Multi-Dialect Workflows ⋆

Leonid Kalinichenko, Sergey Stupnikov, Alexey Vovchenko, and 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. The results presented in this paper contribute to the techniques for conceptual representation of data analysis algorithms as well as processes to specify data and behavior semantics in one paradigm. An investigation of a novel approach for applying a combination of semantically different platform independent rule-based languages (dialects) for interoperable conceptual specifications over various rule-based systems (RSs) relying on the rule-based program transformation technique recommended by the W3C Rule Interchange Format (RIF) is extended here. The approach is coupled also with the facilities for heterogeneous information resources mediation. This paper extends a previous research of the authors [1] in the direction of workflow modeling for definition of compositions of algorithmic modules in a process structure. A capability of the multi-dialect workflow support specifying the tasks in semantically different languages mostly suited to the task orientation is presented. A practical workflow use case, the interoperating tasks of which are specified in several rule-based languages (RIF-CASPD, RIF-BLD, RIF-PRD) is introduced. In addition, OWL 2 is used for the conceptual schema definition, RIF-PRD is used also for the workflow orchestration. The use case implementation infrastructure includes a production rule-based system (IBM ILOG), a logic rule-based system (DLV) and a mediation system.

Keywords: conceptual specification, workflow, RIF, production rule languages, database integration, mediators, PRD, multi-dialect infrastructure

1 Introduction

This work keeps on the intention of developing the facilities for conceptual declarative problem specification and solving in data intensive domains (DID). In [1] it was claimed that conceptual data semantics alone (e.g., formalized in ontology languages based on description logic) are insufficient, so that conceptual representation of data analysis algorithms as well as processes for problem solving are required to specify data and behavior semantics in one paradigm.

⋆ This research has been done under the support of the RFBR (projects 13-07-00579, 14-07-00548) and the Program for Basic Research of the Presidium of RAS.

The results presented in this paper extend the research [1] aimed at the definition and implementation of the facilities for conceptually-driven problems specification and solving in DID aiming at ensuring eventually the following capabilities for expressing the specifications:

1. an ability to provide complete and precise specification of the abstract structure and behavior of the domain entities, their consistency, relationship and interaction;
2. well-grounded diversity of semantics of the modeling facilities providing for the best attainable expressiveness, compactness and precision of the definition of the problem solving algorithm specifications;
3. arrangements for the extensions of the modeling facilities satisfying the changing technological and practical needs;
4. specification independence from implementation platforms (languages, systems);
5. specification independence from concrete information resources (databases, services, ontologies and others) combined with facilities for their semantic integration and interoperability;
6. built-in methodologies for creation of unifying specification languages providing for construction of semantics-preserving mappings of conceptual specifications into their implementations in specific platforms.

The research reported in [1] investigated the conceptual modeling facilities for DID applying rule-based declarative logic languages possessing different, complementary semantics and capabilities combined with the methods and languages for heterogeneous data mediation and integration. Two fundamental techniques were combined: (1) constructing of the unifying extensible language providing for semantics-preserving mapping into it of various information resource (IR) specification languages (e.g., such as DDL and DML for databases); (2) creation of the unified extensible family of rule-based languages (dialects) and a model of interoperability of the programs expressed in such dialects.

The first technique is based on the experience obtained in course of the SYNTHESIS language development [2]. 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 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 into the SYNTHESIS [2]. The canonical information model is constructed as a union of the kernel with such extensions defined for various resource languages. Canonical model is used for development of mediators positioned between the users, conceptually formulating problems in terms of the mediators, and distributed resources. A schema of a subject mediator for a class of problems includes the specification of the domain concepts defined by the respective ontologies.

Another, multi-dialect technique for rule-based programs interoperability applied is based on the RIF standard [3] of W3C. RIF introduces a unified family of rule-based languages together with a methodology for constructing of semantic preserving mappings of specific languages used in various Rule-based Systems (RS) into RIF dialects. Examples of RS include *SILK, OntoBroker, DLV, IBM Websphere ILOG JRules*, and others (more examples can be found at `http://www.w3.org/2005/rules/wiki/Implementations`). From the RIF point of view an IR is a program developed in a specific language of some RS.

