Scientific Hypothesis Database

Fabio Porto and Ana Maria de C. Moura

LNCC – National Laboratory of Scientific Computing

DEXL – Extreme Data Lab

Petropolis, Brazil

{fporto,anamoura}@lncc.br

Abstract: New instruments and techniques used in capturing scientific data are exponentially increasing the volume of data consumed by in-silico research, usually referred to as data deluge. Once captured, scientific data goes through a cleaning workflow before getting ready to analysis that will eventually confirm the scientists hypothesis. The whole process is, nevertheless, complex and takes the focus of the scientist attention away from his/her research and towards solving the complexity associated with managing computing products. Moreover, as the research evolves, products of the exploration become an important source of provenance information and reuse. Based on these observations, we claim that in-silico experiments must be supported by a holistic hypothesis data model. The latter offers a data perspective for scientific hypothesis specification, and a declarative approach for expressing and running simulations. By confronting simulation results with phenomenon observations scientific hypothesis quantitative validation is achieved, guiding parameters tuning and, eventually, hypothesis evolution. This paper presents the hypothesis data model, shows how to run experiments by transforming declarative simulation expressions into a query evaluation plan and discusses the first implementation prototype using the query engine QEF, developed at the DEXL laboratory. A neuroscience scientific model illustrates the applicability of the data model.

Categories and Subject Descriptors: Hypothesis databases, scientific model management system, hypothesis evolution.

1. INTRODUCTION

The availability of important experimental and computational facilities nowadays allows many large-scale scientific projects to produce a never before observed amount of experimental and simulation data. This wealth of data needs to be structured and controlled in a way that readily makes sense to scientists, so that relevant knowledge may be extracted to contribute to the scientific investigation process.

Current data management technologies are clearly unable to cope with scientists' requirements [Stonebreaker et al 2009] despite the efforts the community has dedicated to the area. Such efforts can be measured by the community support to an international conference (SSDBM), running for almost 20 years on scientific and statistical database management, various workshops on associated themes, and important projects such as POSTGRES at Berkeley [Stonebraker and Rowe 1986]. All these initiatives have considerably contributed to extend database technology towards the support to scientific data management.

Giving such a panorama, one may argue about what is missing on the support to scientific applications from a database viewpoint. This paper contributes to this issue by arguing that scientists need an integrated environment for specifying, testing and evolving scientific hypotheses. The latter is a formal attempt to explain an observed phenomenon that can be experimentally validated to confirm or disprove it[1]. From a scientific exploration perspective, the

[1] http://en.wikipedia.org/wiki/Scientific_hypothesis, last access 04/09/2009.

statement of a scientific hypothesis delineates the problem being investigated by means of a causal relationship between an observed phenomenon and other ones trying to explain it. In computational modeling, such relationships are quantitatively expressed through scientific models that formally, most often mathematically, synthesize the understanding about the studied phenomenon. By its turn, *in-silico* experiments (i.e. simulations) require the transformation of scientific models into computational models that attempt to reproduce in software the scientific model, and ultimately the associated scientific hypotheses. Finally, scientists run experiments using computational models loaded with selected parameter values. Experimental results are, in the sequel, confronted with the corresponding phenomenon observation values quantitatively validating the hypothesis.

The described process highlights the different kinds of data produced during a *in-silico* scientific exploration. From the qualitative statement of a scientific hypothesis to its quantitative validation, a complex set of data and metadata are produced and put together as part of the explanation for the scientific exploration. Moreover, the evaluation of *in-silico* experiments involves running the specified computational models to transform and produce new data, a process very similar, but not identical, to query processing in databases. A scientific hypothesis database management system (SHDMS) is responsible for managing data, metadata and *in-silico* experiments associated to a scientific hypothesis. In addition, a SHDMS manages data and metadata associated with the phenomenon the scientific hypothesis attempts to explain. The confrontation of phenomenon observations with simulation results, obtained from a SHDMS, sets the basis for qualitative validation of scientific hypotheses in the context of the SHDMS.

Thus, in this paper a data model and simulation language for scientific hypothesis based exploration is presented. It extends the work in [Porto et al. 2007] that focused on the representation of scientific models by introducing scientific hypotheses, and by showing how this model can be integrated with the Query Engine Framework (QEF) for running experiments. A use case based on a neuroscience research is presented to illustrate the applicability of the model.

The remaining of this paper is structured as follows. Section 2 discusses related work. Section 3 defines the scientific model used to run simulations, and introduces a specific computational model that will be used throughout the paper. Section 4 presents the Hypothesis Data Model (HDM), and shows how to instantiate a scientific application example directly into it. Section 5 describes the framework developed to support hypothesis evaluation. Finally, section 6 concludes the paper with suggestions for future work.

2. RELATED WORK

Data and knowledge management supporting *in-silico* scientific research is a comprehensive topic that has appeared under the eScience label. It encompasses the semantic description of the scientific domain, the experiment evaluation through scientific workflow systems and result analysis through a myriad of different techniques, among other *in-silico* related tasks. Given the broad class of application domains that may benefit from eScience related data management techniques, it has been postulated that there is a small chance that a single solution would cover the diverse set of requirements coming from these domains [Stonebraker et al. 2009]. The semantic description of scientific domains through ontologies [Gruber 1995] has attracted the attention of the scientific community as a means to support collaboration through common conceptual agreement. In this line, GeneOntology[2] is probably the most notorious and successful example of practical adoption of ontologies in the scientific domain. Similarly, scientific

[2] http://www.geneontology.org/

workflows have become the de facto standard for expressing and running *in-silico* experiments, using execution environments, such as [Oinn et al. 2000], [Altintas et al 2004], [Porto et al. 2007]. Despite that, we believe that *in-silico* experiments require a more comprehensive model that can offer scientists a holistic view of his/her research. In particular, we propose a data model with which scientist may define scientific hypotheses, describe scientific models [Hunter 2006] and run simulations using computational models. Integrate hypotheses in a data model is, however, not trivial.

