CONCEPTUAL DECLARATIVE PROBLEM SPECIFICATION AND SOLVING IN DATA INTENSIVE DOMAINS*

L. Kalinichenko[1], S. Stupnikov[2], A. Vovchenko[3], and D. Kovalev[4]

Abstract: Various notations aimed at defining the semantics of a computation in terms of the application domains have been experienced for conceptual modeling. For example, entity-relationship (ER) approach and UML (Unified Modeling Language) diagrams allow one to specify the semantics informally. Ontology languages based on description logic (DL) have been developed to formalize the semantics of data. However, it is now generally acknowledged that data semantics alone are insufficient and still representation of data analysis algorithms is necessary to specify data and behavior semantics in one paradigm. Moreover, the curse of ever increasing diversity of multistructured data models gave rise to a need for their unified, integrated abstraction to make specifications independent of real data in data intensive domains (DID). To overcome these disadvantages, a novel approach for applying a combination of the semantically different declarative rule-based languages (dialects) for interoperable conceptual specifications over various rule-based systems (RSs) relying on the logic program transformation technique recommended by the W3C (World Wide Web Consortium) Rule Interchange Format (RIF) has been investigated. Such approach is coherently combined with the specification facilities aimed at the semantic rule-based mediation intended for the heterogeneous data base integration. The infrastructure implementing the multidialect conceptual specifications by the interoperable RSs and mediating systems (MSs) is introduced. The proof-of-concept prototype of the infrastructure based on the SYNTHESIS MS and RIF standard is presented. The approach for multidialect conceptualization of a problem domain, rule delegation, rule-based programs and mediators interoperability is explained in detail and illustrated on a real nondeterministic polynomial time (NP) complete use-case in the finance domain. The research results are promising for the usability of the approach and of the infrastructure for conceptual, declarative, resource independent and reusable data analysis in various application domains.

Keywords: conceptual specification; W3C RIF; logic rule languages; SYNTHESIS; database integration; mediators; RIF-BLD; RIF-CASPD; multidialect infrastructure; rule delegation

DOI: 10.14357/19922264130412

1 Introduction

For decades, informatics is investigating proper facilities for conceptualization of the diverse application domains, descriptive and predictive data analysis to extract information from data, decision making, and generally, for the problem solving based on such insight. Many times a point was reached when the level of maturity of the conceptual modeling techniques could have been evaluated as satisfactory from the theoretical and practical points of view. But every time new technological developments and research results disproved such optimistic opinions. Now, in the epoch of the data intensive research[5] and system development, a lack of the proper declarative conceptual modeling facilities is felt again. For instance, the call for such facilities has been expressed in the ACM (Association for Computing Machinery) publication "Challenges and Opportunities with Big Data" [2]. In particular, the call considers important to examine a combination including

– facilities for support of variety of data (heterogeneity of data types, representation, and semantic interpretation) and semantic data integration; and
– declarative specifications required to meet the programmability and composition of complex analytical pipelines in an understandable form applying appropriate high-level languages to express the analytics.

*This research has been done under the support of the RFBR (project 11-07-00402-a) and the Program for Basic Research of the Presidium of the Russian Academy of Sciences.

[1]Institute of Informatics Problems, Russian Academy of Sciences, Moscow, Russia, leonidandk@gmail.com
[2]Institute of Informatics Problems, Russian Academy of Sciences, Moscow, Russia, ssa@ipi.ac.ru
[3]Institute of Informatics Problems, Russian Academy of Sciences, Moscow, Russia, itsnein@gmail.com
[4]Institute of Informatics Problems, Russian Academy of Sciences, Moscow, Russia, dm.kovalev@gmail.com

[5]Data Intensive Research (DIR) has emerged as a new paradigm for inferring information from data (known as the 4th paradigm for scientific domains [1]) emphasizing the increasing value of observational, experimental, and computer-generated data in virtually all domains. Similarly, "Big Data" is developing as a recognition of the increasing significance of massive multistructured data in almost each area of the human activity.

This paper is related to such statements investigating a novel methodology and infrastructure supporting conceptually-driven problem specifications and solving in the DID. This research is motivated by aiming at the specification reusability in various domains over different sets of information resources (IRs), widely diverse data, and knowledge semantic integration capability as well as at the accumulation of reproducible data analysis and problem solving methods, related specifications, and experience in various DID.

One of the intended outcomes of the work is to expose current practically reachable limit of declarative conceptual specification construction applying the wealth of various available facilities of logic programming, knowledge representation, semantic Web, and heterogeneous database mediation in a coherent, cooperative way.

Conceptual specifications should be expressed in terms of an application domain independently of design and implementation concerns. Usually, the domain specification understandable and acceptable in the respective community includes a set of coherently tied ontologies together with a conceptual schema of the domain defined on the basis of such ontologies. Conceptual specifications of the applications in this domain are defined in terms of the conceptual schema to express abstract representations of information structures and algorithms of their analysis for the application problem solving.

This research reflects an intention to define and implement facilities for conceptually-driven problems specification and solving in DID to create the perspectives ensuring the following capabilities for expressing the specifications:

- declarative nature of the language facilities involved;
- an ability to provide complete and precise specification of the abstract structure and behavior of the domain entities, their consistency, relationship, and interaction;
- well-grounded diversity of semantics of the modeling facilities providing for proper expressiveness, compactness, and precision of the definition of behavior of the domain entities and problem solving algorithm specifications;
- arrangements for the probable extensions of the modeling facilities for conceptual specifications satisfying the changing technological and practical needs;
- provisions for specification independence from implementation languages and systems;
- provisions for specification independence from specific IRs combined with facilities for their semantic integration and interoperability coping with the increasing diversity of the techniques used for implementation of programs, databases, services, ontologies, and other resources;
- built-in methodologies for creation of unifying specification languages providing for construction of semantic preserving mappings of conceptual specifications into their implementations in specific languages and systems; and
- combination of ease of a human perception with a sufficient efficiency and expressiveness of the implementation for publication in communities intended for specification reuse.

The brief vision of the state-of-the-art in conceptual modeling (not claimed to be exhaustive) is as follows. Various notations aimed at defining the semantics of a computation in terms of the application problems to be solved have been experienced for conceptual modeling for many years. Structured and object-oriented specification notations for conceptual modeling in graphical (diagram) form dominate. Entity-relationship approach models the domain in terms of entities, attributes, and relationships, extended sometimes with cardinality constraints, aggregation, sets, temporality, taxonomic structures. Many parts of UML are the diagrams for specific purposes (e.g., class diagrams, activity diagrams, use cases, state charts, object interaction diagrams) used for the domain modeling and the design phase in software engineering. Such notations often allow one to specify the semantics informally, in terms of natural language; or the semantics is hard-coded using the tools supporting the notation [3]. Various logic specification languages, such as ontology languages based on DL, have been developed to describe the semantics of data. With certain limitations, the ontology languages allow one to define formal models of domain concepts, their taxonomic relations, and constraints over them. They facilitate automated reasoning (inference) over taxonomic structures. However, it is now generally acknowledged that data semantics alone are insufficient and representation of behavior is necessary to specify data and behavior semantics, algorithms for data analysis in one paradigm.

This is just what paper is focused on, emphasizing the importance of declarative nature of the specification facilities. In particular, the approach proposed is aimed at conceptual modeling related to DID applying rule-based declarative languages possessing different, complementary semantics and capabilities combined with the methods for heterogeneous data mediation and integration.

In the work presented, the issues of interoperability of declarative programs defined in the languages having different semantics combined with an ability to integrate various IRs (such as data and knowledge bases, software services, ontologies) for problem solving are investigated on the basis of two fundamental techniques:

(1) constructing of the unifying extensible language providing for semantic preserving representation in it of various IR specification languages (e. g., such as data definition (DDL) and manipulation (DML) languages for databases); and

(2) creation of the unified extensible family of rule-based languages (dialects) and a model of interoperability of the programs expressed in such dialects with different semantics.

The first technique is based on the experience obtained in course of the SYNTHESIS language development [4, 5] accompanied by the methods and facilities for constructing its extensions. The kernel of the SYNTHESIS language is based on the object-frame data model used together with the declarative rule-based facilities in the logic language similar to a stratified Datalog with functions and negation. The extensions of the SYNTHESIS kernel are constructed in such a way that each extension together with the kernel is a result of semantic preserving mapping of some IR language (mainly, a data model language considered to be a combination of the respective DDL and DML) into the SYNTHESIS [6, 7]. The canonical information model (CIM) is constructed as a union of the kernel with such extensions constructed for various IR languages. Canonical information model is used for development of mediators positioned between the users, conceptually formulating problems in terms of the mediators, and various distributed IRs (such as databases and services) needed for a specific application. A schema of a subject mediator for a class of problems includes the specification of the domain concepts defined by the respective ontologies.

Another, multidialect technique for rule-based programs interoperability applied in the current work is based on the RIF standard [8] of W3C. Rule interchange format introduces a unified family of rule-based languages (dialects) together with a methodology for constructing semantic preserving mappings of specific languages used in various RSs[1] into such dialects. Besides, a methodology for extension of the family with the definition of new dialects is also defined as a part of RIF. From the RIF point of view, an IR is a program developed in a specific language of some RS. For the logic dialects, the RIF family should include declarative, high-level languages reflecting in a systematic way the basic semantics of rule-based languages of various existing logic-based RSs. These RSs and their declarative rule-based languages form now a mature technology with decades of theoretical development, practical and commercial use based on logic programming and non-monotonic reasoning.

The approach proposed here is built on the techniques briefly introduced above aiming at construction of declarative specifications containing definitions of the concepts and their relationships in the DID area (conceptual schema), conceptual definitions of problems, and their respective algorithms in declarative form simplifying their reuse. The *conceptually-driven* problem specifications of an application can be defined applying advanced declarative languages, modularization, and composition facilities. Thus, the problem specifications can be considered also as domain-driven, that is, they are defined in terms of the domain independently of IRs (databases, services, workflows, mediators) and implementation notations.

The paper presents the results obtained including the description of an approach and an infrastructure supporting:

- application domain conceptual specification and problem solving algorithms[2] definitions based on the combination of the heterogeneous database mediation technique and rule-based multidialect facilities;
- interoperability of distributed multidialect rule-based programs and mediators integrating heterogeneous databases; and
- rule delegation approach for the peer interactions in the multidialect environment.

The proof-of-concept prototype of the infrastructure based on the SYNTHESIS environment and RIF standards has been implemented. The approach for multidialect conceptualization of a problem domain, rule delegation, rule-based programs, and mediators interoperability is explained in detail and illustrated on a real NP-complete use-case in the finance domain. For the conceptual definition of the use-case problem, Web Ontology Language (OWL) is used for the domain concepts definition and programs in two dialects RIF-BLD (Basic Logic Dialect) and RIF-CASPD (Core Answer Set Programming Dialect [9]) mapped into the SYNTHESIS formula language and answer set programming (ASP) based DLV [10] language, respectively.

The paper[3] is structured as follows. After introduction, the sections containing an overview of the approach and an infrastructure for distributed multidialect rule-based programs support are given. In section 4,

[1]Examples of RSs include SILK, OntoBroker, DLV, IBM ILOG, IRIS, etc. More examples can be found at http://www.w3.org/2005/rules/wiki/Implementations.

[2]In the approach proposed, the logic programming model of computation is assumed.

[3]The approach presented in this paper has been reported for the first time at the ADBIS'2013 conference and annotated in a short publication [9].

the lessons learned during the use-case specification and implementation are summarized. A related work section, the future plans section that outlines the intentions for the future research, and the conclusion which summarizes the results end the paper. The detailed description of the use-case and its implementation is given in an appendix. Such detailed exposition of the results is justified by the novelty of the approach and a necessity to show and make readable all the steps required for the specification and implementation of the use-case applying the languages and the infrastructure proposed.

2 An Approach for Conceptual Problem Specification and Solving Based on the Techniques for Heterogeneous Database Mediation and Multidialect Program Interoperability

According to the above, the problems of integration of various IRs and interoperability of rule-based programs arising during application development are proposed to be resolved on the combination of two techniques:

(1) construction of the unifying extensible language providing for representation in it of various IR languages preserving their semantics and for creation of mediators for integration of heterogeneous IRs; and

(2) creation of the unified family of languages (dialects) for expressing the problem solving algorithms and the model of interoperability of programs represented in such dialects.

2.1 Heterogeneous Database Mediation

In this research, the first technique is based on the results of the SYNTHESIS language project [4] including the methods and means for the development of this language extensions and for its use for database integration.

The *Unifier of information models* [11] is the main facility created for the development of the SYNTHESIS language extensions and the mappings of specific data models into such extensions. Preserving semantics of a specific data model by its mapping into SYNTHESIS in the Unifier is based on the specification *refinement* principle (it is said that specification A refines specification B if it is possible to use A instead of B so that a user of B does not notice such substitution) developed by Abrial [12]. Thus, it is possible to construct provably correct extensions of the SYNTHESIS language and mappings of specific languages into them based on the proof of refinement. Such methodology has been applied for mapping into SYNTHESIS of diverse database languages, service and process specification languages, ontological languages [5–7, 11]. Now, a possibility of mapping into SYNTHESIS of various multistructured data models typical for the Big Data epoch (such as, e. g., various NoSQL models [13], graph-based models [14], triplet data model [15], and array data model [16]) is studied.

The main characteristic properties of the SYNTHESIS language use as the kernel of the extensible language are as follows:

(1) the merge of the kernel extensions constructed for diverse data models is treated as a single language known as the CIM; and

(2) CIM is considered as a basis for the development of mediators positioned between the programs providing for problem solving in terms of mediators (the application domain terms) and specific distributed IRs (such as database, services) selected for the problem solving. The mediator specification for a class of problems includes definitions of the application domain concepts expressed by the respective declarative specifications, specifications of the classes of the domain entities, specifications of types of the instances of the classes mentioned and of their methods, and specification of functions.

A mediator in SYNTHESIS can be defined as a triple (T, S, M) where T is the mediator (target) schema; S is the IR schema; and M is the set of assertions (rules) relating elements of the target schema with elements of the resource schema. It is assumed that IR data model (that is, the respective DDL and DML) has been mapped into the canonical one; thus, both resource and target schemas are defined in the canonical data model. Both resource and canonical data models may allow for the expression of various constraints Σ. In this generality, a data integration setting is (S, T, Σ_r, Σ_{rt}, Σ_t) in which S and Σ_r constitute the IR schema, T and Σ_t form the target schema, and the resource-to-target dependencies included into Σ_{rt} are the assertions of the schema mappings in the mediator. Σ_{rt} should be defined as a set of assertions of weakly-acyclic class [7, 17]. Note that Σ_r includes not only data dependencies corresponding to the IR database in the original data model, but also the dependencies generated in accordance with the extension of the kernel of the canonical data model determining an information-preserving mapping of the resource data model into the canonical one (some categories of such dependencies are shown in [6, 7]).

