Storing Scientific Workflows in a Database

Zoé Lacroix[1,2] Christophe Legendre[2] Spyro Mousses[1]

[1]Translational Genomics Research Institute (TGen)
13028 Shea Blvd, suite 110
Scottsdale AZ 85259, USA
zlacroix@tgen.org
smousses@tgen.org

[2]Arizona State University
PO Box 875706
Tempe AZ 85287, USA
zoe.lacroix@asu.edu
christophe.legendre@asu.edu

ABSTRACT

The use of workflow models to integrate intelligently complex experimental and analytical processes is becoming more and more critical to support scientific discovery. Storing and providing querying capabilities to retrieve, import, re-use, adapt, and reason about workflows are becoming necessary components to workflow architectures supporting collaborative and translational research. We report on the evaluation of ProtocolDB a database that supports workflow design and storage conducted at the Translational Genomics Research Institute (TGen).

1. INTRODUCTION

Scientific discovery relies on an experimental framework that corroborates hypotheses with experiments that are complex reproducible processes generating and transforming large datasets. The methods, implicit in the process, capture the semantics of the data, thus they are responsible for the generation of scientific information and discovery of scientific knowledge. Recording scientific workflows is critical to provide the semantics needed to wrap scientific data from their capture, analysis, publication, and archival. By annotating data with the processes that produce them, the scientist no longer manages data but information and allows their meaningful interpretation and integration. Moreover, a suitable recording of workflows allows workflow re-use, integration, comparison, iterative refinement, and various levels of reasoning.

Scientific workflows are composed of *services* that perform various tasks on a dataflow. Scientific workflows may involve wet lab experiments where services are achieved by laboratory technicians, robots, and machines as well as *in silico* tasks that exploit various data management, analysis, visualization, and publication tools including Web services. Laboratory Information Management Systems (LIMS) [1] support the integration of different functionalities in a laboratory, such as sample tracking (invoicing/quoting), integrated bar-coding, instrument integration, personnel and equipment management, etc. LIMS typically support *wet* workflows that coordinate the management of tasks, samples, and instruments and allow reasoning on business-like parameters such as ordering (e.g., invoicing) and organization (automation and optimization), but they do not offer semantic integration. In contrast scientific workflow systems such as Kepler [2] or Taverna [3] typically express *digital* workflows and execute them on platforms such as grids. They typically focus on syntactic integration in order to produce executable workflows.

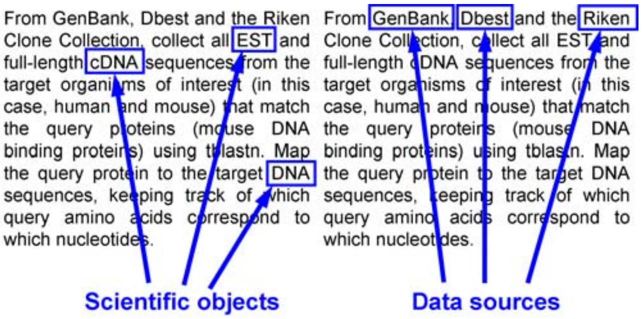
In this paper we report on the experiments conducted at the Translational Genomics Research Institute (TGen), Arizona State University, and other institutes to model and store scientific workflows in a database. In particular, we evaluate the ProtocolDB prototype developed at Arizona State University. We present the data model for workflows and describe the database in Section 2. We present our application cases in Section 3 and report our findings in Section 4. We discuss related work in Section 5 and conclude in Section 6.

2. RECORDING WORKFLOWS

Scientific workflows are often expressed as textual documents structured with steps. The analysis of the description of a scientific protocol is difficult without additional explanations by the scientists. The main difficulties are to understand the process and to identify the scientific aim and implementation of each task. Although the document typically provides a list of numbered steps, each step may involve several scientific tasks, and the complex network, including merges, splits, loops, and revisions, are implicitly expressed in the step descriptions or not included. The various parameters used to calibrate the tools are often missing as they are fixed during the execution of the protocol.

We illustrate the process of structuring a scientific workflow from its traditional textual recording with an alternative splicing workflow provided by our collaborator Dr. Marta Janer, Institute for Systems Biology. The scientific aim (*design*) and the specification of the resources involved in a workflow (*implementation*) are often mixed as illustrated in the step 2 of the alternative splicing pipeline shown in Figure 1. The alternative splicing pipeline is composed of 9 steps and provides a complete characterization of variations in proteins due to splice variation or SNPs evident in repositories of contiguous genome sequence data and expressed sequence tags (ESTs). The pipeline applies secondary structure, tertiary structure, domain motif detection and sequence comparison tools to proteins encoded by genes with alternatively splice forms or SNPs. The step 2 of the alternative pipeline identifies *scientific objects* (left of Figure 1).