Hypothesis modeling has been introduced in databases back in the 80's [Bonner 1990]. In that context, one envisions a hypothetical database state, produced by delete and insert operations, and verifies whether queries are satisfied on that hypothetical state. This approach is, however, far from the experimental semantics settings that we are interested in. Closer to our objective is the logical model proposed in the context of the HyBow project [Racunas et al. 2004a],[Racunas et al. 2004b] for modeling hypotheses in the biology domain. Hypotheses (H) are represented as a set of first-order predicate calculus sentences with free quantifiers. In conjunction with an axiom set specified as rules that models known biological facts over the same universe and experimental data, the knowledge base may contradict or validate some of the sentences in H, leaving the remaining ones as candidates to new discovery. As more experimental data is obtained and rules identified, discoveries become positive facts or are contradicted. In the case of contradictions, hurting rules must be identified and eliminated from the theory formed by H.

The approach adopted by Hybrow supports hypothesis validation in the spirit of what we aim to represent, i.e., a formal definition to be confronted with experimental results and extending the scientific knowledge base. Nevertheless it does not entirely satisfy our requirements. In particular, the adopted model-theoretical approach for hypothesis validation does not seem adequate for representing hypothesis-oriented research that considers quantitative validation of simulation results. Moreover, our work aims at integrating hypotheses with the simulation environment in order to bridge the gap between qualitative and quantitative representation, and to the best of our knowledge, this is the first work that addresses this problem.

3. SCIENTIFIC DATA MODEL

A scientific hypothesis drives research by proposing an explanation for a studied phenomenon. Indeed, according to wikipedia[1], *a scientific hypothesis is used as a tentative explanation of an observation, but which has not yet been fully tested by the prediction validation process for a scientific theory. A hypothesis is used in the scientific method to predict the results of further experiments, which will be used either to confirm or disprove it.* Seen as explanations of phenomena, hypotheses should get as close to the object they explain as possible to the point where one may conceptually replace the other within a modeled distance. In this context, scientific hypotheses play a fundamental role in experimental science by bringing rigor to problem statement and results validation. In addition, *in-silico* experiments validate hypotheses through simulations, whose results, which we name here hypothesis' instances, associate quantitative values to hypotheses. In doing so, scientific hypotheses introduce a valuable contribution by bridging the gap between qualitative description of the phenomenon domain and the corresponding quantitative valuation. Thus, in our modeling approach we aim at coming up with a representation of scientific hypotheses that may be used in qualitative (i.e. ontological) assertions and with minimum tricks can be quantitatively confronted to phenomenon observations. For the sake of completeness, in the next section we revisit the scientific model data model, initially introduced in [Porto et al. 2008]. It serves as the basis for the hypothesis data model.

3.1 Scientific Model

The scientific model data model enables scientists to represent the knowledge associated to a scientific model, its associated computational model, in addition to expressing in-silico experiments. Moreover, the phenomenon being modeled is described and its observations registered.

3.1.1 Observed Phenomenon

The starting point of an in-silico scientific investigation comprehends the specification of the phenomenon one attempts to explain. It is described in the data model as:

Ph (Ph_{id}, OP_h, Title), where: (1)

- Ph_{id} is the phenomenon unique identifier;
- OP_h is a url of a domain ontology, setting the formal conceptual representation of the domain in which the phenomenon is inserted;
- Title assigns a label identifier to the phenomenon.

3.1.2 Scientific Model Example

A scientific model provides a comprehensive description of the scientist interpretation about the observed phenomenon. Of prime importance are a formal representation, possibly using mathematical formulae, and a reference to the phenomenon it attempts to explain. Considering that a phenomenon and its scientific models share the same domain, the phenomenon ontology is attributed to the scientific models (SM) description. In addition, the SM description includes bibliography references and other metadata supporting model presentation. In [Porto et al. 2008] a SM is formally defined, but due to space limitation it will be not presented here.

In order to illustrate a scientific model, consider the Hodgkin&Huxley model [Hodgkin and Huxley, 1952], a scientific model from the neuroscience domain representing the conductance on the membrane of a single neuron, whose formalization is given by the following mathematical equation:

$$\mathbf{I = m^3 h\, g_{Na}\ (E - E_{Na}\ ) + n^4\ g_K\ (E - E_K\ ) + g_L\ (E - E_L\ )},\ \text{where:} \quad (2)$$

g_i, i={Na, K, l} is a time-dependent variable that represents the membrane conductance for sodium, potassium and leakage; E_i models the equilibrium potential for each ion channel; E is the membrane potential; and, n, m and h are parameters controlling the probability of the sodium or potassium gates to be opened. The total ionic current across the membrane of the cell is modeled by the variable I.

3.2 Computational Model

A scientific model suggests a representation of a phenomenon using some formal language. In order to run *in-silico* experiments that simulate the referred phenomenon, a scientist builds a computational representation of the scientific model. Although desirable, an automatic mapping from a formal description of a SM to its computational model is still not feasible, and requires engineering efforts from the scientific group. Nevertheless, once a computation model (CM) has been specified and its software components developed, an engine may read such specification and automatically instantiate an execution on input data. The CM definition in our data model gathers required metadata for such automatic instantiation. Figure 1 illustrates the representation of a computational model implementing the HH scientific model (see equation (2)).