Specifications of mediators are expressed in CIM. Generally, a specification of M in the approach oriented on the definition of the mediator schema T in terms of the application domain independently of schemas of IRs involved is based on the GLAV (Global and Local As View) [18, 19] schema mapping technique. In GLAV, two schema mapping techniques are combined: LAV (Local as View) according to which a schema of IR is defined as a view over the mediator schema and GAV (Global as View) according to which the mediator schema is defined as a global view over the IRs schemas. In LAV, the schemas of IR classes are considered as materialized views over the virtual classes of the mediator. Usually, GAV views are used for resolving various conflicts between the specifications of IRs and of the mediator. In such cases, the GAVs provide the rules for transformation of the results of a query over IRs to their representation in the mediator. The GLAV technique provides for stability of the mediator specification in time of the specific IRs change or the change of the actual IR presence (remove of an IR, adding of new resources, and so on). Also, the GLAV technique provides for scalability of mediators with respect to the number of IRs involved in them. It worth mentioning that the mediators, in their turn, can play the role of IRs towards the mediators of the higher level.

The following characterization of the GLAV setting is applied in the SYNTHESIS project. The sound GLAVs are assumed [18]. The GLAV mapping is constructed as a composition of two schema mappings — GAV mapping $M_{\mathrm{rg}} = (S_r, S_g, \Sigma_{\mathrm{rg}})$ (from resource schemas to GAV schema) and LAV mapping $M_{\mathrm{gt}} = (S_g, T_{\mathrm{rt}}, \Sigma_{\mathrm{gt}})$ (from GAV schema to mediator (target) schema). GLAV is formed by a definition of the composition $M_{\mathrm{rg}} \circ M_{\mathrm{gt}}$ by means of the GAV and LAV dependencies (Σ_{rg} and Σ_{gt}, respectively). The GLAV mappings are formed showing how two parts of the schema (resource and target) are linked together by the Σ_{rt} dependencies.

Queries and programs over the mediator are expressed declaratively in the SYNTHESIS formula language (resembling Datalog) in terms of the mediator schema T [4]. An example of the mediator definition can be found in appendix (see subsections A3.1 and A3.2).

2.2 Diverse Rule-Based Programs Interoperability

Another fundamental principle used in this research is based on a multidialect technique intended for providing diverse rule-based programs interoperability and known as the W3C RIF standard [8]. The range of intended applications of the RIF standard (besides the Semantic Web) includes also development of the intelligent information systems as well as knowledge representation in various application domains. The family of dialects proposed in RIF has a common core (the *Core dialect*) and an ensemble of the dialects extending the core and forming the directed acyclic graph. An arc of the graph is directed from the dialect being extended to the dialect extending it.

According to RIF, any specific RS can act in two independent roles — the role of *provider* and the role of *consumer*. The first one indicates that RS is capable to transform its native rule-based programs into the programs in the respective dialect; the second one indicates that RS is capable to accept a program defined in a proper dialect and transform it into the program in the native RS language. Thus, to include any rule-based language into the family of interoperable dialects, it is required to provide the respective RS with two semantic preserving transformers — from its own language into the proper dialect (the role of provider) and from the proper dialect into its own language (the role of consumer). In contrast to the development of the SYNTHESIS language extensions, each dialect extending other dialects may have its own, different of others, semantics. For example, for logic rule-based languages, each dialect can use its own semantics (e. g., the classic first-order semantics), well-founded semantics, stable model semantics, etc. [14]. Unlike the SYNTHESIS language, in which for development of the language extensions and respective semantic preserving language mappings, it is required to establish the refinement relation between the specifications, in RIF for the logic rule-based languages, it is required to establish the *entailment* relation between the specifications. The entailment relation between formulas f and g, $f| = g$ (f entails g) is valid if and only if g is true for all models where f is true.

In the RIF standard, only the basic, simple dialects have been fixed. Thus, RIF-BLD [20] corresponds to Horn logic with various syntactic and semantic extensions. RIF-PRD (Production Rules Dialect) [21] specifies a production rules dialect to enable the interchange of production rules. RIF-PRD captures the main aspects of various production rule systems. Production rules are defined using *ad hoc* computational mechanisms, which are not based on a logic. The condition language of RIF-PRD is defined as a common subset of RIF-BLD and RIF-PRD. RIF-Core (Core Dialect) is a subset of both RIF-BLD and RIF-PRD thus enabling limited rule exchange between logic rule dialects and production rules.

Significant contribution of RIF consists in inclusion into the standard of the Framework for Logic Dialects RIF-FLD [22]. RIF-FLD represents quite general logic language including the broad set of syntactic and semantic constructs frequently used in logic. On the basis of the framework, concrete dialects having specific semantics can be defined. Definitions of all dialects in the standard are strictly formalized. Rule interchange format defines the general conception of the unified language construction as a family of dialects intended for provision of the interoperability of the numerous RSs. On the basis of the RIF-FLD framework, several dialects have been designed in the world. They need still to pass through the procedure of approval by W3C. Among these dialects are the following. RIF-CLPWD (Core Logic Programming Well-founded Dialect [23]) uses WFS [24] with the default negation and functions. RIF-CASPD [25] uses the semantics of ASP, known also as the stable model semantics [26]. Difference of semantics and application areas of WFS and ASP is a typical example showing different capabilities of the rule-based programs expressible in the languages with such semantics. The WFS and ASP language classes correspond to two philosophically different approaches in logic programming: WFS keeps the idea of defining a single (well-founded) model for a program while ASP provides for the computation of the set of preferred models that is more than query answering. Thus, the ASP-based systems are specifically oriented on solving the complex, combinatorial (NP-complete) problems. The ASP application areas include: diagnosis, information integration, constraint satisfaction, reasoning about actions (including planning), routing, and scheduling, health care, biomedicine and biology, text mining and classification, question answering, etc. In contrast to the ASP systems, the WFS-based systems incorporate three-valued logic are computationally complete and can be used as the universal logic programming facilities. The joint usage of programs with different semantics for problem solving is obviously reasonable.

The RIF standard has also defined the necessary concepts to ensure compatibility of RIF with resource description framework (RDF) and OWL [27], in spite of dissimilarity of their syntaxes and semantics. Rule interchange format uses its frame syntax to communicate with RDF/OWL. These frames are mapped onto RDF triples and a joint semantics is defined for the combination. The basic idea is that rules and OWL will view each other as "black boxes" with well-defined interfaces through exported predicates. The OWL-based ontologies will export some of their classes and properties, while rule-based knowledge bases will export some of the predicates that they define. Each type of the knowledge base will be able to refer to the predicates defined in the other knowledge bases and treat them extensionally as collections of facts. Ontologies are assumed to provide commonly shared conceptualization of a domain. There may be different application-specific rule programs for different applications in the domain.

In the use-case considered in the paper, the following languages for conceptual specifications are used: OWL 2, RIF-BLD and RIF-CASPD (the examples of specifications in these languages can be found in the appendix, section A2).

2.3 Basic Ideas of the Conceptual Problem Solving Approach Built on the Symbiosis of the Above

The idea of the conceptual problem solving approach considered in the paper consists in developing a modular infrastructure in which alongside with the mediators integrating data and services in the SYNTHESIS-like setting, the modules representing knowledge and declarative rule-based programs over various resources will be introduced. The infrastructure is based on the following principles:

- *the multidialect construction of the conceptually-driven problem specifications combining the SYNTHESIS and RIF approaches*. The specifications are represented as a functional composition of declarative modules, each based on its own language (dialect) with an appropriate semantics. Semantics of a conceptual definition in such setting becomes a multidialect one. To the SYNTHESIS language, one of the existing RIF dialects (or a new, specific dialect) should be put into correspondence (currently, OWL is used for the conceptual schema specification together with the RIF-BLD for the mediator rules to be interpreted in SYNTHESIS);
- *the specification modules are treated as peers*. Rule-based program modules are included into the specification alongside with the mediators that can be supported by various MSs. Interoperability of such combination of modules is based on P2P (peer-to-peer) and W3C RIF interoperability techniques;
- *combination of integration and interoperability*. The IR integration can be provided in the scope of individual mediator modules. Interoperability is provided between the modules supporting different dialects according to the RIF methodology; and
- *rule-based specifications on different levels of the infrastructure*. Rule-based, inference providing modules can be used for declarative programming over the mediators. In this case, the mediators support schema mapping for semantic integration of the IRs supported by various MSs.

In the paper, the ideas of construction of the infrastructure providing for conceptual declarative programming and heterogeneous IR mediation interoperating in the multidialect environment are presented.

The infrastructure generally outlined in Fig. 1 provides for implementation of conceptual specifications on the basis of the CIM-like kernel and its extensions used in the mediators combined with declarative languages of the rule-based programming systems possessing diverse semantics and capabilities. Such environment provides for

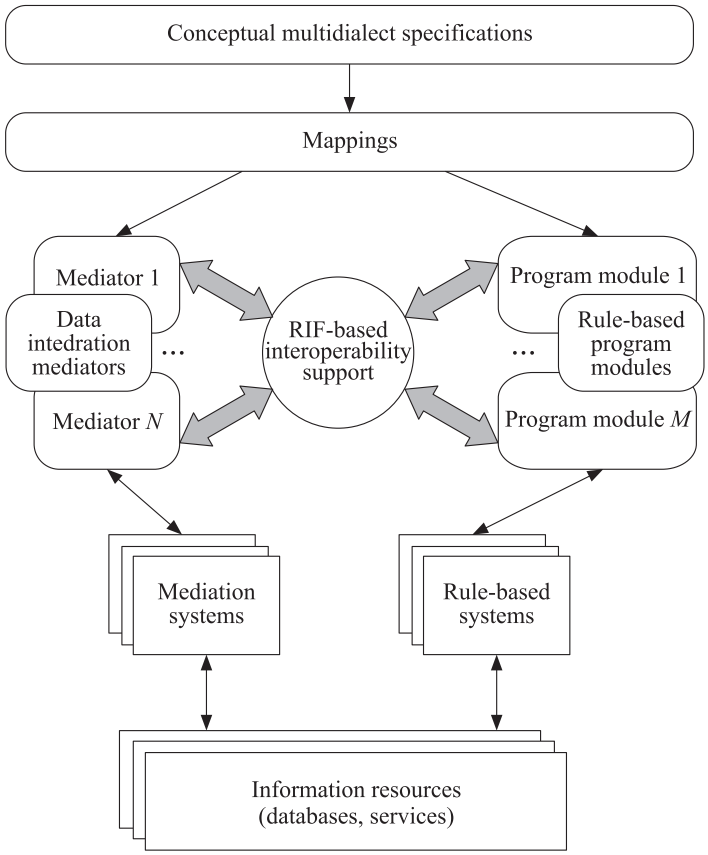

Figure 1 Infrastructure for conceptual problem solving

programming over heterogeneous resources (e.g., over databases and services) accessed through the mapping of specific resource schemas into the target schemas of the mediators. Multidialect program is implemented by means of transformation of conceptual specifications into modular, component-based P2P program represented on the languages of the mediation and rule-based systems. Interaction of components of such distributed program is carried out by means of the delegation technique (see subsection 3.3).

An approach for transformation of conceptual specifications into the distributed mediators and modules interpreted in RSs with different semantics interoperating in P2P style by the rule delegation in the multidialect W3C RIF-based environment is presented in section 3.

Practical value of the results obtained is demonstrated with a use-case in the finance domain. The important result of the research conducted consists also in accumulation of the experience of usage of the multidialect specification facilities for *conceptually-driven problem specifications* in specific domains and implementation of such specifications in the prototype that has been developed.

3 Infrastructure of the Multidialect Environment for Distributed Rule-based Programs and Mediators Interoperability

3.1 Conceptual Programming over the Conceptual Schema

The aim of the novel infrastructure proposed is a conceptual programming of problems in the RIF dialects and an implementation of conceptual specifications using declarative languages of the RSs and MSs. Conceptual multidialect logic programs specify the algorithms for problem solving in a subject domain. They are implemented using their transformation into RS or MS programs.

Conceptual specification of a problem (class of problems) is defined in the context of a subject domain and consists of a set of *RIF-documents* (document is a specification unit of RIF). Each document contains the groups of rules. Conceptual specification of a problem is an abstract program of a problem solving.

The subject domain conceptualization is performed using OWL 2 [28] ontologies containing entities of the domain and their relationships (Fig. 2, right-hand part). Thus, the *conceptual schema* of the domain can be formed. Conceptual specification is defined over conceptual schema: names of the entities (classes and attribites) and only they are used in the rules of the RIF-documents as extensional predicates.

Ontologies are imported into the RIF-documents specifying an import profile, for instance, *OWL Direct*. A profile defines a semantics of an OWL. The profiles are formally defined in [27].

Modular construction of the conceptual specification is based on the techniques of document import and link. Each RIF-document is defined using a dialect which is a *specialization* of the RIF Framework for Logic Dialects (RIF-FLD) [22]. Documents *import* other documents having the same semantics (the *Import* directive) or *link* documents defined using other dialects and having different semantics (remote module directive *Module*).

Documents refer to predicates from imported documents (using the name of an imported module *module:predicate*) as well as to predicates from remote modules using *remote terms f@r* [22]. Here, f means a term and r means a reference to a remote module.

Retrieving results of problem solving is performed by querying the virtual knowledge/database formed by combined infrastructure of mediators and rule-based systems.

Queries are formulated in terms of the conceptual schema and the intentional predicates of the conceptual specification of a problem. The result is a Boolean value (if the formula is closed) or a collection of tuples of values of free variables of the formula.

3.2 Mapping of the Peer Schemas into the Conceptual Specification

In the proposed infrastructure, (1) the logic programs implementing RIF-documents of the conceptual schema in specific RSs and (2) the mediators supporting virtual collections of facts as the result of heterogeneous databases integration are considered as peers.

A schema S_R of a peer R is a set of entities (classes or relations and their attributes) corresponding to extensional and intensional predicates of the resources.

The RS or the MS of each peer R should be a *conformant* D_R *consumer* where D_R is the respective RIF dialect (Fig. 2, left-hand part). Conformance is formally defined using formula entailment and language mappings [22].

The peer R is *relevant* to a RIF-document d of a conceptual specification of a problem (see Fig. 2, right-hand part) if:

- D_R is a subdialect of the document d dialect; and
- entities of the peer schema S_R (if they exist) are *ontologically relevant*[1] to entities of the conceptual schema the names of which are used in d for extensional predicates.

The schema of a relevant peer is mapped into the conceptual schema. The mapping establishes the correspondence of the conceptual entities referenced in the document d to their expressions in terms of entities of the schema S_R using logic rules of the D_R dialect. These schema mapping rules constitute separate RIF-document (Fig. 2, middle part). An example of mapping is given in appendix (see subsection A3.1).

