Organization of Virtual Experiments in Data-Intensive Domains: Hypotheses and Workflow Specification

© Dmitry Kovalev © Leonid Kalinichenko © Sergey Stupnikov

Federal Research Center «Computer Science and Control» of Russian Academy of Sciences, Moscow, Russia

dkovalev@ipiran.ru lkalinichenko@ipiran.ru sstupnikov@ipiran.ru

Abstract. Organization and management of virtual experiments in data-intensive research has been widely studied in the several past years. Authors survey existing approaches to deal with virtual experiments and hypotheses, and analyze virtual experiment management in a real astronomy use-case. Requirements for a system to organize virtual experiments in data intensive domain have been gathered and overall structure and functionality for system running virtual experiments are presented. The relationships between hypotheses and models in virtual experiment are discussed. Authors also illustrate how to conceptually model virtual experiments and respective hypotheses and models in provided astronomy use-case. Potential benefits and drawbacks of such approach are discussed, including maintenance of experiment consistency and shrinkage of experiment space. Overall, infrastructure for managing virtual experiments is presented.

Keywords: virtual experiment, hypothesis, conceptual modeling, data intensive domains.

1 Introduction

Data intensive research (DIR) is evolving according to the 4th paradigm of scientific development and reflects the fact that modern science is highly dependent on knowledge extraction from massive datasets [5]. Data intensive research is multidisciplinary in its nature, bringing in many separate principles and techniques to handle complex data analysis and management. Up to 80% of researcher's time is spent on management of raw and analytical data, including data collection, curation and integration. The rest part requires knowledge inference from collected data in order to test proposed hypotheses, gather novel information and correctly integrate it. Although, it is the core of scientific work, it takes just 20% of researcher's time. To overcome that, a new approach for handling multidisciplinary DIR is needed.

Large-scale scientific experiments besides data processing issues are highly sophisticated– they include workflows, models and analytical methods. Every implementation of DIR can be treated as virtual experiment over massive collections of data. In [7] a survey is presented discussing different approaches to experiment modeling and how its core artifacts – hypotheses, can be specified. The use of conceptual representation of hypotheses and their corresponding implementation is emphasized, thus leading to the need of proper tools.

The article aims at developing methods and tools to support the execution and conceptual modeling of virtual experiment and designing infrastructure to manage it.

Article is structured as follows. In Section 2 related works are discussed. Section 3 explains why systems from section 2 are not enough, and introduces real-world use-case coming from astronomy. In section 4 main notions are defined. Section 5 provides infrastructure and functionality of system components is proposed. Section 6 concludes the article.

2 Related works

Systems with explicit representation of hypotheses are being rapidly developed during last several years [2–4, 6, 10]. Authors analyzed 3 different systems for executing virtual experiments and hypotheses: Hephaestus, Upsilon-DB and SDI. Some requirements for organizing and managing virtual experiments were extracted during the analysis. Although these platforms provide some important insights into defining and handling hypotheses, they miss some important features. First, they do not describe the perception of automatically derived hypotheses by domain experts, do not track their evolution, and do not discuss experiment design principles.

Hephaestus. It is a system for running virtual experiments over existing collections of data. It provides independence from resources and the system rewrites its queries into data source queries. System hides underlying implementation details from user, letting him work only with Hephaestus language. The language itself is a SQL-like language and is used to specify virtual experiment and underlying hypotheses.

Hephaestus separates two different classes of hypotheses: top-down and bottom-up. Top-down hypotheses are the one introduced by the researcher, while bottom-up hypotheses are derived from data. System supports the discovery of bottom-up hypotheses by looking for the correlation in data. These hypotheses are then ranked by some score (e. g. p-value of some statistical test) and the one with highest are passed to the researcher. Yet the system does not support automatical finding of causality, which is an important requirement for the future work. Hephaestus emphasizes the role of

Proceedings of the XIX International Conference "Data Analytics and Management in Data Intensive Domains" (DAMDID/RCDL'2017), Moscow, Russia, October 10–13, 2017

the expert in understanding which relationships should be further studied and which should not be chased. Hephaestus also computes metrics about experiments to estimate significance adequate to abandon further computation. System is used in testing clinical trials. The system does not catch the evolution of hypotheses or experiments yet.