In this context, the Environmental ontology and the Domain ontology contribute to disambiguate CM specifications. The Environmental ontology describes the execution

environment associated to the CM, including: programming language specification, programming environmental parameters, libraries, documentation, input and output parameters, initialization procedure, and executing programs. Parameters that are read or produced by the simulation are structured into a XML document, whereas set-up values are expressed as attribute-value pairs.

In order to capture output produced by the underlying program, an *outputWrapper* class must be provided to transform the output into a set of attribute value pairs. The output produced by the wrapper is mapped into a XML structure.

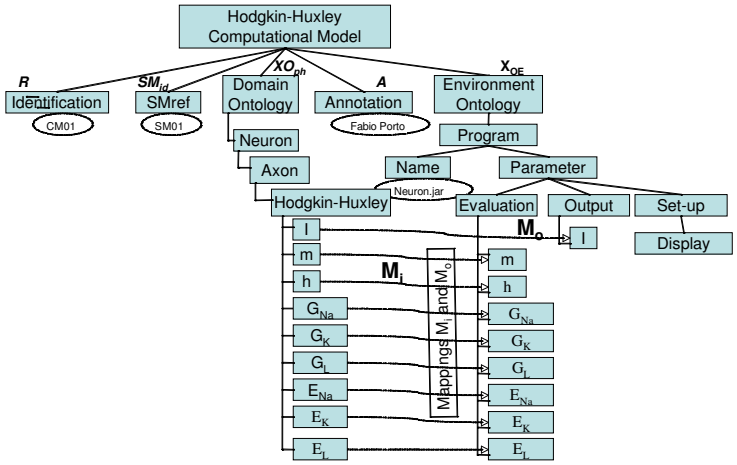

Fig. 1. A Hodgkin-Huxley computational model

A CM is formally defined as a 7-tuplet, such as:

$$\mathcal{CM}(\mathcal{CM}_{id}, \mathcal{SM}_{id}, \mathcal{XO}_E, \mathcal{XO}_{Ph}, \mathcal{M}_i, \mathcal{M}_o, \mathcal{A}), \text{ where:} \quad (3)$$

CM_{id} is the CM resource identification; SM_{id} is a reference to the associated scientific model; XO_E and XO_{Ph} are the XML serializations of the Environmental and Phenomenon ontologies, respectively[3]; M_i and M_o are the mappings between the underlying program input and output parameters (structured into two formats: XML and attribute-value pairs) and the corresponding domain ontology properties (XML tree leave nodes). Finally, A corresponds to annotations identifying authoring information.

3.3 Simulations

Simulations are *in-silico* experiments run to assess the quantitative distance of the scientific hypothesis with respect to the observed phenomenon. By analogy with databases, where users' data is intentionally expressed in queries, we call ***simulation query*** the specification of a simulation.

Let us define a simulation database $\mathcal{DB}_S = \{\mathcal{VCM}_1, \mathcal{VCM}_2, \ldots, \mathcal{VCM}_m\}$, where $\mathcal{VCM}_i$, $1 < i \leq m$, are n-ary data views on the computational model. Given a computational model CM_i, a corresponding n-ary data view $\mathcal{VCM}_i$ abstracts the software program behavior associated to the CM, by exposing its input and output parameters as data attributes and completely hiding its implementing programs. This is similar to modeling user-defined functions as relations in

[3] The serialization of Ontologies into an XML structure follows a detailed technique not presented here. The main intuition is to form a tree structure having concepts as nodes and aiding in semantically qualifying program's parameters.

databases [Chauduri and Shim, 1993]. Consider, for instance, the data view below corresponding to the CM of the HH model (Figure 1):

$$\textbf{HodgkinHuxley}(i:(m, n, h, g_{Na}, g_K, g_L, E_{Na}, E_K, E_L), o:(I)) \quad (4)$$

The HodgkinHuxley data view presents one attribute for each input/output CM parameter. Querying a data view requires binding attributes in the input parameter set (prefixed with i:) to input values and retrieving results associated to output parameters (prefixed with o:). In this context, a simulation query (S) interrogates a data view $\mathcal{VCM}_i$ by providing binding values and obtaining results from output attributes.

3.3.1 Simulation Language

In-silico experiments are commonly expressed using workflow or some sort of script languages. We aim to leverage the expression of simulations by providing a high-level query language with which scientists may express a large class of workflows, notably those that can be modeled as direct acyclic graphs.

In this context, a simulation query is specified as expression in non-recursive Datalog [Ullman, 1988] comprising a head and a body. The body is a boolean expression composed of a conjunction of predicates, whereas the head specifies a predicate holding literals and variables containing the expected simulation results, necessarily appearing in one of the predicates in the body. Users interface with simulation queries by providing the input parameters and set-up values needed for the evaluation of the predicates, and obtaining in return the output values in the head of the query.

3.3.2 Simulation Predicate

A simulation query predicate is specified as:

$$\mathbf{S_i}\,(\,(V_i,W_i)\,;\,(X_i',\,X_o')\,;\,(I_i,O_i)\,;\,I_S\,) \quad (5)$$

In (5), S_i labels the simulation query predicate and associate it with the corresponding CM data view ($\mathcal{VCM}_{id}$) resource identification. V_i and W_i are the two sets of variables defined to refer to values provided as input or produced as output when running the underlying CM program. The set of input and output parameter's values are provided by the XML documents X'_i and X'_o, respectively. Note that the associated CM definition specifies the schemas for X_i and X_o. For example, using the CM in Figure 1, the X_i document can be obtained from the result of the XPath expression "/CM/DomainOntology" over the Hodgkin&Huxley CM element XO_{Ph}, and by filling its leaf nodes with the input values. Thus, /Neuron/Axon/Hogking-Huxley/m = 0,1 illustrates a possible value assignment for the input parameter *m*. I_i and O_i are the mappings defining the correspondence (see definition 1) between the input and output variables in V_i and W_i and the input and output parameter values in X_i and X_o. Finally, I_S represents simulation set-up parameters.