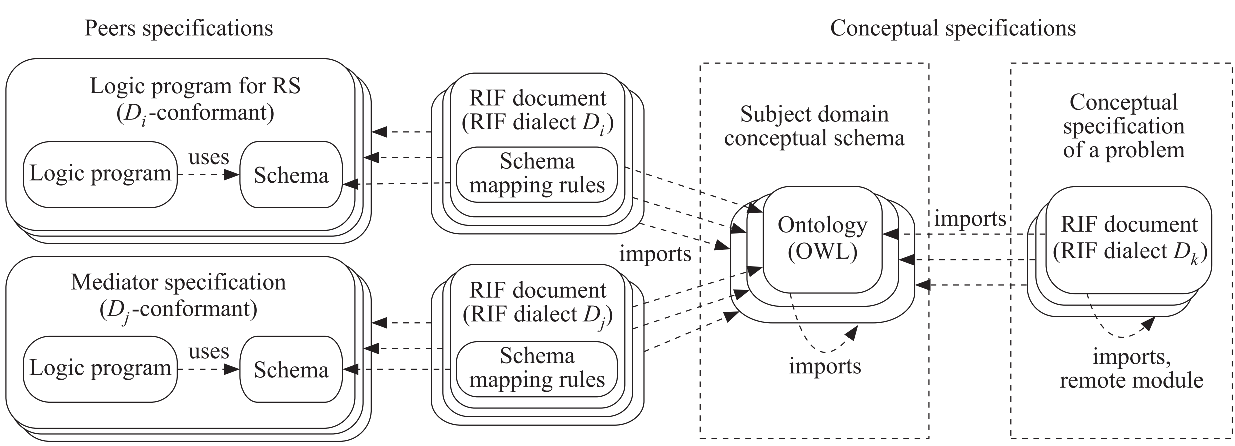

Figure 2 Conceptual and peer specifications

[1] In this paper, the methods for ontological matching of the schemas [15] are not considered.

3.3 Implementation of the Conceptual Specification

Programs defined in the conceptual specification are implemented in P2P environment formed by relevant peers which are related to conceptual specification by mapping rules (see Fig. 2, middle part).

Peers communicate using a technique for distributed execution of the logic programs. The basic notion of the technique is *delegation* — transferring facts and rules from one peer to another. A peer is installed on a *node* of the multidialect infrastructure. A node is a combination of a wrapper, an RS or an MS, and a peer (Fig. 3). A wrapper transforms programs and facts from the specific RIF dialect into the language of the RS or MS and *vice versa*. A wrapper also implements the delegation mechanism. A definition of the delegation is given in the latter part of this subsection. Transferring facts and rules among peers is performed in RIF dialects. The wrappers implement an interface *RIFNodeWrapper* (see Fig. 3).

A special component (*Supervisor*) of the proposed architecture stores shared information of the environment, i. e., conceptual specifications related to the domain and to the problem, a list of the relevant resources, RIF-documents combining logic rules for the conceptual specification, and a resource schema mapping.

Implementation of the conceptual specification includes the following steps:

- rewriting the conceptual documents into the RIF-programs of the peers performed by the *Supervisor*. A rewriting includes also (1) replacing the document identifiers (used to mark predicates) by peer identifiers; and (2) adding schema mapping rules to programs (see Fig. 2, middle part);
- a transfer of the rewritten programs to nodes containing peers relevant to the respective conceptual documents. The transfer is performed by the *Supervisor* by calling the method *loadRules* of the respective node wrappers;
- a transformation of the RIF-programs into the concrete RS or MS languages. The transformation is performed by the *NodeWrapper* or by the RS or MS itself (if the RS or MS supports the respective RIF dialect); and
- an execution of the produced programs in P2P environment.

During the process of rewriting the conceptual schema into the resource programs, the relationships between RIF-documents of the conceptual schema defined by remote or imported terms are replaced by relationships between peers also defined by remote or imported terms. To implement remote and imported terms, a *rule delegation* mechanism is used — transferring facts and rules from one peer to another. The details of rule delegation approach implemented in the current prototype of the multidialect infrastructure are provided below.

In general case, the programs transferred to some peer may include *nonlocal* rules. These rules contain remote or imported terms. On the contrary, *local* rules do not contain such terms. For simplicity, only remote terms are mentioned in the latter part of this subsection. To make possible an execution of a program on a node, the program should be *normalized*, i. e., transformed into an equivalent program including only local rules and *delegation rules*. There are two kinds of delegation rules for a peer n:

(1) *fact delegation rule* like $p@m(X)$: -$q@n(X)$ where p and q are the predicate names; X is the variable list; and m is the peer different from n. The rule means that all the facts turning q into *true* have to be transferred to the peer m as facts turning p into *true*. A fact is a term $p(v_1, \ldots, v_n)$ such

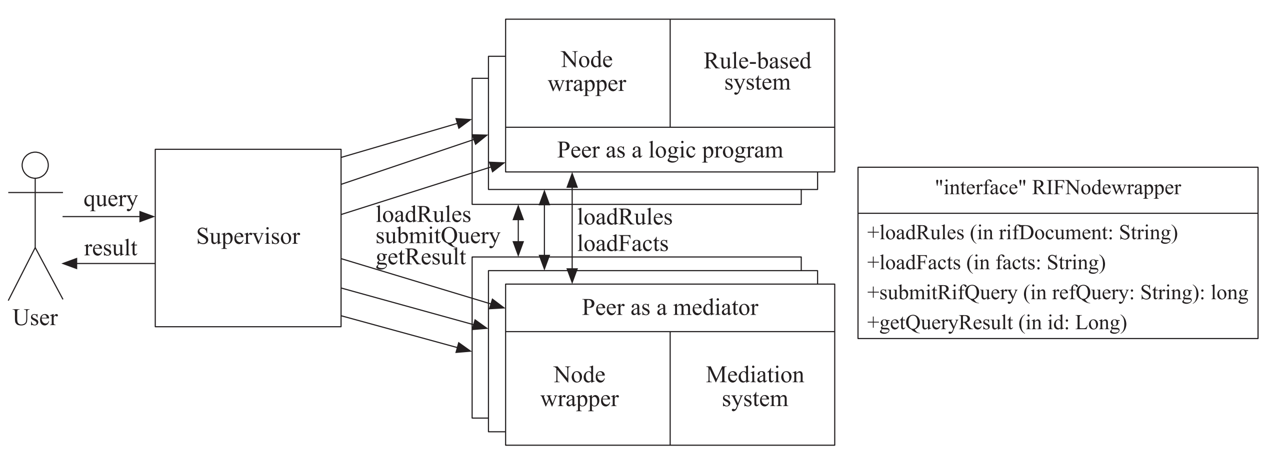

Figure 3 Peer-to-peer multidialect architecture

that $p(v_1, \ldots, v_n) = true$ where $v_1, \ldots, v_n$ are the values of variables $X = x_1, \ldots, x_n$; and

(2) *delegation rule* like $q@n(X)$: -$p@m(X)$. The rule has to be transferred to the peer m where it becomes a fact delegation rule.

The procedure of *normalization* (Algorithm 1) of a program pr on a peer n is as follows. For each occurrence of a remote term in a rule r, a fact delegation rule is added to pr aimed at importing facts from another peer to the peer n or at exporting facts from n to another peer. The rule includes a new local predicate created for storing imported or exported facts. Each occurrence of a remote term in a rule r is replaced by an occurrence of the respective local predicate. In that way, the rule r is transformed into a local rule.

Algorithm 1. $Normalization(pr, n)$
Input: pr is a program — a set of rules; n is a peer name
Output: normalized program pr
  for each rule $r \in pr$ like $head(r)$: -$body(r)$. **do**
   if the $head(r)$ of r is a remote term $p@m(X)$ **then**
    // remove r from pr
    $pr \leftarrow pr \backslash \{r\}$
    // add new fact delegation rule,
    // here, p_m is a new local predicate
    $pr \leftarrow pr \cup \{p@m(X)$: - $p_m(X).\}$
    // add new local rule
    $pr \leftarrow pr \cup \{p\_m(X)$: - $body(r)\}$
   elseif the $head(r)$ is a local predicate **and** the $body(r)$ contains an occurrence of a remote term $q@l(Y)$
   then
    for each remote term $q@l(Y)$ contained in the $body(r)$ **do**
     // add new fact delegation rule
     // here, q_l is a new local predicate
     $pr \leftarrow pr \cup \{q\_l(Y)$: - $q@l(Y).\}$
     // replace the occurrence of the remote term in r
     // by a local term $q\_l(Y)$
     $r \leftarrow head(r)$: - $body(r)[q@l(Y) \rightarrow q\_l(Y)]$.
    end for
   endif
  end for
return pr

The algorithm of execution of a program on a node containing a peer n looks as follows (Algorithm 2). The program stored on the peer is normalized. Delegation rules produced during normalization are transferred to the respective peers. After that, the algorithm waits for the facts from all the peers to which the delegation rules were transferred. During waiting, some delegation rules can be obtained from other peers (no mutual recursion between programs sent to different peers is presumed). When all facts are got, they become a part of the peer. Then, local RIF-rules of the peer are transformed into the RS or MS rule language and executed in the RS or MS. The last step consists in transferring the required facts to other peers in accordance with the fact delegation rules.

Algorithm 2. $Execute(n)$
Input: n is a peer name;
  $prog(n)$ is a program stored at n at the moment (additional rules can be loaded into $prog(n)$ during execution of $Execute$)
Output: $result$ — a set of tuples which are the result of the executing program
Local variables
  prn — a variable to store a normalized program
  pre — a variable to store local normalized rules
  $pre1$ — a variable to store a program transformed into the RS or MS rule language
  $prn \leftarrow Normalization(prog(n), n)$
  for each fact delegation rule $r \in prn$ like $q\_l(Y)$: -$q@l(Y)$
   // transfer rule $q\_l@n(Y)$: -$q(Y)$ to peer l
   $l.loadRule($"$q\_l@n(Y)$: -$q(Y)$"$)$
   $pre \leftarrow pre \backslash \{r\}$
  end for
  wait until for all fact delegation rules, $r \in prn$ like $q\_l(Y)$: -$q@l(Y)$
   a respective peer l has transferred facts like $q(v_1, \ldots v_n)$ to peer n as facts like $q\_l(v_1, \ldots v_n)$
   for each fact delegation rule, $r \in prn$ like $q\_l(Y)$: -$q@l(Y)$
   $load$ facts from predicate q_l into the peer n
  end for
   // put all normalized local rules into pre
  $pre \leftarrow \{r | r \in prn \vee r$ is local$\}$
  $transform$ the program pre into the resource rule language program $pre1$
  $result \leftarrow execute$ the program $pre1$ on the resource (mediator or RS) connected to n
  for each fact delegation rule $r \in prog(n)$ like $p@m(X)$: - $q(X)$
   // transfer facts from local predicate q into remote
   // predicate p at the peer m
   $m.loadFacts(\{p(x) | x \in q(x)\})$
  end for
return $result$

A rule delegation mechanism described in this section resembles one proposed in WebdamLog [30].

4 Lessons Learned During the Use Case Implementation

To provide a proof of the multidialect infrastructure concept, a use case in the financial domain has been implemented. The problem to be solved in the use case is called *investment portfolio diversification problem*. The detailed description of the use case is included in the appendix. In this section, the important lessons learned during the implementation of the use case are summarized.

During the implementation of the *investment portfolio* use case in the multidialect infrastructure, the conceptual specifications were defined: a conceptual schema of the financial domain (using OWL 2) and conceptual specifications of the problem (using RIF dialects BLD and CASPD). Conceptual specification was defined over conceptual schema: names of the entities (classes and attributes) of conceptual schema and only they were used in the rules of the conceptual specification as extensional predicates. Conceptual specifications were implemented using a multidialect infrastructure which includes a mediation system integrating the financial data and an ASP [26] RS.

Application specification in the multidialect infrastructure is started with the informal definition of a problem (see appendix, section A1). Then, a conceptual schema of a subject domain is defined using OWL 2 (see section A2). Due to the fact that OWL 2 does not provide facilities to specify functions, it is necessary to model the functions related to the collections of facts of the domain as OWL classes (e. g., *correlation*):

```
Class(correlation)
  DataProperty(series1  domain(correlation)
    range(DatedValue))
  DataProperty(series2  domain(correlation)
    range(DatedValue))
  DataProperty(corr  domain(correlation)
    range(xsd:double))
```

Input and output parameters of the functions are modeled as data or object properties of the respective classes (for instance, series1, series2, and corr). The types of the parameters are modeled as range of the respective properties.

Lesson 1. Functions related to the collections of facts in conceptual schema of the domain have to be modeled as OWL classes.

The classes of the conceptual schema are used as predicates in rules which constitute documents of the conceptual specification of the problem. Logic rules over taxonomic structures of the schema may require existential quantifier in their heads (see section A2):

```
Exists ?ts( And(?ts#gex:tickers
?ts[symbol->?symbol]) ):-
  And(?t#srt:stockRates  ?t[ticker->?symbol]
      ?t[isInTop500->?inTop500]
      External(pred:boolean-equal(?inTop500
      true))
  )
```

The rule states that for each object ?t of stockRates class there exists an object ?ts of a class tickers. A predicate e#s denotes that the element e belongs to the set s.

Built-in predicates (like Boolean equality pred:boolean-equal) and functions are used as *external terms* of RIF. The symbol External indicates that an atomic formula or a function term is defined externally (neither in the current RIF-document nor in the remote RIF-document). Built-in predicates are defined in a special part of the RIF standard.

Lesson 2. Conceptual specification of a problem using RIF logic rules over taxonomic structures may require existential quantifier in the rule heads. The RIF-BLD dialect intended for serving as the basic logic dialect does not provide such facility. Respective extension of RIF-BLD is suggested.

Lesson 3. Conceptual specification of a problem often requires usage of built-in predicates and functions as external terms.

During the application of the RIF-CASPD dialect [31] for conceptual specification, it was found out that the ASP programs require a specific facility — so-called *weak constraints* [10]. Such constraints should be satisfied if it is possible, but their violation does not invalidate the models. For instance, a weak constraint used in the specification of the portfolio problem looks as follows (see section A2):

```
Forall ?X( :~ prt:nonPortfolio(?X))
```

The rule states that only such stable models that minimize the truth set of the predicate nonPortfolio are considered as a result of the logic program (that includes the mentioned rule). The difference of the rule with a strong constraint like

```
Forall ?X( :- prt:nonPortfolio(?X))
```

consists in that in a program with the strong constraint the predicate nonPortfolio(?X) has to be turned to *false* for all ?X. Weak constraint requires the set of all possible values of ?X that turns nonPortfolio(?X) into *true* to be of minimum size.

Lesson 4. Conceptual specification using ASP-like dialects requires a specific kind of logic rules — *weak constraint*. The RIF-CASPD dialect intended for serving as the dialect supporting ASP does not provide such facility. Respective extension of RIF-CASPD is suggested.

To implement the conceptual schema, a mediator intended to provide the collection of facts implied by the schema was created. The mediator integrates the respective financial data. The schema of the mediator (see subsection A3.2) was created by mapping the conceptual schema into the CIM (the SYNTHESIS language [32]). The classes of the conceptual schema were mapped into types and/or classes of the mediator schema, including their attributes. It appeared that some of the attributes (e. g., *isInTop500*) of the conceptual schema taking into account implementation reasons better to interpret by the methods of the mediator schema. It appeared also that the definition of *isInTop500* as a method on the mediator layer made it simpler to establish semantic mappings (GLAV views) between the mediator and the resources.

Another peculiarity of the conceptual schema implementation worth mentioning here is that the classes of the conceptual schema (namely, correlation, mentioned earlier in this section) can be mapped into a function of the mediator schema:

```
{ correlation; in: function;
    params: {+s1/{set; type_of_element:
    DatedValue;},
          +s2/{set; type_of_element:
          DatedValue;}, -corr/real };
};
```

Lesson 5. The attributes of the conceptual schema can be mapped into attributes or functions of the schemas of the mediators implementing the conceptual specifications. The classes of the conceptual schema can be mapped into functions of the schemas of the mediators.