VLDB '09, August 24-28, 2009, Lyon, France.

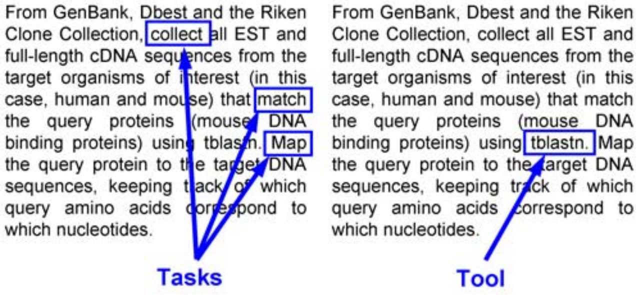


From GenBank, Dbest and the Riken Clone Collection, collect all EST and full-length cDNA sequences from the target organisms of interest (in this case, human and mouse) that match the query proteins (mouse DNA binding proteins) using tblastn. Map the query protein to the target DNA sequences, keeping track of which query amino acids correspond to which nucleotides.

From GenBank, Dbest and the Riken Clone Collection, collect all EST and full-length cDNA sequences from the target organisms of interest (in this case, human and mouse) that match the query proteins (mouse DNA binding proteins) using tblastn. Map the query protein to the target DNA sequences, keeping track of which query amino acids correspond to which nucleotides.

Figure 1 - S*cientific objects* (left) and *data sources* (right).

Scientific objects specify the input and output of the tasks involved in protocol steps, and may correspond to conceptual classes in a domain ontology. The input of each task is retrieved from a particular data source as illustrated on the right side of Figure 1. The tasks and tools involved in the protocol steps are shown in Figures 2.



From GenBank, Dbest and the Riken Clone Collection, collect all EST and full-length cDNA sequences from the target organisms of interest (in this case, human and mouse) that match the query proteins (mouse DNA binding proteins) using tblastn. Map the query protein to the target DNA sequences, keeping track of which query amino acids correspond to which nucleotides.

From GenBank, Dbest and the Riken Clone Collection, collect all EST and full-length cDNA sequences from the target organisms of interest (in this case, human and mouse) that match the query proteins (mouse DNA binding proteins) using tblastn. Map the query protein to the target DNA sequences, keeping track of which query amino acids correspond to which nucleotides.

Figure 2 - *Scientific tasks* (left) and *tools* (right)

Scientific workflows record both the scientific aim of each task and the description of its implementation. The scientific aim is often implicit and the design of the implementation is typically driven by the resources known by the scientist rather than by the resources that would best meet the workflow needs. To offer flexibility, in ProtocolDB each scientific workflow is decomposed into two components: *desig*n and *implementation*. Both the design and the implementation of a scientific workflow are composed of coordinated tasks. Each task of the *design* is defined by its input, output, and description. When an ontology is available to describe the scientific objects and tasks involved in a scientific protocol, the input and output of each protocol design task are defined by their respective concept classes. The description of the task may be a relationship defined between the input and output conceptual classes or a description of a relationship not defined in the ontology. The *implementation* describes the selection of resources used to implement each task of the protocol. The input of a protocol implementation task is the description (including name, URL, format, etc.) of the data source or the dataset for the input entries. The input data are instances of the input conceptual class for the corresponding design task. The output of a workflow implementation task is the description (including name, URL, format, etc.) of the data source linked to by the application or the dataset produced by the application implementing the task. The output data are instances of the output conceptual class of the corresponding design task. A similar distinction of the scientific aim from its implementation was noted by [4] who distinguish *conceptualization* and *specification*, and by [5] who refers to *abstract* and *concrete* workflows.