Definition 1: Correspondence assertions in I_i and O_i are specified as $x = *Path*, where $x is a variable in $\{V_i \cup W_i\}$ and *Path* is an XPath[4] expression pointing to a data element in X_k', k={i,o}, whose leaf node is either an input parameter value or an output value.

3.3.3 Simulation Query

A simulation query combines the head and its body into a clause as illustrated in (6).

---

[4] http://www.w3.org/TR/xpath, last access 26/04/2008.

$$S(K) := S_1((V_1, W_1) ; (X_{i1}, X_{o1}) ; (I_1, O_1) ; I_{S1}) \wedge$$
$$S_2((V_2, W_2) ; (X_{i2}, X_{o2}) ; (I_2, O_2) ; I_{S2}) \wedge \quad (6)$$
$$\ldots. \wedge$$
$$S_n((V_n, W_n) ; (X_{in}, X_{on}) ; (I_n, O_n) ; I_{Sn})$$

An example of a simulation query is given in Figure 2. This particular query returns the total ionic current across the membrane ($I) according to the parameters values specified in the input document $HHCM01_I$. As discussed before, the user must provide a mapping from each query variable to the corresponding data element of the domain ontology XML serialization document. In this example, the input and output XML documents, $\mathbf{X_i}$ and $\mathbf{X_o}$, are illustrated by documents $HHCM01_I$ and $HHCM01_O$, respectively, both of type Neuron.

```
S($I, $z) := VCM01((i:($m,$h,$n,$gNa,$gK,$gL,$ENa,$EK,$EL),o:($I));
                    (HHCM01I, HHCM01O);
                  ( $m = /Neuron/Axon/ Hodgkin-Huxley/m,
                    ….[5],
                  $I = /Neuron/Axon/ Hodgkin-Huxley/I) )      ∧
              CM02[6] (($I , $z); (ACM02I, ACM02O);
                  ( $z=/Analysis/result) )
```

Fig. 2. A simulation query example.

4. HYPOTHESIS MODELLING

In order to integrate scientific hypotheses into the scientific model data model, we formally define a Hypothesis Data Model ($\mathcal{HDM}$), taking into account the following conceptual definitions:

Definition 2: phenomenon – represents a set of phenomena that scientists wish to explain;
Definition 3: scientific hypothesis – represents a set of explanations for a given phenomenon;

4.1 Hypothesis Data Model

A $\mathcal{HDM}$ describes an experiment domain and is defined as:

$\mathcal{HDM} = \{\mathcal{Ph}, \mathcal{PhO}, \mathcal{H}, \mathcal{E}, \mathcal{SM}, \mathcal{VCM}, \mathcal{SQ}\}$, where:

- $\mathcal{Ph}$ – is a set of phenomena, as defined in (1);
- $\mathcal{PhO}$ – is a set of phenomenon observations, which corresponds to a temporal recording of a phenomenon, quantitatively described by its attribute values. The specification in (1) for phenomenon observations is extended in the $\mathcal{HDM}$ as $\mathcal{ph_i}(ob_{id}, date, \mathcal{V}, \mathcal{U}, \mathcal{A})$, $\mathcal{ph_i} \in \mathcal{Ph}$, (7) where :

ph_i is a phenomenon observation set label; ob_{id} is an observation identifier; $\mathcal{V} = <v_1, v_2, \ldots, v_k>$; $\mathcal{U} = <a_1, a_2, \ldots, a_l>$), with $a_i, v_j \in \mathcal{D}_m, \mathcal{D}_n$, respectively, $\mathcal{D}_m, \mathcal{D}_n \subseteq \mathcal{D}$, for all $1 \leq i \leq k$, $1 \leq j \leq l$; $\mathcal{V}$ represents a list of initial set-up values and $\mathcal{U}$ a list of phenomenon comparable attributes. Finally, $\mathcal{A}$ is a list of annotations;

[5] The remaining mappings are not shown due to lack of space.
[6] The CM02 computational model has purposely not been described

- $\mathcal{H}$ – is a set of hypotheses, representing possible explanations of identified phenomena. The set of scientific hypotheses are specified as $\boldsymbol{h}$ (h_{id}, *title*, $\mathcal{P}h_{id}$, f, sq_{id}), (8) where:

h_{id} is an hypothesis identifier; *title* is a text describing the hypothesis; $\mathcal{P}h_{id}$ is a phenomenon identifier; f is a comparison function, used to measure the accuracy of hypotheses with respect to observations; and sq_{id} is an identifier for the corresponding simulation query schema as defined in (10). Observe that a simulation query establishes a consequent-antecedent relationship between the predicate appearing in the head of the clause with those in its body. The former provides the values for the corresponding experiment instance used in hypothesis validation, whereas the latter explains the hypothesis computation by means of a conjunction of other experiments;

- $\mathcal{E}$ – is a set of experiments. An experiment instance holds initial set-up values, in conformation with those of the phenomenon it attempts to simulate, and attribute values computed by the evaluation of the corresponding simulation query. These values are compared against phenomenon observation attribute values through a hypothesis comparison function. Experiments ($\mathcal{E}_i$) are formally specified as $\boldsymbol{\mathcal{E}_i}$ ($\mathcal{E}_{id}$, h_{id}, *date*, $\mathcal{X}_i$, *ob*, *dist*), (9) where:

