Rule-based Multi-dialect Infrastructure for Conceptual Problem Solving over Heterogeneous Distributed Information Resources*

Leonid Kalinichenko, Sergey Stupnikov, Alexey Vovchenko, Dmitry Kovalev

Institute of Informatics Problems, Russian Academy of Sciences, Moscow, Russia
leonidandk@gmail.com, ssa@ipi.ac.ru, itsnein@gmail.com,
dm.kovalev@gmail.com

Abstract. An approach for supporting a possibility of applying a combination of the semantically different logic rule languages (dialects) for interoperable conceptual programming over various rule systems (RS) and relying on the logic program transformation technique recommended by the W3C Rule Interchange Format (RIF) is presented. We show also how to combine such approach with the heterogeneous data base integration applying semantic rule mediation. The basic functions of the infrastructure developed for implementation of the multi-dialect conceptual specifications by the interoperable relevant RS programs are defined. The detailed description of the infrastructure application for solving complex combinatorial problem as two interoperable sub-applications defined in two RIF dialects – BLD (transformed into a logic program over database integrating mediator) and CASPD (transformed into a DLV declarative program solving NP-complete part of the application) is given. The research results show the usability of the approach and of the infrastructure for declarative resource independent and re-usable data analysis in various application domains.

Keywords: conceptual specification, RIF, logic rule languages, database integration, mediators, BLD, CASPD, multi-dialect infrastructure, rule delegation

1 Introduction

The paper investigates a novel methodology and infrastructure supporting conceptually-driven problems specification and solving aiming eventually at the specifications reusability in various applications over different sets of data, widely diverse data and knowledge semantic integration capability, and for accumulation of reproducible data analysis and problem solving methods and experience in various application domains.

Specifically the approach proposed is aimed at conceptual modeling of data intensive domains (DID) in rule-based dialects possessing different, complementary se-

* 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 RAS.

mantics and capabilities combined with the methods for heterogeneous data mediation and integration. Besides that, the approach might also be applicable for the programmability and composition of complex analytical pipelines in an understandable form applying appropriate high-level languages to express the analytics intended for inferring knowledge from data [1].

In the work presented the issues of interoperability and integration of various information resources (such as data and knowledge bases, software services, ontologies) in time of problem solving are investigated on the basis of two approaches: 1) constructing of the unifying extensible language providing for semantic preserving representation in it of various information resource (IR) languages; 2) creation of the unified extensible family of rule-based languages (dialects) and a model of interoperability of the programs in such dialects. The first approach is based on the developed at IPIRAN the SYNTHESIS language [2-3] accompanied by methods and facilities for constructing of 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 object Datalog with functions and negation. Over such kernel the extensions are constructed in such a way that each extension together with the kernel is a result of semantic preserving mapping of some IR language into the SYNTHESIS [4-5]. As a result the canonical information model is constructed to be used for development of subject mediators positioned between the users, conceptually formulating problems in terms of the mediator, 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 definition of subject domain concepts defined by the respective ontological specifications. A mediator in SYNTHESIS can be defined as a triple (*T, S, M*), where *T* is the mediator (target) schema, *S* is a set of resource schemas, and *M* is a set of assertions (rules) relating elements of the target schema with elements of the resource schema. *M* is defined as a set of assertions of weakly-acyclic class [5-6] and is based on the GLAV technique combining LAV (Local as View) and GAV (Global as View) approaches.

Another, multi-dialect approach for IR interoperability applied in the current work is based on the RIF (Rule Interchange Format) recommendation [7] of W3C. RIF introduces a unified rule-based language (dialect) family together with a methodology for constructing of semantic preserving mappings in such dialects of specific languages used in various Rule-based Systems (RS). Preserving semantics during dialect mapping means preserving *entailment* of formulae (*f entails g* iff *g* is true on all models on which *f* is true). For inclusion of a rule-based language of a specific RS into a set of interoperable dialects it is required to develop for this RS two semantic preserving transformers – from the RS language into a RIF dialect (a *supplier* role) and from the dialect into the RS language (a *consumer* role). Every RIF dialect can have its own semantics different from other dialects. For the present RIF the recommendations are defined for the very basic dialects. E.g., the RIF-BLD (Basic Logic Dialect [8]) corresponds to the Horn logic with some extensions. Its subdialects that still are expected to be accepted as the W3C recommendations are the RIF-CLPWD (Core Logic Programming Well-founded Dialect [9]), which uses well-founded semantics (WFS) with the default negation and functions, and RIF-CASPD (Core Answer Set Pro-