In ProtocolDB workflows are first expressed in terms of a domain ontology where each task expresses a specific aim. The ontology can be specified prior to the workflow design or generated from the workflow entry. A *design* workflow is defined top-down from a conceptual design task with an input and output expressed as complex types (record, set, list) in terms of the concepts of the ontology. This step allows the characterization of the dataflow. The user may either select concepts already entered in the domain ontology or the concepts used to describe the input and output of the workflow will be entered as new concepts in the workflow ontology. Each design task may be split either successively or in parallel into two design tasks. The splitting process (successor or parallel) is constrained by the dataflow already defined. That is if a workflow W has an input I and output O and is split with a succession $W_1 \otimes W_2$, then the input of W_1 is automatically assigned to I and the output of W_2 is automatically assigned to O. Similarly, if W is split with two parallel tasks W_1, and W_2, then the input (resp. output) of W_1 (resp. W_2) is included[1] in I (resp. O). A design workflow is mapped to one (or more) *implementation* workflows. Each design task is mapped to an implementation protocol: a service (basic task), a succession or a parallel composition of two implementation protocols. Workflows (designs and implementations), ontologies (conceptual graph) and resources (service graph) are stored in a relational database.

3. APPLICATION CASES

3.1 Protein superposition workflow

A protein is an organic compound made of a chain of amino acids. Scientific discovery often relies on the identification of similarities between an object of interest and other objects whose properties are already known. Techniques that capture similarities at the level of the amino acid sequence (alignment) may miss protein structural similarities. Other techniques focus on structural superposition of proteins. The problem of protein superposition is the spatial orientation of the structure. The *protein structure superposition workflow* was designed by Dr. Nathalie Meurice at the Translational Genomics Research Institute. It was modeled and recorded in ProtocolDB in 2008. In this section we illustrate the process and discuss the various analysis of the workflow that can be conducted with ProtocolDB.

The scientific aim of the protein structure superposition protocol is expressed in three successive conceptual steps. Initially a single task is declared with a complete description of the inputs and outputs of the workflow. The input of the protocol (and of the first conceptual task) consists of two proteins: a reference protein (*rP*) and an adapting protein (*aP*) that will be superposed onto *rP*. Both proteins have a structure (*rS*, *aS*), each with a specific initial orientation. Furthermore the output would also include the final orientation of the adapting protein (*aS'*). Each input and output of a task of the design workflow (left of Figure 3) is a collection of conceptual variables whereas the task names are relationships in the domain ontology. This task is then

[1] Here ‘inclusion’ refers to sub-typing.

further divided into three tasks of which two are in parallel and one sequential. The parallel tasks extract the structure of *rP* and *aP*. The third task which is sequentially placed after the first two tasks consists in finding the final orientation of the adapting protein (*aS'*) such that *rS* and *aS'* have the optimum superposition. The output of the protocol is the pair of proteins in their optimal coordinate system for superposition.

The implementation phase consists of the selection of resources (database accesses and tools) that are connected into an executable protocol. Some conceptual tasks may easily be implemented by a single resource. For example, the conceptual task *Has Structure* can be mapped to a database where the protein structure may be retrieved given a PDB[2] identifier. Other tasks may require the composition of several resources into a physical workflow for their implementation. This is the case of the *Re-Orientation* task.

