Proceedings of the 8th International Conference on Ontologies, DataBases, and Applications of Semantics (ODBASE 2009), Portugal, pp.992-1009.

Ontology-driven Provenance Management in eScience: An Application in Parasite Research

Satya S. Sahoo[1], D. Brent Weatherly[2], Raghava Mutharaju[1], Pramod Anantharam[1], Amit Sheth[1], Rick L. Tarleton[2],

[1] Kno.e.sis Center., Computer Science amd Engineering Department, Wright State University, Dayton, OH 45435 USA, [2] Tarleton Research Group, CTEGD, Univeristy of Georgia, Athens, GA 30602, USA
{sahoo.2, mutharaju.2, anantharam.2, amit.sheth}@wright.edu,
{dbrentw,tarleton}@uga.edu

Abstract. Provenance, from the French word "*provenir*", describes the lineage or history of a data entity. Provenance is critical information in scientific applications to verify experiment process, validate data quality and associate trust values with scientific results. Current industrial scale eScience projects require an end-to-end provenance management infrastructure. This infrastructure needs to be underpinned by formal semantics to enable analysis of large scale provenance information by software applications. Further, effective analysis of provenance information requires well-defined query mechanisms to support complex queries over large datasets. This paper introduces an ontology-driven provenance management infrastructure for biology experiment data, as part of the Semantic Problem Solving Environment (SPSE) for *Trypanosoma cruzi* (*T.cruzi*). This provenance infrastructure, called *T.cruzi* Provenance Management System (PMS), is underpinned by (a) a domain-specific provenance ontology called Parasite Experiment ontology, (b) specialized query operators for provenance analysis, and (c) a provenance query engine. The query engine uses a novel optimization technique based on materialized views called materialized provenance views (MPV) to scale with increasing data size and query complexity. This comprehensive ontology-driven provenance infrastructure not only allows effective tracking and management of ongoing experiments in the Tarleton Research Group at the Center for Tropical and Emerging Global Diseases (CTEGD), but also enables researchers to retrieve the complete provenance information of scientific results for publication in literature.

Keywords: Provenance management framework, provenir ontology, Parasite Experiment ontology, provenance query operators, provenance query engine, eScience, Bioinformatics, *T.cruzi* parasite research

1 Introduction

Life sciences domain is witnessing an exponential increase in availability of scientific data through use of industrial scale experiment protocols, easy access to distributed data resources, and computational tools. This deluge of data will benefit the scientific community only if it can be effectively analyzed to gain new research insights. The correct interpretation of scientific results requires analysis of metadata describing how the data was generated, for example, the material and methods used, date on which the data was generated, and the research context. This category of metadata describing the history or lineage of a dataset is called provenance, which is derived from the French word "*provenir*" meaning "to come from". Provenance information enables validation of data, verification of the experiment protocols that generated the data, and association of trust values with scientific results.

Information provenance has been recognized as a hard problem in computing science [1], and there are many challenges being addressed by the provenance research community [2] [3]. For example, interoperability of provenance information from different sources is essential since integration of scientific results from disparate sources requires the analysis of their associated provenance information. Recent initiatives to create a common provenance model to facilitate interoperability include the upper level provenance ontology called *provenir* [4] and the open provenance model (OPM) [5]. Another important issue in provenance research is the ability to analyze the provenance information using robust query mechanisms. Provenance analysis will enable scientists to make informed decisions about the reliability of results from experiments. These challenges in provenance management has been the focus of extensive research efforts in the database [6], scientific workflow [7], and more recently in the Semantic Web community [3].

Recently, a provenance management framework (PMF), underpinned by Semantic Web standards, has been proposed to manage provenance information in large eScience projects [8]. The PMF consists of:

(a) The *provenir* upper level provenance ontology modeled in OWL-DL [9]. *Provenir* ontology was developed using the Open Biomedical Ontologies (OBO) foundry principles
(b) A set of specialized operators to query provenance information and facilitate analysis of provenance information
(c) A query engine to support the provenance operators implemented on an Oracle RDF database

We have used this framework in a real world eScience project for parasite research to create an end-to-end provenance management infrastructure. The next section presents an overview of this eScience project.

1.1 *T.cruzi* Semantic Problem Solving Environment Project

The *T.cruzi* Semantic Problem Solving Environment (SPSE) is a collaborative bioinformatics research project involving researchers at the Kno.e.sis Center, Wright State University, the Tarleton Research Group at CTEGD, University of Georgia, and the National Center for Biomedical Ontologies (NCBO) at Stanford University. The primary objective of the project is to create an ontology-driven integrated environment to facilitate identification of vaccine, diagnostic, and chemotherapeutic targets in the human pathogen *Trypanosoma Cruzi* (*T.cruzi*) [10]. *T.cruzi* is a protozoan parasite and a relative of other human pathogens that cause African sleeping sickness and leishmaniasis. Approximately 18 million people in Latin America are infected with this parasite.

Parasite researchers use data from multiple sources, namely “wet-lab” experiment protocols (for example, expression profiling, and proteome analysis), external databases (for example, UniProtDB [11], TriTrypDB [12]), and published literature (for example, PubMed [13]). These datasets not only have different representation formats but also use different methods for data generation and curation. Existing approaches use tedious manual techniques to integrate these datasets from multiple sources. The *T.cruzi* SPSE aims to utilize Semantic Web technologies to integrate local and external datasets to answer biological questions at multiple levels of granularity (Figure 1).