gramming Dialect [10]) which uses answer set programming semantics (ASP), known also as the stable model semantics [11]. WFS and ASP are used for different purposes. ASP-based systems are specifically oriented on solving of complex combinatorial (NP-complete) problems whereas WFS-based systems are computationally complete and can be used as the general purpose logic programming facilities. RIF recommendations have also defined the necessary concepts to ensure compatibility of RIF with RDF and OWL [12], in spite of dissimilarity of their syntaxes and semantics.

The paper presents the results obtained including the description of an approach and an infrastructure supporting (a) the application domain conceptual specification and problem solving algorithms definitions based on the combination of the heterogeneous database mediation technique and rule-based multi-dialect facilities; (b) interoperability of distributed multi-dialect rule-based programs and mediators integrating heterogeneous databases; (c) rule delegation approach in the multi-dialect environment. The infrastructure based on the SYNTHESIS environment and RIF standards has been implemented. The approach for multi-dialect conceptualization of the application, rule delegation and rule-based programs and mediators interoperability is explained in detail and demonstrated on a real NP-complete application chosen from the finance domain. For the conceptual definition of the problem we use OWL for the domain concepts definition and programs in two dialects RIF BLD and RIF-CASPD mapped into the SYNTHESIS mediator and ASP-based DLV [13] program respectively.

The paper is structured as follows. After the introduction, the section containing an overview of the approach and an infrastructure for distributed multi-dialect rule-based programs support is given. In the third section a detailed description of the application example is provided. A related work section and the conclusion which summarizes the results and outlines plans for the future work close the paper.

2 Infrastructure of the Multi-dialect Environment for Distributed Rule-based Programs Interoperability

2.1 Conceptual Programming and Conceptual Schema

The aim of the infrastructure proposed is a conceptual programming of problems in RIF dialects and an implementation of conceptual programs using declarative languages of the RSs. These languages possess different capabilities and semantics and provide for programming over heterogeneous resources of data, programs and ontologies. Access to the resources is provided by concrete RSs or subject mediators. Conceptual multi-dialect logic programs specify the algorithms for problem solving in a subject domain. They are implemented using their transformation into a RS or a mediator programs.

Conceptual schema of a problem (class of problems) is defined in the frame of a subject domain and consists of a set of *RIF-documents* (document is a specification unit of RIF). Every document contains groups of rules. The subject domain conceptualization is performed using OWL 2 ontologies containing entities of the domain and

their relationships (Fig. 1). Ontologies constitute the *conceptual specification* of the domain. Name of the entities (classes and attributes) are used in the rules of the RIF-documents. Ontologies are imported into RIF-documents specifying an import profile, for instance, *OWL Direct*. A profile defines a semantics of an OWL ontology. Profiles are formally defined in [12].

Every RIF-document is defined using a dialect which is a *specialization* of the RIF Framework for Logic Dialects (RIF-FLD) [14]. 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* [14]. Here *f* means a term and *r* means a reference to a remote module.

Retrieving results of problem solving is performed by querying the conceptual schema of a problem. A query is a formula over the conceptual schema. A query is formulated over a RIF-document of a schema using the dialect of the document. The result is a Boolean value (if the formula is closed) or a collection of tuples of values of free variables of the formula.

2.2 Resources Relevance to a Problem and Mapping of their Schemas into the Conceptual Specification

In the frame of the proposed infrastructure (1) the logic programs implementing RIF-documents of the conceptual schema in specific RSs and (2) the subject mediators supporting collections of facts as the result of heterogeneous databases integration are considered as resources. Besides that the database of facts may be connected to RSs.

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

The RS of every resource R should be a *conformant* D_R *consumer*, where D_R is a RIF dialect. Conformance is formally defined using formula entailment and language mappings [14].