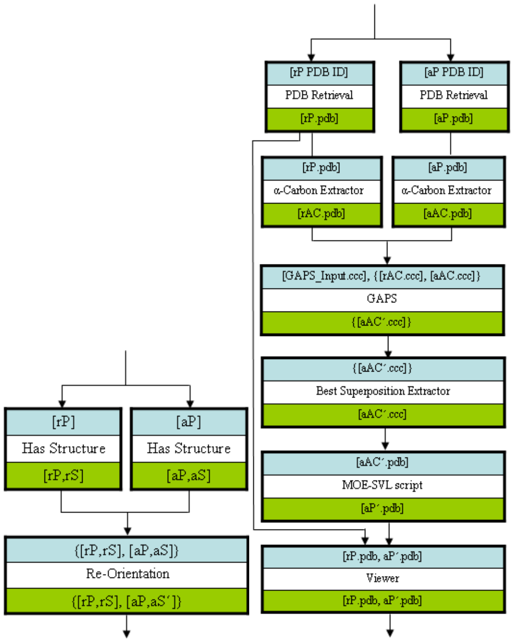


Figure 3 - Design (left) and implementation (right) workflows

The first task of the protein structure superposition protocol consists of the retrieval of each protein structure. Protein structures are retrieved from PDB using the protein identifiers and downloaded in PDB data format. The second task consists in extracting the α-carbon chain from each of the two input protein structures; this task is accomplished by a specific routine (called α-carbon Extractor in Figure 3). The third task is performed by GAPS[3] and produces several relative orientations of the adapting protein with respect to the reference protein, each protein being depicted at the user-defined representation level (here: α-carbon, coarse level) [6]. The best superposition extractor routine extracts the adapting protein α-carbon atoms with the best scoring orientation. This in turn becomes the input for a script written in Scientific Vector Language (SVL) that interfaces with the Molecular Operating Environment (MOE) platform[4]. This SVL script converts the best orientation α-carbon structure of the adapting protein (*aS'*) back into its original full protein form. Finally reference and adapting protein structures can be viewed superposed through a molecular viewer and further analyzed by the scientist. It is important to note that syntactic discrepancies exist between the inputs and outputs of the different tools. The first two parallel conceptual steps are each mapped to the PDB retrieval step.

The implementation workflow is mapped to the design workflow as follows. The input of the conceptual step is a protein name mapped to the input keyword of the PDB retrieval call. The third conceptual task is mapped to a whole implementation workflow that first extracts the α-carbon chains and then computes all orientations of the adapting protein to select the optimal one for visualization.

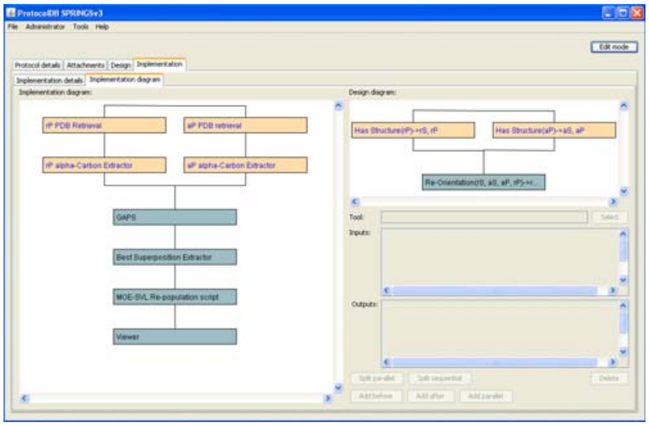


Figure 4 - Protein superposition workflow in ProtocolDB

The protocol in its present condition can be entered and stored in ProtocolDB (see Figure 4) but cannot be executed until syntactic interoperability issues are resolved. Indeed, once each conceptual task is mapped to a resource or an implementation workflow, the workflow is semantically characterized but not yet specified syntactically yet. Unlike format-driven approaches that select resources that can be composed together into an executable workflow, the ProtocolDB approach favors the semantics of the resources rather than their syntax (input/output data formats). We discuss mechanisms to produce efficient executable in Section 4.

3.2 Sub-cloning workflow

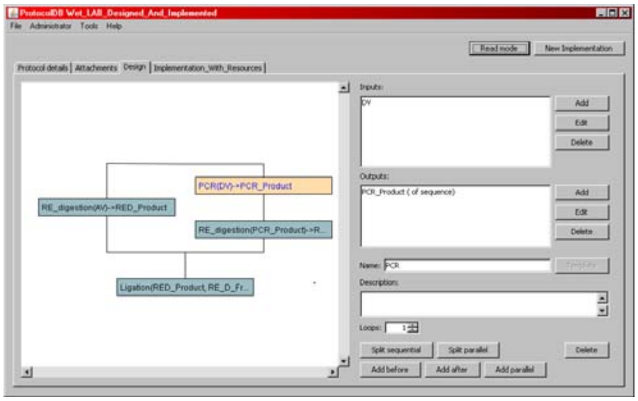
The sub-cloning workflow was designed by Dr. Sukru Tuzmen at TGen in 2008. The aim of the *sub-cloning workflow* is to transfer a sequence of interest present in a vector (donor vector) into another vector (acceptor vector). The design workflow entered in ProtocolDB is displayed in Figure 5. The input (resp. output) of the workflow is a plasmid construct containing a sequence of interest, i.e., insert (resp. another plasmid construct which has accepted the insertion of the sequence of interest). The first design task is a PCR step that aims at producing large

[2] The Protein Data Bank (PDB) is a public repository available at http://www.wwpdb.org/ that contains 3-D structural data of large biological molecules, such as proteins and nucleic acids.

[3] GAPS is a tool dedicated to protein similarity.