In [1] the first results obtained were presented including the description of an approach and an infrastructure supporting:

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

The proof-of-concept prototype of the infrastructure based on the SYNTHESIS environment and RIF standards has been implemented. The approach for multi-dialect conceptualization of a problem domain, rule delegation, rule-based programs and mediators interoperability were explained in detail and illustrated on an use-case in the finance domain [1]. For the conceptual definition of the use-case problem the OWL was used for the domain concepts definition and two RIF logic dialects RIF-BLD [4] and RIF-CASPD [5] were used and mapped for implementation into the SYNTHESIS formula language and the ASP-based DLV [6] language respectively.

The results obtained so far are quite encouraging for future work: they show that the mentioned in the beginning capabilities (1-6) sought for conceptual modeling become feasible. This paper reports the results of extending the research in the direction of modeling of the processes for the problem solving following the approach briefly outlined above. These results include extensions of the infrastructure and specification languages considered in [1] to the workflow level keeping the same approach and paradigm as well as aiming at the capabilities of the conceptualization (1-6) that were stated in [1] and mentioned in the beginning of the introduction.

For investigation of such extension w.r.t. the choice of rule-based languages it was decided not to go outside the limits of the existing set of the published RIF dialects. Such decision would allow to retain well-defined semantics of the conceptual rule-based languages with a possibility to check preservation of their semantics by various languages of the implementing systems.

The production rule dialect RIF PRD [7] has been chosen as the language for the workflow modeling in such a way that the tasks of the workflow can have multi-dialect rule-based representation (as defined in [1]). This paper reporting the results of such investigation is structured as follows. To make the paper self-contained, the next section provides a brief overview of the infrastructure supporting multi-dialect programming defined in details in [1]. Here we stress that this infrastructure is suitable for the workflow tasks specification. Workflow-oriented extension of the multi-dialect infrastructure is considered in the section 3. Use case implementation in the proof-of-concept prototype is given in the section 4. Related works are reviewed in the section 5. Conclusion summarizes contributions of the research.

2 Basic Principles of the Workflow Tasks Representation in the Multi-Dialect Infrastructure

Every workflow task (besides those that for pragmatic reasons are defined as externally specified functions) is assumed to be represented in the novel infrastructure defined in details in [1]. Conceptual programming of tasks is performed using the RIF dialects (now not only logic but also production rule dialects can be used). Conceptual tasks are implemented by their transformation into the rule-based programs of the respective RSs and mediation systems (MSs). *Conceptual specification of a task* is defined in the context of a subject domain and consists of a set of RIF-documents (document is a specification unit of RIF). The *conceptual schema* of the domain is defined using OWL 2 [8] ontologies. Such usage of ontology is analogous to [9], however it is specifically important in the multidialect environment due to the formally defined compatibility between RIF and OWL. The ontologies contain entities of the domain and their relationships (Fig. 1, right-hand part). Conceptual specification of a task is defined over conceptual schema. Ontologies are imported into the RIF-documents specifying an import profile, for instance, *OWL Direct*. Documents *import* other documents having the same semantics (the *Import* directive), *link* documents defined using other dialects and having different semantics (remote module directive *Module*) or *refer* to entities contained in other documents using *external terms*.

Semantics of a conceptual task definition in such setting become a multi-dialect one. The specification modules of a task are treated as peers. Mediation modules are assumed to be defined in RIF-BLD for representation of the mediator rules (to be interpreted in SYNTHESIS) supporting schema mapping and semantic integration of the information resources. Multi-dialect task is implemented by means of transformation of conceptual specifications into modular, component-based P2P program represented in the languages of the mediation (MSs) and rule-based systems (RSs) with the respective semantics. Interoperability of logic rule components of such distributed program is carried out by means of the delegation technique ([1] section 3.3). Production rule components are considered as external functions, interoperability is achieved through the mechanism of external terms.

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 resource implementing the peer R. The RS or the mediation system (MS) of every peer R should be a conformant D_R consumer, where D_R is a respective RIF dialect (Fig. 1, left-hand part). Conformance is formally defined using formula entailment and language mappings [3].