Upsilon-DB. System enables researcher to code and manage deterministic scientific hypotheses as uncertain data. It uses internal database to form hypotheses as relations and adds uncertainty parameter. Later, that uncertainty parameter is used to rank hypotheses using Bayes rule. Provided approach can be treated as complementary to classical statistical approach. The systems allows to work with two types of uncertainty - theoretical, which is brought by competing hypotheses, and empirical uncertainty, which appears because of alternative datasets used. The system introduces algorithm to rank hypotheses using observed data. This is done because several competing hypothesis can explain the same observation well and some score to distinguish them is needed. When new data becomes available, this score can be adjusted accordingly.

Hypotheses have mathematical representation and authors provide method to translate its mathematical representation into relations in database. The simulations are also treated as data and respective relations are put inside the same database as hypotheses. Authors emphasize the need to support and develop the extraction of hypotheses from data and methods to sample both hypotheses and data. They illustrate that systems such as *Eureqa* [8] can be used to learn formula representation from data.

Following example is presented in the paper: authors present three different laws describing free fall and some simulated data. They rank hypotheses accordingly.

SDI. Platform is used to support scientific experiments. The system has the ability to integrate open data, reuse observed data and simulation data in the further development of experiments. The system enables multiple groups of researchers to access data and experiments simultaneously. Components of the framework are developed in such way that they could be deployed, adapted and accessed in individual research projects fast. SDI requires the support of lineage, provenance, classification, indexing of experiments and data, the whole cycle of obtaining data, curating and cleaning it, building experiments to test hypotheses over massive data, aggregating results is supported over long periods of time. The use of semantics is required by the system.

3 Astronomical Use-case

Surveyed systems do not cover several important issues, including interaction between hypotheses in single experiment, tracing experiment evolution, perception of automatically derived hypotheses and formulas by field experts.

Authors' further experience on how to deal with virtual experiments and hypotheses is based on *Besancon Galaxy Model (BGM)*. BGM is based on "the population synthesis approach … aims at assembling current scenarios of galaxy formation and evolution, theories of stellar formation and evolution, models of stellar atmospheres and dynamical constraints, in order to make a consistent picture explaining currently available observations of different types (photometry, astrometry, spectroscopy) at different wavelengths".

BGM which is being developed for more than 35 years represents a complex computational artifact, described in a series of [1, 11, 12]and presented in several major releases. Such a development represents a unique experience for catching the evolution scenarios for the model, changes to the model introduced both by using new observations (e.g. Hypparcos and Tycho-2 surveys) and the theoretical progress in the field. Both small changes to parameters of the model and huge improvements of the whole process were also made during the lifetime of the model. Also, the BGM authors enabled the community to change some parts of the model.

Due to the great experience collected by the BGM authors in the respective articles and associated code, now there is a possibility to collect the requirements for the system to supports experiments and provide rationale to choosing the appropriate methods and adequate techniques for the infrastructure.

BGM takes as input hypotheses and their parameters. The examples of such hypotheses are star formation rate (SFR), initial mass function (IMF), density laws, evolutionary tracks and so on [1]. As the model is evolving, new values for hypotheses parameters, even new parameters have been introduced into the BGM, e.g. for the IMF hypothesis in the last realization there has not only been tests of several new values of the hypothesis, but also separation of 2-slope and 3-slope instances of IMF is done.

It is very important to explicitly catch the relationship between several hypotheses in VE. Hypotheses and their parameters can be interrelated. For example, stellar birthrate function is derived from both IMF and SFR functions and local volume density function is based on provided density law. The relationships between hypotheses put constraints on the tuning of their parameters – model can quickly become.

Parameters of a single hypothesis can be linked to each other directly through equations. There are also indirect connections of parameters of several hypotheses, e.g. SFR parameter correlates with the slopes of IMF. This implies that one could not give the best solution for a particular variable without correlating it with others. So, there is a need to support for a correlation search between hypotheses parameters and to store relationships between parameters of a single hypothesis.

Not all model ingredients are allowed to be changed by the user. This is done because if some hypothesis is changed in the model and no further adjustments for the dependant hypotheses are made a model consistency is

broken. Furthermore, the model has a property of being self-consistent meaning that when input values change, if it is possible hypotheses derived by the one changed are properly adjusted in order not to break fundamental equations of astronomy. Therefore, derived by relationship needs to be modeled. Also, system component should enable the adjustment and calibration of any hypothesis available in the model.