The resource R is *relevant* to a RIF-document d of a conceptual schema if:

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

The schema of a relevant resource is mapped into the subject domain specification. This means that conceptual entities referenced in the document d are expressed in terms of entities of the schema S_R using logic rules of the D_R dialect. These rules constitute separate RIF-document (Fig. 1).

---

[1] In this paper we do not consider methods for schema ontological matching [15].

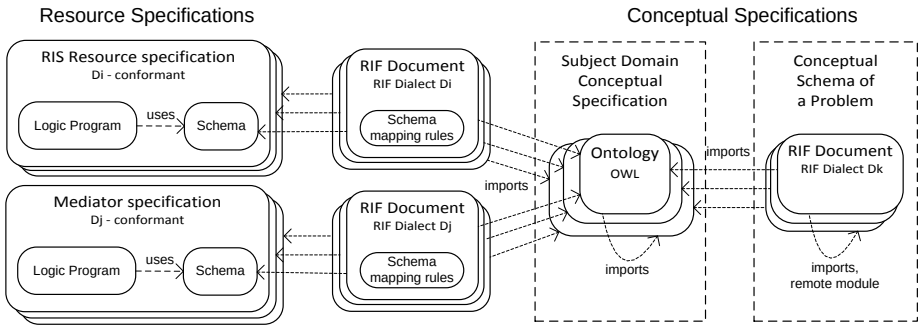

Fig. 1. Conceptual schema and resource specifications

2.3 Implementation of the Conceptual Schema Programs

Programs of the conceptual schema are implemented in P2P environment formed by relevant resources which are related to conceptual specification by mapping rules (Fig. 1).

Resources are peers (nodes) of the P2P environment. 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 a combination of a wrapper, a RS or a mediator, a logic program and possibly a collection of facts (Fig. 2). A wrapper transforms programs and facts from the supported RIF dialect into the language of the RS or mediator and vice versa. A wrapper also implements the delegation mechanism. A definition of the delegation is given in the latter part of this section. Transferring facts and rules among peers is performed using RIF dialects. Wrappers implement an interface *RIFNodeWrapper* (Fig. 2).

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

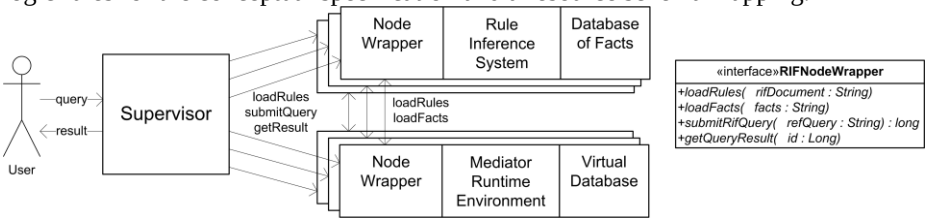

Fig. 2. Peer-to-peer multi-dialect network architecture

Implementation of the conceptual schema includes the following steps:

- Rewriting of the conceptual schema into the RIF-programs of resources. Rewriting is performed by the *Supervisor* component. A rewriting includes (1) replacing the document identifiers (used to mark predicates) by peer identifiers and (2) adding schema mapping rules to programs (Fig. 1).

- A transfer of the rewritten programs to peers containing resources relevant to the respective conceptual documents. The transfer is performed by the *Supervisor* component by calling the method *loadRules* of the respective node wrappers.
- A transformation of the RIF-programs into the concrete RS languages. The transformation is performed by the *NodeWrapper* component or by the RS itself (if the RS supports the respective RIF dialect).
- An execution of the produced programs on peers.

During the process of rewriting of the conceptual schema into the resource programs a structure of a real P2P network is formed. A virtual node corresponding to a RIF-document of the conceptual schema is replaced by one or more peers corresponding to resources relevant to the document. Relationships between virtual nodes defined by remote or imported terms are replaced by relationships between real peers also defined by remote or imported terms. To implement remote and imported terms a *rule delegation* mechanism is used. This mechanism is similar to one proposed in WebdamLog [16] (for more details look at the *Related Work* section).