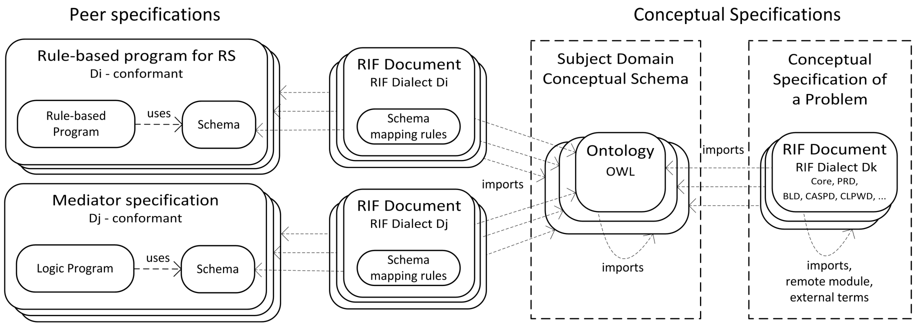

Fig. 1. Conceptual schema and peer specifications

The peer R is relevant to a RIF-document d of a conceptual specification of a problem (Fig. 1, right-hand part) if (a) D_R is a subdialect of the document d dialect (subdialect is a language obtained from some dialect by removing certain syntactic constructs and imposing respective restrictions on its semantics [4]; every program that conforms with the subdialect also conforms with the dialect) and (b) entities of the peer schema S_R (if they exist) are *ontologically relevant* 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 referred in the document d to their expressions in terms of entities of the schema S_R using rules of the D_R dialect. These schema mapping rules constitute separate RIF-document (Fig. 1, middle part).

Peers communicate using a technique for distributed execution of the rule-based 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 multi-dialect infrastructure. A node is a combination of a wrapper, an RS or an MS, and a peer (to save space, we refer a reader for the details to the paper [1], 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. Transferring facts and rules among peers is performed in the RIF dialects.

A special component (*Supervisor*) of the architecture defined in [1] 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 rules for the conceptual specification and a resource schema mapping.

Implementation of the conceptual specification includes the following steps:

– Rewriting of the conceptual documents into the RIF-programs of the peers performed by the *Supervisor*. The rewriting includes also (1) replacing the

document identifiers (used to mark predicates) by peer identifiers and (2) adding schema mapping rules to programs (Fig. 1, 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).
- An execution of the produced programs in P2P environment.

During the process of rewriting of 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 to transfer facts and rules from one peer to another. The details of *rule delegation* approach including description of the related algorithms are provided in [1].

3 Workflow-Oriented Extension of the Multi-Dialect Infrastructure

The aim of the infrastructure proposed is a conceptual programming of problems in the RIF-dialects and an implementation of conceptual specifications using rule-based languages of the RSs and MSs. One of the objectives of this particular paper is to introduce an extension of the existing multi-dialect infrastructure [1] aiming at the conceptual specification of rule-based workflows.

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. Besides the documents expressed in the logic dialects of RIF, the documents expressed in the production rule dialect (RIF-PRD) also can be a part of conceptual specification of a problem. In particular, these documents are aimed to express a process of solving the problem as the production rule-based workflow.

A workflow consists of a set of tasks orchestrated by specific constructs (*workflow patterns* [10], for instance *sequence, split, join*) defining the order of tasks execution. The specification of such orchestration is called here a *workflow skeleton.* A skeleton is defined using RIF-PRD production rules. Workflows and workflow patterns can be represented using production rules in various ways, e.g. as in [10][19]. The approach applied in this paper to represent workflows requires the extension of RIF-PRD dialect by several built-in predicates (they are considered to be a part of *wkfl* namespace referenced by `http://www.w3.org/2014/rif-workflow-predicate#` URI similarly to *func* and *pred* namespaces defined in [23] for built-in functions and predicates of RIF).

Predicates *wkfl:variable-definition* and *wkfl:variable-value* allow to specify workflow variables and their values and thus to organize the data flow within a workflow. Predicates *wkfl:parameter-definition* and *wkfl:parameter-value* allow to