The mapping of conceptual schema into the mediator schema is formally specified using schema mapping rules constituting a separate RIF-document as shown in Fig. 3, middle part. These rules are similar to GAV views which represent classes of the conceptual schema as views over the mediator schema. This allows one to consider the schema mapping rules just as a consistent part of the logic program that have to be executed in a peer (e. g., in a mediator).

The document containing schema mapping rules refers to the conceptual schema and to the mediator schema. Both schemas are identified by URIs:

```
Import(<http://synthesis.ipi.ac.ru/
        optimalSecurityPortfolio>
        <http://www.w3.org/ns/entailment/
         OWL-Direct>)
Prefix(srt <http://synthesis.ipi.ac.ru/
            optimalSecurityPortfolio#>)
Prefix(fs <http://synthesis.ipi.ac.ru/
            FinanceServices#>)
```

Entities of the mediator schema are referred in the schema mapping rules using external terms of the RIF, for instance, External(?t#fs:stockRates) (see subsection A3.1). This means that the mediator schema is neither OWL-ontology or RDF-document nor RIF-document and possesses its own semantics.

Mapping of the *correlation* class of the conceptual schema into the correlation function of the mediator schema is defined using a rule with a frame predicate in the head and functional predicate in the body (see subsection A3.1):

```
And(?c#correlation
   ?c[?series1->series1 ?series2->series2
   ?corr->corr]) :-
External(pred:numeric-equal(
   fs:correlation(?series1 ?series2) ?corr))
```

Lesson 6. Entities of the peer (mediator) schemas have to be referred in schema mapping rules using external terms of the RIF.

Lesson 7. Logic programs defined in the native languages of RS or MS look more elegant and concise than the respective programs defined using RIF-dialects. The present authors consider this as a price paid for providing the interoperability of multidialect programs with different semantics. More elegant notation to apply RIF for conceptual specifications should be investigated.

It appeared that the mediation system used does not support all the features of the RIF-BLD (terms with named arguments, conjunctions of atoms in the head of a rule, and some others). DLV system also does not support all the features of the CASPD dialect (terms with named arguments, conjunctions of atoms in the head of a rule, frame terms, subclass terms, membership terms, and some others). These systems can be the consumers only for the subsets of the respective dialects.

Lesson 8. According to RIF, one could have expected that the rule-based programs represented in a specific RIF-dialect should be independent on the RS (or MS) consumer languages at least in the class of RSs (or MSs) conforming to this dialect. For instance, any consumer in the class of the ASP systems should be considered as conformant to the CASPD dialect. Noncompliance with such condition makes the conceptual modeling in the RIF dialects difficult: it is required to choose a specific RS (or MS) in advance and to adjust the dialect to be used appropriately (as it was shown in Lessons 2 and 4 or in the observation above showing that it is unlikely that an RS or MS would support all the features of some standard dialect). Practically, it means that at least the RIF-dialect standards should include instructions for the dialect adjustment making possible to produce appropriate subdialects or superdialects at the early stage of the conceptual design.

5 Related Work

Among various future research issues predicted in [33] to obtain reasonable solutions in the next decades, the conceptual modeling has been positioned as the basis for interoperability and for shareable information services. The solution proposed in the present paper is tightly related to the issues listed in [33].

Ambiguity and incompleteness of structured and object-oriented notations for conceptual modeling in graphical form causes the unceasing process of generating various contributions suggesting formal meaning to the diverse graphical constructions, enhancing their expressiveness, e. g., as in several recent papers [34–36]. The present paper suggests more radical formalization decisions.

Diverse ontology centered conceptual modeling tools have been developed. Just a couple of them to mention.

KAON2[1] is an infrastructure for managing OWL-DL, SWRL (Semantic Web Rule Language), and F-Logic ontologies. Its main feature is its own inference engine. KAON2 supports the SHIQ(D) subset of OWL-DL, DL-safe subset of the SWRL, making reasoning decidable [32], as well as the function-free subset of F-Logic. This approach is related to the multidialect architecture considered here due to a provision of using several different languages treated as ontological notations. ICOM 3.0 tool [37] for conceptual modeling is aimed at the support of the design of multiple extended ER or UML diagrams with inter- and intramodel constraints. Reasoning support via DL to help validate the models is provided. The system is helpful for validating integrated models (support for the integration of models organized in several ontologies is provided). But it is difficult to improve such notations as ER or UML completely; they allow only to specify the semantics informally, in terms of natural language; or the semantics are hard-coded using the tools supporting the notation.

Rule-based modeling is applied in various DIS. For example, in biology, rule-based modeling has increasingly attracted attention due to enabling a concise and compact description of biochemical systems. Proteins, individual cells, and cell populations denote different levels of an organizational hierarchy, each of them with its own dynamics. Multilevel modeling is concerned with describing a system at these different levels and relating their dynamics. The approach proposed facilitates developing and maintaining multilevel models that, for instance, interrelate intracellular and intercellular dynamics [38].

The idea used in the infrastructure prototype for ontology-based access (using OWL 2) to the mediators integrating heterogeneous databases resembles the idea of the MASTRO-I system [39] in which QuOnto engine supporting inference in the DL-Lite ontologies and conjunctive query answering is positioned above the federated database. The latter is the result of relational databases integration.

The rule exchange using RIF-PRD is discussed in [40, 41]. In such case, the production rule systems share the same operational semantics opposed to the present approach studying the problem of rule exchange between systems with different semantics.

Several papers on the use of database query languages for specifying declarative distributed programs and managing data in distributed environment have been published [30, 42–44]. In contrast to multidialect approach, a single declarative language is used in each of the proposed systems. Usually, it is a conventional Datalog extended with the notion of localization and possibly other nondatalog constructs [42]. In the multidialect approach, localization is specified with the RIF remote and imported terms.

Conceptual notion of *delegation* applied in the present approach is similar to the notion of delegation in Webdamlog defined as the "possibility of installing a rule at another peer. In its simplest form, delegation is essentially a remote materialized view. In its general form, it allows peers to exchange rules, i. e., knowledge beyond simple facts, and thereby provides the means for a peer to delegate work to other peers" [16]. Actually, the current implementation supports delegation as a remote materialized view. Extending the approach for delegation of knowledge is a future work. The idea of program normalization is similar to the rule localization rewriting the step described in [42].

6 Future Plans

A number of related issues remain open. Some of them that are planned to be investigated are listed below.

Conceptual specification of processes. How to define compositions of algorithmic modules in a process structure (like Petri nets, UML activity diagrams, the Business Process Modeling) can be found in [45] where process modeling semantics, service discovery, mediation, and composition are considered. Such notations as WSDL-S, SAWSDL, OWL-S, and WSMO (Web Service Modeling Ontology) can be applied for that. For example, OWL-S is a semantic web service description language that borrows ontological operators from OWL and provides a service upper ontology to describe web services and service processes in a standard manner. The WSMO and the combination of UML

[1] http://kaon2.semanticweb.org/.

and Object Constraints Language (OCL) are viewed as promising alternatives to OWL-S. Notation WSMO, like OWL, is also based on the first-order logic but employs an object-oriented style of modeling based on frames. Unlike OWL-S, WSMO includes service and ontology specification formalisms with well-defined formal semantics, whereas OWL-S relies on OWL for ontology specifications and on languages such as SWRL and KIF (Knowledge Interchange Format) for formal semantics. The present authors plan to investigate an approach for reusable process specifications in conceptual terms compliant with the infrastructure proposed.

Multiple layers of specifications. Metamodeling as a kind of modeling across multiple layers uses an "instance-of" relationships between layers. UML/MOF/MDA by OMG is an example of a strict metamodeling hierarchy in which metamodels introduce higher levels of abstractions for specification and language concepts. Similar hierarchy can be introduced also in ontologies (e. g., upper ontologies above the domain ontologies in one and the same language). The models can be used in a descriptive or prescriptive manner [46]. In this paper, the metamodeling issues are not considered though in future plans, the authors are closer to the Ontology-based software Development Environment (ODE) approach of [47].

Hybrid specifications. Conceptual modeling should likely explicitly involve hybrid aspects of information (i. e., the coexistence of formal reasoning and "informal" human interactions). Correspondingly, "hybrid ontologies" will need to be modeled and deployed within their supporting social implementation environments [48].

Modeling hypotheses in the DIS experiments. Evaluating the conceptual model and infrastructure proposed in the paper, the following far-reaching DIS related ideas are planned to be taken into account [49]. As the DIS research evolves, references to previous results and specifications are needed as a source of provenance data. *In-silico*[1] experiments must be supported by a hypotheses model that describes the elements involved in a scientific exploration and supports hypotheses assessment. Adopting a conceptual specification perspective to represent hypotheses would allow high-level references to experiments and provides support for hypotheses evolution. Such hypotheses model would support scientists in describing, running simulations and interpreting their results [49]. In conjunction with an axiom set specified as rules that model known facts over the same universe and experimental data, the knowledge base may contradict or validate some of the sentences in the hypotheses. In the case of contradictions, the rules that caused the problems must be identified and eliminated from the theory formed by the hypotheses [50]. The conceptual specification model proposed in the present paper looks suitable to such intention providing for a possibility of hypotheses representation as a set of first-order predicate calculus sentences applying the multiplicity of the dialects required.

7 Concluding Remarks

The approach presented is an attempt of introducing the multidialect interoperable conceptual programming over various semantically different rule-based programming systems relying on the logic program transformation technique recommended by W3C RIF. It is also shown how to coherently combine such approach with the heterogeneous data bases integration applying the semantic mediation. Thus, the data independence of conceptual specifications is provided. The results obtained so far are quite encouraging for future work aimed at reaching the conceptual specifications reusability in various applications over different sets of data, as well as for sharing and accumulation of reproducible data analysis and problem solving methods and experience in various DID. At the same time, the RIF-dialect standards are suggested to be made more flexible including into them the instructions for the dialect adjustment. This makes possible to produce appropriate subdialects or superdialects at the early stage of the conceptual design (see section 4, Lesson 8).

APPENDIX A

The Use-case for the Multidialect Infrastructure

A1 Investment Portfolio Diversification Problem

The capabilities of the multidialect architecture are illustrated with the solution of the *investment portfolio diversification problem* [51]. The portfolio is a collection of securities (such as equities or bonds) of companies, and its size is the number of securities in the portfolio. The task is to build a diversified portfolio of maximum size. Diversification means that the prices of the securities in portfolio should be almost independent of each other. If the price of one security falls, it will not significantly affect the prices of other. Thus, the risk of a portfolio sharp decrease is significantly reduced.

[1] *In silico* is an expression used to mean "performed on computer or via computer simulation." The phrase was coined in 1989 as an analogy to the Latin phrases *in vivo*, *in vitro*, and *in situ*, which are commonly used in biology and refer to experiments done in living organisms, outside of living organisms, and where they are found in nature, respectively (refer: Differences between *in vitro*. *in vivo*, and *in silico* studies. 2013. Available at: http://mpkb.org/home/patients/assesing_literature/in_vitro_studies (accessed December 2, 2013)).

The input data for the problem is a set of securities and respective time series of indicators of the security price for each security. Time series for each security is a set of pairs (d, v) where d is a date and v is an indicator of the security price (for instance, closing price).

For a pair of time series $X = \{(d_1, x_1), \ldots, (d_n, x_n)\}$ and $Y = \{(d_1, y_1), \ldots, (d_n, y_n)\}$, their *correlation* (a measure of similarity) can be calculated. The Pearson correlation[1] r_{XY} was used between X and Y

$$r_{XY} = \frac{\sum_{i=1}^{n} (x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n} (x_i - \overline{x})^2 \sum_{i=1}^{n} (y_i - \overline{y})^2}}$$

where $\overline{x}$ and $\overline{y}$ are the means of $x_1, \ldots, x_n$ and $y_1, \ldots, y_n$.

For the diversified portfolio, the securities which time series are noncorrelated should be used. Noncorrelation of the time series means that their correlation is less than some predetermined price correlation value. This predetermined value is a parameter of the portfolio problem. The lower is the correlation — the lower is the risk measure of a sharp decrease of the portfolio value.

So, the predetermined correlation serves as the maximum risk measure. In practice, specified correlation may differ for various types of securities.

The output data for the problem is a subset of securities of the maximum size, for which the pairwise correlation will be less than the predetermined one.

The problem is divided into the following tasks:

(1) computation of the security pairwise correlations (for specified dates); and

(2) calculation of the maximum satisfying subset of securities.

To solve the first task, the financial services *Google Finance*[2] and *Yahoo! Finance*[3] are considered, both of which provide current and historical information about various securities: stock prices, currencies, bonds, stock indexes, etc. In particular, the information provided includes various indicators of the security price for all trading days of last decades. The mediation system is used to solve the problem of resource integration [5].

The second task can be formulated as follows. Let G be a graph where the vertices are the securities, and an edge between two securities exists if absolute value of their correlation is less than a specified number. So, any two securities connected by an edge are considered as noncorrelated. In that case, the problem of finding the portfolio of the maximum size is exactly the problem of finding a maximum clique in an undirected graph. Indeed, a clique in a graph is a subset of its vertices such that every two vertices in the subset are connected by an edge. All the vertices (securities) in a clique are connected by edges (noncorrelated) and any clique is a candidate for a portfolio. A maximal clique is a maximal portfolio.

Finding a maximum clique in an undirected graph is a well-known NP-complete problem[4].

DLV [10] is one of the ASP logic programming systems well-suited for solving such problems [31]. Answer set programming is based on the stable model (answer set) semantics of logic programming [26]. In ASP, the search problems are reduced to computing stable models, and *answer set solvers* — programs for generating stable models — are used to perform search. With regard to solving the maximum clique problem, ASP methods are applicable nowadays for graphs with thousands of vertices.

A2 Conceptual Specification of the Application Domain and the Problem

Conceptual schema (ontology) of the application domain of historical prices of securities is written in the simplified[5] OWL functional syntax [28]:

```
Ontology(<http://synthesis.ipi.ac.ru/optimalSecurityPortfolio>
    Class(stockRates)
        DataProperty(ticker  domain(stockRates) range(xsd:string))
        DataExactCardinality(1  ticker  stockRates)
        ObjectProperty(rates  domain(stockRates)  range(DatedValue))
        DataProperty(isInTop500  domain(stockRates)  range(xsd:boolean))
        DataExactCardinality(1  isInTop500  stockRates)

    Class(DatedValue)
        DataProperty(value  domain(DatedValue)  range(value  xsd:double))
        DataExactCardinality(1  value  DatedValue)
        DataProperty(date  domain(DatedValue)  range(xsd:date))
        DataExactCardinality(1  date  DatedValue)
```

[1]Refer: De Smith, M. J. 2013. *Statistical analysis handbook*. Avalable at: http://www.statsref.com/HTML/pearson_product_moment_correla.html (accessed December 2, 2013).

[2]https://www.google.com/finance

[3]http://finance.yahoo.com/

[4]Refer: Cormen, T. H., T. Leisercon, R. L. Rivest, and C. Stein. 2009. NP-complete problems. *Introduction to algorithms*. 3rd ed. Cambridge: The MIT Press. 1292 p.