In the general case, programs transferring 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 section. To make possible an execution of a programs on a peer 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*:

- *fact delegation rule* like *p@m(X) :- q@n(X)*, where *p, q* are predicate names, *X* is a variable list, *m* is a 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;
- *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 the *normalization* of a program *pr* on a peer *n* is the following. For every rule *r* like *head(r):- body(r)* of the program *pr*, if the *head(r)* of *r* is a remote term *p@m(X)*, then *r* is replaced by fact delegation rule *p@m(X):- p_m(X)* (where *p_m* is a new local predicate) and a rule *p_m(X) :- body(r)*. If the *head(r)* is a local predicate and the body *body(r)* of *r* contains an occurrence of a remote term *q@l(Y)*, then *r* is replaced by fact delegation rule *q_l(Y):- q@l(Y)* (where *q_l* is a new local predicate) and a rule *head(r) :- body(r)[q@l(Y)→q_l(Y)]*. In the body of the latter rule the occurrence of the remote term is replaced by a local term *q_l(Y)*.

The algorithm of execution of a program *pr* on a peer is the following. The program *pr* 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 may be obtained from other peers. When all facts are got they are loaded into the database of the peer with a program *pr*. Then local RIF-rules of the program are transformed into the resource rule language and executed in the RS resource or mediator runtime environment. The last step consists in transferring the required facts to other peers in accordance with the fact delegation rules.

3 The Use-case for the Multi-dialect Infrastructure

3.1 Investment Portfolio Diversification Problem

The capabilities of the multi-dialect architecture are illustrated with the solution of the *investment portfolio diversification problem* [17]. The portfolio is a collection of securities (such as equities or bonds), 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 are 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.

The input data to the problem is a set of securities and corresponding time series (such as closing price) for each security. Also the predetermined price correlation value is specified. It serves as maximum risk measure of a sharp reduction of the portfolio value. In practice, specified correlation may differ for various types of securities (for example, for stocks it is lower than for bonds). The output is the maximum size subset of securities, for which the pairwise correlation will be less than the specified one. The Pearson correlation r_{XY} between time series X and Y written as x_i and y_i (where $i = 1, \ldots, n$ and $\overline{x}$, $\overline{y}$ are means of X, Y) is used:

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

The problem is divided into the following tasks:

1. Computation of the security pairwise correlations (for specified dates).
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 stock prices, currencies, bonds, stock indexes, etc. Mediator integration environment is used to solve the problem of resource integration [3].

Second task can be formulated as follows. Let G be a graph where the vertices are the securities, and the edge between the two securities exists if absolute value of their correlation is less than a specified number. So, this is a well-known NP-complete problem – finding a maximum clique in an undirected graph. ASP logic programming systems, e.g. DLV [13], are well-suited for solving such problems [11].

3.2 Conceptual Specification of the Application Domain and Problem Schema

Application domain conceptual specification (ontology) of security historical prices is written in the OWL simplified[4] functional syntax [18]:

---

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

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

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

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

The *stockRates* class is used to denote securities, which are characterized by identifiers (attribute ticker) and time series of historical prices. 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 schema of the problem includes two 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. The document is defined in the RIF-BLD[5] dialect [8]:

```
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 ?tp ?symbol, ?ticker(
    Exists ?ts( And(?ts#gex:tickers  ?ts[symbol -> ?symbol]) ):-
      And(?t#srt:stockRates  ?t[ticker->?symbol])
    Forall ?m ?n ?c ?ticker1 ?ticker2 ?start ?end ?rates1 ?rates2
      ?dv1 ?dv2 ?date1 ?date2 ?series1 ?series2 ?corr (
      Exists ?e (
        And(?e#gex:noncorrelated ?e[start->?ticker1 end->?ticker2])):-
        ?m#srt:stockRates  ?m[ticker->?ticker1  rates->?rates1]
        ?n#srt:stockRates  ?n[ticker->?ticker2  rates->?rates2]
        ?dv1#?rates1  ?dv1[date -> ?date1]
        ?dv2#?rates2  ?dv2[date -> ?date2]
        External(pred:date-greater-than-or-equal(?date1  2012-01-01))
```

[5] Dialect is extended with the possibility to use the existential quantifier in the head of a rule.

```
          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:numeric-less-than(?ticker1  ticker2))
) ) )
```

The first rule of the document defines the predicate-collection *tickers*, which element attribute *symbol* 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*.