$\mathcal{E}_i \in \mathcal{E}$; $\mathcal{E}_{id}$ is an experiment identifier; h_{id} is a scientific hypothesis identifier, as in (8); $\mathcal{X}_i$ is an XML document holding values for the underlying CM setup and input parameters; *ob* is the identification of a set of phenomenon observations defining the observations comparison set and *dist* is a measure of distance, computed by a hypothesis comparison function f between the experiment instance and the explained phenomenon observations *phj*. Finally,

- $\mathcal{SM}$ – is a set of scientific models;
- $\mathcal{VCM}$ – is a set of computational models data views;
- $\mathcal{SQ}$ – is a set of simulation queries used to compute experiments. Different experiments can be evaluated by reusing a stored simulation query and applying different parameter values to it. The simulation query set is specified as $\mathcal{SQ}$ (sq_{id}, *listVCM*, *headlist*, $\mathcal{X}$, *query-text*), (10) where: sq_{id} identifies the simulation query; *listVCM* is a list of computational model data views associated with the experiment; *headlist* is a list of variables in the head of the clause; $\mathcal{X}$ is an XML document structuring the set of setup and parameter elements appearing in the simulation query; and *query-text* includes the text associated with the query simulation.

4.2 A Running Example

Let us consider a small modification of the scenario presented in section 3.1.2. Suppose we want to feed a scientific visualization application with the temporal variation on the value of the ionic current (I), in other words, the ionic current is a function of time. The result is a time series showing the variation of the ionic current during an interval of time Δt. In addition, we will assume that independent scientific models are conceived to model the ionic current on each gate (i.e. sodium, potassium and leakage). In this revised scenario, the formulae in (2) can be re-written as:

$$\mathbf{I = \int_{(1\text{-}d)} m^3 h\,(E - E_{Na})\,g_{Na}\,(t)\,dt + \int_{(1\text{-}d)} n^4\,(E - E_K)\,g_K\,(t)\,dt + \int_{(1\text{-}d)} (E - E_L)\,g_L\,(t)\,dt} \quad (11).$$

In (11), d is the duration of the simulation and the membrane conductance g_i is a function of the simulation time instant. The ionic current on each gate is modeled by a different scientific model, leading to the following computational models, expressed by its computation model label and the corresponding mathematical formulae:

ionicChannel$_{Na}$: $(\int_{(1\text{-}d)} m^3 h\,(E - E_{Na})\,g_{Na}\,(t)\,dt)$;

$ionicChannel_K$: ($\int_{(1\text{-}d)} n^4 (E - E_K)\, g_K\, (t)\, dt$);
$ionicChannel_L$: ($\int_{(1\text{-}d)} (E - E_L)\, g_L\, (t)\, dt$).

Given this new scenario, a scientist may formulate the following hypothesis concerning Hodking-Huxley model: *The total ionic current on a membrane compartment is a time function of the ionic current on the sodium, potassium and leakage channels*.

In the following, we illustrate the use of the $\mathcal{HDM}$ model, $\mathcal{HDM} = \{\mathcal{Ph}, \mathcal{PhO}, \mathcal{H}, \mathcal{E}, \mathcal{SM}, \mathcal{VCM}, \mathcal{SQ}\}$, based on the example expressed in (11). Initially, we present the phenomenon being studied, represented by its schema definition and instance, respectively:

$\mathcal{Ph}$ ($\mathcal{Ph}_{id}$, $O_{\mathcal{Ph}}$, *title*), as in (1) and instantiated as
$\mathcal{Ph}_{totalionic current}$ (Ph_{TIC},O_{TIC},*"TotalIonicCurrent"*).

Next, we specify the phenomenon observations ($\mathcal{PhO}$)

$\mathcal{Ph}_i$ (ob_{id}, *date*,{<*m,n,h*>,<g_{NA}, g_K, g_L,*d*>< I_{NA}, I_K, I_L>, < *totalIonic*>}, $\mathcal{A}$), as in (7), where <*m,n,h*> are setup values; <g_{NA}, g_K, g_L,*d*> and <I_{NA}, I_K, I_L, *totalIonic*> are input and output parameters, respectively. Note that *d* refers to the observation duration in milliseconds. With respect to the example in (11), < *totalIonic*> corresponds to the comparable attribute of phenomenon Ph_{TIC} as in (7). It is instantiated as:
totalIonicCurrent (ob_1, *"01/03/2010", {*<m_1,n_1,h_1>,<0.039,6.0,2.73,1445>,<0.00025, 0.00040, 0.00050>,<0.001>},*"first measurement");* (12)
totalIonicCurrent (ob_2, *"03/03/2010",{* <m_1,n_1,h_1>,<0.039,6.0,2.73,1460>,<0.00028, 0.00037, 0.00048>, <0.002>},*"second measurement");* (12)

In (12), two observations of the phenomenon *totalIonicCurrent* are depicted. The observations include an identifier, the date of the observation, a set of initial state values, specifying the context on which the phenomenon was observed, and a comparable attribute that quantitatively describes it. The latter serves as the basis for assessing hypotheses.

A scientist formulates hypotheses that may or not be validated when compared to observations. Referring to our running example, the total ionic current phenomenon hypothesis ($\mathcal{H}$) would be specified as:

$\mathcal{h}$ (h_{id}, *title*, $\mathcal{Ph}_{id}$, *f*, sq_{id}) as in (8), instantiated as
$\mathcal{h}$ (h_1 , "*The total ionic current on a membrane compartment…*", Ph_{TIC}, f_i, sq_{id}); (13)

Observe that Ph_{TIC} identifies the phenomenon associated with this hypothesis. The scientific hypothesis text formulation appears as free text in the hypothesis definition. Additionally, sq_{id} identifies the simulation query that computes instances of hypothesis (i.e. experiments). The evaluation of an experiment tries to simulate the phenomenon Ph_{TIC}. In this context, an experiment instance set corresponds to the set of results obtained by evaluating the simulation query associated with the hypothesis, and forms the basis for quantitative hypothesis validation.