[5]To save space, "Declaration" keyword is omitted; property, domain and range declarations are combined.

```
    Class(correlation)
        DataProperty(series1  domain(correlation)  range(DatedValue))
        DataProperty(series2  domain(correlation)  range(DatedValue))
        DataProperty(corr  domain(correlation)  range(xsd:double))
        DataExactCardinality(1  corr  correlation)
)
```

The stockRates class is used to denote securities, which are characterized by:

- identifier (attribute ticker);
- time series of historical prices (attribute rates);
- belonging to S&P 500 list of companies (attribute isInTop500).

The S&P 500 is a stock market index maintained by the Standard & Poor's, comprising 500 large-cap American companies.

The *correlation* class is the correlation of time series pairs. For each class instance, the value of *corr* attribute equals to the correlation of its attributes series1 and series2 (time series).

The conceptual specification of the problem includes two RIF documents that correspond to the specified tasks. The first of the documents (name *gex* is the local prefix of the document) contains a program that calculates the correlation graph of securities (predicate *noncorrelated*) based on the prices in a given period of time. For each pair of securities, a correlation of their time series (defined in section A1) of historical prices is calculated that is why the graph is called "correlation graph." But as the candidates for the elements of the portfolio, only noncorrelated securities (which correlation is lower than a predefined value) are considered. That is why the predicate is called "*noncorrelated*." The document is defined in the RIF-BLD[1] dialect [20]:

```
Document( Dialect(RIF-BLD)
    Import(<http://synthesis.ipi.ac.ru/optimalSecurityPortfolio>
            <http://www.w3.org/ns/entailment/OWL-Direct>)
    Prefix(srt  <http://synthesis.ipi.ac.ru/optimalSecurityPortfolio#>)
    Prefix(gex  <http://synthesis.ipi.ac.ru/graphExtraction#>)
    Group(
    Forall ?t ?symbol ?ticker ?inTop500(
        Exists ?ts( And(?ts#gex:tickers  ?ts[symbol -> ?symbol]) ):-
            And(?t#srt:stockRates  ?t[ticker->?symbol]
                ?t[isInTop500->?inTop500]
                External(pred:boolean-equal(?inTop500  true))
            )
    )

        Forall ?m ?n ?c ?ticker1 ?ticker2 ?start ?end ?rates1 ?rates2
            ?top1 ?top2 ?dv1 ?dv2 ?date1 ?date2 ?series1 ?series2 ?corr (
            Exists ?e (
                And(?e#gex:noncorrelated ?e[start->?ticker1 end->?ticker2])):-
                ?m#srt:stockRates  ?n#srt:stockRates
                ?m[ticker->?ticker1  rates->?rates1  isInTop500->?top1]
                ?n[ticker->?ticker2  rates->?rates2  isInTop500->?top2]
                ?dv1#?rates1  ?dv1[date -> ?date1]
                ?dv2#?rates2  ?dv2[date -> ?date2]
                External(pred:date-greater-than-or-equal(?date1  2012-01-01))
                External(pred:date-less-than-or-equal(?date1  2012-12-31))
                External(pred:date-greater-than-or-equal(?date2  2012-01-01))
                External(pred:date-less-than-or-equal(?date2  2012-12-31))
                ?c#srt:correlation ?c[corr->?corr
                        series1->?rates1 series2->?rates2]
                External(pred:numeric-greater-than(?corr  -0.25)
                External(pred:numeric-less-than(?corr  0.25))
                External(pred:boolean-equal(?top1  true))
                External(pred:boolean-equal(?top1  true))
                External(pred:string-less-then(?ticker1  ?ticker2))
    ))
  )
)
```

[1]The RIF-BLD dialect is extended with the possibility to use the existential quantifier in the head of a rule.

The first rule of the document defines the predicate-collection tickers, in which the attribute symbol element runs through the list of security identifiers. Here, e#p is a predicate denoting membership of element e in collection p; predicate e[a->v]}means that the value of attribute a of object e is v. Securities taken only from S&P 500 list are put into tickers collection (attribute isInTop500 equals to true). This means that only large companies from S&P 500 are considered as candidates for portfolio elements.

The second rule defines predicate noncorrelated, which is a noncorrelation relation between securities. Object ?e belongs to a relation noncorrelated if

– both securities belong to S&P top 500 list;
– the absolute value of correlation between the time series if historical prices of securities for given attribute values e.start and e.end is less than 0.25. Here, 0.25 is the value chosen for the use-case as the predetermined maximal correlation value (see section A1); and
– identifiers of securities id1 and id2 are in lexicographic order (id1 < id2). This is required to prevent equivalent pairs (id1, id2) and (id2, id1) to be both presented in a relation corresponding to the predicate noncorrelated — only one pair is sufficient.

Dates in time series range from January 1, 2012 to December 31, 2012.

The second document (prt) contains a program that computes the maximum clique in a graph of correlations. The document is defined in the RIF-CASPD[1] dialect [25]:

```
Document( Dialect(RIF-CASPD)
    Import(<http://synthesis.ipi.ac.ru/optimalSecurutyPortfolio>
        <http://www.w3.org/ns/entailment/OWL-Direct>)
    Module(<http://synthesis.ipi.ac.ru/graphExtraction#>)
   Prefix(prt  <http://synthesis.ipi.ac.ru/portfolio#>)
    Prefix(gex  <http://synthesis.ipi.ac.ru/graphExtraction#>)
    Group (
        Forall ?X(Or(prt:portfolio(?X) prt:nonPortfolio(?)) :- tickers@gex(?X))
        Forall ?X ?Y( :- And(prt:portfolio(?X)  prt:nonPortfolio(?X)))
        Forall ?X ?Y( :- And(prt:portfolio(?X)  prt:portfolio(?Y)
                                        (Naf noncorrelated@gex(?X ?Y))) )
        Forall ?X( :~ prt:nonPortfolio(?X))
        prt:portfolio(?X).
) )
```

The program defines a predicate portfolio, whose values are the security identifiers in the portfolio, and predicate nonPortfolio, whose values include all other securities under consideration.

The first rule states that the only securities considered are the securities which turn to truth the predicate tickers in document gex.

The second rule states that a security cannot simultaneously belong and not belong to the portfolio. So, the predicates portfolio and nonPortfolio divide the set of all securities into two nonintersecting sets.

The third rule claims that any two securities in portfolio are noncorrelated (according to noncorrelated predicate in the document gex). Naf here means a sort of negation — *negation as failure* [52]. This exactly means that securities belonging to portfolio form a clique in a graph of noncorrelated securities.

The fourth rule is a weak constraint, which minimizes the number of securities not belonging to the portfolio. So, the portfolio itself is maximized.

Obviously, these four rules declaratively specify the problem of finding the maximum cliques in a graph of noncorrelated securities.

Fifth rule is just a predicate portfolio(?X) with a free variable ?X. This rule forms a result of the program, that is, a collection of sets of identifiers of securities. Each set corresponds to a particular solution of the problem (a stable ASP-model for the program in the document prt) and forms a maximal portfolio.

A3 Peers of the Use-Case Infrastructure

For implementation of the conceptual specification of the problem of the investment portfolio diversification, the two peers should be produced. One of them is the mediator used for integration of data produced by the *Google Finance* and the *Yahoo! Finance* services.

Another one is a program for the rule-based programming system DLV [10]. The mediator produces a virtually integrated collection of facts, and DLV is a system for executing ASP-programs. Initially, the nodes do not contain any logic programs.

A3.1 The mediator schema

The mediator schema implements the conceptual specification of the application domain by expressing its semantics. Schema is written in the SYNTHESIS [4] — the canonical information model used for the mediator specifications:

[1]The CASPD dialect is extended with the operation :∼ for a weak constraint. Such constraints should be satisfied if possible, but their violation on some tuples of variables values does not invalidate the models [13].

```
{ FinanceServices; in: module;
  type:
  { DatedValue; in: type;
     date: time;
     value: real;
  },
  { StockRates; in: type;
     ticker: string;
     rates: {set; type_of_element: DatedValue;};
     isInTop500:
     { in: function;
         params: {-returns/boolean};
     };
  };

  class_specification:
  { stockRates; in: class;
     instance_section: StockRates;
  };
  function:
  { correlation; in: function;
    params: {+s1/{set;
    type_of_element: DatedValue;},
    +s2/{set;
    type_of_element: DatedValue;}, -corr/real };
  };
}
```

The schema includes types DatedValue and StockRates, a class stockRates, and a function correlation. Semantics of class stockRates, type DatedValue, and function *correlation* correspond to the semantics of classes stockRates, DatedValue, and correlation belonging to optimalSecurityPortfolio ontology (see section A2).

The correspondence between the mediator schema and the conceptual schema is described by the schema mapping rules (see Fig. 3, middle part) constituting a separate RIF-document:

```
Document( Dialect(RIF-BLD)
    Import(<http://synthesis.ipi.ac.ru/optimalSecurityPortfolio>
            <http://www.w3.org/ns/entailment/OWL-Direct>)
    Prefix(srt  <http://synthesis.ipi.ac.ru/optimalSecurityPortfolio#>)
    Prefix(fs  <http://synthesis.ipi.ac.ru/FinanceServices#>)
    Group(
        And(?t#srt:stockRates
            ?t[?ticker-> srt:ticker ?top-> srt:isInTop500
                    ?rates-> srt:rates]) :-
        And(External(?t#fs:stockRates)
            External(?t[?ticker->fs:ticker ?top->fs:isInTop500
                ?rates->fs:rates]) )

        And(?dv#srt:DateValue ?dv[?v->srt:value ?d->srt:date]) :-
        And(External(?dv#fs:DateValue)
            External(?dv[?v->fs:value ?d->fs:date])     )

        And(?c#correlation
            ?c[?series1->series1 ?series2->series2 ?corr->corr]) :-
        External(pred:numeric-equal(
            fs:correlation(?series1 ?series2) ?corr))
    )
)
```

The document consists of three rules. The first one defines a correspondence between class stockRates of the optimalSecurityPortfolio ontology (denoted as srt:stockRates) and class stockRates of the FinanceServices mediator module (denoted as fs:stockRates) as well as correspondences between their attributes. The second and the third rules define correspondences between ontology class srt:DatedValue and mediator type fs:DatedValue and between ontology class srt:correlation and mediator function fs:correlation.

Entities of the mediator schema and conceptual schema are in one-to-one correspondence, the names and the semantics of the relevant entities are the same. Also, the mediator is a conformant RIF-BLD consumer. Thus, the mediator FinanceServices is relevant to the RIF-document gex of the conceptual schema.

A3.2 Resource integration by the mediator

The mediator integrates two resources: *Google Finance* and the *Yahoo! Finance services*. For both resources, specific wrappers (the *Google Finance* wrapper and the *Yahoo! Finance* wrapper) were implemented to embed the resources in the mediation system [53].

Schema of the *Google Finance* service presented for uniformity in the canonical information model (the SYNTHESIS language) looks as follows:

```
{ GoogleFinance; in: module;
   type:
   { GoogleQuote; in: type;
      symbol: string;
      Date: time;
      Open: real;
      High: real;
      Low: real;
      Close: real;
      Volume: integer;
   };

   function:
   { groupByTicker; in: function;
      params: {+symbol/string,
      +date/time, +close/real,
       -rates/{set; type_of_element:
        DatedValue;}}
   },
   { isInTop500; in: function;
      params: {+ticker/string,
       -returns/boolean};
   };

   class_specification:
   { historicalData; in: class;
      instance_section: GoogleQuote;
   };
}
```

The schema includes a type GoogleQuote, which includes attributes that match the identifier of the company (symbol), date (Date), and different indicators of the security price. Daily closing price of the shares (Close) is used to form the portfolio.

The function top500 reflects the S&P 500 index: if the value of the ticker parameter belongs to S&P 500 list, then the function returns true and false otherwise.

The function groupByTicker is aimed to group flat instances of the GoogleQuote type by ticker attribute to produce a time series of an indicator of the security price for all securities identified by ticker.

Functions of the schema are implemented by the *Google Finance* wrapper.

Schema of the *Yahoo! Finance* service presented in the canonical model of subject mediators looks as follows:

```
{ YahooFinance; in: module;
   type:
   { YahooQuote; in: type;
      symbol: string;
      Date: time;
      Open: real;
      High: real;
      Low: real;
      Close: real;
      Volume: integer;
      Adj_Close: real;
   };

   function:
   { groupByTicker; in: function;
      params: {+symbol/string,
      +date/time, +close/real, -rates/{set;
       type_of_element: DatedValue;}};
   },
   { isInTop500; in: function;
      params: {+ticker/string,
       -returns/boolean};
   };

   class_specification:
   { historicalData; in: class;
      instance_section: YahooQuote;
   };
}
```

The schema includes a type YahooQuote similar to GoogleQuote and functions groupByTicker and isInTop500 similar to respective functions of *GoogleFinance* schema.

Integration of the resources in the mediator has been provided using GLAV-mappings. The GLAV-mapping is a combination of GAV-mappings and LAV-mappings. The GAV-mapping looks as follows:

```
financeData(x/[ticker, rates]) :-
 group_by({ticker}, [ticker, p1: partition],
  GoogleFinance.historicaldata(
   x/[ticker: symbol, date: Date,
    value: Close]))&
 group_by({ticker}, [ticker, p2: partition],
   YahooFinance.historicaldata(
    x/[ticker: symbol, date: Date,
     value: Close])&
  is_equal(rates, union(p1, p2))
 ).
```

An intermediate predicate financeData (used to bind GAV and LAV-mappings) is expressed as a view over resource classes GoogleFinance.historicaldata and YahooFinance.historicaldata. Flat data structures of the resource classes are turned into hierarchical data structures of the mediator class using group_by operation. Curly braces in group_by operation enclose the grouping attribute (ticker). Square brackets in group_by operation enclose attributes of objects in a collection formed by group_by operation [4]. Partition is a default attribute storing grouped partitions. Here, resource attributes Date and Close are grouped in the partition attribute. Partitions from Google and Yahoo resource classes are merged into rates attribute of stockRates class (union operation). Attributes of the resource classes are renamed properly in order to meet the structure of the mediator class (for instance, Close attribute is renamed into value).

The LAV mappings look as follows:

```
financeData(x/[ticker, rates]) :-
   stockRates(x/[ticker, rates]).
GoogleFinance.isInTop500(x.ticker, b) ->
   isInTop500(x, b)
YahooFinance.isInTop500(y.ticker, b) ->
   isInTop500(y, b)
```

Here, financeData predicate is expressed as a simple view over mediator schema class stockRates, and isInTop500 resource functions are expressed through mediator isInTop500 function using *functional inverse rules* [19].

A3.3 DLV system

DLV system is a conformant RIF-CASPD consumer, and initially, the system does not contain any logic program or facts. Document prt does not contain any occurrence of an extensional predicate corresponding to an entity in the conceptual schema. Thus, the system is relevant to the RIF-document prt of the conceptual schema, since the relevance condition (see subsection 3.2) is reduced to the relation of dialects.

A4 Portfolio Problem Infrastructure

In the provided example, the infrastructure (Fig. 4) includes two nodes corresponding to the mediation system (called fsv) and to a rule-based programming system DLV (called dlv).