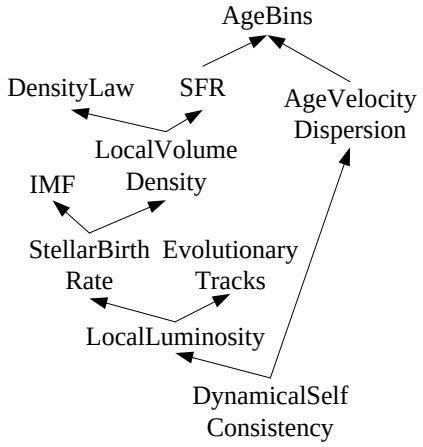

Figure 1 BGM Hypotheses Lattice with derived by relationship

Apart from explicit hypotheses, there are also implicit hypotheses in the model. They are not described in the articles and are tacit. The example of such hypothesis is that no stars come from outside of the Galaxy. It is important to explicitly store such hypotheses and understand how to extract such hypotheses from publications and data sources.

Workflow is used to implement BGM experiment specifying when each model which conforms to related hypotheses should be invoked. The workflow has also evolved since the first version, e.g. for thin disk treatment new activities dependent on IMF and SFR hypotheses are introduced. This development can only be tracked using publications. Some activities in model structure require the usage of statistical methods, tests and tools, which are used on both local hypotheses and on the general simulations from the whole experiment.

As the number of experiments is huge due to the increasing size of competing hypotheses family, now not all of the possible are run against the whole sky. Studying the ways to reduce the number of experiments which give the best fit and to choose when and if to abandon further computations of experiment is a major part of requirements to the new system. Using the information from experiment run done both locally and by other research groups can be helpful in achieving that goal.

Some researches of data-intensive analysis emphasize the role of error bars. As the data in astronomy is provided usually with errors, the BGM uses special methods to work with such type of uncertainty. A component supporting statistical tools which works with error bars is a major requirement for the infrastructure.

4 Hypotheses and Models in Virtual Experiment

4.1 Main Notions

Extracted information needs to be formally specified. For that, authors define additional artifact – virtual experiment. It is a tuple **<O, H, M, R, W, C>**, where **O** is a domain ontology. Domain ontology is a set of concepts and relationships in applied domain formally specified with some language.

H is a set of hypotheses specifications and relationships between them. H is a part of ontology and uses concepts from it. Together they form the ontology of virtual experiment. Hypothesis is a proposed explanation of a phenomenon that still has to be rigorously tested.

M is a set of models. Each model is a set of functions. Every model implements a hypothesis specification. If model generates expected behavior of some phenomenon, it is said that model and respective hypothesis are supported by observations.

R: H -> M, is a mapping from the set of hypotheses and into the models.

W is a workflow. Workflow is a set of tasks, orchestrated by specific constructs (workflow patterns - split, join, etc.). Each task represents a function with predefined signature, which invokes models from M. Workflow implements experiment specifying when each model that conforms to related hypotheses should be invoked.

C is a configuration for each experiment run. It consists of a total mapping from workflow tasks into sets of function parameter values.

There exist a lot of possible hypotheses representations – mathematical models, Boolean networks, ontologies, predicates in first-order logic, etc. Authors use ontologies to specify hypotheses.

Possible relationships between hypotheses are *competes_with*, which is used to relate competing hypotheses and *derived_by* to relate two hypotheses, one of which was used to derive another. *Derived_by*can be used to form hypotheses lattice [9] – algebraic structure with partial order relation. Hypotheses derived from a single hypothesis are atomic, otherwise – complex (see Fig. 1).

Model, which implements hypothesis, should conform to the hypothesis specification. If model generates expected behavior of some phenomenon, it is said that model and respective hypothesis are supported by observations.

4.2 Remarks on methodology

Since hypotheses become the core artifact of virtual experiment, there is a shift in treating data to successfully manage it. Fig. 2 depicts the process of specifying virtual experiment.