According to (9), the experiment instance would be represented as:

$\mathcal{E}_i$ ($\mathcal{E}_{id}$, h_{id} , *date*, {<*m,n,h*>,<g_{NA}, g_K, g_L,*t*>< I_{NA}, I_K, I_L>, < *totalIonic*>}, *ob*, *dist*), instantiated as

$E_{totalIonicCurrent}$ (e_1,h_1, *"01/04/2010", {*<m_1,n_1,h_1>,<0.039,6.0,2.73,1440>,<0.00028, 0.00037, 0.00048>,<0.003>},< ob_1, ob_2>,0.005) (14).

The computational models designed to simulate the *ionic channels* and to compute the *total ionic current* are exposed to evaluation through computational model data views (see section 3.3). Thus, in this running example the following data views with their schema are specified: *totalIonicCurrent* (*i*:($\$I_{NA}$, $\$I_K$, $\$I_L$),o:($\$totalIonic$)); $ionicChannel_{Na}$ (i:($\$t$, $\$m$, $\$h$, $\$g_{NA}$),o:($\$I_{NA}$)); $ionicChannel_K$ (i:($\$t$, $\$n$, $\$g_K$),o:($\$I_K$)); $ionicChannel_L$ (i:($\$t$, $\$g_L$),o:($\$I_L$)).

This set of data views are used in a simulation query to quantitatively compute the scientific

hypothesis that explains the *total ionic current* phenomenon. According to (10), the simulation query is expressed as:

$\mathcal{SQ}$(*sq$_{id}$, list$\mathcal{VCM}$, headlist, X, query-text*) instantiated as

$\mathcal{SQ}$(sq$_1$,{<*totalIonicCurrent, ionicChannel$_{Na}$, ionicChannel$_K$, ionicChannel$_L$*>}, {*totalIonic*}, {<*m, n, h, g$_{NA}$, g$_K$, g$_L$,t*>< *I$_{NA}$, I$_K$, I$_L$, totalIonic*>},"*totalIonicCurrent ($totalIonic)= totalIonicCurrent (I_{NA}$, I_K$, I_L$, $totalIonic)* $\wedge$ *ionicChannel$_{Na}$ ($t, $m, h, g_{NA}, $\underline{I}$_{NA})* $\wedge$ *ionicChannel$_K$ ($t, n, g_K,$\underline{I}$_K)* $\wedge$ *ionicChannel$_L$ (t, g_L, $\underline{I}$_L)*") (15);

In our example, evaluation between experiment observations and hypotheses is done as following: observation results (ob_1,ob_2) corresponding to *0.001* and *0.002* values in (12) are confronted with 0.003 in (14), and with the distance 0.005 value computed by the hypothesis comparison function f_i(13), provided by the scientist. This comparison result will show if the hypothesis simulation is approved or not. In negative case, new simulations can be specified.

The proposed model distinguishes three aspects of a scientific investigation: phenomenon experiment, simulation and formal representation. Table 1 presents a synthesis of the hypothesis data model classified according to these groups.

Table 1- Main elements to be considered in the $\mathcal{HDM}$

Phenomenon experiment	Phenomenon ($\mathcal{Ph}$)	Phenomenon observation ($\mathcal{PhO}$)			
Simulation	Hypothesis ($\mathcal{H}$)	Experiment ($\mathcal{E}$)	Computational Model ($\mathcal{CM}$)	Data View ($\mathcal{VCM}$)	Simulation query ($\mathcal{SQ}$)
Formal representation	Scientific Model ($\mathcal{SM}$)				

The elements of the hypothesis data model have been presented. In the next section, a discussion concerning the simulation query evaluation is presented.

5. HYPOTHESIS EVALUATION – FROM SIMULATION TO WORKFLOW EXECUTION

The quantitative validation of a scientific hypothesis is based on the evaluation of the associated experiment and achieved by comparing its results with the phenomenon observation comparable attribute values, as illustrated in Figure 3. In this section, the evaluation of hypotheses is discussed.

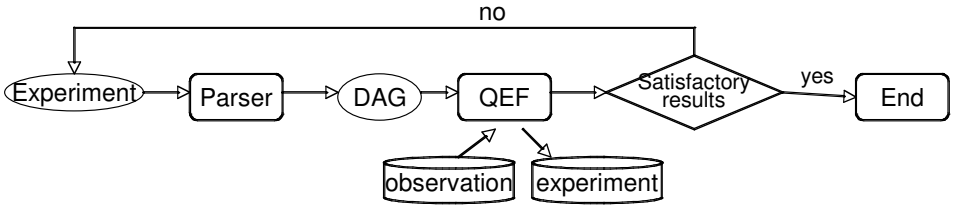

Fig. 3. Hypothesis evaluation process

QEF [Porto et al. 2007] is a software framework designed to support the evaluation of queries involving data transformation operations scheduled according to a data pipeline structure. Using QEF, non-typical database applications can be leveraged to a declarative invocation and take advantage of the optimizations already available in QEF, as for instance, operator parallelism. Data transformation operations are modeled as operators of a specific algebra and combined into

a valid expression in the form of a query evaluation plan (QEP) [Silberschatz et al. 2010]. Concerning the support for data representation, QEF makes it possible the definition of new data types and uses wrappers to adapt data-source data structures to a user data type embedded in a tuple envelope. Thus, operators read and write tuples that hold user data according to their data types. The combination of supporting specific algebra with the adaptation to data-source data structure produces a very powerful evaluation environment extensible for different application domains.