[4] SVL and MOE are products of Chemical Computing Group, Montreal, Canada (http://www.chemcomp.com/).

amounts of material and verifying the quality of the donor vector (DV). Starting with a small amount of DV material, a PCR is conducted in order to amplify the DV-integrated sequence of interest (or insert). Then, a digestion step is performed using the same enzymes to cut both the acceptor vector (AV) and the insert. A gel extraction of the digested products is performed (filtering task) in order to get rid of small sequences generated by the last enzymatic reaction and because they will spoil the ligation step. The two final extracted products, i.e., AV and sequence of interest, are linked together using the T4 Ligase enzyme (ligation step). The product of the ligation is run on an electrophoresis gel in order to check whether the expected vector is present or not in the mix of the final reaction. If it is, the newly formed vector (AV with sequence of interest) has to be amplified for further usage. The bacterial transformation is used for the final step of the sub-cloning protocol. The workflow design is thus composed of four design tasks: PCR, two tasks of digestion, and a ligation. The dataflow of the workflow is composed of the translational flow (here sequences) and the various parameters needed to implement each task. These parameters are not specified at the design level but will be specified at implementation when the services are selected.



Inputs: DV

Outputs: PCR_Product (of sequence)

Name: PCR

Loops: 1

Figure 5 - Sub-cloning design workflow

The expected implementation of the sub-cloning workflow is a *wet* implementation composed of eight tasks as described in Figure 6. This implementation workflow describes the process where each task is implemented by one or more services. Services are represented as basic implementation tasks with a name, input, and output description. Each service is also mapped to the ontology. Its input (resp. output) is expressed in terms of concepts and the service itself may be mapped to an existing conceptual relationship. For example, the thermocycler device available at the laboratory requires eight reagents: RNase/DNase free water, PCR buffer, MgCl2++, mix of dNTP (diNucleotide TriPhosphate), Taq polymerase, Primers Forward and Reverse, and finally the template (sequence). The complex datatype that describe the input of the thermocycler is expressed in terms of four concepts in the ontology: solvent, ion, sequence, enzyme, and as follows:

[solvent,solvent,ion,sequence,enzyme,[sequence,sequence],sequence]

Its output is a set of sequences or {sequence}. While the input sequence corresponds to the input of the workflow, the other inputs: RNase/DNase free water, PCR buffer, MgCl2++, mix of dNTP (diNucleotide TriPhosphate), Taq polymerase, Primers Forward and Reverse of respective conceptual type solvent, solvent, ion, sequence, enzyme, and [sequence, sequence] are outputs of service providers such as BioLabs and Invitrogen.

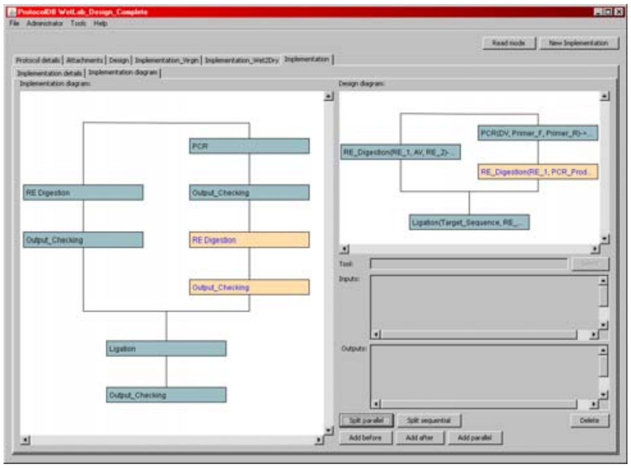


Figure 6 – *Wet* implementation workflow

For example, the design task PCR links two complex conceptual types [sequence] and {sequence} therefore a workflow that implements it receives a sequence as input and produces a set of sequences. The design task PCR is linked to its implementation composed of nine services as follows $T_{PCR}\equiv (S_1 \oplus S_2 \oplus S_3 \oplus S_4 \oplus S_5 \oplus S_6 \oplus S_7 \oplus S_8) \otimes S_9$ where S_9 is the thermocycler service (see Figure 7).

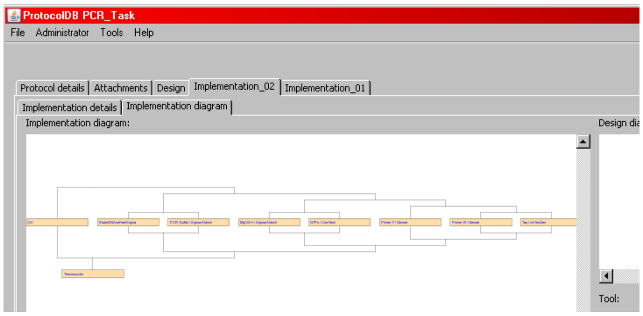


Figure 7 - Implementation of PCR design task

The sub-cloning workflow was implemented at TGen with the laboratory thermocycler, however the other services could be selected with respect to their ability to optimize criteria of interest, e.g., cost and quality, as discussed in Section 4.