The second rule defines predicate *noncorrelated*, which is a noncorrelation relationship between securities. Object *?e* belongs to a collection *noncorrelated* if the absolute value of correlation between the prices time series of securities for given attribute values *e.start* and *e.end* is less than 0.25. 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 RIF-CASPD[6] dialect [10].

```
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))  ) )
```

The program defines a predicate *portfolio*, whose values are the security ids in the portfolio, and predicate *nonPortfolio*, whose values include all other securities under consideration. The first rule states that the only securities considered are securities which turn to truth predicate *tickers* in document *gex*. The second rule states that a securities can't simultaneously be in and not in the portfolio. The third rule claims that it is not possible for the pair of securities to be in the portfolio, but not in the graph of not correlated securities in document *gex*. Fourth rule is a weak constraint, which minimizes the number of securities not belonging to the portfolio.

A possible user query to the conceptual schema, namely, to the document *prt* is the following: *portfolio(?X)*. The query result is a collection of sets of identifiers of secu-

[6] Dialect is extended with operation :~ for a weak constraint. Such constraints should be satisfied if it is possible, but their violation does not invalidate the models [DLV].

rities. 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.

3.3 Resources of the Multi-dialect Infrastructure for the Problem

Resources for the problem of the investment portfolio diversification are subject mediator, in which the *Google Finance* and the *Yahoo! Finance* services are integrated, and a program in rule-based programming system DLV [13]. The mediator presents a virtual collection of facts, and DLV is a service for executing ASP-programs. Initially, the nodes do not contain logic programs.

Subject mediator schema implements the conceptual specification of application domain by expressing its semantics. Schema is written in the SYNTHESIS [2] - the canonical model of subject mediators:

```
{ FinanceServices; in: module;
  type:
  { DatedValue; in: type;
    date: {time; from: {yy}; to: {dd};};
    value: real;  };
  class_specification:
  { stockRates; in: class;
    instance_type: {
      ticker: string;
      rates: {set; type_of_element: DatedValue;}; }; };
  function:
  { correlation; in: function;
    params: {+s1/{set; type_of_element: DatedValue;},
           +s2/{set; type_of_element: DatedValue;}, -corr/real }; };
}
```

The schema includes a type *DatedValue*, a class *stockRates*, a function *correlation*. Their semantics corresponds to semantics of classes *DatedValue*, *stockRates*, and *correlation* belonging to *optimalSecurityPortfolio* ontology (section 3.2).

Integration of the resources in the mediator is provided using mappings that are logical rules binding mediator and service schemas. For instance, the *stockRates* class from the mediator is associated with the *google.finance.historicaldata* class from *Google Finance* with the rule shown on Fig. 3.

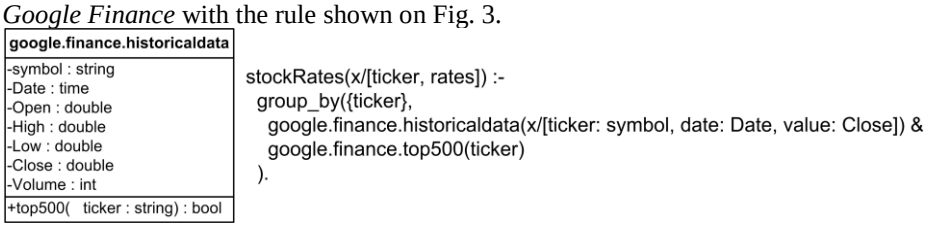

Fig. 3. Google *Finance* service schema and its mapping into the mediator schema

In this use-case the mediator class includes data about securities of companies belonging only to the S&P 500 stock market index. The S&P 500 is maintained by the Standard & Poor's, comprising 500 large-cap American companies. 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. Data from the resource class are grouped by the attribute *ticker* in the view. Attributes of the resource class are renamed properly in order to meet the structure of the mediator class.

Schemas of the financial services consist of a single type, which includes attributes that match the ID of the company, date and different indicators of the security price. Daily closing price of the shares (*Close*) is used to form the portfolio.