In the hypothesis data model, experiments are specified according to a simulation query. In this context, evaluating an experiment corresponds to evaluating the underlying simulation query with input parameters. Once a simulation query associates computational model data views in the form of a conjunction of predicates, its evaluation is equivalent to that of a conjunctive query, in which each CM data view predicate takes the form of a dependent-join operator [Florescu, D. et al. 1999]. The latter considers a relation with limited access to its data, only available through the binding of input values to query input variables. In the context of a CM data view evaluation, input parameters are bound to values, the CM is computed, and the resulting output is returned.

Completing the simulation query evaluation strategy, predicates share variables that establish a producer-consumer relationship. Given a variable in a simulation query, there is a single predicate in which it is either bound to a literal value or associated to an output parameter of the CM data view. All the remaining occurrences of the variable in the query consume its value. From a query evaluation point of view, such semantics implements a producer-consumer relationship between the predicate that associates a value to the variable and the others that read its content.

Thus, by modeling simulation query predicates as dependent-join operators and by establishing an evaluation order according to a producer-consumer relationship among predicates, a simulation query is modeled as a directed graph, in which nodes represent algebraic operators and directed edges define the producer-consumer relationship. Figure 4 depicts a directed graph representation of the simulation query in (15).

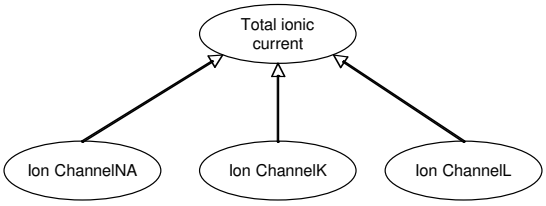

Fig. 4. A simulation query evaluation graph

5.1 Evaluating the Directed Graph Plan

Clearly, the graph in Figure 4 differs from traditional query evaluation plan, by presenting nodes with n producers, $n > 2$. In particular, if the CM data view *TotalIonicCurrent* is to be modeled as a dependent-join it should not be a ternary operator. Conversely, some graphs may present a 1xN producer-consumer relationship, in which the output of an operator is split into n consumers, $n \geq 1$. In order to support both data exchange models, two new operators are introduced: Merge and Split. These operators are classified as control operators [Ayres et al. 2003] (in opposition to algebraic operators) as their function is to control the dataflow between nodes, instead of applying data transformations. The Merge control operator behavior consumes one tuple from each of its producers, merging them into a single tuple and returning it to its consumer operator. As a result, for a single tuple request from its consumer the Merge operator submits one tuple request for each of its producers. The Split control operator offers the inverse

behavior. Observe that Merge and Split appear as patterns in workflow languages [Van Der Aalst 2003]. Figure 5 presents the graph of Figure 4 transformed into a query evaluation plan. Note that the bushy topology of the QEP in Figure 5 is an optimization strategy to enable running the CMs in parallel. A pipeline deep left topology is also viable. In this scenario each dependent-join is a producer for the subsequent one, eliminating the need fort the Merge operator.

Finally, simulation queries update the *Experiment* information with the input-output values and the distance between simulation comparable attributes and the phenomenon observations, computed by the function associated to the corresponding hypothesis, as illustrated in Figure 6.

The *Apply* operator invokes the user defined function *f* computing the distance between the *totalioniccurent* computed by the simulation and the set of phenomenon observations as specified in the hypothesis. The topology of the QEP in Figure 6 does not change for different experiments. Thus, it can be transformed into a template to be added to the top of the directed graph produced by analyzing the simulation query. In order to illustrate the template structure, a dashed box in Figure 6 delineates the part of the QEP to be inserted into a simulation query plan.

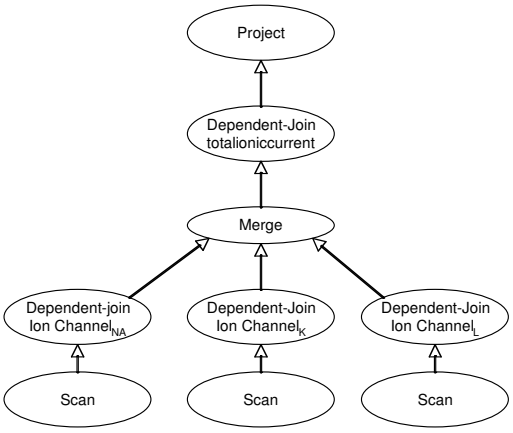

Fig. 5. Query evaluation plan

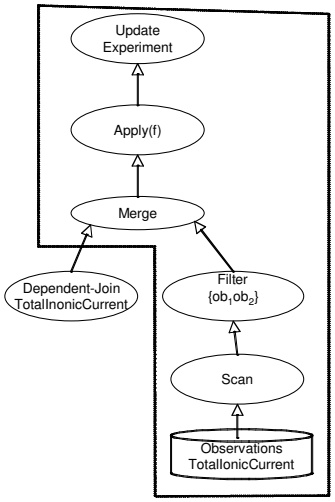

Fig. 6. Updating experiment

Once a QEP has been produced according to QEF language, the simulation can be run using the standard query execution iterator model [Graefe 2003].

6. CONCLUSION

Managing in-silico simulations has become a major challenge for eScience applications. As science increasingly depends on computational resources to aid solving extremely complex questions, it becomes paramount to offer scientists mechanisms to manage the wealth of knowledge produced during a scientific endeavor.