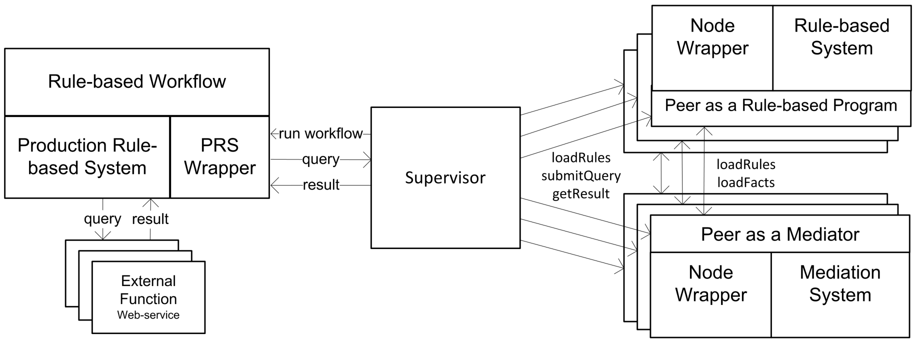

Fig. 2. Extended multi-dialect infrastructure

specify workflow parameters and their values and thus to define the interface of a workflow in terms of input and output parameters. Using of workflow parameters and variables is illustrated in the next section.

Predicate *wkfl:end-of-task(?arg)*, where *?arg* is an identifier of a task, turns into true if a task *?arg* has been completed. The predicate allows to orchestrate the order of execution of workflows tasks using conditions and actions of production rules. For instance, *AND-Split* workflow pattern is represented in RIF-PRD by the following production rule template using *wkfl:end-of-task* predicate:

```
If Not(External(wkfl:end-of-task(A)))
Then Do (Act(A)  Assert(External(wkfl:end-of-task(A))))
If And(Not(External(wkfl:end-of-task(B))) External(wkfl:end-of-task(A)))
Then Do (Act(B)  Assert(External(wkfl:end-of-task(B))))
If And(Not(External(wkfl:end-of-task(C))) External(wkfl:end-of-task(A)))
Then Do (Act(C)  Assert(External(wkfl:end-of-task(C))))
```

The template includes three rules for tasks A, B and C respectively. $Act(A)$, $Act(B)$ and $Act(C)$ denotes actions associated with tasks A, B and C. Orchestration (tasks B and C are executed concurrently right after task A is completed) is specified using *wkfl:end-of-task* predicate in conditions and *Assert* actions of rules. More complicated patterns like OR-, XOR- splits and joins, structured loops, subflows and others are represented in RIF-PRD similarly.

Workflow tasks can be specified as:

- separate RIF-documents in various logic RIF-dialects (this is the way how multi-dialect infrastructure [1] is extended with workflow capabilities);
- separate RIF-documents in the RIF-PRD dialect;
- set of production rules embedded into the workflow skeleton;
- external functions treated as "black boxes".

Semantics of tasks specified as multi-dialect logic programs are defined in accordance with the RIF-FLD [3] standard and standards for the respective RIF-dialects. Semantics of tasks specified as production rule programs are defined in accordance with the RIF-PRD standard. Semantics of external functions "are assumed to be specified externally in some document" [3].

All kinds of tasks (except those that are embedded into a workflow skeleton) are referenced in the workflow skeleton as *external terms* [3] like *External(t)* where term t is defined by an external resource identified by internationalized resource identifier (IRI) [3].

Workflows defined in the conceptual specification are implemented in the environment shown on Fig. 2. P2P environment [1] intended to implement logic programs is extended with a production rule-based system — PRS in short (for instance, a production system compliant with the OMG Production Rule Representation [18]) and with external functions, implemented as web-services. Implementation of the conceptual specification includes the following steps:

- Transfer of the conceptual RIF-documents constituting a workflow skeleton to the production rule-based system node (performed by the *Supervisor* component).
- Transformation of the conceptual RIF-documents constituting a workflow skeleton into the language of the production rule-based system (performed by the *PRS Wrapper* component).
- Transferring RIF logic programs related to tasks to the relevant nodes of the environment and transformation of the RIF-programs into the concrete RS or MS languages [1].
- Execution of the workflow.

The interface of the *Supervisor* includes methods for submitting and executing a workflow represented as a set of RIF-documents, and for getting the result of the workflow execution.

4 Multi-dialect Workflow Use-case

Motivation of the use case that illustrates the proposed approach comes from the finance area. The use case extends the investment *portfolio diversification problem* defined in ([1], Appendix) by adding workflow orchestration applying the production rules dialect RIF-PRD. The idea of the portfolio diversification problem is as follows. The portfolio is a collection of securities of companies, and its size is the number of securities in the portfolio. The problem 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 reduced.