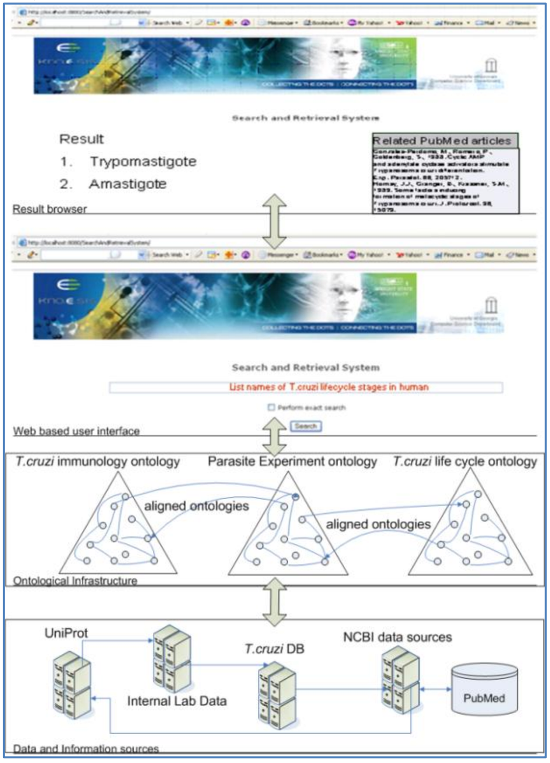

Search and Retrieval System
Result
1. Trypomastigote
2. Amastigote
Related PubMed articles
Result browser
Search and Retrieval System
List names of T.cruzi lifecycle stages in human
Perform exact search
Search
Web based user interface
T.cruzi immunology ontology
Parasite Experiment ontology
T.cruzi life cycle ontology
aligned ontologies
aligned ontologies
Ontological Infrastructure
UniProt
Internal Lab Data
T.cruzi DB
NCBI data sources
PubMed
Data and Information sources

Figure 1: A schematic representation of the *T.cruzi* SPSE

A coherent integration of the disparate datasets in *T.cruzi* SPSE requires the analysis of the associated experimental conditions in which the datasets were generated. To achieve this objective a provenance management infrastructure called *T.cruzi* Provenance Management System (PMS) has been implemented. In this paper, we describe creation of this infrastructure using the theoretical underpinning of the PMF [8]. The key contributions of the paper are described below:

1. Creation of an end-to-end provenance management infrastructure for parasite research called *T.cruzi* PMS.
2. Development of a domain-specific provenance ontology for *T.cruzi* PMS called Parasite Experiment (PE) ontology. The PE ontology models provenance information of experiment protocols used in parasite research. The PE ontology

extends the *provenir* upper level provenance ontology defined in the PMF [8] to facilitate interoperability with provenance ontologies in other domains.

3. An evaluation of the *T.cruzi* PMS capabilities to answer provenance queries over experiment datasets generated in the Tarleton research group is presented. The provenance queries are executed using provenance query operators implemented in a query engine over an Oracle RDF database.
4. The scalability of the *T.cruzi* PMS is also demonstrated in terms of both increasing sizes of data and complexity of provenance queries using a novel optimization technique based on materialized views.

1.2 Outline of the Paper

Section 2 describes the challenges faced in management of provenance information using the current infrastructure in the Tarleton research group. Section 3 describes the architecture of the *T.cruzi* PMS and introduces the Parasite Experiment ontology to model provenance of experiment protocols. Section 4 discusses the query infrastructure for provenance analysis in *T.cruzi* PMS. Section 5 presents the evaluation results for *T.cruzi* PMS. Section 6 correlates the work described in this paper with existing work in provenance management and Section 7 concludes with summary and future work.

2 Challenges in Provenance Management for *T.cruzi* SPSE

An important approach to the study of *T.cruzi* infection is the use of reverse genetics to create avirulent (non-virulent) strains of the *T.cruzi* parasite in the laboratory. The creation of such parasite strains requires the identification of genes that control a core biochemical function. These genes can be deleted from the genome of the parasite (gene "knock-out") in order to ablate the biochemical function, possibly resulting in an avirulent strain. The two experiment processes used in creation of *T.cruzi* avirulent strains are (a) Gene Knockout (GKO) Protocol, and (b) Strain Project (SP) Protocol.

The next section describes the two experiment protocols and the associated provenance information that need to be captured along with the experiment results.

2.1 Provenance Information in Gene Knockout and Strain Project Experiment Protocols

Given a list of genes for creation of potential avirulent *T.cruzi* strains, each gene forms an input to the GKO experiment protocol. To totally ablate (or at a minimum reduce) the function of genes, each of the alleles of the genes are targets of knock-out (Figure 2). The output of the GKO experiment protocol is a "knockout construct plasmid", which is created using the appropriate sequences of the target gene and a chosen antibiotic resistance gene. This plasmid is used in the SP experiment protocol to create a new strain of *T.cruzi* (Figure 3).

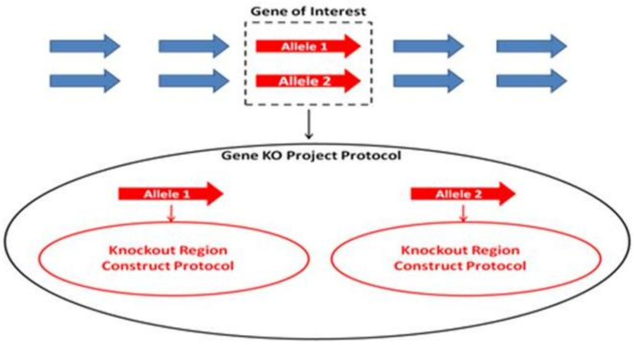


Figure 2: Alleles of target genes are used to create knockout plasmids

The SP Protocol is composed of three sub-processes (described in Figure 3) namely, Transfection, Drug Selection, and Cloning. Briefly, during transfection the Knockout Construct Plasmid will replace the target gene in the *T. cruzi* genome with a selected antibiotic resistance gene resulting in a "Transfected Sample". The expression of the antibiotic resistance gene will allow parasites that were successfully transfected to survive drug treatment (Selection) with an antibiotic such as Neomycin. Researchers treat the Transfected Sample with the antibiotic for the period of time that kills all parasites in a non-transfected sample. Individual parasites within the resulting "Selected Sample" are then cloned to create "Cloned Samples" which are then used to infect model animals such as mice to assess strain phenotype and attenuation.