Entities of the mediator's schema and conceptual specifications are in one-to-one correspondence, the names and the semantics of the relevant entities are the same. Also the subject mediator is a conformant RIF-BLD consumer. Thus, the subject mediator *FinanceServices* is relevant to the RIF-document *gex* of the conceptual schema. Entities of mediator's schema are trivially mapped into the entities of the conceptual specification:

```
Document( Dialect(RIF-BLD)
  Import(<http://synthesis.ipi.ac.ru/optimalSecurutyPortfolio>
         <http://www.w3.org/ns/entailment/OWL-Direct>)
  Prefix(srt  <http://synthesis.ipi.ac.ru/optimalSecurityPortfolio>)
  Prefix(fsv  <http://synthesis.ipi.ac.ru/resources/FinanceServices#>)
  Group(
    Forall ?m (?m#srt:stockRates :- ?m#fsv:stockRates)
    Forall ?c (?c#srt:correlation :- ?c#fsv:correlation)  ) )
```

DLV resource is a conformant RIF-CASPD consumer, and its schema is empty - initially resource 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 resource is relevant to the RIF-document *prt* of the conceptual schema, since the relevance condition (Section 2.2) is reduced to the relation of dialects.

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

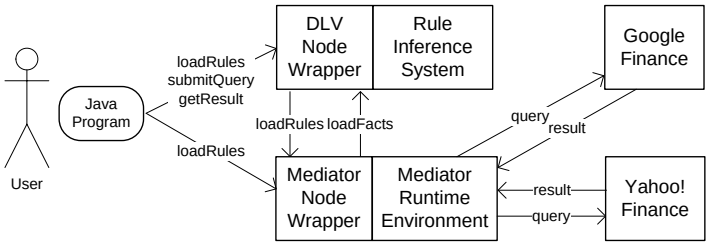

Fig. 4. Portfolio problem infrastructure

Prefixes of the documents of the conceptual schema in the rules are replaced with the actual node prefixes during rewriting of the conceptual programs into the pro-

grams over P2P network nodes. Thus, one of the rules of the *prt* document is 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 ?Y( :- And(dlv:portfolio(?X)  dlv:portfolio(?Y)
                     (Naf  noncorrelated@fsv(?X ?Y))) )  ) )
```

Other rules from both documents are rewritten similarly. The role of the supervisor is further functioned by a Java application, which sends rewritten programs to the nodes. Rewritten *gex* document is sent to the mediator node, rewritten *prt* document is sent to the *dlv* node.

After that node adapters automatically execute distributed programs in accordance with the algorithm described in section 2.3. First, program normalization is done. Thus, the non-local rule mentioned above

```
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 turn the *tickers* and the *noncorrelated* predicates to *true*. After receiving all necessary facts the program computing portfolio is executed on the *dlv* node.

Before the execution the normalized programs written in RIF dialects were automatically transformed by means of the ATL [19] into the logic languages supported on the nodes – the SYNTHESIS and the DLV, respectively.

The maximal models of the *portfolio* predicate, found during the execution of the program, are the diversified portfolios of maximal size containing securities of companies from the S&P 500. As the result of the program execution the 11 stable models containing 10 ground atoms each were generated. Several examples of the models can be found below. Atoms contain symbols of different companies, e.g. *duk* and *sbux* denote *Duke Energe* and *Starbucks Corp.* respectively. To save space we outline the models as sets of company identifiers (like *sbux*) related to the security ground terms in *portfolio(sbux)*. Note that models have nonempty intersections:

Model 1: {cah, duk, el, etr, hot, lm, psa, tjx, tyc, unh}

Model 2: {ba, bmy, duk, el, kss, lh, psa, stt, tjx, viab}

4 Related Work

There is an extensive literature on the use of database query languages for specifying declarative distributed programs and managing data in distributed environment [16], [20-22]. In contrast to multi-dialect 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 non-datalog constructs [22]. In the multi-dialect approach location is specified with RIF remote and imported terms.

Conceptual notion of *delegation* introduced in our approach coincides with 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, current implementation supports 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 rewrite step described in [22].

5 Conclusion

The approach presented is the first attempt of introducing the multi-dialect interoperable conceptual programming over various semantically different rule-based programming systems relying on the logic program transformation technique recommended by W3C RIF. We show also how to 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 planning aimed at reaching of 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 application domains.