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). The financial services Google Finance (`https://www.google.com/finance`) and Yahoo! Finance (`http://finance.yahoo.com/`) are considered. They include various indicators of the security price for all trading days of the last decades. For the diversified portfolio the securities having non-correlated time series should be used. Non-correlation of the time series means that their correlation is less than some predetermined price correlation value. The output data for the problem is a set of subsets of securities of the maximum size, for which the pairwise correlation will be less than the predetermined one.

The maximum satisfying subset of securities is calculated in the following way. Let G be a graph where the vertices are the securities. 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 non-correlated. In such case, the problem of finding the portfolio of the maximum size is exactly the problem of finding a maximum clique in an undirected graph. A maximal clique is a maximal portfolio. Note that several different maximal portfolios can be found.

The conceptual specification of the use case [1] used two RIF-dialects: RIF-BLD and RIF-CASPD. The use case was implemented in the environment containing a mediation system used as a platform for RIF-BLD [4] and ASP-based DLV system [6] – a platform for RIF-CASPD. The RIF-BLD was used to specify the problem of data integration, and RIF-CASPD – the problem of finding a maximum clique in an undirected graph.

The portfolio use case is extended in this work in the following way. The goal is not only to build a set of diversified portfolios, but to choose the "best" of them according to some criteria. There are several approaches to choose the most appropriate portfolio.

The most recognized one is based on the Markovitz portfolio theory [11]. The idea is to choose the portfolio, which has the maximum risk/return ratio. The most well-known metric to operate with risk/return is Sharpe-ratio [12]: $(r_p - r_f)/\sigma^2$. Here r_p denotes the expected return of the portfolio, r_f denotes a risk free rate, σ^2 denotes a portfolio standard deviation (risk). The more the Sharpe-ratio, the better the investment is.

Another approach is based on an idea that with the advent of social networks, it became possible to monitor ideas, sentiments, actions of people and lots of available information has to do with the markets and investments. In [13] Bollen et al. draw the connection between the mood of investor tweets and the move of Dow Jones Index, stating that correlation between them is more that 80%. The idea of using tweets to assess market movements has been implemented in several hedge funds.

Combining these two strategies could provide benefits of both of them, which leads to the following problem statement: having S&P500 (a stock market index maintained by the Standard & Poor's, comprising 500 large-cap American companies) list of companies, compute the diversified portfolio of maximum size with the best riskreturn and sentiment ratios. Fig. 3 demonstrates the workflow of the extended portfolio problem. It contains six tasks:

– *getPortfolios*. A set of diversified portfolio candidates is computed. The multi-dialect task specification consists of two RIF-documents in BLD and CASPD dialects ([1], Appendix). Portfolios received as a result contain only

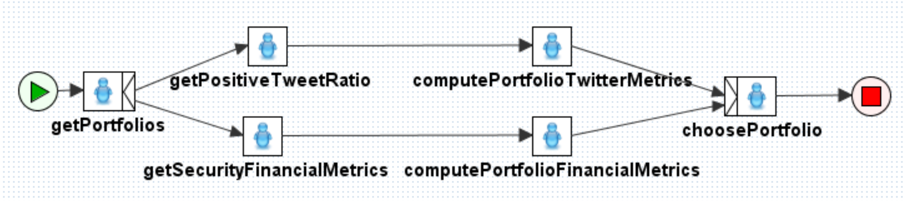

Fig. 3. Portfolio workflow

security tickers, they have to be augmented by financial and sentiments ratios.

- *getPositiveTweetRatio*. This task is responsible for computing a sentiment ratio of tweets for every security. Every tweet is assessed to be positive, negative or neutral. The task is specified as a call of external function.
- *computePortfolioTwitterMetrics*. The portfolio sentiment ratio is computed as the average of its securities sentiment ratio. The task is specified using RIF-PRD.
- *getSecurityFinancialMetrics*. For every security in a portfolio the financial rates (the *expected return* and the *standard deviation*) are calculated on the basis of historical rates of securities specified as an OWL 2 class of the ontology of the application domain. The task is specified using RIF-BLD dialect.
- *computePortfolioFinancialMetrics*. The computation of the portfolio expected return, risk, and Sharpe-ratio is done within this task. The task is specified using RIF-PRD dialect.
- *choosePortfolio*. The best portfolio is chosen according to maximizing the (*Sharpe ratio* $*$ *sentiment ratio*) product. The task is specified using RIF-PRD dialect.