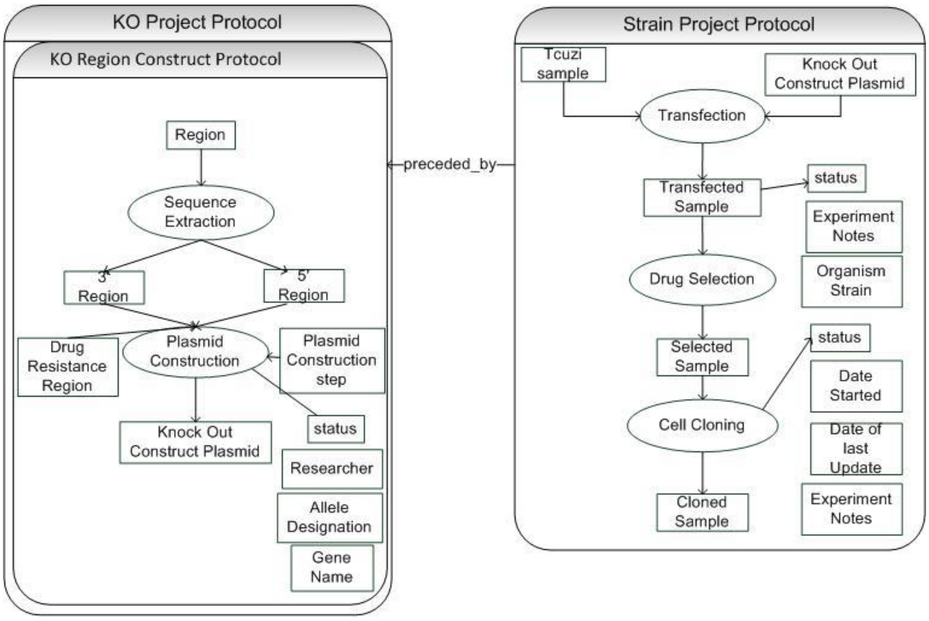


Figure 3: Schematic representation of GKO and SP experiment protocols

At the end of above two protocols we not only obtain a new avirulent *T. cruzi* strain, but also a plethora of data that need to be stored and analyzed. The process to create a new strain may take many months, and at each step, important provenance information must be collected and stored. This provenance information can be used by the technicians and project managers to track the progress of the experiments and is also important for publications of results in literature. Specific examples of provenance information that must be collected include the samples identifier, names and annotation information for the targeted genes, justification for knockout, plasmid constructs, antibiotic resistance genes, transfection methods (e.g. sonication, electroporation), number of transfection attempts, selection antibiotic, period of selection, and the ultimate success of knocking-out the gene from the genome.

The collection, representation, storage, and querying of the provenance information is difficult using the existing infrastructure in the Tarleton research group. In the next section, we discuss these challenges using a set of example provenance queries.

2.2 Querying Provenance Information of Experiment Data

The provenance information collected during GKO and SP experiments is used by multiple users with different requirements:

1) Technicians performing the lab-related work,
2) Project managers or principal investigators who want to track progress and/or view strains successfully created,
3) New researchers such as visiting faculty or post-docs who want to learn the lab-specific methods, and
4) Researchers in the parasite research community who can infer phenotype of the related organisms that they study from the work done on *T. cruzi*.

The current informatics infrastructure in the Tarleton research group is built using multiple relational databases that are accessed via custom web pages to store, track, and view data for a project. We describe how an example set of provenance queries are executed using the existing infrastructure.

Query 1: *List all groups using"target_region_plasmid_Tc00.1047053504033.170_1" target region plasmid*?
Current Approach: This query cannot be performed by a user using the current infrastructure in the Tarleton research group. The informatics specialist has to search for data from different databases and then create a set of customized SQL queries and scripts to answer the query.

Query 2: *Find the name of the researcher who created the knockout plasmid "plasmid66"?*
Current Approach: Answering this query requires access to three tables from two database schemas and use of PHP-based web tools. A custom query builder tool is used to search for plasmids with identifier "plasmid66". Next, in a three step process, including searching for "Strain" record associated with the given plasmid identifier and gene details, the name of the researcher is located.

Query 3: "c*loned_sample66" is not episomal. How many transfection attempts are associated with this sample?*
Current Approach: Using a custom query builder tool, a SQL query joining two tables is generated to list the strains with cloned samples with confirmed episomal insertion of the plasmid. From this list the user can obtain the number of transfection attempts to create the strain.

Query 4: *Which gene was used create the cloned sample "*c*loned_sample66"*?
Current Approach: To answer this query the researcher again uses a custom query builder to select the "KO Gene Name" in a tabular result view and search for "cloned_sample66". The custom query builder performs the joins necessary to combine data from three tables and creates the SQL query automatically. However, changes to the underlying database require modification of the PHP code by the informatics specialist.

These example queries demonstrate the limitations of current infrastructure that either cannot answer a query (Query 1) or requires the user to follow a multi-step process to retrieve the result. These limitations, especially the manual effort required, assume significance in a high-throughput experiment environment with multiple concurrent projects and the need to integrate provenance information across projects to guide future research. In the next section, we describe the ontology-driven provenance management infrastructure that has been created to address these challenges.

3 *T.cruzi* Provenance Management System

The *T.cruzi* PMS infrastructure addresses four aspects of provenance management in the *T.cruzi* SPSE project (Figure 4):

1. **Provenance Capture** – The provenance information associated with SP and GKO experiment protocols are captured via web pages used by researchers during an experiment. This data is transformed into RDF instance data corresponding to the PE ontology schema.
2. **Provenance Representation** – The parasite experiment (PE) ontology is used to model the provenance information of GKO and SP experiment protocols. The integrated provenance information, from both these experiment protocols, is represented as "ground" RDF graph, that is, without any blank nodes [14].
3. **Provenance Storage** – The provenance information is stored in an Oracle 10g (release 10.2.0.3.0) RDF database management system (DBMS). Oracle 10g was chosen due to its widespread use in biomedical informatics applications [15] [16] and it satisfied the requirements of the *T.cruzi* PMS. For example, it supports the full SPARQL query specification [17], use of RDFS [14] as well as user-defined reasoning rules, and is a proven platform for large scale RDF storage [18].We note that the *T.cruzi* PMS can be implemented over any RDF DBMS that support the above listed requirements.
4. **Provenance Query Analysis** – In addition to storage of provenance information, the *T.cruzi* PMS supports querying of provenance information, using a set of

specialized provenance query operators. The query operators are implemented in a query engine deployed over the Oracle RDF database.

