What-If Analysis and Debugging Using Provenance Models of Scientific Workflows

Gulustan Dogan

***Abstract*—Scientific papers are the results of complex experiments done using large datasets. A researcher reading a scientific paper will not totally comprehend the ideas without learning the steps of the experiment and understanding the dataset. As this is an accepted fact, the idea of including the experimental work while publishing scientific papers has been around for many years. First, the steps were written as computer scripts and data was distributed assuming that all scientists were skilled programmers with intensive computer knowledge. Since this was not an efficient solution, the idea of scientific workflows arose. Scientific workflows illustrate the experimental steps taken to produce the scientific papers and provenance models capture a complete description of evaluation of a workflow. As provenance is crucial for scientific workflows to support reproducibility, debugging and result comprehension, they have been an increasingly important part of scientific workflows. In our paper, we argue that scientific workflow systems should support what-if analysis and debugging in order to allow users do modifications, see the results without actually running the workflow steps and be able to debug the workflows.**

***Index Terms*—Escience, provenance, scientific workflows, visualization.**

I. Introduction

Today scientific works contain several complex steps and there is a higher need for an automation to illustrate the steps they follow and present the data they use [1]. Traditional way of keeping laboratory notebooks is not an efficient way anymore because scientists want to share their experiments with their colleagues, they want to be able to easily reproduce, duplicate and maintain their scientific work and data. This goal has been named as "reproducible research" by computer and computational scientists [2].

The motivation of reproducible search led the geophysicist Jon Claerbout to the idea of standard of makefiles for construction of all the computational results in papers published by Stanford Exploration Project in 1990s [3]. After that time, various solutions have been proposed such as a markup language that can produce all of the text, figures, code, algorithms, and settings for the computational research. However the solutions often assumed that the scientists are skilled programmers with high computer knowledge. As a result, these attempts failed to become a standard because not all scientists have the programming skills that these approaches require. At this point, commercial and open source scientific workflow systems started to be developed to allow scientists automate the steps taken during their research without going into burdens of scripting [4]. We can list some popular scientific workflow management systems as myGrid/Taverna [5], Kepler [6], VisTrails [7], and Chimera [8].

Provenance is defined broadly as the origin, history, and chain of custody, derivation or process of an object. In other disciplines such as art, archaeology, provenance is crucial to value an artifact as being authentic and original. In computational world, as all kinds of information is easily changed, provenance becomes important way of keeping track of alterations [9]. Although scientific workflows will contribute to all science fields by their feasible characteristics, provenance management should be a concern too in order to have an understanding of how the results are obtained. Therefore workflow systems automatically capture provenance information during workflow creation and execution to support reproducibility [10]. Having this motivation, workflow provenance has been studied by several approaches, but research pointing out the fact that workflows with provenance models should support what-if analysis has not been done yet. What-if analysis refers to a set of actions which will help scientists forecast what will happen if they change a parameter, a function, a dataset in their experiments. For instance a researcher who has built a scientific workflow looking for common DNA patterns in cancer patients might want to run the same research on a different dataset. Experiments working on big data can take several days, it is time consuming for the researcher to run the experiment and then get an error. Debugging can take a lot of time. However the what-if analysis tool that we propose collects the execution graphs and labels them as bad-good runs and builds intelligence. With our tool when the scientist connects the workflow to a different dataset, based on the repository of good-bad runs, our tool makes a prediction of what can go wrong. This gives an insight to the scientist without running the experiment and saves time and effort.

II. Background

The idea of documenting the provenance of a data item comes from the arts, but recently science has taken a great deal of interest in documenting the steps, data sets and processes used in a research result. When the programs and datasets all resided within a lab or closed set of people, there was importance in documenting the data and process but now it has become almost the imperative. In this section we would like to give some background on workflow provenance.

Information gathered during workflow execution can be structured as a workflow provenance. Workflow provenance captures a complete description of evaluation of a workflow, and this is crucial to verification [11]. In addition, it can be

Manuscript received August 17, 2015; revised October 21, 2015.

Gulustan Dogan is with the Yildiz Technical University, Istanbul, Turkey (e-mail: dogangulus@gmail.com).

DOI: 10.7763/IJET.2016.V8.930

used for fault tolerance adding debugging support, performance optimization by allowing modifications, result reproduction from earlier runs to explain unexpected results and experiment preparation for publication [12]. There are several steps in creating the final results on which the conclusions are based. The methodology that they used in deriving the results is almost as important as the results. It is required to document accurately the data sets and programs (application and user written) that were used in the development of the paper. For example a paper could have made use of laboratory-developed data sets that were analyzed by a statistical or through an algorithmically derived analysis. The methodology may mean using especially selected software or it may mean a simulation or a statistical test from a particular program's output. Often the processes that are used come from outside the laboratory or research center and the one used is one of very many possibilities. A research result may employ software that is produced elsewhere and the authors may have made a choice of which software to use based on knowledge of which works best. All this should be documented and is as part of the provenance.