Workflow skeleton is specified as a RIF-PRD document importing the ontology of the application domain. To save space we provide below the orchestration rules only for the task *getPortfolios*:

```
Document( Dialect(RIF-PRD)
 Import(<http://synthesis.ipi.ac.ru/portfolio/ontology#>
        <http://www.w3.org/ns/entailment/OWL-Direct>)
 Prefix(ont  <http://synthesis.ipi.ac.ru/portfolio/ontology#>)
 Prefix(svc  <http://synthesis.ipi.ac.ru/portfolio/services#>)
Group 2 ( Do(
Assert(External(wkfl:parameter-definition(startDate xsd:string IN)))
Assert(External(wkfl:parameter-definition(endDate xsd:string IN)))
Assert(External(wkfl:variable-definition(ps List<ont:Portfolio> IN))))
Group 1 (
Forall ?sd ?ed such that( External(wkfl:parameter-value(startDate  ?sd))
 External(wkfl:parameter-value(endDate  ?ed))  )
( If Not(External(wkfl:end-of-task(getPortfolios)))
```

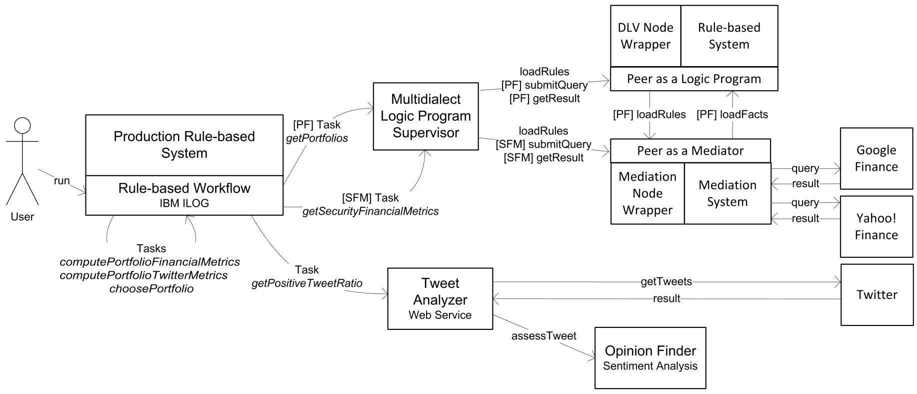

Fig. 4. Portfolio problem implementation structure

```
Then
Do( Modify(External(
       wkfl:variable-value(ps External(svc:getPortfolios(?sd ?ed)))
Assert(External(wkfl:end-of-task(getPortfolios)))  )  )  )
```

Production rules of the document are divided into two groups. The first group with priority 2 contains rules defining workflow parameters and variable. Parameters are *start date* and *end date* of historical rates used for calculation of *portfolio metrics*. Workflow variable *ps* denotes a set containing *portfolio candidates*.

The second group with priority 1 contains the orchestration rules – workflow skeleton. The only orchestration rule provided in the example above corresponds to the task *getPortfolios*. The external function *getPortfolios* encapsulates a multi-dialect logic program calculating portfolio candidates ([1], Appendix). A *Modify* action is used to call the function and to put the returned result into the *ps* variable.

The implementation structure of the use case is shown on Fig. 4.

The RIF-PRD workflow skeleton was transformed into a program in the ILOG [14] language combining production rules and workflow facilities (like *fork* and *sequence*). The ILOG program was executed in the *IBM Operational Decision Manager* tool.

The *computePortfolioTwitterMetrics*, *computePortfolioFinancialMetrics*, and *choosePortfolio* tasks are implemented as production rules in ILOG.

The *getSecurityFinancialMetrics* task uses the same instance of the mediation system as the *getPortfolios* task. The reason is that financial metrics are calculated using the historical rates of the securities. This is exactly the information that is extracted by the mediation system from *Google Finance* and *Yahoo! Finance*. The difference between two tasks is that the *getPortfolios* is implemented as a submission of a query to the DLV node, but the *getSecurityFinancialMetrics* is implemented as a submission of a different query to the Mediation Node.