First, hypotheses are extracted from articles. Usually, it is text or formulas. Sometimes, there is a need to provide external hypotheses and substitute existing ones. Next step is to define mapping between hypotheses and models, which implement these hypotheses, and build some workflow specifying the sequence of tasks. Forming a research lattice is a next step. Virtual experiment needs configuration and execution plan. After that, one can launch virtual experiment.

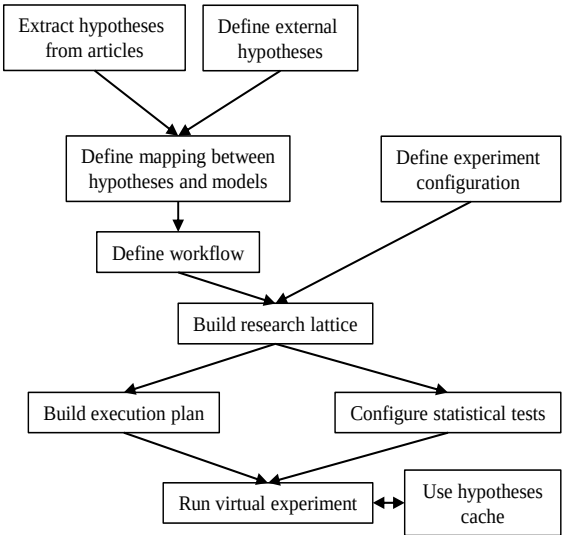

Figure 2 Methodology to form virtual experiment

4.3 Virtual Experiment Specification

Conceptual schema to define virtual experimentis provided. It is written with the simplified OWL functional syntax (*Declaration* keyword is omitted; property, domain, and range declarations are combined).Virtual experiment (*VirtualExperiment* class) has associated set of *hypotheses*, single *workflow*, *observed_data* against which experiment will run and *probability*, which describes how well underlying model suits observed data. Closer probability is to 1, better the underlying model simulates phenomenon.

```
Ontology(<http://synthesis.ipi.ac.ru/virtual_experiment/ontology>

Class(VirtualExperiment)

ObjectProperty(Hypothesis

domain(VirtualExperiment) range(Hypothesis))

ObjectMinCardinality(1 Hypothesis
     VirtualExperiment)

DataProperty(workflow domain(VirtualExperiment)
range(xsd:anyURI))

DataMinCardinality(1 workflow
     VirtualExperiment)

DataProperty(mediatordomain(VirtualExperiment)
range(xsd:anyURI))

DataMinCardinality(1 mediator
     VirtualExperiment)

     DataProperty(probability

       domain(VirtualExperiment)
range(xsd:float))

     DataExactCardinality(1          probability
     VirtualExperiment))
```

Hypothesis is specified in the same ontology as virtual experiment. Every hypothesis has *name*, *description*, *author(s)* and associated *articles*. It also has a model associated with it. Following [4] associated *probability* of hypothesis is introduced.

Several hypotheses explaining one and the same phenomena are called *competing*. Also hypothesis can be derived by some other hypothesis. Hypotheses lattice is formed with *derived_by* relationship on hypotheses space.

```
Class(Hypothesis)

DataProperty(probability domain(Hypothesis)
     range(xsd:float))

DataExactCardinality(1probabilityHypothesis)

DataProperty(name domain(Hypothesis)
     range(xsd:string))

DataExactCardinality(1name   Hypothesis)

DataProperty(description domain(Hypothesis)
     range(xsd:string))

DataMinCardinality(1descriptionHypothesis)

DataProperty(author
domain(Hypothesis)range(xsd:string))

DataMinCardinality(1authorHypothesis)

DataProperty(article domain(Hypothesis)
     range(xsd:anyURI))

DataMinCardinality(1article    Hypothesis)

DataProperty(model domain(Hypothesis)
Hypothesis)

DataExactCardinality(1 model range(xsd:anyURI))



Class(HypothesisMetaClass)

ClassAssertion(HypothesisMetaClassHypothesis)

ObjectProperty(competes
domain(HypothesisMetaClass)

range(HypothesisMetaClass))

ObjectProperty(derivedBydomain(HypothesisMetaClass)

range(HypothesisMetaClass))
```

4.4 Hypotheses Specification

Examples of hypotheses and their relationships come from Besancon Galaxy Model (BGM). For the sake of clarity not all hypotheses in BGM are specified. All of the BGM hypotheses are treated as subclasses of Hypothesis class.