References

1. Challenges and Opportunities with Big Data. A community white paper developed by leading researchers across the United States, 2012, http://cra.org/ccc/docs/init/bigdatawhitepaper.pdf
2. L. A. Kalinichenko, S.A. Stupnikov, D.O. Martynov. SYNTHESIS: A language for canonical information modeling and mediator definition for problem solving in heterogeneous information resource environments. — M.: IPIRAN, 2007. – 171 p.
3. Kalinichenko L.A., Briukhov D.O., Martynov D.O., Skvortsov N.A., Stupnikov S.A. Mediation Framework for Enterprise Information System Infrastructures. Proc. of the 9th International Conference on Enterprise Information Systems ICEIS 2007. - Funchal, 2007. -- Volume Databases and Information Systems Integration. - P. 246-251.

4. Kalinichenko, L.A. Methods and Tools for Equivalent Data Model Mapping Construction. In: Bancilhon, F., Tsichritzis, D., Thanos, C. (eds.) EDBT 1990. LNCS, vol. 416, pp. 92–119. Springer, Heidelberg (1990)
5. Kalinichenko L.A., Stupnikov S.A. Synthesis of the Canonical Models for Database Integration Preserving Semantics of the Value Inventive Data Models, ADBIS 2012, LNCS, vol. 7503, pp. 223-239, Springer, 2012
6. Fagin, R., Kolaitis, P.G., Miller, R.J., Popa, L.: Data exchange: semantics and query answering. Theoretical Computer Science 336, 89–124 (2005)
7. RIF Overview (Second Edition) W3C Working Group Note 5 February 2013, Eds. H. Boley, M. Kifer.
8. RIF Basic Logic Dialect (Second Edition). W3C Recommendation, 5 February 2013, Eds. H. Boley, M. Kifer.
9. RIF Core Logic Programming Dialect Based on the Well-founded Semantics, 2009. RuleML specification 13 August 2010, Ed. Michael Kifer, http://ruleml.org/rif/RIF-CLPWD.html
10. RIF Core Answer Set Programming Dialect / Eds. S. Heymans, M. Kifer. – 2009. – http://ruleml.org/rif/RIF-CASPD.html
11. M. Gelfond. Answer sets. In: Handbook of Knowledge Representation, Elsevier, 2008. – pp. 285-316.
12. RIF RDF and OWL Compatibility (Second Edition). W3C Recommendation, 5 February 2013, Eds. J. de Bruijn, C. Welty.
13. N. Leone, G. Pfeifer, W. Faber, T. Eiter, G. Gottlob, S. Perri, F. Scarcello. The DLV System for Knowledge Representation and Reasoning. ACM Transactions on Computational Logic 7(3), 2006. – pp. 499–562.
14. RIF Framework for Logic Dialects (Second Edition). W3C Recommendation, 5 February 2013, Eds. H. Boley, M. Kifer.
15. P. Shvaiko, J. Euzenat. A survey of schema-based matching approaches. Journal on Data Semantics, IV:146-171, 2005.
16. S. Abiteboul, M. Bienvenu, A. Galland, et al. A rule-based language for Web data management. In Proceedings 30th ACM Symposium on Principles of Database Systems, ACM Press, 2011. – pp. 283–292.
17. W. Sharpe, G. J. Alexander, J. W. Bailey. Investments. Prentice Hall. 1998.
18. OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax (Second Edition). W3C Recommendation 11 December 2012, Eds. C. Bock et. al.
19. ATL Project, http://www.eclipse.org/m2m/atl/
20. P. Alvaro, W.R. Marczak, et al. Dedalus: Datalog in time and space. Technical Report EECS-2009-173, University of California, Berkeley, 2009.
21. S. Grumbach and F. Wang. Netlog, a rule-based language for distributed programming. In M. Carro and R. Pena, editors, Proceedings 12th International Symposium on Practical Aspects of Declarative Languages, vol. 5937 of LNCS, pages 88–103, 2010.
22. B. T. Loo, T. Condie, M. Garofalakis, D. E. Gay, J. M. Hellerstein, P. Maniatis, R. Ramakrishnan, T. Roscoe, and I. Stoica. Declarative networking: language, execution and optimization. ACM SIGMOD Conference Proceedings, pages 97–108, 2006.