W.; Storey, V.C. Eds.): Proc. 19th Int. Conf. on Conceptual Modeling (ER2000), Salt Lake City, 2000. Springer, 2000; pp.1-15.

[Chimaera] http://www-ksl-svc.stanford.edu:5915/doc/chimaera/chimaera-docs.html - URL was accessed on May 8, 2001.

[Er98] Ermolayev, V. A.: Object Oriented Dynamic Data Modeling and Active Data Dictionaries: Some Crosspoints. In: Lecture Notes of Zaporozhye State University 1(2):53-63 (1998).

[Ga95] Garcia-Molino, H. et. al.: The TSIMMIS Approach to Mediation: Data Models and Languages. Proc. of the NGITS (Next Generation Information Technologies and Systems), Naharia, 1995.

[GF92] Genesereth, M.R.; Fikes, R.E. et.al.: Knowledge Interchange Format Version 3.0 Reference Manual. Logic-92-1, Stanford University Logic Group, 1992.

[GK00] Gavrilova, T. A.; Khoroshevsky, V.F.: Knowledge Bases of Intelligent Systems. PITER, St. Petersburg, 2000 (in Russian)

[Go97] GoÒi, A.; Illarramendi, A.; Mena, E.; Blanco, J.M.: Monitoring the Evolution of Databases in Federated Relational Database Systems. In (Conrad, S.; Hasselbring, W.; Heuer, A.; Saake, G. Eds.): Proc. of Int. CAiSE'97 Workshop on Engineering Federated Database Systems (EFDBS'97), Barcelona, 1997. Otto-von-Guericke-Universität Magdeburg, Fakultät für Informatik, Preprint Nr. 6, 1997; pp.81-91.

[Ka83] Kalinichenko, L.A.: Methods and Means for Integration of Heterogeneous Databases. SCIENCE, Moscow, 1983 (in Russian).

[Ka93] Kalinichenko, L. A.: SYNTHESIS: the Language for Definition, Design and Programming of Interoperable Environments for Heterogeneous Information Resources. Institute for Problems of Informatics, Russian Academy of Sciences, Moscow, 1993 (in Russian).

[Kl00] Klein, M.; Fensel, D.; van Harmelen, F.; Horrocks, I.: The Relation between Ontologies and Schema languages: Translating OIL-specifications in XML-Schema. In (Horn, W. Ed.): Proc. of 14th European Conf. on Artificial Intelligence (ECAI'00), Berlin, 2000. IOS Press, Amsterdam, 2000.

[NM00] Noy, N. F.; Musen M.: PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment. In: Proc. of the 17th National Conf on AI / 12th Conf. on Innovative Applications of AI (AAAI/IAAI 2000); AAAI Press / The MIT Press, 2000; pp.450-455.

[OKBC98] OKBC standard. Version 2.0-3, April 1998. http://www.ai.sri.com/~okbc/ - URL was accessed on May 8, 2001.

[Ol99] Oliver, D.E.; Shahar, Y.; Musen, M.A.; Shortliffe, E.H.: Representation of Change on Controlled Medical Terminologies. Artificial Intelligence in Medicine 15(1):53-76, (1999).

[Om01] Omelayenko, B.: Syntactic-level Ontology Integration Rules for E-commerce. In: Proc. of the 14th Int. Conf. FLAIRS 2001, Key West, 2001. AAAI Press, 2001 (to appear).

[On-To-Knowledge] http://www.ontoknowledge.org/wpdes.shtml#wp1 - URL was accessed on May 8, 2001.

[Ontolingua] http://www.ksl.stanford.edu/software/ontolingua/project-papers.html - URL was accessed on May 8, 2001.

[Sh00] Sheth, A. et.al.: OBSERVER: An Approach for Query Processing in Global Information System based on interoperation across Pre-existing Ontologies. Distributed and Parallel Databases 8(2): 223-271 (2000).

[St93] Storey, V. C.: Understanding semantic relationships. The Very Large Data Bases (VLDB) Journal 2(4):455-488 (1993).

[St97] Storey, V. C.; Chiang, R. H. L.; Dey, D.; Goldstein, R. C.; Sundaresan, S.: Database design with Common Sense Business Reasoner. ACM TODS 22(4): 471-512 (1997).

[Us00] Uschold, M.: Creating, Integrating and Maintaining Local and Global Ontologies. In (Horn, W. Ed.): Proc. of 14th European Conf. on Artificial Intelligence (ECAI'00), Berlin, 2000. IOS Press, Amsterdam, 2000.

[VMS94] Van der Vet, P.E.; Mars, H.J.; Speel, P.H.: The Plinius Ontology of Ceramic Materials. In (Cohn, A.G. Ed.): Proc. of 11th European Conf. In Artificial Intelligence (ECAi'94), Amsterdam, 1994. John Wiley and Sons, 1994.

[WS00] Wong, H.-C.; Sycara, K.: A Taxonomy of Middle-agents for the Internet. In: Proc. of the 4th International Conference on Multi-Agent Systems (ICMAS'2000), Boston, 2000.

[Za97] Zaniolo, C.; Ceri, S.; Faloutsos, C.; Snodgrass, R.T.; Subrahmanian, V.S.; Zicari, R.: Advanced Database Systems. Morgan Kaufmann Publishers, San Francisco, 1997.