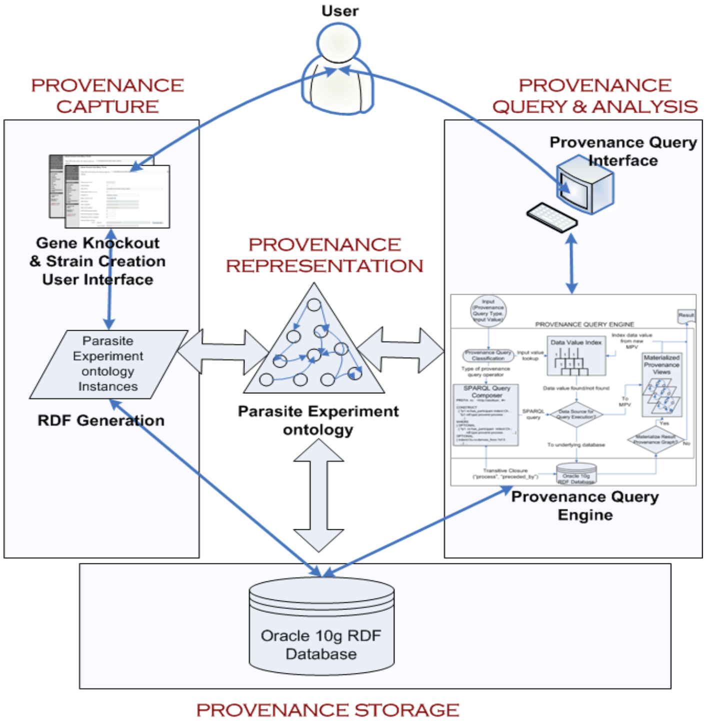

User
PROVENANCE CAPTURE
PROVENANCE QUERY & ANALYSIS
Provenance Query Interface
Gene Knockout & Strain Creation User Interface
PROVENANCE REPRESENTATION
Parasite Experiment ontology Instances
RDF Generation
Parasite Experiment ontology
Input (Provenance Query Type, Input Value)
Result
PROVENANCE QUERY ENGINE
Provenance Query Classification
Input value lookup
Data Value Index
Index data value from new MPV
Type of provenance query operator
Materialized Provenance Views
SPARQL Query Composer
Data value found/not found
To MPV
SPARQL query
Data Source for Query Execution?
Yes
Materialize Result Provenance Graph?
No
To underlying database
Transitive Closure ("process", "preceded_by")
Oracle 10g RDF Database
Provenance Query Engine
Oracle 10g RDF Database
PROVENANCE STORAGE

Figure 4: The architecture of the *T.cruzi* PMS addressing four aspects of provenance management

In this section we focus on the PE ontology that forms the key component of the *T.cruzi* PMS (Figure 4). Provenance information includes significant domain-specific information (for example, trypsin enzyme is used for proteolysis of a protein sample). But, a single monolithic provenance ontology for different domains is clearly not feasible. Hence, a modular approach is proposed in the PMF [8] and involves integrated use of multiple ontologies, each modeling provenance metadata specific to a particular domain. For example, the ProPreO ontology [19] represents proteomics domain-specific provenance. These provenance ontologies extend the *provenir* upper-level provenance ontology to facilitate interoperability. We present a brief overview of the *provenir* ontology.

3.1 Provenir ontology – Upper Level Provenance ontology

The *provenir* ontology has been created using the OBO Foundry principles [20]. Using the two primitive philosophical ontology concepts of "occurrent" and "continuant" [21], *provenir* defines three basic classes namely, `data`[1], `agent`, and

[1] We use the `courier` font to denote ontology classes and properties

`process` (Figure 5). The two base classes, `data` and `agents` are defined as specialization (sub-class) of the continuant class. The third base class `process` is a synonym of occurrent class.

The datasets that undergo modification in an experiment are modeled as `data_collection` class and the parameters that influence the execution of experiments are modeled as `parameter` class. Both these classes are sub-classes of the `data` class. The `parameter` class has three sub-classes representing the spatial, temporal, and thematic (domain-specific) dimensions, namely `spatial_parameter`, `temporal_parameter`, and `domain_parameter` (Figure 5).

Instead of defining a new set of properties, a set of fundamental properties defined in the Relation ontology (RO) [21] have been reused and adapted in *provenir* ontology (Figure 5). For example, "`part_of`", "`contained_in`", "`preceded_by`", and "`has_participant`". The *provenir* ontology is defined using OWL-DL [9] with an expressivity of $\mathcal{ALCH}$; further details of the ontology are described in [8]. In the next section we describe the PE ontology that extends the *provenir* ontology for *T.cruzi* PMS.

3.2 Parasite Experiment ontology

The PE ontology models the provenance information associated with GKO and SP experiment protocols, described in Section 2.2. New classes and properties are added to PE ontology while ensuring that any new construct does not contradict constructs in the *provenir* ontology. The PE ontology is modeled using the OWL-DL [9] language and contains 94 classes and 27 properties (23 object and 4 datatype properties) with a description logic (DL) expressivity of $\mathcal{ALCHF(D)}$. We now describe the different components of the PE ontology (Figure 5).

First, we discuss the modeling of process entities that constitute the GKO and SP experiment protocols. Two classes namely, `gene_knockout_process` and `strain_creation_process`, are created as subclass of `provenir:process`[2] class, to model generic gene knockout and strain creation experiment processes. The `knockout_project_protocol` and `strain_creation_protocol` classes represent the particular protocols used in the Tarleton research group. The GKO and SC protocols consist of multiple sub-processes, which are also modeled in PE ontology, for example `sequence_extraction`, `plasmid_construction`, `transfection`, `drug_selection`, and `cell_cloning` (Figure 5).

Next, we describe the PE ontology concepts that model the datasets and parameters used in the GKO and SC experiment protocols. A novel feature of the *provenir* ontology is the distinct modeling of `provenir:data_collection` (that represents data entities that undergo processing in an experiment) and `provenir:parameter` (that represents parameters that influence the behavior of a

[2] Provenir classes and properties are represented using the provenir namespace as `provenir:process` where `provenir` resolves to http://knoesis.wright.edu/provenir/provenir.owl

process or agent). This distinction is maintained in the PE ontology, for example, given the transfection process its input value Tcruzi_sample is modeled as a subclass of provenir:data_collection class whereas the parameter value transfection_buffer is modeled as a sub-class of the provenir:parameter class. Further, the parameter values in PE ontology are categorized along the space, time, and theme (domain-specific) dimensions (Figure 5).