Prefixes of the documents of the conceptual schema in the rules are replaced with the actual node prefixes during rewriting the conceptual programs into the programs over the P2P network nodes. For instance, the rules of the prt document were rewritten as follows:

```
Document( Dialect(RIF-CASPD)
    Module(<http://synthesis.ipi.ac.ru/resources/FinanceServices#>)
    Prefix(dlv  <http://synthesis.ipi.ac.ru/resources/DLV#>)
    Prefix(fsv  <http://synthesis.ipi.ac.ru/resources/FinanceServices#>)
    Group (
        Forall ?X(Or(dlv:portfolio(?X) dlv:nonPortfolio(?)) :-
                            tickers@fsv(?X))
        Forall ?X ?Y( :- And(dlv:portfolio(?X)  dlv:nonPortfolio(?X)))
        Forall ?X ?Y( :- And(dlv:portfolio(?X)  dlv:portfolio(?Y)
                                    (Naf  noncorrelated@fsv(?X ?Y))) )
        Forall ?X( :~ dlv:nonPortfolio(?X))
        dlv:portfolio(?X).
    )
)
```

Rules from the document gex were rewritten similarly. Rewritten documents are sent by the supervisor to the proper nodes: gex to the mediation node, and prt to the dlv node.

After that, the node wrappers automatically execute distributed programs in accordance with the algorithm described in subsection 3.3. First, the program normalization is done. Thus, the nonlocal rule mentioned above

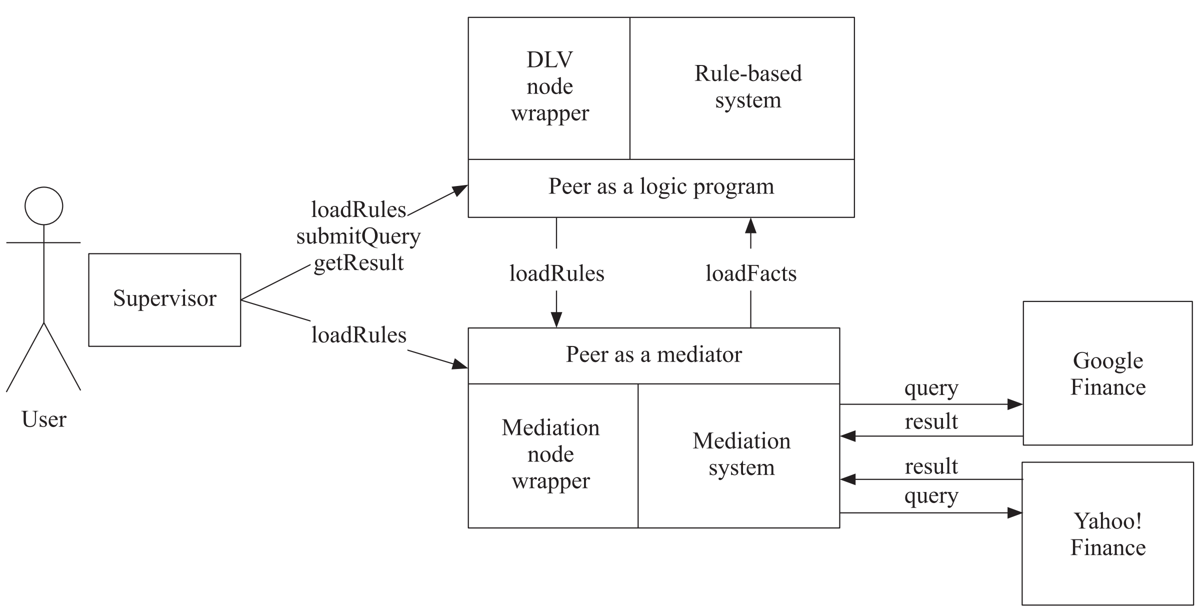

Figure 4 Portfolio problem infrastructure

```
Forall ?X ?Y( :- And(dlv:portfolio(?X)
dlv:portfolio(?Y)
   (Naf  noncorrelated@fsv(?X ?Y))) )
```

is transformed into a delegation rule

```
noncorrelated_fsv(?X ?Y) :-
  noncorrelated@fsv(?X ?Y)
```

and a local rule

```
Forall ?X ?Y( :- And(dlv:portfolio(?X)
  dlv:portfolio(?Y)
     (Naf  noncorrelated_fsv(?X ?Y))) )
```

Remaining rules from both documents are normalized in a similar way.

After that, the normalized program is executed. The fact delegation rule, which sends the noncorrelated predicate to the dlv node, is transmitted to the fsv node. The bodies of the rules of the program on the fsv node do not contain remote terms; so, the program on the node can be executed without waiting for the facts from the other nodes. In contrast, the dlv node has to wait for the arrival of the facts from the fsv node, which turns the tickers and the noncorrelated predicates to true. After receiving all necessary facts, the program computing portfolio is executed in the dlv node.

Before the execution, the normalized programs written in RIF dialects were automatically transformed into the logic languages supported on the nodes — the SYNTHESIS and the DLV, respectively. Transformations were implemented using a model transformation language and toolkit ATL (ATLAS Transformation Language) [54].

Toolkit ATL is a hybrid of declarative and imperative languages. An ATL transformation program is composed of rules that define how source model elements are matched and navigated to create and initialize the elements of the target models.

The result of the transformation of the prt document into the SYNTHESIS language looks as follows:

```
tickers(ts/[symbol]) :-
    stockRates(t/[symbol: ticker]) &
    isInTop500(t, inTop500) & inTop500 = true.

noncorrelated(e/[start: ticker1,
 end: ticker2]) :-
stockRates(m/[ticker1: ticker, rates1: rates,
    top1: isInTop500]) &
stockRates(n/[ticker2: ticker, rates2: rates,
    top2: isInTop500]) &
in_set(dv1, rates1) & in_set(dv2, rates2) &
dv1.date = date1 & dv2.date = date2 &
date1 >= '2012-01-01' & date1 <= '2012-12-31' &
date2>= '2012-01-01' & date2 <= '2012-12-31' &
correlation(c/[corr, rates1: series1,
rates2: series2 ]) &
corr >= -0.25 & corr <= 0.25 &
top1 = true & top2 = true &
ticker1 < ticker2.
```

Note that existential quantifier in heads of the rules is discarded during the mapping. According to the semantics of the SYNTHESIS logic rules, the variables in the head of a rule which are not presented in the body of the rule are bounded by existential quantifier by default. The RIF predicates of belonging of a variable to a class (like `?t\#stockRates`) are combined with frame predicates (like `?t[ticker->?symbol]`) and mapped into a predicate-collection [4] of the SYNTHESIS language (like stockRates(t/[symbol: ticker])).

The result of the transformation of the gex document into DLV language looks as follows:

```
portfolio(X) v nonPortfolio(X) :- tickers(X).
:- portfolio(X), portfolio(Y),
     not noncorrelated(X, Y).
:~ nonPortfolio(X).
portfolio(?X).
```

A5 Result of the Use Case Program Execution

The maximal models of the portfolio predicate, found during the execution of the program on the dlv node, are the diversified portfolios of maximal size containing securities of companies from the S&P 500 for 2012. As the result of the program execution, the 11 stable models containing 10 ground atoms each were generated (the maximum size of a diversified portfolio appeared to be equal to 10). The models can be found below. Atoms contain symbols of different companies, e. g., DUK and SBUX denote *Duke Energy* and *Starbucks Corp.*, respectively. The full list of abbreviations is provided in the table. The models are outlined as the sets of company identifiers (like SBUX) related to the security ground terms in portfolio(SBUX). Note that the models have nonempty intersections:

Model 1: {CAH, DUK, EL, ETR, HOT, LM, PSA, TJX, TYC, UNH}
Model 2: {CAH, DUK, EL, ETR, HOT, LM, PSA, TJX, UNH, VNO}
Model 3: {CAH, DUK, EL, ETR, HON, LM, PSA, TJX, UNH, VNO}
Model 4: {CAH, DUK, EL, ETR, LM, PSA, TJX, TYC, UNH, WDC}
Model 5: {BMY, DUK, EL, FDX, IPG, KSS, PSA, STT, TJX, VIAB}
Model 6: {BA, BMY, DUK, EL, IPG, KSS, PSA, STT, TJX, VIAB}
Model 7: {BA, BMY, DUK, EL, KSS, LH, PSA, STT, TJX, VIAB}
Model 8: {AGN, BA, BMY, DUK, EL, KSS, LH, PSA, TJX, VIAB}
Model 9: {BA, BMS, BMY, DUK, EL, KSS, LH, PSA, SBUX, TJX}
Model 10: {BMY, BSX, DUK, EL, FDX, GAS, KMI, MDLZ, RRC, TJX}
Model 11:{BA, BMY, DUK, FMC, GIS, KIM, LH, LTD, MNST, SBUX}

Tickers symbols and company names

Ticker	Company	Ticker	Company
CAH	Cardinal Health Inc.	STT	State Street Corp.
DUK	Duke Energy	VIAB	Viacom Inc.
EL	Estee Lauder Cos.	BA	Boeing Company
ETR	Entergy Corp.	LH	Laboratory Corp. of America Holding
HOT	Starwood Hotels & Resorts	AGN	Allergan Inc
LM	Legg Mason	BMS	Bemis Company
PSA	Public Storage	SBUX	Starbucks Corp.
TJX	TJX Companies Inc.	BSX	Boston Scientific
TYC	Tyco International	GAS	AGL Resources Inc.
UNH	United Health Group Inc.	KMI	Kinder Morgan
VNO	Vornado Realty Trust	MDLZ	Mondelez International
HON	Honeywell Int'l Inc.	RRC	Range Resources Corp.
WDC	Western Digital	FMC	FMC Corporation
BMY	Bristol-Myers Squibb	GIS	General Mills
FDX	FedEx Corporation	KIM	Kimco Realty
IPG	Interpublic Group	LTD	L Brands Inc.
KSS	Kohl's Corp.	MNST	Monster Beverage

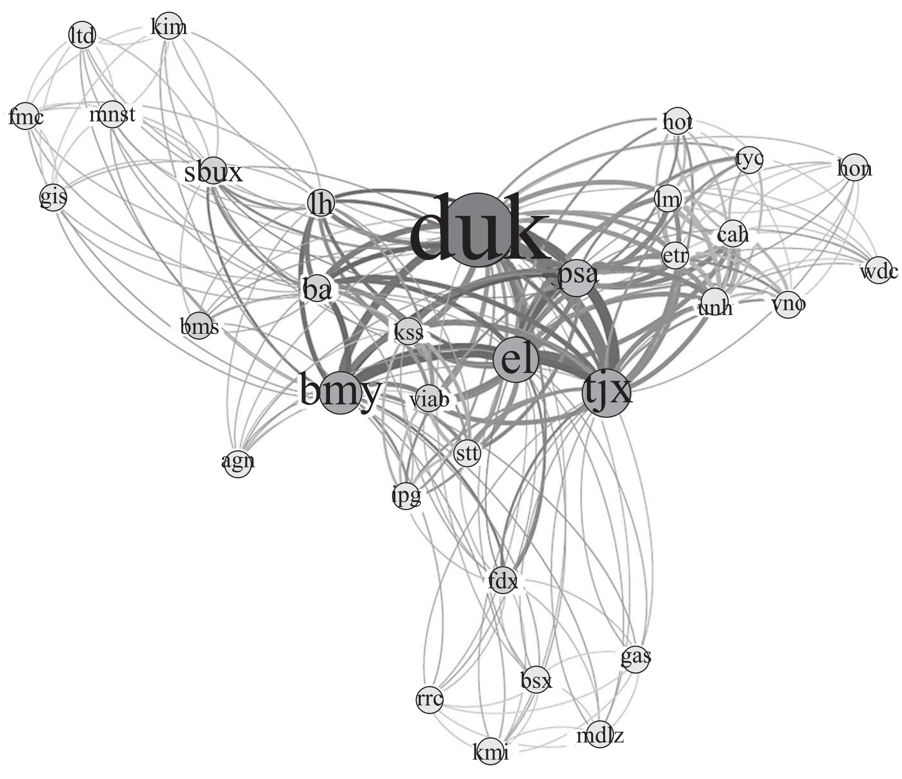

Figure 5 Models and their elements

Models and their elements are illustrated in Fig. 5[1].

The circles in the figure denote companies. The lines connecting two circles denote that two companies belong to the same model. The larger is the circle, the larger is the number of models to which a company belongs. The thicker is the line, the larger is the number of models to which a pair of companies belongs to.

References

1. Hey, T., S. Tansley, and K. Tolle, eds. 2009. *The fourth paradigm: Data-intensive scientific discovery*. Redmond: Microsoft Research. 252 p.
2. Challenges and opportunities with big data. A community white paper developed by leading re-

[1] Figure was created using a graph visualization platform *Gephi* (https://gephi.org/). The *ForceAtlas* layout algorithm was applied.

searchers across the United States. 2012. Available at: http://cra.org/ccc/docs/init/bigdatawhitepaper.pdf (accessed November 21, 2013).