The *getPositiveTweetRatio* task is implemented as a Java-program wrapped by a web service. First it collects tweets using the *Twitter Streaming API*. After that a sentiment analysis is done by the *Polarity Classifier* of the *OpinionFinder* tool [15] which assesses if tweet is positive, negative or neutral. Finally the sentiment ratio for every security in a portfolio is calculated.

Detailed specifications of the use case including ontologies, logic programs, production rules, workflow specification and implementation are provided in Appendix of [1] and in a technical report [16]. The technical report includes also results obtained by one of the workflow runs.

5 Related Work

Two types of workflow models, namely abstract and concrete were identified [17]. In the abstract model, a workflow is described in an abstract form, without referring to specific resources. In this paper we propose workflow representation in abstract and platform independent (PIM) form.

A classification model for scientific workflow characteristics [10] contributes to better understanding of scientific workflow requirements. The list of structural patterns discovered during this analysis (including sequential, parallel, parallel-split, parallel-merge, mesh) influenced our choice of the workflow patterns.

The OMG standard [18] reflects an attitude to production rules from the industrial side providing an OMG MDA PIM model with a high probability of support at the PSM (Platform-Specific Model) level from the rule engine vendors. Similar capabilities though formally defined are used as the basis for the production rule dialect RIF-PRD [7].

Some vendors of such production rule engines have extended their languages with the workflow specification capabilities. IBM has extended ILOG to provide the ruleflow capability. Microsoft supports Windows Workflow Foundation as a platform providing the workflow and rules capabilities. Examples of specific formalisms for PIM rule-based process specifications are provided also in [19].

Comparing to the known variants of the PIM production rule representations, selection of the RIF-PRD production rule dialect we consider well-grounded: (1) the RIF-PRD is formally defined; (2) RIF ensures support of interoperability of modules written in different rule-based dialects with different semantics; (3) RIF provides foundations for PIM to PSM semantic preserving transformation; (4) RIF also pro-vides an ability for specification of the concepts in application domain terms combining rule-based specifications with the OWL ontologies.

Importance of providing the inter-dialect interoperation is advocated in [20] for combining the functionalities of production systems and logic programs for abductive logic programming (ALP). The ALP framework gives a model-theoretic semantics to both kinds of rules and provides them with powerful proof procedures, combining backward and forward reasoning.

Papers related to RIF-PRD experimentations are focused mainly on the issue of the PRD programs transformation to an implementation system. In [21] a case study of bridging the ILOG Rule Language (IRL) to RIF-PRD and vice versa is considered. In [22] implementation of RIF-PRD in three different paradigms: Answer Set Programming, Production Rules and Logic Programming (XSB) is investigated.

The contribution of this paper w.r.t. previous works of the authors [1] consists in extensions of the infrastructure and specification languages considered in [1] to the workflow level.

6 Conclusion

Progress in the investigation of the infrastructure [1] for the conceptual multi-dialect interoperable programming in the abstract, rule-based, platform independent notations is reported. We present an extension of the coherent combination of the multi-dialect rule-based programming technique recommended by the W3C RIF with the approach for unifying modeling of heterogeneous data bases for their semantic mediation. The extension of the infrastructure and specification languages considered in [1] in the direction of the workflow modeling is presented.

Sticking to the limits of the existing set of the published RIF dialects, we present a capability of the multi-dialect workflow support with the tasks specified in semantically different languages mostly suited to the task orientation. We present a realistic problem solving use case containing the interoperating tasks specified in several platform independent rule-based languages: RIF-CASPD, RIF-BLD, RIF-PRD. In addition, OWL 2 is used for the conceptual schema definition, RIF-PRD is applied for the workflow orchestration. The platforms selected for implementation of the tasks include: DLV, SYNTHESIS, IBM ILOG. Such approach retains well-defined semantics of the platform independent rule-based languages with a possibility to check preservation of their semantics by various languages of the implementing systems. The principle of independence of tasks from the specific IRs is carried out by the heterogeneous database mediation facilitates contributing to the re-use of tasks and workflows. Alongside with the further extension of the approach, in the future work we plan to apply the conceptual multi-dialect programming philosophy for support of the experiments in data intensive sciences. In particular, we plan to investigate modeling hypotheses in astronomy representing them as a set of rules applying the multiplicity of the dialects required.