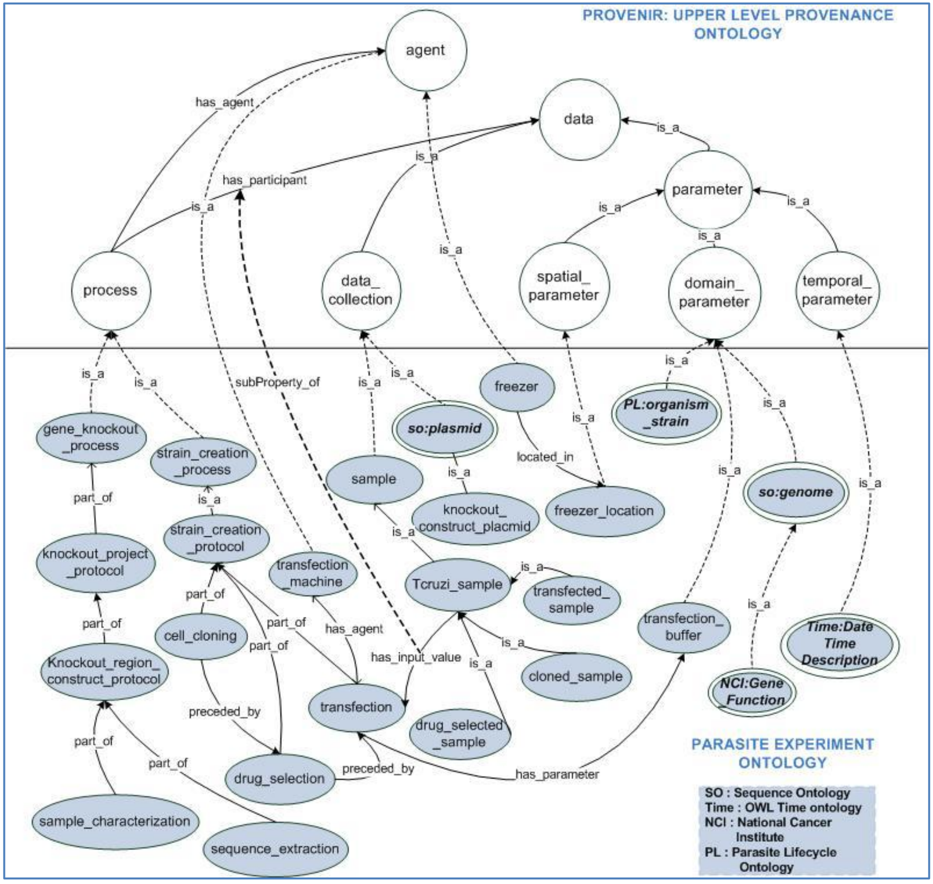


Figure 5: The PE ontology extends the provenir ontology to model domain-specific provenance for experiment protocols in *T.cruzi* research

The third set of PE ontology concepts extend the provenir:agent class to model researchers and instruments involved in an experiment. For example, transfection_machine, microarray_plate_reader are instruments modeled as subclass of provenir:agent; researcher is an example of human agent; and knockout_plasmid is an example of a biological agent.

Finally, we describe the properties used to connect the PE ontology classes. In addition to the eleven relationships in *provenir* ontology, new object and datatype properties specific to GKO and SP experiment protocols were created. For example, four new object properties are defined to model the similarity relationships between two genomic regions, namely `is_paralogous_to`, `is_orthologous_to`, `is_homologous_to`, and `is_identical_to`.

In addition to extending the *provenir* ontology, the PE ontology re-uses classes from existing ontologies listed at the NCBO, which is discussed in the next section.

3.3 Interoperability with Existing ontologies

The NCBO currently lists 137 publicly available biomedical ontologies [22] and it is important that new ontologies, such as PE ontology, are interoperable with these existing ontologies. Hence, the PE ontology re-uses relevant classes and relationships from four public ontologies namely, Sequence ontology (SO) [12], the National Cancer Institute (NCI) thesaurus [23], Parasite Life Cycle ontology (PL) [24], and the W3C OWL Time ontology [25] (Figure 5).

The SO models biological sequences and is a joint effort by genome annotation centers and users of sequence annotation data [12]. The PE ontology re-uses multiple SO classes, including `so:plasmid`, `so:genome` along with its subclasses such as `so:chromosome`, `so:gene`, and `so:flanking_region`. Similarly, `NCI:gene_function`, `PL:organism_strain`, `Time:DateTimeDescription` are some of the other classes re-used in PE ontology from the NCI, PL, and OWL Time ontology respectively. In addition, PE ontology also re-uses the object property `PL:has_base_strain` from PL ontology. Therefore, the PE ontology not only allows interoperability with domain-specific provenance ontologies by extending the *provenir* ontology, but also ensures interoperability with existing biomedical ontologies listed at NCBO.

In the next section, we describe the query capabilities of the *T.cruzi* PMS that uses PE ontology for query composition and query optimization.

4 Query Infrastructure of *T.cruzi* PMS: Provenance Query Operators and Query Engine

The capture and storage of provenance information is of limited use without an effective query mechanism to enable provenance analysis. The query capability of *T.cruzi* PMS is constituted of two components, namely:

a) **Provenance Query Operators**: A set of specialized query operators for provenance information.

b) **Provenance Query Engine**: A query engine to support the provenance query operators over an Oracle 10g RDF database using SPARQL query language. The query engine uses a novel materialized view-based optimization technique to ensure scalability with increasing size of data as well as complexity of queries.