3. Kappel, G., M. Wimmer, W. Retschitzegger, and W. Schwinger. 2011. Leveraging model-based tool integration by conceptual modeling techniques. *The evolution of conceptual modeling*. Eds. R. Kaschek and L. M. L. Delcambre. Lecture notes in computer science ser. Berlin, Heidelberg: Springer-Verlag. 6520:254–84.
4. Kalinichenko, L. A., S. A. Stupnikov, and D. O. Martynov. 2007. SYNTHESIS: A language for canonical information modeling and mediator definition for problem solving in heterogeneous information resource environments. M.: IPIRAN. 171 p.
5. Kalinichenko, L. A., D. O. Briukhov, D. O. Martynov, N. A. Skvortsov, and S. A. Stupnikov. 2007. Mediation framework for enterprise information system infrastructures. *9th Conference (International) on Enterprise Information Systems ICEIS 2007 Proceedings*. Funchal. Vol. Databases and information systems integration. 246–51.
6. Kalinichenko, L. A. 1990. Methods and tools for equivalent data model mapping construction. *Advances in database technology — EDBT'90*. Eds. F. Buncilhon, C. Thanos, and D. Tsichritzis. Lecture notes in computer science ser. Berlin, Heidelberg: Springer-Verlag. 416:92–119.
7. Kalinichenko, L. A., and S. A. Stupnikov. 2012. Synthesis of the canonical models for database integration preserving semantics of the value inventive data models. *Advances in database and information systems*. Eds. T. Morzy, T. Härder, R. Wrembel, *et al.* Lecture notes in computer science ser. Berlin, Heidelberg: Springer-Verlag. 7503:223–39.
8. Boley, H., and M. Kifer, eds. 2013. RIF overview. W3C working group note. 2nd ed. Available at: http://www.w3.org/TR/rif-overview/ (accessed November 21, 2013).
9. Kalinichenko, L., S. Stupnikov, A. Vovchenko, and D. Kovalev. 2013. Rule-based multi-dialect infrastructure for conceptual problem solving over heterogeneous distributed information resources. New trends in databases and information systems. *Selected Papers of the 17th European Conference on Advances in Databases and Information Systems and Associated Satellite Events. Advances in Intelligent Systems and Computing* 241:61–68.
10. Leone, N., G. Pfeifer, W. Faber, T. Eiter, G. Gottlob, S. Perri, and F. Scarcello. 2006. The DLV system for knowledge representation and reasoning. *ACM Trans. Comput. Log.* 7(3):499–562.
11. Kalinichenko, L. A., and S. A. Stupnikov. 2008. Constructing of mappings of heterogeneous information models into the canonical models of integrated information systems. *ADBIS 2008 Proceedings*. Pori: Tampere University of Technology. 106–22.
12. Abrial, J.-R. 1996. The B-book: Assigning programs to meanings. Cambridge: Cambridge University Press. 816 p.
13. Skvortsov, N. A. 2012. Otobrazhenie modeley dannykh NoSQL v ob"ektnye specifikatsii [Mapping of NoSQL data models to object specifications]. *Trudy 14-y Vserossiyskoy nauchnoy konferentsii "Elektronnye biblioteki: Perspektivnye metody i tekhnologii, elektronnye kollektsii"' RCDL 2012* [*14th Russian Conference on Digital Libraries RCDL 2012 Proceedings*]. *CEUR Workshop Proceedings* 934:53–62.
14. Stupnikov, S. A. 2013. Otobrazhenie grafovoy modeli dannykh v kanonicheskuyu ob"ektno-freymovuyu informatsionnuyu model' pri sozdanii sistem integratsii neodnorodnykh informatsionnykh resursov [Mapping of a graph data model into an object-frame canonical information model for the development of heterogeneous information resources integration systems]. *Trudy 15-y Vserossiyskoy nauchnoy konferentsii "Elektronnye biblioteki: Perspektivnye metody i tekhnologii, elektronnye kollektsii" RCDL 2013* [*15th Russian Conference on Digital Libraries RCDL 2013 Proceedings*]. Yaroslavl: P. G. Demidov Yaroslavl State University. 193–202.
15. Skvortsov, N. A. 2013. Otobrazhenie modeli dannykh RDF v kanonicheskuyu model' predmetnykh posrednikov [Mapping of RDF data model into the canonical model of subject mediators]. *Trudy 15-y Vserossiyskoy nauchnoy konferentsii "Elektronnye biblioteki: Perspektivnye metody i tekhnologii, elektronnye kollektsii" RCDL 2013* [*15th Russian Conference on Digital Libraries RCDL 2013 Proceedings*]. Yaroslavl: P. G. Demidov Yaroslavl State University. 202–9.
16. Stupnikov, S. A. 2013. Verifitsiruemoe otobrazhenie modeli dannykh, osnovannoy na mnogomernykh massivakh, v ob"ektnuyu model' dannykh [A verifiable mapping of a multidimensional array data model into an object data model]. *Inform. Appl.* 7(3):22–34.
17. Fagin, R., P. G. Kolaitis, R. J. Miller, and L. Popa. 2005. Data exchange: Semantics and query answering. *Theor. Comput. Sci.* 336:89–124.
18. Lenzerini, M. 2002. Data integration: A theoretical perspective. *ACM Symposium on Principles of Database Systems (PODS) Proceedings*. 233–46.
19. Kalinichenko, L. A., D. O. Martynov, and S. A. Stupnikov. 2004. Query rewriting using views in a typed mediator environment. *Advances in databases and information systems*. Eds. G. Gottlob, A. Benczur, and J. Demetrovics. Lecture notes in computer science ser. Berlin, Heidelberg: Springer-Verlag. 3255:37–53.
20. Boley, H., and M. Kifer, eds. 2013. RIF basic logic dialect. W3C recommendation. 2nd ed. Available at: http://www.w3.org/TR/rif-bld/ (accessed November 21, 2013).
21. De Sainte Marie, C., G. Hallmark, and A. Paschke, eds. 2013. RIF production rule dialect. W3C recommendation. 2nd ed. Available at: http://www.w3.org/TR/rif-prd/ (accessed November 21, 2013).
22. Boley, H., and M. Kifer, eds. 2013. RIF framework for logic dialects. W3C recommendation. 2nd ed. Available at: http://www.w3.org/TR/rif-fld/ (accessed November 21, 2013).
23. Kifer, M., ed. 2010. RIF core logic programming dialect based on the well-founded semantics. Available at: http://ruleml.org/rif/RIF-CLPWD.html (accessed November 21, 2013).

24. Van Gelder, A., K. A. Ross, and J. S. Schlipf. 1991. The well-founded semantics for general logic programs. *J. ACM* 38(3):620–50.
25. Heymans, S., and M. Kifer, eds. 2009. RIF core answer set programming dialect. Available at: http://ruleml.org/rif/RIF-CASPD.html (accessed November 21, 2013).
26. Gelfond, M., and V. Lifschitz. 1988. The stable model semantics for logic programming. *Logic Programming: 5th Conference and Symposium Proceedings*. 1070–80.
27. De Bruijn, J., and C. Welty, eds. 2013. RIF RDF and OWL Compatibility. W3C recommendation. 2nd ed. Available at: http://www.w3.org/TR/rif-rdf-owl/ (accessed November 21, 2013).
28. Motik, B., P. F. Patel-Schneider, and B. Parsia, eds. 2012. OWL 2 Web ontology language structural specification and functional-style syntax. W3C recommendation. 2nd ed. Available at: http://www.w3.org/TR/owl2-syntax/ (accessed November 21, 2013).
29. Shvaiko, P., and J. Euzenat. 2005. A survey of schema-based matching approaches. *J. Data Semantics* IV:146–71.
30. Abiteboul, S., M. Bienvenu, A. Galland, *et al*. 2011. A rule-based language for Web data management. *30th ACM Symposium on Principles of Database Systems Proceedings*. ACM Press. 283–92.
31. Gelfond, M. 2008. *Answer sets. Handbook of knowledge representation*. Elsevier. 285–316.
32. Motik, B., and R. Rosati. 2007. Closing semantic Web ontologies. University of Manchester Report. Available at: http://www.cs.ox.ac.uk/people/boris.motik/pubs/mr06closing-report.pdf (accessed November 21, 2013).
33. Chen, P. P., B. Thalheim, and L. Y. Wong. 1999. Future directions of conceptual modeling. *Conceptual modeling*. Eds. P. P. Chen, J. Akoka, H. Kangassulu, and B. Thalheim. Lecture notes in computer science ser. Berlin, Heidelberg: Springer-Verlag. 1565:287–301.
34. Costal, D., C. Gómez, and G. Guizzardi. 2011. Formal semantics and ontological analysis for understanding subsetting, specialization and redefinition of associations in UML. *Conceptual modeling — ER 2011*. Eds. M. Jeusfeld, L. Delcambre, and T. W. Ling. Lecture notes in computer science ser. Berlin, Heidelberg: Springer-Verlag. 6998:189–203.
35. Martinez, Y., C. Cachero, M. Matera, S. Abrahao, and S. Lujan. 2011. Impact of MDE approaches on the maintainability of Web applications: An experimental evaluation. *Conceptual modeling — ER 2011*. Eds. M. Jeusfeld, L. Delcambre, and T. W. Ling. Lecture notes in computer science ser. Berlin, Heidelberg: Springer-Verlag. 6998:233–46.
36. Castro, L. F. Baião, and G. Guizzardi. 2011. A semantic qriented method for conceptual data modeling in OntoUML based on linguistic concepts. *Conceptual modeling — ER 2011*. Eds. M. Jeusfeld, L. Delcambre, and T. W. Ling. Lecture notes in computer science ser. Berlin, Heidelberg: Springer-Verlag. 6998:486–94.
37. Fillottrani, P. R., E. Franconi, and S. Tessaris. 2011. The ICOM 3.0 intelligent conceptual modelling tool and methodology. Semantic Web. *IOS Press*. 1–14. Available at: http://www.semantic-web-journal.net/sites/default/files/swj105_1.pdf (accessed November 21, 2013).
38. Maus, C., S. Rybacki, and A. M. Uhrmache. 2011. Rule-based multi-level modeling of cell biological systems. *BMC Syst. Biol.* 5:166. Available at: http://www.biomedcentral.com/content/pdf/1752-0509-5-166.pdf (accessed November 21, 2013).
39. Calvanese, D., G. De Giacomo, D. Lembo, M. Lenzerini, A. Poggi, and R. Rosati. 2007. MASTRO-I: Efficient integration of relational data through DL ontologies. *2007 Workshop (International) on Description Logic (DL 2007) Proceedings*. *CEUR Workshop Proceedings*. 250.
40. Cosentino, V., M. D. Del Fabro, and A. El Ghali. 2012. A model driven approach for bridging ILOG rule language and RIF. *RuleML 2012 Proceedings*. *CEUR Workshop Proceedings*. 874. Available at: http://ceur-ws.org/Vol-874/paper9.pdf (accessed November 21, 2013).
41. Gonzalez-Moriyon, G., L. Polo, D. Berrueta, and C. Tejo-Alonso. 2012. Final steel industry public demonstrators. ONTORULE deliverable D5.5. Available at: http://ontorule-project.eu/outcomes%3Ffunc=fileinfo&id=94.html (accessed November 21, 2013).
42. Loo, B. T., T. Condie, M. Garofalakis, D. E. Gay, J. M. Hellerstein, P. Maniatis, R. Ramakrishnan, T. Roscoe, and I. Stoica. 2006. Declarative networking: Language, execution and optimization. *ACM SIGMOD Conference Proceedings*. 97–108.
43. Grumbach, S., and F. Wang. 2010. Netlog, a rule-based language for distributed programming. *Practical aspects of declarative languages*. Eds. M. Carro and R. Peña. Lectire notes in computer science ser. Berlin, Heidelberg: Springer-Verlag. 5937:88–103.
44. Alvaro, P., W. R. Marczak, N. Conway, J. M. Hellerstein, D. Maier, and R. Sears. 2010. Dedalus: Datalog in time and space. *Datalog reloaded*. Eds. O. de Moor, G. Gottlob, T. Furche, and A. Sellers. Lectures notes in computer science ser. Berlin, Heidelberg: Springer-Verlag. 6702: 262–81.
45. Grossmann, G., R. Thiagarajan, M. Schrefl, and M. Stumptner. 2011. Conceptual modeling approaches for dynamic Web service composition. *The evolution of conceptual modeling*. Eds. R. Kaschek and L. M. L. Delcambre. Lecture notes in computer science ser. Berlin, Heidelberg: Springer-Verlag. 6520:180–204.
46. Henderson-Sellers, B. 2011. Random thoughts on multi-level conceptual modelling. *The evolution of conceptual modeling*. Eds. R. Kaschek and L. M. L. Delcambre. Lecture notes in computer science ser. Berlin, Heidelberg: Springer-Verlag. 6520:93–116.
47. Falbo, R. A., F. B. Ruy, and R. D. Moro. 2005. Using ontologies to add semantics to a software engineering environment. *SEKE 2005 Proceedings*. Curran Associates. 151–56.

48. Berre, A., and M. Missikoff (Moderators). 2011. Panel: Modeling for the future Internet. *Conceptual modeling — ER 2011*. Eds. M. Jeusfeld, L. Delcambre, and T. W. Ling. Lecture notes in computer science ser. Berlin, Heidelberg: Springer-Verlag. 6998: 526–27.
49. Porto, F., and S. Spaccapietra. 2011. Data model for scientific models and hypotheses. *The evolution of conceptual modeling*. Eds. R. Kaschek and L. M. L. Delcambre. Lecture notes in computer science ser. Berlin, Heidelberg: Springer-Verlag. 6520: 285–305.
50. Racunas, S. A., N. H. Shah, I. Albert, and N. V. Fedoroff. 2004. Hybrow: A prototype system for computer-aided hypothesis evaluation. *Bioinformatics* 20(1):257–64.
51. Sharpe, W., G. J. Alexander, and J. W. Bailey. 1998. *Investments*. Prentice Hall. 962 p.
52. Gabbay, D. M. 2012. What is negation as failure? *Logic programs, norms and action*. Eds. A. Artikis, R. Craven, N. K. Çiçekli, *et al*. Lecture notes in computer science ser. Berlin, Heidelberg: Springer-Verlag. 7360:52–78.
53. Briukhov, D. O., A. E. Vovchenko, V. N. Zakharov, O. P. Zhelenkova, L. A. Kalinichenko, D. O. Martynov, N. A. Skvortsov, and S. A. Stupnikov. 2008. Arkhitektura promezhutochnogo sloya predmetnykh posrednikov dlya resheniya zadach nad mnozhestvom integriruemykh neodnorodnykh raspredelennykh informatsionnykh resursov v gibridnoy grid-infrastrukture virtual'nykh observatoriy [The middle ware architecture of the subject mediators for problem solving over a set of integrated heterogeneous distributed information resources in the hybrid grid-infrastucture of virtual observatories]. *Inform. Appl.* 2(1):2–34.
54. ATL Project. 2013. Available at: http://www.eclipse.org/atl/ (accessed November 21, 2013).

КОНЦЕПТУАЛЬНЫЕ ДЕКЛАРАТИВНЫЕ СПЕЦИФИКАЦИИ И РЕШЕНИЕ ЗАДАЧ В ОБЛАСТЯХ С ИНТЕНСИВНЫМ ИСПОЛЬЗОВАНИЕМ ДАННЫХ

Л. Калиниченко[1], С. Ступников[2], А. Вовченко[3], Д. Ковалев[4]

[1]Институт проблем информатики Российской академии наук, Москва, Россия, leonidandk@gmail.com
[2]Институт проблем информатики Российской академии наук, Москва, Россия, ssa@ipi.ac.ru
[3]Институт проблем информатики Российской академии наук, Москва, Россия, itsnein@gmail.com
[4]Институт проблем информатики Российской академии наук, Москва, Россия, dm.kovalev@gmail.com

Аннотация: В сфере концептуального моделирования долгое время исследовались разнообразные нотации, предназначенные для определения семантики вычислений в терминах предметных областей. Подход «сущность–связь» и диаграммы UML позволяют определять семантику лишь неформально. Онтологические языки, основанные на дескриптивной логике, разрабатывались для формализации семантики данных. Однако сейчас общепризнано, что одной лишь семантики данных недостаточно — требуется еще и представление алгоритмов анализа данных для спецификации данных и поведения в одной парадигме. Более того, все усиливающееся разнообразие разноструктурированных моделей данных вызывает потребность в их унифицированной, интегрированной абстракции для получения спецификаций, независимых от реальных данных в предметных областях с интенсивным использованием данных. С целью преодоления названных недостатков предлагается новый подход к применению семантически различных языков на правилах (диалектов) для создания интероперабельных концептуальных спецификаций над различными системами на правилах. Подход основан на на технике преобразования логических программ, рекомендованной *Форматом обмена правилами* (RIF) W3C. Этот подход гармонично сочетается со спецификациями, предназначенными для определения семантических посредников на правилах, обеспечивающих интеграцию неоднородных баз данных. Определена инфраструктура, реализующая мультидиалектные концептуальные спецификации при помощи интероперабельных систем на правилах и систем поддержки посредников. Представлен подтверждающий предложенные концепции прототип инфраструктуры, основанный на системе поддержки посредников СИНТЕЗ и стандарте RIF. Подход к мультидиалектной концептуализации предметной области, делегированию правил, интероперабельности программ на правилах и посредников подробно рассмотрен и проиллюстрирован на реальном примере NP-полной задачи в финансовой области. Результаты исследования свидетельствуют о применимости подхода и инфраструктуры для концептуального, декларативного, независимого от ресурсов и повторно используемого анализа данных в различных предметных областях.