References

1. Kalinichenko, L. A., Stupnikov, S. A., Vovchenko, A. E., Kovalev, D. Y. Conceptual Declarative Problem Specification and Solving in Data Intensive Domains. Informatics and applications 7(4), 112–139, IPI RAN, Moscow (2013), `http://synthesis.ipi.ac.ru/synthesis/publications/13ia-multidialect`
2. 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.: IPIRAN, 171 p. (2007)

3. Boley, H., Kifer, M. (eds.): RIF Framework for Logic Dialects. W3C Recommendation, 2nd edn. (February 5, 2013)
4. Boley, H., Kifer, M. (eds.): RIF Basic Logic Dialect. W3C Recommendation, 2nd edn. (February 5, 2013)
5. Heymans, S., Kifer, M. (eds.): RIF Core Answer Set Programming Dialect. (2009), http://ruleml.org/rif/RIF-CASPD.html
6. Leone, N., Pfeifer, G., Faber,W., Eiter, T., Gottlob, G., Perri, S., Scarcello, F.: The DLV System for Knowledge Representation and Reasoning. ACM Transactions on Computational Logic 7(3), 499-562 (2006)
7. de Sante Marie, C., Hallmark, G., Paschke, A. (eds.): RIF Production Rule Dialect. W3C Recommendation, 2nd edn. (February 5, 2013)
8. Bock, C. et. al. (eds.): OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax. W3C Recommendation, 2nd edn. (December 11, 2012)
9. Calvanese, D. et al. Ontology-based database access. In: Proceedings of the Fifteenth Italian Symposium on Advanced Database Systems. pp. 324-331 (2007)
10. Ramakrishnan, L., Plale, B.: A Multi-Dimensional Classification Model for Scientific Workfow Characteristics. In: Proceedings of the 1st International Workshop on Workflow Approaches to New Data-centric Science. ACM, New York (2010)
11. Markowitz, H. M.: Portfolio Selection: Efficient Diversification of Investments. Wiley (1991)
12. Sharpe, W. F.: Mutual Fund Performance. J. Business 39 (S1): 119-138 (1966)
13. Bollen, J., Maoa, H., Zeng, X.: Twitter mood predicts the stock market. J. Comp. Sci. 2(1) (2011)
14. IBM WebSphere ILOG JRules Version 7.0. Online documentation. http://pic.dhe.ibm.com/infocenter/brjrules/v7r0/index.jsp
15. Wilson, T., Wiebe, J., Hoffmann, P.: Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis. In: Proceedinhgs of the conference on Human Language Technology and Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Stroudsburg, pp. 347–354 (2005)
16. Kalinichenko, L. A., Stupnikov, S. A., Vovchenko, A. E., Kovalev, D. Y. Multi-Dialect Workflows: A Use Case. Technical Report. IPI RAN, Moscow (2014), http://synthesis.ipi.ac.ru/synthesis/projects/RuleInt/Multidialect-Workflows-Use-Case.pdf
17. Yu, J., Buyya, R.: A taxonomy of scientific workflow systems for grid computing. ACM SIGMOD Records, 34(3):44-49 (2005)
18. Production Rule Representation (PRR), Version 1.0. OMG Document Number: formal/2009-12-01 (2009), http://www.omg.org/spec/PRR/1.0
19. Boukhebouze, M, Amghar, Y., Benharkat, A-N. and Maamar, Z.: A rule-based approach to model and verify flexible business processes. Int. J. Business Process Integration and Management, 5(4), pp. 287-307 (2011)
20. Kowalski, R., Sadri, F.: Integrating Logic Programming and Production Systems in Abductive Logic Programming Agents. RR 2009. LNCS, vol. 5837, Springer, pp. 1–23 (2009).
21. Cosentino, V., Del Fabro, M.D., El Ghali, A.: A model driven approach for bridging ILOG Rule Language and RIF. In: Proceedings of the 6th International Symposium on Rules, RuleML 2012. CEUR-WS.org, vol. 874, pp. 96-102 (2012)
22. Veiga, F. D. J. Implementation of the RIF-PRD. Master thesis. Universidade Nova de Lisboa (2011)
23. Polleres, A., Boley, H., Kifer, M. (eds.): RIF Datatypes and Built-Ins 1.0 W3C Recommendation, 2nd edn. (February 5, 2013)