4. FINDINGS

4.1 Workflow data model

The two layer model with a semantic conceptual representation mapped to one or more implementations was adopted once the benefits of the approach were measured. The benefits of using a domain ontology to describe scientific workflows include the ability of (1) linking several implementations to a given design (versions, simulations, etc.), (2) comparing different workflows by defining notions of similarity, (3) re-using and importing existing workflows, and (4) annotating execution data and reasoning on data provenance.

The adoption of the approach was first difficult because scientists are not yet familiar with ontologies and they often design their workflows with an implementation-driven approach. They select the resources they know and combine them in a way that they know how. The ProtocolDB method requires to thinking a workflow in an orthogonal way and record first the scientific objects involved and the scientific aim of each task and that phase was not natural. However, the ability to record several implementations was welcome as workflows are often revised.

A challenge that needs to be addressed is a generic semantic representation of *services* that supports workflow design and recording and that is compatible with semantic data annotation. Because the dataflow can be annotated with a domain ontology and because the scientific aim of each service is to transform the dataflow, services should be represented in terms of a domain ontology. In ProtocolDB the input and output of each service is mapped to a complex datatype (i.e., set, list, record) of conceptual basic types (e.g., sequence, gene). The service itself may be mapped to a conceptual relationship. The domain ontology extended with the complex conceptual datatypes and the services constitute a service graph or *semantic map* [7]. From that initial mapping, the dataflow of workflows is expressed in terms of the domain ontology. With this approach, at execution time, the dataflow can be automatically annotated with the concepts describing the input and output of each service therefore generating information that can further be analyzed for provenance or integrated with the results of other workflows. But existing formats to represent services such as Web services and BioMoby are not compatible with a semantic map [8]. Moreover the various metadata that could be used to predict the performance of workflow implementations are not expressed in existing formats.

None of the workflows we collected at TGen required loops but there is a vivid discussion among our scientific collaborators about the need for looping operators on the model. The current prototype does not model loops in workflows and none of the workflows studied during the evaluation phase included a loop. It is not clear yet what looping functionality would be necessary to extend the approach.

The top-down method for workflow design was not friendly to our scientific collaborators. This combined with binary operators did not meet users' expectations. This is because scientists design their workflows more easily from the bottom-up and wish to be able to connect more than two tasks at once without having to decompose the process into multiple binary steps.

4.2 Reasoning on workflows

The support for resource discovery was a desirable functionality. The conceptual workflow expressed in terms of a domain ontology becomes a resource discovery query. The first reasoning functionally that extends ProtocolDB [9] consists in mapping the design workflow to the semantic map to identify resources semantically suitable to implement each task. This is the reasoning approach of BiOnMap [10-11] that uses a deductive database to record the service graph and ontology in the extensional database and the reasoning rules against the ontology as the intentional database. Given a resource discovery query, BiOnMap returns various semantically suitable workflow implementations. In addition to exact mappings that allows the identification of services that exactly match the workflow design, BiOnMap can relax some constraints regarding the specificity of the service. Indeed, if a class C_1 is a sub-class of class C_2 in the ontology, if a design task *T* requires a service that takes C_1 and returns output *C*, if there is no service available with input C_1, and if there is a service *S* with input C_2 and output C, then service *S* can be selected to implement *T*. Implementation workflows answers of resource discovery queries are semantically equivalent to the design workflow. However, it is desirable to be able to transform automatically these implementation workflows into executable workflows.

Syntactic operability is the first requirement that needs validation for execution. In the ProtocolDB approach, that means that task compositions are validated not only semantically (this is a default feature of ProtocolDB) but also syntactically (when data formats correspond). This is achieved by finding existing connectors in the service graph that map the output format of a service to the input format of the next service in the workflow or by generating connectors automatically. Because the two data formats are mapped to the domain ontology, a schema mapping tool can exploit this semantic information to generate a connector more easily.