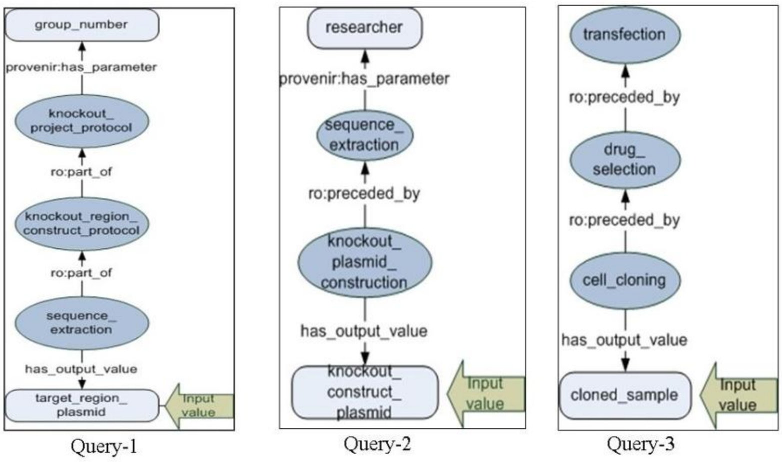
4.1 Provenance Query Operators

The provenance query operators are based on the classification of provenance queries proposed in PMF [8], that categorizes provenance queries into three classes:

1. **Query for provenance metadata**: Given a data entity, this category of queries returns the complete set of provenance information associated with a data entity.
2. **Query for dataset using provenance information**: An opposite perspective to the first category of query is, given a set of constraints defined over provenance information retrieve a set of data entities satisfying these constraints.
3. **Operations on provenance information**: This category of queries defines operations over the provenance metadata such as comparing or merging of provenance information.

Using this classification scheme, a set of specialized query operators has been defined [8] namely, (a) **provenance ( )** – to retrieve provenance information for a given dataset, (b) **provenance_context ( ) –** to retrieve datasets that satisfy constraints on provenance information, (c) **provenance_compare ( )** – given two datasets, this query operator determines if they were generated under equivalent conditions by comparing the associated provenance information, and (d) **provenance_merge ( )** – to merge provenance information from different stages of an experiment protocol. The formal definition of these query operators is described in [8].

In contrast to the existing informatics infrastructure in the Tarleton research group, the *T.cruzi* PMS uses the provenance query operators (implemented in a query engine) to execute provenance queries. Given an input value, the query operators compose the corresponding SPARQL query pattern. Figure 6 describes the use of the *provenance ()* query operator to answer the example provenance queries introduced in Section 2.2.



Query-1 Query-2 Query-3

Figure 6: Use of *provenance ()* query operator to answer example provenance queries (from Section 2.2)

The provenance query operators are implemented in a query engine over an Oracle 10g RDF database. We describe the details of this provenance query engine in the next section.

4.2 Provenance Query Engine and Materialized Provenance Views

The provenance query engine is designed as a Java-based Application Programming Interface (API) for use with any RDF DBMS that supports SPARQL query language and rule-based reasoning. The query engine uses the formal definition of the provenance query operators [4], to automatically compose the corresponding query expression in SPARQL syntax.

Provenance queries are path computations over RDF graphs and are expensive operations that require computation of fixed paths, recursive pattern-based paths and neighborhood retrieval. As discussed in [8], a straightforward implementation does not scale with the large scale datasets for complex provenance queries, hence a new class of materialized views called materialized provenance views (MPV) have been defined in PMF [8].

Theoretically, the MPV correspond to a single logical unit of provenance in a given domain, for example one complete experiment cycle in *T.cruzi* domain. A logical unit of provenance information is identified using the domain-specific ontology used for an application. The MPV in *T.cruzi* PMS is defined using the PE ontology as a set of processes starting with the `sequence_extraction` class and terminating with the `cell_cloning` class (Figure 7). An important advantage of defining an MPV using the PE ontology is the ability of a single MPV to satisfy all queries for data entities created or used in a single experiment cycle.

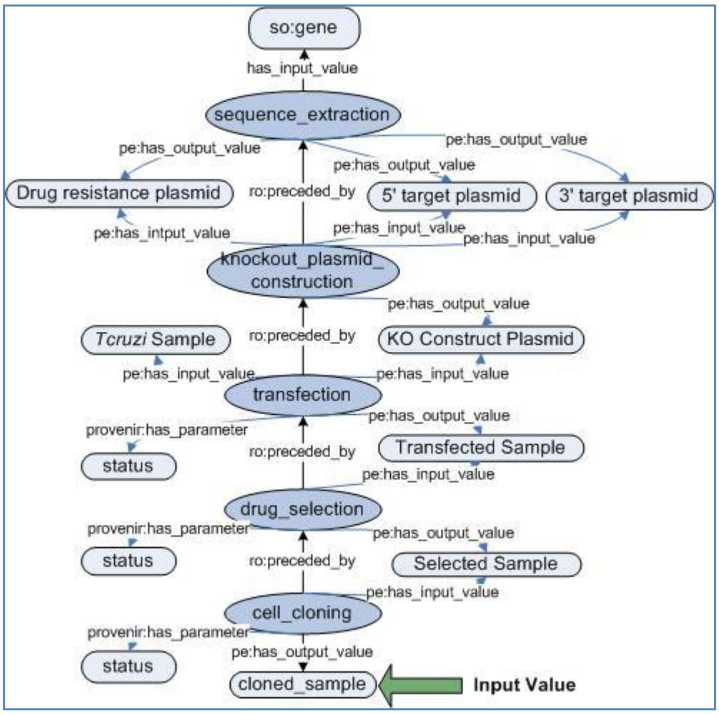


Figure 7: The result of Query 4 corresponds to a Materialized Provenance View (MPV) in *T.cruzi* PMS

The query engine uses a B-tree index to identify query inputs that can be satisfied by a MPV instead of being executed against the underlying database. The use of MPV results in significant gain in query response time with increasing data size and complexity of provenance query expression pattern. The next section discusses the evaluation results of the provenance queries (introduced in Section 2.2).

5 Evaluation and Results

The objective of our evaluation of the *T.cruzi* PMS is three-fold:

1. Verify that the example provenance queries (Section 2.2) can be answered correctly in the *T.cruzi* PMS
2. Evaluate the scalability of *T.cruzi* PMS with increasing size of RDF data
3. Evaluate the ability of the *T.cruzi* PMS to answer increasingly complex provenance queries.

5.1 Experiment Setup, Queries, and Dataset

The experiments were conducted using Oracle10g (Release 10.2.0.3.0) DBMS on a Sun Fire V490 server running 64-bit Solaris 9 with four 1.8 GHz Ultra Sparc IV processors and 8GB of main memory. The database used an 8 KB block size and was configured with a 512 MB buffer cache.