Initial Mass Function is the mass distribution of a given population of stars and is represented by standard power law. Due to construction of the hypothesis in the BGMIMF has a mathematical representation as a piecewise function with 2 or 3 pieces (slopes) where it is defined for mass regions. As there are just 2 possible sizes of the piecewise function, we put this into two disjoint subclasses. There are restrictions on available mass to Sol mass ratio. For IMF, authors test 10 different

versions of a hypothesis, 4 of them are 2-slope functions and 6 of them are 3-slope function. All of tested hypotheses are competing. Example instance from each subclass is given.

```
Class(Slope)

DataProperty(alpha domain(Slope)
range(xsd:float))

DataProperty(minMass domain(Slope)
range(xsd:float))

DataProperty(maxMass domain(Slope)
range(xsd:float))

DataExactCardinality(1 alpha Slope)

DataExactCardinality(1 minMassSlope)

DataExactCardinality(1 maxMassSlope)

SubClassOf(IMF Hypothesis)

ObjectProperty(Slopes domain(IMF) range(Slope))

DataProperty(availableMass domain(IMF)
     range(xsd:float))

DataExactCardinality(1 availableMass  IMF )

DataProperty(outputStarMass domain(IMF)
     range(xsd:float))

DataExactCardinality(1 outputStarMass IMF )

SubClassOf(ThreeSlopeIMF IMF)

ObjectExactCardinality(3 Slopes  ThreeSlopeIMF)

SubClassOf (TwoSlopeIMF IMF)

ObjectExactCardinality(2 Slopes  TwoSlopeIMF )

DisjointClasses(TwoSlopeIMFThreeSlopeIMF)


ObjectPropertyAssertion(competes TwoSlopeIMF
IMF)

ObjectPropertyAssertion(competes ThreeSlopeIMF
IMF)

ClassAssertion(Slope HaywoodSlope1)

DataPropertyAssertion(alpha HaywoodSlope1
"1.7"^^xsd:float)

DataPropertyAssertion(minMass HaywoodSlope1
"0.09"^^xsd:float)

DataPropertyAssertion(maxMass HaywoodSlope1
"1.0"^^xsd:float)

ClassAssertion(Slope HaywoodSlope2)

DataPropertyAssertion(alpha HaywoodSlope2
"2.5"^^xsd:float)

DataPropertyAssertion(minMass HaywoodSlope2
"1.0"^^xsd:float)

DataPropertyAssertion(maxMass HaywoodSlope2
"3.0"^^xsd:float)

ClassAssertion(Slope HaywoodSlope3)

DataPropertyAssertion(alpha HaywoodSlope3
"3.0"^^xsd:float)

DataPropertyAssertion(minMass HaywoodSlope3
"3.0"^^xsd:float)

DataPropertyAssertion(maxMass HaywoodSlope3
"120.0"^^xsd:float)

ClassAssertion(ThreeSlopeIMFHaywoodIMF)

ObjectPropertyAssertion(Slopes
HaywoodIMFHaywoodSlope1)

ObjectPropertyAssertion(Slopes HaywoodIMF
HaywoodSlope2)

ObjectPropertyAssertion(Slopes HaywoodIMF
HaywoodSlope3)
```

Star Formation Rate, $\Psi(t)$ represents the total mass of stars born per unit time per unit mass of Galaxy. Star formation rate has subclasses for representing constant $\Psi(t) = C$ and exponential function $\Psi(t) = \exp\{-\gamma t\}$ where γ is a parameter. Authors tested several competing hypotheses - two possible values for gamma (0.12 and 0.25) and one constant value. They can be stated as instances of respective classes.

```
SubClassOf(SFR Hypothesis)

DataProperty(time domain(SFR) range(xsd:float))

DataExactCardinality(7 time  SFR )

DataProperty(bornStarMass domain(SFR)
     range(xsd:float))

DataExactCardinality(7 bornStarMass SFR )

SubClassOf(ConstantSFR SFR)

DataProperty(constant domain(ConstantSFR)
range(xsd:float))

DataExactCardinality(1 constant )

SubClassOf(ExponentSFR SFR)

DataProperty(gamma domain(ExponentSFR)
range(xsd:float))

DataExactCardinality(1 gamma)

DisjointClasses(ExponentSFRConstantSFR)

ObjectPropertyAssertion(competes ConstantSFR
SFR)

ObjectPropertyAssertion(competes ExponentSFR
SFR)

ClassAssertion(ExponentSFRRobinSFR)

DataPropertyAssertion(gamma RobinSFR
     "0.12"^^xsd:float)
```