Scientists expressed the need to be able to conduct some test executions on workflow implementations to compare the results obtained from alternative service selections. Nevertheless, the execution of scientific workflows requires significant resources (effort, time, and supplies) and their optimization is often critical to the success of the research project. The ability of simulating digitally a workflow prior to its execution in the laboratory was demonstrated with the sub-cloning workflow. This method was used to evaluate the dataflow and the parameters used.

The benefits of exploiting resource metadata to predict the performance of the implementation was also a desirable feature. The fact that the method could use any quantitative measure for a cost function is definitely a benefit of the approach. However, the identification and collection of the metadata of interest to perform those predictions seems to raise some difficulties. In the wet lab, the comparison of the performance of implementations can be based on the characteristics (e.g., cost, quality, effectiveness) of reagents and products or tools used to perform a task. The selection of a method may depend on the allocated budget to the experiment, or on the supplies or equipment already available in the laboratory. Reasoning on the performance of each implementation provides the guidance needed to select the most effective implementation. This approach is similar to the functionality provided by LIMS although reasoning methods can evaluate optimized optimizations with respect to criteria specified by the user. A similar analysis can be used for *in silico* protocols as well as mixed ones.

5. RELATED WORK

Laboratory Information Management Systems (LIMS) support the integration of different functionalities in a laboratory, such as sample tracking (invoicing/quoting), integrated bar-coding, instrument integration, personnel and equipment management, etc. LIMS typically support *wet* workflows that coordinate the management of tasks, samples, and instruments and allow reasoning on business-like parameters such as ordering (e.g., invoicing) and organization (automation and optimization), but they do not offer resource planning with respect to a customized set of metrics. In contrast the ProtocolDB approach offers the ability to select the services that best fit the users' needs by evaluating the performance of each service in the workflow and ranking the options so that users may select the services that best meet their needs.

Scientific workflow systems such as Kepler [2] or Taverna [3] describe the scientific process from experiment design, data capture, integration, processing, and analysis that leads to scientific discovery. They typically express *digital* workflows and execute them on platforms such as grids. Workflows systems do not provide resource discovery functionalities such as presented in this paper where service composition plans are ranked with respect to a customized metrics. In addition, a large amount of scientific workflows mix wet and digital tasks. Experiments are first designed and simulated with digital resources in order to predict the quality of the result or to identify the parameters suitable for the expected outcome. The ProtocolDB approach can provide service discovery and planning for any kind of workflows, even those that mix manual (wet) tasks and digital ones.

Business processes are modeled with Business Process Modeling Notation (BPMN) and the Unified Modeling Language (UML). BPMN represents objects (events, activities, and gateways), connections (sequence, message, and association), and classifications and annotations of activities. However, the model does not support the semantic or syntactic description of the data flow. UML represents objects, attributes, operations, and relationships as well as the dynamic behavior of a system. Workflows could be represented in UML at a significant cost in terms of complexity and friendliness to the users. Indeed, the UML framework is composed of 13 different diagrams each capturing a viewpoint on a system. UML does not easily represent a conceptual workflow linked to multiple implementations and versions.

Service discovery systems typically support the identification of services suitable to implement a specific task. Criteria for discovery are typically syntactic (e.g., input or output format of a Web service) or semantic (i.e., what the service does) but very little has been done to support discovery through additional metadata. The optimization of service selection typically handles a single measure and has focused on quality. In contrast the ProtocolDB approach aims at supporting multi-dimensional performance criteria based on a variety of metadata attached to services. This approach offers multiple views of the workflow and provides the ability to compare different implementations.

6. CONCLUSION AND FUTURE WORK

We report on the testing of workflow modeling and recording with ProtocolDB[5] in the context of scientific applications at TGen. The two-layer representation exploiting a domain ontology to capture the scientific aim of each task and a service graph to specific the implementation is a valuable feature of the approach. However, the user interface needs significant improvement to make the internal representation of workflow more transparent and more friendly to the user. Although scientific workflows are not all completely digital, the ability to generate workflows in a format compatible with Taverna or Kepler for their execution is a desirable feature. Our current work is devoted to the integration of reasoning functionalities by integrating BiOnMap deductive approach to support resource discovery and syntactic validation.