The dataset (Table I) corresponds to a number of experiment cycles and were generated in the Tarleton research group. The standard RDFS entailment rules and two user defined rules were used to create new inferred triples (Table I). The first user-defined rule asserts that "If the input value of a `process` (p1) is same as output value of another `process` (p2), then p1 is linked to p2 by property `ro:preceded_by`". The second user-defined rule asserts that "If a `process` (p1) is part of another `process` (p2) and pa1 is a `parameter` for p2, then pa1 is also a parameter for `process` (p1).

Table 1. The four RDF datasets used to evaluate scalability of *T.cruzi* PMS

Dataset ID	Number of RDF Inferred Triples	Total Number of RDF Triples
DS 1	2,673	3,553
DS 2	3,470	4,490
DS 3	4,988	6,288
DS 4	47,133	60,912

The SPARQL query patterns corresponding to the example provenance queries represent varying levels of query patterns complexity in terms of total number of variables, the total number of triples, and use of the SPARQL OPTIONAL function [26]. This complexity is also called "expression complexity" [27], and Table II lists the expression complexity of the example queries expressed in SPARQL syntax.

Table 2. The four queries (Section 2.2) used to evaluate scalability of query engine

Query ID	**Number of Variables**	**Total Number of Triples**	**Nesting Levels using OPTIONAL**
Query 1: Target plasmid	25	84	4
Query 2: Plasmid_66	38	110	5
Query 3: Transfection attempts	67	190	7
Query 4: cloned_sample66	67	190	7

5.2 Experiment 1

This experiment involved the verification of the results for the four queries executed using the T.cruzi PMS by the informatics specialist in the Tarleton research group. The results of the four queries are:

1) *"Group1" used "target_region_plasmid_Tc00.1047053504033.170_1" target region plasmid to create cloned samples*

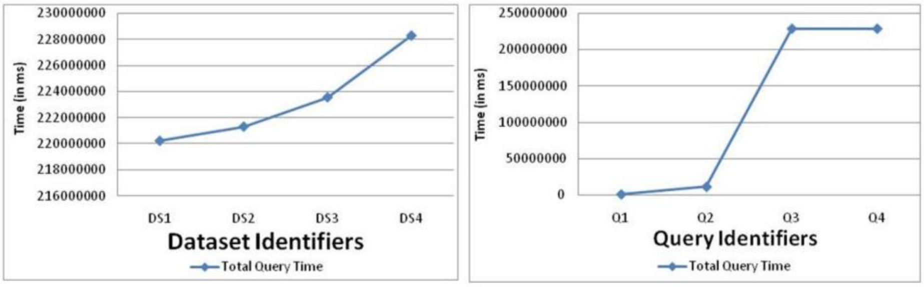
2) *Researcher with user ID = "1" created the knockout plasmid "plasmid66"*

3) *"Cloned sample 66", which is not episomal, involved 1 transfection attempt.*

4) *Gene with identifier "Tc00.1047053506727.100" was used create the cloned sample "cloned_sample66".*

5.3 Experiment 2

The four queries, Q1 to Q4 (in Table II), were executed against a fixed RDF dataset, DS4 (in Table I) to evaluate the performance of query engine for provenance queries with increasing complexity. Figure 8 (a) shows that the response time increases with increasing complexity of the provenance queries. Similarly, to evaluate the performance of the query engine with increasing size of data, query Q4 (in Table II) is executed against the four RDF datasets, DS1 to DS4 (in Table I). Figure 8 (b) shows that the response time of the query engine increases with increasing size of RDF data.



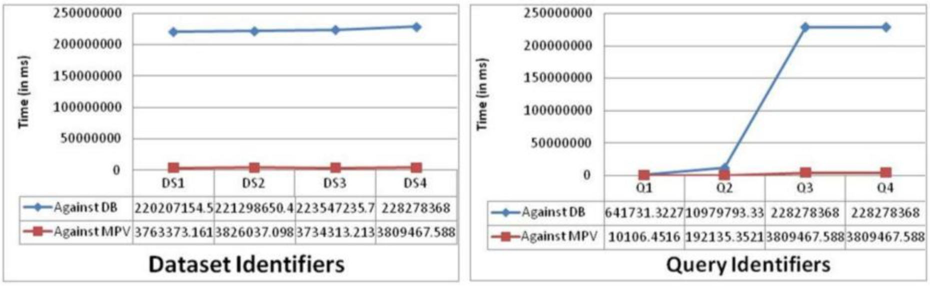
(a) (b)

Figure 8: The response time for provenance query engine with (a) increasing size of RDF dataset and (b) increasing complexity of queries

The two sets of results demonstrate the need for effective optimization techniques to enable practical use of the query engine in the *T.cruzi* PMS. The next experiment discusses the results of using the MPV for query optimization.

5.4 Experiment 3

Using the results of "Experiment 2" as baseline, this experiment discusses the significant improvement in response time of the provenance query engine using MPV. Figure 9 (a) shows the comparative results for provenance queries with increasing complexity executed against the underlying database and the MPV. Similar to "Experiment 2", Figure 9 (a) describes the result of executing the four queries (in Table II) using the fixed dataset DS 4 (in Table I). The MPV used in Figure 9 (a) corresponds to the provenance result of "*Cloned sample 66*" consisting of 139 RDF triples and occupying 27KB space. We note that this single MPV is used to answer all the four queries, Q1 to Q4 (in Table II). Figure 9 (b) shows the benefit of using MPV for query Q4 (in Table II) over increasing size of RDF datasets, DS1 to DS4 (in Table I).



(a)

(b)

Figure 9: Comparative results for (a) increasing size of RDF datasets over the underlying database and MPV and (b) provenance queries with increasing complexity

The results demonstrate that the use of MPV leads to significant improvement in response time, for both increasing complexity of provenance queries and increasing size of RDF dataset.