BGM apart from model ingredients has also implicit hypotheses, which are not marked as ingredients. For example, 1) thin disk is divided into seven age bins; 2) no stellar population comes from the outside of the galaxy. For the first example we can specify additional class AgeBins which has exactly seven age bins.

```
SubClassOf(AgeBins Hypothesis)

DataProperty(ageBin domain(AgeBins)
range(xsd:integer))

DataExactCardinality(7 ageBin  AgeBins)
```

It is more difficult to deal with the second one. As a possible solution, additional hypothesis could later be specified.

Hypotheses lattice is modeled with derived Byobject property. Some classes can be specified using Equivalent Classes construction. Hypotheses lattice for BGM was created manually, but later it should be constructed automatically by system for executing experiments. (Part of) hypotheses lattice for BGM is shown in Fig. 1.

```
ObjectPropertyAssertion(derivedBy SFR AgeBins)

ObjectPropertyAssertion(derivedByAgeVelocityDis
persionAgeBins)

ObjectPropertyAssertion(derivedBy SFR
LocalVolumeDensity)

ObjectPropertyAssertion(derivedByDensityLawLoca
lVolumeDensity)
```

For IMF class and there are relations between slopes, output Mass and available Mass. Based on available Mass parameter alpha is chosen and then output Mass is computed. If available Mass is inside the respective interval, alpha is taken and output Mass is computed. Next, post-condition for ExponentSFR is written. It says that born stars should have mass respective to the exponential equation. Other pre- and post-conditions are specified in the same manner.

```
Document(

Group(Forall ?IMF  ?am ?s ?om ?a ?min ?max (

AND (?IMF[AvailableMass -> ?am Slopes -> ?s
outputStarMass ->

    ?om]  ?s[alpha -> ?a minMass -> ?min
maxMass -> ?max]) :-

  AND (External(pred:numeric-greater-than(?am

External(func:numeric-
multiply(?mincon:solMass)))

External(pred:numeric-less-than(?am
External(func:numeric-

multiply(?max con:solMass))))

))))

Forall ?ExponentSFR ?g ?t ?m(

?ExponentSFR[Gamma -> ?g Time->?t BornStarMass-
> ?m]:- AND(

External(pred:numeric-equal(?m

External(func:numeric-exponent(func:numeric-
multiply(

"-1.0"^^xsd:float?t)?g)))))))
```

4.5 Workflow Specification

The model of mass determination consists of a local mass normalization, the simulation of the local neighborhood and calculating vertical density distribution. These tasks can be further divided into several subtasks:

1. *getRSVDensity*. Relative density is calculated using Einasto density law. After that for each population this density ρ(r, l, b, i) is integrated in the vertical direction ratio of surface to volume density (RSV) is computed.
2. *getSurfaceDenisty*. For each thin disk subcomponent surface density is calculated and then summed. Surface density of each age subcomponent has to be proportional to the intensity of SFR in its respective age bin.
3. *getVolumeDensity*. Volume stellar mass densities are calculated and summed Total volume is checked to fit the observations.
4. *adjustSurfaceDensity*. If the difference occurs surface and volume density are adjusted and recomputed.
5. *getLNSimulations*. Provided with specific hypotheses (IMF, SFR, evolutionary tracks and so on) stars and their parameters are simulated in the local neighborhood.
6. *getAliveStarsRemnants*. Stars are splitted into alive stars and remnants. Remnants -are possible stars for which the age and mass combination was not on the evolutionary tracks.
7. *solvePotentialEquation*. Poisson equation is solved with the input of stellar content of thin disk.
8. *constrainPotential*. Calculated potential should be constrained by observed Galactic rotation curve. The central mass and corona parameters are computed in such a way that the potential reproduces the observed rotation curve.
9. *calculatePotentialParameters*. Based on the calculated potential central mass parameters and corona parameters are computed.
10. *solveBoltzmannEquation*. Boltzmann collisionless equation for an isothermal and relaxed stellar population is solved in order not to break fundamentals of the model.
11. *checkDynamicalConsistency*. As equations in 6,7,8 are solved separately, the potential does not satisfy both constraints. These tasks should be run until the changes in the potential and other parameters are less than 0.01.