7. ACKNOWLEDGMENTS

We wish to thank all the students of the Professional Master Program in Computational BioSciences who tested ProtocolDB. Our greatest thanks to our multiple collaborators from TGen and other institutes who have provided time and effort to evaluate of the system. In particular we wish to thank Marta Janer, Institute for Systems Biology, Jeffrey W White. U.S. Water Conservation Laboratory, U.S. Department of Agriculture and our multiple collaborators at TGen over the years including Anna Joy, Dominique Hoelzinger, Christian Beaudry, Mike Berens, Joachim Petit, Maliha Aziz, Jeff Kiefer, Meraj Aziz. We acknowledge and thank particularly Nathalie Meurice for providing the protein superposition workflow and Sukru Tuzmen for the sub-cloning workflow used for the evaluation. We acknowledge Piotr Wlodarczyk for the implementation of the first ProtocolDB prototype. This research was partially supported by the National Science Foundation[6] (grants IIS 0431174, IIS 0551444, IIS 0612273, and CNS 0849980).

8. REFERENCES

[1] N. Townsend, M. Waugh, M. Flattery, and P. Mansfield, "LIMS: Meeting the challenge of modern business," *American Laboratory*, vol. 33, 6, pp. 34, 2001.

[2] B. Ludascher, I. Altintas, C. Berkley, D. Higgins, E. Jaeger, M. Jones, E. A. Lee, J. Tao, and Y. Zhao, "Scientific Workflow Management and the KEPLER System," *Concurrency and Computation: Practice and Experience, Special Issue on Scientific Workflows*, vol. 18, 10, pp. 1039-65, 2005.

[3] T. Oinn, M. Addis, J. Ferris, D. Marvin, M. Senger, M. Greenwood, T. Carver, K. Glover, M. R. Pocock, A. Wipat, and P. Li, "Taverna: a tool for the composition and enactment of bioinformatics workflows," *Bioinformatics*, vol. 20, 17, pp. 3045-54, 2004.

[4] N. Hashmi, S. Lee, and M. P. Cummings, **"**Abstracting Workflows : Unifying Bioinformatics Task Conceptualization and Specification Through Semantic Web Services,**"** In Proc. W3C Workshop on Semantic Web for Life Sciences, Cambridge, Massachusetts, USA, 2004.

[5] E. Deelman, J. Blythe, Y. Gil, C. Kesselman, G. Mehta, S. Patil, M.-H. Su, K. Vahi, M. Livny, "Pegasus: Mapping Scientific Workflows onto the Grid," In Proc. Grid

[5] The ProtocolDB system is currently under development at Arizona State University and is available at bioinformatics.eas.asu.edu/siteProtocolDB/indexProtocolDB.htm.

[6] Any opinion, finding, and conclusion or recommendation expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

Computing,, Nicosia, Cyprus, Lecture Notes in Computer Science, Springer, vol. 3165, pp. 11-20, 2004.

[6] J. Mestres, "Gaussian-based alignment of protein structures: deriving a consensus superposition when alternative solutions exist, " *Journal of Molecular Modeling*, vol. 6, pp. 539–549, 2000.

[7] H. Ménager, Z. Lacroix, and P. Tufféry, "Bioinformatics Services Discovery Using Ontology Classification," In Proc. First IEEE International Workshop on Service Oriented Technologies for Biological Databases and Tools, In conjunction with ICWS/SCC, Salt Lake City, Utah, USA, IEEE Press, pp. 106-113, 2007.

[8] Maliha Aziz, Zoé Lacroix, "ProtocolDB: classifying resources with a domain ontology to support discovery", In Proc. 1st International Workshop On Resource Discovery, in conjunction with IIWAS, ACM, pp. 462-469, 2008.

[9] M. Kinsy, Z. Lacroix, C. Legendre, P. Wlodarczyk, and N. Yacoubi Ayadi, "ProtocolDB: Storing Scientific Protocols with a Domain Ontology," In Proc. Web Information Systems Engineering - International Workshops, Nancy, France, Lecture Notes in Computer Science, Springer, vol. 5333, pp. 998-1009, 2007.

[10] N. Yacoubi-Ayadi, Z. Lacroix, and M.-E. Vidal, "A Deductive Approach for Resource Interoperability and Well-Defined Workflows," In Proc. Workshop on Semantic Web and Web Semantics, Lecture Notes in Computer Science, Springer, pp. 998-1009, 2008.

[11] N. Yacoubi-Ayadi, Z. Lacroix, and M.-E. Vidal, "BiOnMap: a deductive approach for resource discovery," In Proc. 1st International Workshop On Resource Discovery, in conjunction with IIWAS, ACM, pp. 477-482, 2008.