6 Related Work

Provenance has been studied in both the eScience [7] and the database community [6]. In the eScience community, provenance management has been addressed primarily in the context of workflow engines [2] [5], but recent work has argued for use of domain semantics in eScience provenance [3]. Simmhan et al. [7] survey the provenance management issues in eScience. The database community has also addressed the issue of provenance and defined various types of provenance, for example "why provenance" [28] and "where provenance" [28]. Database provenance is also described as fine-grained provenance [6]. A detailed comparison of PMF (that underpins the *T.cruzi* PMS) with both workflow and database provenance is presented in [8].

The Semantic Provenance Capture in Data Ingest Systems (SPCDIS) [29] is an example of eScience project with dedicated infrastructure for provenance management. In contrast to the *T.cruzi* PMS, the SPCDIS project uses the proof markup language (PML) [30] to capture provenance information. The Inference Web toolkit [30] features a set of tools to generate, register and search proofs encoded in PML. Both *T.cruzi* PMS and the SPCDIS have common objectives but use different approaches to achieve them, specifically the *T.cruzi* PMS uses an ontology-driven approach with robust query infrastructure for provenance management.

7 Conclusion

This paper introduces an in-use ontology-driven provenance management infrastructure for parasite research called *T.cruzi* PMS. The following conclusions are drawn from our experience described in this paper:

1. Domain-specific provenance ontologies, such as PE ontology, are the key component for an eScience provenance management infrastructure. Further, by extending the *provenir* ontology, the PE ontology can interoperate with other domain-specific provenance ontologies to facilitate sharing and integration of provenance information from different domains and projects.
2. The provenance query operators effectively support provenance queries and provide the users with a well-defined and robust mechanism to execute complex provenance queries.
3. The *T.cruzi* PMS, using MPV-based query optimization, is a scalable infrastructure for increasing data size as well as complexity of provenance queries.

In future, we plan to integrate other experiment protocols in the Tarleton research group such as proteome analysis and sample characterization in the *T.cruzi* PMS.

Acknowledgments. This work is funded by NIH RO1 Grant# 1R01HL087795-01A1. We would like to thank Flora Logan for her extensive help in developing the Parasite Experiment ontology.

References

[1] Society BC. Grand challenges in computing research (BCS Survey); 2004.

[2] http://twiki.ipaw.info/bin/view/Challenge/WebHome.

[3] Sahoo SS, Sheth, A., Henson, C. Semantic Provenance for eScience: Managing the Deluge of Scientific Data. IEEE Internet Computing 2008;12(4):46-54.

[4] Sahoo SS, Barga, R.S., Goldstein, J., Sheth, A. . Provenance Algebra and Materialized View-based Provenance Management: Microsoft Research Technical Report; 2008 November.

[5] http://twiki.ipaw.info/bin/view/Challenge/OPM.

[6] Tan WC. Provenance in Databases: Past, Current, and Future. IEEE Data Eng. Bull. 2007;30(4):3 -12

[7] Simmhan YL, Plale, A.B., Gannon, A. D. A survey of data provenance in e-science SIGMOD Rec. 2005;34(3):31 - 36

[8] Sahoo SS, Barga, R.S., Goldstein, J., Sheth, A.P., Thirunarayan, K. "Where did you come from...Where did you go?" An Algebra and RDF Query Engine for Provenance Kno.e.sis Center, Wright State University; 2009.

[9] http://www.w3.org/TR/owl-features/. 22 Jan 2008

[10] http://knoesis.wright.edu/research/semsci/projects/tcruzi/.

[11] http://www.uniprot.org/.

[12] Aurrecoechea C, M. Heiges, H. Wang, Z. Wang, S. Fischer, P. Rhodes, J. Miller, E. Kraemer, C. J. Stoeckert, Jr., D. S. Roos, and J. C. Kissinger. ApiDB: integrated resources for the apicomplexan bioinformatics resource center. Nucleic Acids Research 2007;35(D):427-30.

[13] http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=PubMed.

[14] http://www.w3.org/TR/rdf-mt/#defentail. 22 Jan 2008

[15] Kelly BK, Anderson, P. E., Reo, N. V., DelRaso, N. J. , Doom, T. E., Raymer, M. L. A proposed statistical protocol for the analysis of metabolic toxicological data derived from NMR spectroscopy. In: 7th IEEE International Conference on Bioinformatics and Bioengineering (BIBE 2007); 2007; Cambridge - Boston, Massachusetts, USA; 2007. p. 1414-1418.

[16] http://www.oracle.com/technology/industries/life_sciences/olsug.html.

[17] http://www.w3.org/TR/rdf-sparql-query. 22 Jan 2008

[18] Chong EI, Das, S., Eadon, G., and Srinivasan, J. An efficient SQL-based RDF querying scheme. In: 31st international Conference on Very Large Data Bases; 2005 August 30 - September 02; Trondheim, Norway: VLDB Endowment; 2005. p. 1216-1227.

[19] Sahoo SS, Thomas, C., Sheth, A., York, W. S., and Tartir, S. Knowledge modeling and its application in life sciences: a tale of two ontologies. In: Proceedings of the 15th international Conference on World Wide Web WWW '06 2006 May 23 - 26; Edinburgh, Scotland; 2006. p. 317-326.

[20] http://obo.sourceforge.net/.

[21] Smith B, Ceusters W, Klagges B, Kohler J, Kumar A, Lomax J, et al. Relations in biomedical ontologies. Genome Biol 2005;6(5):R46.

[22] http://bioontology.org.

[23] http://ncit.nci.nih.gov.

[24] http://www.sanger.ac.uk/Users/mb4/PLO/.

[25] Hobbs JR, Pan, F. Time Ontology in OWL In: W3C Working Draft; 2006.

[26] Pérez J, Arenas, M., Gutiérrez, C. . Semantics and Complexity of SPARQL. In: Int'l Semantic Web Conf. (ISWC '06); 2006; Athens, GA; 2006. p. 30-43.

[27] Vardi M. The Complexity of Relational Query Languages. In: 14th Ann. ACM Symp. Theory of Computing (STOC '82); 1982; 1982. p. 137-146.

[28] Buneman P, Khanna , S., Tan, W.C. Why and Where: A Characterization of Data Provenance. In: 8th International Conference on Database Theory; 2001; 2001. p. 316 - 330

[29] http://spcdis.hao.ucar.edu/.

[30] http://iw.stanford.edu/2.0/.