Workflow is specified as a RIF-PRD document. The ontology for virtual experiment and BGM ontology are imported. Rules in the document are separated into two groups. The first group with priority 2 is used to define workflow input and output parameters and variables. Part of specification describes several hypotheses passed as input parameters and calculated local surface density for each age bin as output. GetLocalSurfaceDensity task is specified in a group with priority 1. Task gets as input SFR hypothesis and total surface density vector (initially a guess) and multiplies provided values. Task checks if Xor of dependent tasks is done.

```
Document( Dialect(RIF-PRD)

Base(<http://synthesis.ipi.ac.ru/virtualexperim
ent/workflow#>)

Import(<http://synthesis.ipi.ac.ru/virtualexper
iment/ontology#>)

Import(<http://synthesis.ipi.ac.ru/bgm/ontology
#>)Prefix(bgm<http://sy

nthesis.ipi.ac.ru/bgm/ontology#>)

Prefix(ve<http://synthesis.ipi.ac.ru/virtualexp
eriment/ontology#>)

Group 2 (

Do(

Assert(External(wkfl:parameter-
definition(sfrbgm:SFRIN)))

Assert(External(wkfl:parameter-
definition(imfbgm:IMF IN)))

Assert(External(wkfl:parameter-definition(avd
```

```
bgm:AgeVelocityDispersionIN)))
Assert(External(wkfl:parameter-definition(dl
bgm:DensityLaw IN)))
Assert(External(wkfl:parameter-definition(et
bgm:EvolutionaryTracks IN)))
Assert(External(wkfl:variable-
definition(lsdsList(xsd:float)
IN)))
Assert(External(wkfl:variable-definition(clsds
List(xsd:float)
      OUT)))
Assert(External(wkfl:variable-value(clsd
List())))
Group 1 (
Do(
Forall ?sfr?bsm?lsd ?lsds ?clsd ?clsds such
that (
External(wkfl:variable-value(lsds ?lsds))
External(wkfl:variable-value(clsds ?clsds))
External(wkfl:variable-value(sfr ?sfr))
      ?lsd#?lsds
      ?clsd#?clsds
      ?sfr[bornStarMass -> ?bsm]
( IfOr(Not(External(wkfl:end-of-
task(getRSVDensity)))
External(wkfl:end-of-
task(adjustSurfaceDensity)) )
Then Do( Modify(?clsd ->External(func:numeric-
multiply(?bsm
      ?lsd)) )
Assert(External(wkfl:end-of-
task(getSurfaceDensity))) ))
```

4.6 Choosing parameters of hypotheses for virtual experiment execution

Since some hypotheses can take quite a few values, the number of possible models can reach thousands. This poses a question about the order of model execution and how to make these executions effective (and not to recompute previous unchanged results). For that we use special structures to cache and store results. The system can put model execution in some order and use the results of previous executions. This could drastically increase the speed of model computation, especially on big amount of data. To implement this we use properties of hypotheses lattices.

The researcher can run several experiments finding the probability of each, which can be later queried by other researchers. For example, following query takes two experiments, which have underlying models best explaining observed data, and fixed values for hypothesis SFR and workflow specified by URI. Since there could be thousands of possible experiments, there is a need to order them by their probability. As in [3] we don't want the researched to bury in thousands of possible models and just take several best ones.

```
SELECT ?experiment
WHERE {
     ?experiment probability ?probability .
     ?experiment workflow ?workflow .
     ?experiment Hypothesis ?hypothesis .
     ?hypothesis name ?name .
     FILTER(?name = 'RobinSFR' && ?workflow =
URI)
}
ORDER BY desc(?probability)
LIMIT 2
```

5 Requirements for Infrastructure for Managing Virtual Experiments

In a series of experiment run it is important to keep track on evolution of models, hypotheses and experiments, as well as identifying new data sources.

Operations to manipulate virtual experiments and its components need to be defined. Next, the system needs to capture dependencies (competes, derived by) between hypotheses, invariants in single hypothesis. Correlations between parameters of several hypotheses should also be considered.