Ключевые слова: концептуальная спецификация; W3C RIF; логические языки на правилах; СИНТЕЗ; интеграция баз данных, посредники; RIF-BLD; RIF-CASPD; мультидиалектная инфраструктура; делегирование правил

DOI: 10.14357/19922264130412

Работа выполнена при поддержке Российского фонда фундаментальных исследований (проект 11-07-00402-а) и Программы фундаментальных исследований Президиума Российской академии наук.

Литература

1. The fourth paradigm: Data-intensive scientific discovery / Eds. T. Hey, S. Tansley, K. Tolle. — Redmond: Microsoft Research, 2009. 252 p.
2. Challenges and opportunities with big data. A community white paper developed by leading researchers across the United States, 2012. Available at: http://cra.org/ccc/docs/init/bigdatawhitepaper.pdf (accessed November 21, 2013).
3. *Kappel G., Wimmer M., Retschitzegger W., Schwinger W.* Leveraging model-based tool integration by conceptual modeling techniques // The evolution of conceptual modeling. Eds. R. Kaschek and L. M. L. Delcambre. — Lecture notes in computer science ser. — Berlin, Heidelberg: Springer-Verlag, 2011. Vol. 6520. P. 254–284.
4. *Kalinichenko L. A., Stupnikov S. A., Martynov D. O.* SYNTHESIS: A language for canonical information modeling and mediator definition for problem solving in heterogeneous information resource environments. — M.: IPI-RAN, 2007. 171 p.
5. *Kalinichenko L. A., Briukhov D. O., Martynov D. O., Skvortsov N. A., Stupnikov S. A.* Mediation framework for enterprise information system infrastructures // 9th Conference (International) on Enterprise Information Systems ICEIS 2007 Proceedings. — Funchal, 2007. Vol. Databases and information systems integration. P. 246–251.
6. *Kalinichenko L. A.* Methods and tools for equivalent data model mapping construction // Advances in database technology — EDBT'90. Eds. F. Buncilhon, C. Thanos, and D. Tsichritzis. — Lecture notes in computer science ser. — Berlin, Heidelberg: Springer-Verlag, 1990. Vol. 416. P. 92–119.
7. *Kalinichenko L. A., Stupnikov S. A.* Synthesis of the canonical models for database integration preserving semantics of the value inventive data models // Advances in database and information systems / Eds. T. Morzy, T. Härder, R. Wrembel, *et al.* Lecture notes in computer science ser. Berlin, Heidelberg: Springer-Verlag, 2012. Vol. 7503. P. 223–239.
8. RIF overview. W3C working group note / Eds. H. Boley, M. Kifer. — 2nd ed., 2013. Available at: http://www.w3.org/TR/rif-overview/ (accessed November 21, 2013).
9. *Kalinichenko L., Stupnikov S., Vovchenko A., Kovalev D.* Rule-based multi-dialect infrastructure for conceptual problem solving over heterogeneous distributed information resources. New trends in databases and information systems // Selected Papers of the 17th European Conference on Advances in Databases and Information Systems and Associated Satellite Events. Advances in Intelligent Systems and Computing, 2013. Vol. 241. P. 61–68.
10. *Leone N., Pfeifer G., Faber W., Eiter T., Gottlob G., Perri S., Scarcello F.* The DLV system for knowledge representation and reasoning // ACM Trans. Comput. Log., 2006. Vol. 7. No. 3. P. 499–562.
11. *Kalinichenko L. A., Stupnikov S. A.* Constructing of mappings of heterogeneous information models into the canonical models of integrated information systems // ADBIS 2008 Proceedings. — Pori: Tampere University of Technology, 2008. P. 106–122.
12. *Abrial J.-R.* The B-Book: Assigning programs to meanings. — Cambridge: Cambridge University Press, 1996. 816 p.
13. *Скворцов Н. А.* Отображение моделей данных NoSQL в объектные спецификации // Труды 14-й Всеросс. научн. конф. «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» RCDL 2012. — Переславль-Залесский: Ун-т г. Переславля, 2012. Т. 934. С. 53–62.
14. *Ступников С. А.* Отображение графовой модели данных в каноническую объектно-фреймовую информационную модель при создании систем интеграции неоднородных информационных ресурсов // Труды 15-й Всеросс. научн. конф. «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» RCDL 2013. — Ярославль: Ярославский гос. ун-т им. П. Г. Демидова, 2013. С. 193–202.
15. *Скворцов Н. А.* Отображение модели данных RDF в каноническую модель предметных посредников // Труды 15-й Всеросс. научн. конф. «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» RCDL 2013. — Ярославль: Ярославский гос. ун-т им. П. Г. Демидова, 2013. С. 202–209.
16. *Ступников С. А.* 2013. Верифицируемое отображение модели данных, основанной на многомерных массивах, в объектную модель данных // Информатика и её применения, 2013. Т. 7. Вып. 3. С. 22–34.
17. *Fagin R., Kolaitis P. G., Miller R. J., Popa L.* Data exchange: Semantics and query answering // Theor. Comput. Sci., 2005. Vol. 336. P. 89–124.
18. *Lenzerini M.* Data integration: A theoretical perspective // ACM Symposium on Principles of Database Systems (PODS) Proceedings, 2002. P. 233–246.
19. *Kalinichenko L. A., Martynov D. O., Stupnikov S. A.* Query rewriting using views in a typed mediator environment // Advances in databases and information systems / Eds. G. Gottlob, A. Benczur, J. Demetrovics. — Lecture notes in computer science ser. — Berlin, Heidelberg: Springer-Verlag, 2004. Vol. 3255. P. 37–53.

20. RIF basic logic dialect. W3C recommendation / Eds. H. Boley, M. Kifer. — 2nd ed., 2013. Available at: http://www.w3.org/TR/rif-bld/ (accessed November 21, 2013).
21. RIF production rule dialect. W3C recommendation / Eds. C. De Sainte Marie, G. Hallmark, A. Paschke. — 2nd ed., 2013. Available at: http://www.w3.org/TR/rif-prd/ (accessed November 21, 2013).
22. RIF framework for logic dialects. W3C recommendation / Eds. H. Boley, M. Kifer. — 2nd ed., 2013. Available at: http://www.w3.org/TR/rif-fld/ (accessed November 21, 2013).
23. RIF core logic programming dialect based on the well-founded semantics / Ed. M. Kifer. — 2010. Available at: http://ruleml.org/rif/RIF-CLPWD.html (accessed November 21, 2013).
24. *Van Gelder A., Ross K. A., Schlipf J. S.* The well-founded semantics for general logic programs // J. ACM, 1991. Vol. 8. No. 3. P. 620–650.
25. RIF core answer set programming dialect / Eds. S. Heymans, M. Kifer. — 2009. Available at: http://ruleml.org/rif/RIF-CASPD.html (accessed November 21, 2013).
26. *Gelfond M., Lifschitz V.* The stable model semantics for logic programming // Logic programming. 5th Conference and Symposium Proceedings, 1988. P. 1070–1080.
27. RIF RDF and OWL compatibility. W3C Recommendation / Eds. J. De Bruijn, C. Welty. — 2nd ed., 2013. Available at: http://www.w3.org/TR/rif-rdf-owl/ (accessed November 21, 2013).
28. OWL 2 Web ontology language structural specification and functional-style syntax. W3C recommendation / B. Motik, P. F. Patel-Schneider, B. Parsia. — 2nd ed., 2012. Available at: http://www.w3.org/TR/owl2-syntax/ (accessed November 21, 2013).
29. *Shvaiko P., Euzenat J.* A survey of schema-based matching approaches // J. Data Semantics, 2005. Vol. IV. P. 146–171.
30. *Abiteboul S., Bienvenu M., Galland A., et al.* A rule-based language for Web data management // 30th ACM Symposium on Principles of Database Systems Proceedings. — ACM Press, 2011. P. 283–292.
31. *Gelfond M.* Answer sets. Handbook of knowledge representation. — Elsevier. 2008. P. 285–316.
32. *Motik B., Rosati R.* Closing semantic Web ontologies. University of Manchester Report, 2007. Available at: http://www.cs.ox.ac.uk/people/boris.motik/pubs/mr06closing-report.pdf (accessed November 21, 2013).
33. *Chen P. P., Thalheim B., Wong L. Y.* Future directions of conceptual modeling // Conceptual modeling / Eds. P. P. Chen, J. Akoka, B. Thalheim. — Lecture notes in computer science ser. — Berlin, Heidelberg: Springer-Verlag, 1999. Vol. 1565. P. 287–301.
34. *Costal D., Gmbóxomez C., Guizzardi G.* Formal semantics and ontological analysis for understanding subsetting, Specialization and redefinition of associations in UML // Conceptual modeling — ER 2011 / Eds. M. Juesfeld, L. Delcambre, T. W. Ling. — Lecture notes in computer science ser. — Berlin, Heidelberg: Springer-Verlag, 2011. Vol. 6998. P. 189–203.
35. *Martιnez Y., Cachero C., Matera M., Abrahao S., Lujan S.* Impact of MDE approaches on the maintainability of Web applications: An experimental evaluation // Conceptual Modeling — ER 2011 / Eds. M. Jeusfeld, L. Delcambre, T. W. Ling. — Lecture notes in computer science ser. — Berlin, Heidelberg: Springer-Verlag, 2011. Vol. 6998. P. 233–246.
36. *Castro L., Baião F., Guizzardi G.* A semantic oriented method for conceptual data modeling in OntoUML based on linguistic concepts // Conceptual modeling — ER 2011 / Eds. M. Jeusfeld, L. Delcambre, T. W. Ling. — Lecture notes in computer science ser. — Berlin, Heidelberg: Springer-Verlag, 2011. Vol. 6998. P. 486–494.
37. *Fillottrani P. R., Franconi E., Tessaris S.* The ICOM 3.0 intelligent conceptual modelling tool and methodology. Semantic Web. — IOS Press, 2011. P. 1–14. Available at: http://www.semantic-web-journal.net/sites/default/files/swj105_1.pdf (accessed November 21, 2013).
38. *Maus C., Rybacki S., Uhrmache A. M.* Rule-based multi-level modeling of cell biological systems // BMC Syst. Biol., 2011. Vol. 5. P. 166. Available at: http://www.biomedcentral.com/content/pdf/1752-0509-5-166.pdf (accessed November 21, 2013).
39. *Calvanese D., De Giacomo G., Lembo D., Lenzerini M., Poggi A., Rosati R.* MASTRO-I: Efficient integration of relational data through DL ontologies // 2007 Workshop (International) on Description Logic (DL 2007) Proceedings. CEUR Workshop Proceedings, 2007. P. 250.
40. *Cosentino V., Del Fabro M. D., El Ghali A.* A model driven approach for bridging ILOG rule language and RIF // RuleML 2012 Proceedings. CEUR Workshop Proceedings, 2012. P. 874. Available at: http://ceur-ws.org/Vol-874/paper9.pdf (accessed November 21, 2013).
41. *Gonzalez-Moriyon G., Polo L., Berrueta D., Tejo-Alonso C.* Final steel industry public demonstrators. ONTORULE deliverable D5.5. — 2012. Available at: http://ontorule-project.eu/outcomes%3Ffunc=fileinfo&id=94.html (accessed November 21, 2013).
42. *Loo B. T., Condie T., Garofalakis M., Gay D. E., Hellerstein J. M., Maniatis P., Ramakrishnan R., Roscoe T., Stoica I.* Declarative networking: Language, execution and optimization // ACM SIGMOD Conference Proceedings, 2006. P. 97–108.
43. *Grumbach S., Wang F.* Netlog, a rule-based language for distributed programming // Practical aspects of declarative languages / Eds. M. Carro, R. Peña. — Lecture notes in computer science ser. — Berlin, Heidelberg: Springer-Verlag, 2010. Vol. 5937. P. 88–103.
44. *Alvaro P., Marczak W. R., Conway N., Hellerstein J. M., Maier D., Sears R.* Dedalus: Datalog in time and space // Datalog reloaded / Eds. O. de Moor, G. Gottlob, T. Furche, and A. Sellers. — Lecture notes in computer science ser. — Berlin, Heidelberg: Springer-Verlag, 2010. Vol. 6702. P. 262–281.

45. *Grossmann G., Thiagarajan R., Schrefl M., Stumptner M.* Conceptual modeling approaches for dynamic Web service composition // The evolution of conceptual modeling / Eds. R. Kaschek, L. M. L. Delcambre. — Lecture notes in computer science ser. — Berlin, Heidelberg: Springer-Verlag, 2011. Vol. 6520. P. 180–204.

46. *Henderson-Sellers B.* Random thoughts on multi-level conceptual modelling // The evolution of conceptual modeling / Eds. R. Kaschek, L. M. L. Delcambre. — Lecture notes in computer science ser. — Berlin, Heidelberg: Springer-Verlag, 2011. Vol. 6520. P. 93–116.

47. *Falbo R. A., Ruy F. B., Moro R. D.* Using ontologies to add semantics to a software engineering environment // SEKE 2005 Proceedings. — Curran Associates, 2005. P. 151–156.

48. Berre A., Missikoff M. (Moderators). Panel: Modeling for the future Internet // Conceptual modeling — ER 2011 / Eds. M. Jeusfeld, L. Delcambre, T. W. Ling. — Lecture notes in computer science ser. — Berlin, Heidelberg: Springer-Verlag, 2011. Vol. 6998. P. 526–527.

49. *Porto F., Spaccapietra S.* Data model for scientific models and hypotheses // The evolution of conceptual modeling / Eds. R. Kaschek, L. M. L. Delcambre. — Lecture notes in computer science ser. — Berlin, Heidelberg: Springer-Verlag, 2011. Vol. 6520. P. 285–305.

50. *Racunas S. A., Shah N. H., Albert I., Fedoroff N. V.* 2004. Hybrow: A prototype system for computer-aided hypothesis evaluation // Bioinformatics, 2004. Vol. 20. No. 1. P. 257–264.

51. *Sharpe W., Alexander G. J., Bailey J. W.* Investments. — Prentice Hall, 1998. 962 p.

52. *Gabbay D. M.* What is negation as failure // Logic Programs, norms and action / Eds. A. Artikis, R. Craven, N. K. Çiçekli, *et al.* — Lecture notes in computer science ser. — Berlin, Heidelberg: Springer-Verlag, 2012. Vol. 7360. P. 52–78.

53. *Брюхов Д. О., Вовченко А. Е., Захаров В. Н., Железкова О. П., Калиниченко Л. А., Мартынов Д. О., Скворцов Н. А., Ступников С. А.* Архитектура промежуточного слоя предметных посредников для решения задач над множеством интегрируемых неоднородных распределенных информационных ресурсов в гибридной грид-инфраструктуре виртуальных обсерваторий // Информатика и её применения, 2008. Т. 2. Вып. 1. С. 2–34.

54. ATL Project. 2013. Available at: http://www.eclipse.org/atl/ (accessed November 21, 2013).