This paper presented a semantic based hypothesis model that aims at integrating scientific hypotheses and the computational models used to execute them, associated with the phenomenon scientists wish to explain. Scientific hypotheses are explanations of observable phenomena expressed through the results of computer simulations, which can be compared against phenomena observations. The model allows scientists to record the existing knowledge about an observable investigated phenomenon, including a formal mathematical interpretation of it, if one exists. Additionally, it intends to serve as the basis for the formal management of the scientific exploration products, as well as supporting models evolution and model sharing.

In order to illustrate the applicability of the model proposal, a hypothesis using a specific scientific model was formulated, from which some hypotheses were generated. The experiment instances resulted from the simulation queries applied over this model enabled to compare them with a set of experimental phenomena previously observed. A first prototype of the HDM together with the simulation query language were implemented on top of the QEF system to evaluate associated experiments. The system was designed in the context of a scientific model management system architecture with a set of minimal services that scientists may expect from such an environment.

There are various opportunities for future work. Qualitative modeling of hypotheses within an ontological context, hypothesis evolution and the adequacy of a scientific environment based on the hypothesis data model must be investigated. Moreover, the simulation query language enables a reduced set of scientific workflow constructs. Investigating the expressivity of datalog like rule languages for modeling scientific workflows is an interesting topic. Finally, there is a huge space for dynamic optimization of experiment evaluation.

Acknowledgments. This work has been partially supported by CNPq (Conselho Nacional de Pesquisa), Proc. 309502/2009-8 and 382.489/2009-8.

REFERENCES

AYRES F. V. M., PORTO F., MELO R. N.: An extensible machine to support new query evaluation models. SBBD 2003, Manaus, Brazil, pp. 371-380. (*in Portuguese*).

ALTINTAS I., BERKLEY C., JAEGER E., JONES, M. LUDASCHER B., KEPLER M. *An Extensible System for Design and Execution of Scientific Workflows***.** In *SSDBM*, 2004**.**

BONNER A. J. *Hypothetical Datalog: Complexity and Expressibility***.** Theoretical Computer Science 76 (1990), pp. 3-51, North-Holland. Racunas S.A., Shah N.H., Albert I., Fedoroff N.V. *Hybrow: a Prototype System for Computer-Aided Hypothesis Evaluation.* Bioinformatics, Vol.20, Suppl.1 2004a, pp. 257-264.

CHAUDURI S., SHIM K. *Query Optimization in the Presence of Foreign Functions***,** Proc. of the 19th Very Large Database Conference, Dublin, Ireland, 1993, pp. 529-542.

FLORESCU D., LEVY, A., MANOLESCU, I. AND SUCIU, D. *Query optimization in the presence of limited access patterns*, SIGMOD '99: Proceedings of the 1999 ACM SIGMOD international conference on Management of data, pp. 311-322, Philadelphia, Pennsylvania, US, 1999.

GRAEFE G., *Query Evaluation Techniques for Large Databases*. ACM Comput. Surv. 25(2): 73-170 (1993)

GRUBER T. R. *Toward Principles for the Design of Ontologies Used for Knowledge Sharing***.** Stanford University. International Journal of Human-Computer Studies 1995.

HODGKIN A., HUXLEY A. *A Quantitative Description of Ion Currents and its Applications to Conduction and Excitation in Nerve Membranes,* J. Physiol. (Lond.), 117:500-544, 1952.

HUNTER J. *Scientific Models – A User-oriented Approach to the Integration of Scientific Data and Digital Libraries***,** VALA 2006, Melbourne, February, 2006**.**

OINN T., GREENWOOD M., ADDIS M. *Taverna: Lessons in Creating a Workflow Environment for the Life Sciences***.** Concurrence Computation : Pract. Exper., 1-7, 2000**.**

POSPISCHIL M., TOLEDO-RODRIGUEZ M., MONIER C., PIWKOWSKA Z., BAL T., FRÉGNAC Y., MARKRAM H., DESTEXHE A. *Minimal Hodgkin-Huxley type models for different classes of cortical and thalamic neurons,* Biological Cybernetics, 99:427-441, 2008.

PORTO F., TAJMOUATI O. SILVA V.F.V, SCHULZE,B., AYRES F.M. *QEF Supporting Complex Query Applications*. 7th Int'l Symposium on Cluster Computing and the Grid, Rio de Janeiro, Brazil, pp. 846-851, 2007.

PORTO F., MACEDO J. A. F., TAMARGO J. S., ZUFFEREY Y. W., VIDAL V. P., SPACCAPIETRA S. *Towards a Scientific Model Management System*. ER Workshops 2008: 55-65.

RACUNAS S.A., SHAH N.H., ALBERT I., FEDOROFF N.V. *Hybrow: a Prototype System for Computer-Aided Hypothesis Evaluation*, Bioinformatics, Vol.20, Suppl.1 2004a, pp. 257-264.

RACUNAS S., GRIFFIN C., SHAH N. *A Finite Model Theory for Biological Hypotheses***.** Proc. of the 2004 IEEE Computational Sytems Bioinformatics Conferences, 2004b.

STONEBREAKER, M., BECLA, J., DEWITT, D., et al. *Requirements for Science Data Base and SciDB*. Conference on Innovative Data Systems Research, CIDR, 2009.

SILBERCHATZ A., KORTH H.F., SUDARSHAN S**.** *Database System Concepts*, McGraw-Hill, 6th edition, 2010.

TAMARGO J. S**.** *Scientific Model Computation.*. Master Project, EPFL, Switzerland, 2008.

ULLMAN J. *Principles of Database and Knowledge-Base Systems***.** Vol.1, Computer Science Press, 1988.

VAN DER AALST W.M.P., TER HOFSTED A.H.M., KIEPUSEZEWSKI, B., BARROS A.P**.** *Workflow Patterns,* Distributed and Parallel Databases, 14, 5-51, 2003.