Second, infrastructure should contain components responsible for automatic extraction of dependencies between hypotheses, parameters in single and multiple hypotheses. Obtained data is used in deciding which experiments should be abandoned and also used in keeping hypotheses in a single experiment consistent.

Third, one needs components for maintaining experiment consistency and constraining the number of possible experiments as well as defining the metric which is used to define if experiment poorly explains phenomena and abandon further computations. Methods for removing poor experiments based on previous experiments runs are also required. Experiments and hypotheses should stay consistent when parameters of a hypotheses change.

As soon as several hypotheses in some experiments could explain some phenomena well and due to errors in data, researcher needs to deal with uncertainty and needs methods to rank experiments and competing hypotheses on massive datasets.

While experiment could change slightly from a previous experiment run (e. g. one hypothesis parameter changes), system should store some data about previous executions. Methods for understanding which parts of experiments should be recomputed and which are not should be developed as well. Defining structures to store results of previous experiments and query these results is important. Since there could be thousands of possible experiments system should use a method to form a plan to execute experiments in such way that stored results are mostly used and no additional recomputations are made.

Some stages will investigate and adopt or reject certain values such as a velocity hypothesis, then continue. The design of the paths to be followed is called experimental design that as in the scientific method is the hardest part of the analysis. In principle, as in many systems, Hephaestus could pursue multiple paths in parallel using some metric to determine when to abandon a path. Some have criticized DeepDive and others for following a single path.

Reducing computational experiments (what we call virtual experiments), as mentioned above using metrics to estimate significance adequate to abandon further computation.

6 Conclusion

The article aims at developing a new approach to managing virtual experiment. Hypotheses are becoming core artifacts of that approach. By analyzing existing systems and use case requrements are extracted. Formal specification of the determination of the mass model from BGM is presented in the OWL syntax.

Further work should be concentrated on developing metasystem for handling hypotheses, models and other metadata in virtual experiment.

Acknowledgments

This research was partially supported by the Russian Foundation for Basic Research (projects 15-29-06045, 16-07-01028).

References

[1] Czekaj, M. et al.: The Besancon Galaxy Model Renewed I. Constraints on the Local Star Formation History from Tycho Data. *arXiv preprint arXiv:1402.3257*. (Feb. 2014)

[2] Demchenko, Y. et al.: Addressing Big Data Issues in Scientific Data Infrastructure. Collaboration Technologies and Systems (CTS), 2013 International Conference on, pp. 48-55 (2013)

[3] Duggan, J., Brodie, M.L.: Hephaestus: Data Reuse for Accelerating Scientific Discovery. (2015).

[4] Gonçalves, B. et al.: Y-DB: A system for Data-driven Hypothesis Management and Analytics. (Nov. 2014)

[5] Hey, A.J. et al. eds.: The fourth paradigm: data-intensive scientific discovery. Microsoft Research Redmond, WA (2009)

[6] Kalinichenko, L. et al.: Rule-based Multi-dialect Infrastructure for Conceptual Problem Solving over Heterogeneous Distributed Information Resources. New Trends in Databases and Information Systems. Springer International Publishing, pp. 61-68 (2013)

[7] Kalinichenko, L.A. et al.: Methods and Tools for Hypothesis-Driven Research Support: A Survey. Informatics and Application, 9 (1), pp. 28-54 (2015)

[8] Ly, D.L.,Lipson, H.: Learning Symbolic Representations of Hybrid Dynamical Systems. J. of Machine Learning Research.,13, pp. 3585-3618, Dec (2012)

[9] Porto, F. et al.: A Scientific Hypothesis Conceptual Model. Advances in Conceptual Modeling, pp. 101-110. Springer (2012)

[10] Porto, F., Schulze, B.: Data Management for eScience in Brazil. Concurrency and Computation: Practice and Experience, 25 (16), pp. 2307-2309 (2013)

[11] Robin, A., Crézé, M.: Stellar Populations in the Milky Way-A Synthetic Model. Astronomy and Astrophysics, 157, pp. 71-90 (1986)

[12] Robin, A.C. et al.: A Synthetic View on Structure and Evolution of the Milky Way. Astronomy & Astrophysics, 409 (2), pp. 523-540. (2003)