One of the hallmarks of scientific research is that others can duplicate it. This allows validation and moreover presents the additional research ideas that a paper creates. Although the processes are straightforwardly connected, having clear workflow and data provenance would enhance the reproducibility of the paper's results or additional examinations.

To efficiently document the steps, the methodology and the data used, workflow tools in scientific research and computational research is becoming more and more accepted. To more clearly illustrate the uses of workflows, we would like to give examples. Bioinformatics is an area with various different datasets; most of the time researchers have to combine data between the databases, which is a very time-consuming and hard task. In a research on a disease in cattle caused by flies injecting parasites to the cattle's blood, scientists work on trying to figure out the resistant genes. This research includes many teams in different locations, hence many datasets and complicated processes. They create workflows of this experiment with Taverna workflow management system [5] and publish their paper with the workflow. The scientists can clearly observe the advantages of this approach, their colleagues reading the paper can easily comprehend their work as they see the steps of the experiment, new members of the teams quickly get adapted to the project and data manipulation is automated. An example of the usage of workflows in computations will be a Web Service Workflow. Most workflow systems use web service technology intensively since nowadays web services are the most popular way of opening specific information to web users. A web service is invoked, the web service actor outputs the retrieved data sequence and it is displayed in multiple formats such as XML, HTML. This is a very tedious task because it involves data retrieval and manipulation but when these processes are illustrated as a workflow presenting the steps clearly, the work is easily referenced and repeated.

After scientific workflow management tools emerged, the developers began to observe that people are emailing the workflows to each other or putting workflows to web sites. This observation led them to the idea of a community group like myExperiment.org and Kepler Component Repository [6] that will serve as an open repository where people can share their workflows, search for workflows. These repositories have various workflows from many disciplines such as biology; computer science, social sciences, physics and they adapt a social web approach. Some of the workflows that are stored in the repositories can be listed as follows to give the reader an idea about their diversity, a disease recovery workflow which finds a disease relevant to a query string, a workflow retrieving a number of sequences from mouse, human, rat species using biology soaplab services, aligning them and returning a plot of the alignment result, a workflow executing a web service remotely to extract gene sequences and returning them in XML format, a multihop routing simulation workflow, a Lotka-Volterra workflow solving the classic Lotka-Volterra predator prey dynamics model.

The literature generally divides provenance into data and workflow provenance [13]. Data provenance gives a detailed record of the derivation of a piece of data that is the result of a transformation step [14]. Workflow provenance is the information or metadata that characterizes the processing of information from input to output [10]. Workflow provenance will be the concern of this paper and we will elaborate more on it in the other sections.

III. Related Work

Not much work has been done on debugging in scientific workflows. There is some research that points to the open areas in scientific workflows. In one of these works, the authors have stated that the workflow that the user has provided and the executed workflow might have some differences as the workflow management system might make some changes to execute the workflow. The fact that the differences between the user-specified workflow and executed workflow should be saved for debugging purposes is stated. By saving the two workflows, the user can trace how the specification they provided evolved into an executable sub-workflow. Although they have mentioned for a need for debugging in their work, they have focused on data management challenges in their work [15].

Altintas *et al*. have investigated the provenance support in Kepler workflow systems [16]. They have a provenance recorder that sends out notifications of events and occurring errors. By using provenance their system can report which actors were executing with which inputs at the time of the error. But their debugging mechanism is different than the one that we are proposing in this work. Rather than informing the user the possibly wrong designed parts of the workflow, it communicates to the user in which step the execution failed. Our model is different in the sense that rather than presenting a point of failure to the user, we inform the user about the potentially wrong subgraphs. This gives the user a broader view and helps the user in designing the workflow correctly. As we do mining over multiple historical workflows, we build an intelligence of the workflow and this module also becomes a base for what-if analysis framework.

A system called Panda is demonstrated in another work [17]. Their debugging mechanism is manual. User traces provenance logs to find out the cause for an error. Our

proposed system is superior to their mechanism as our system looks to past executions to give an educated guess of what the errors can be. It saves more time to the user and certainly a task too complicated to be done manually.

To our knowledge, any research on what-if analysis on scientific workflows has not been done yet.

IV. What-If Analysis and Debugging in Workflows

It is a fact that to be generally accepted a work should be repeatable by others. When reports of a duty is published with the workflow, a user reading the report might get curious about the results if some factors are changed or due to the possible occurrence of unexpected changes in the environment a need to repeat the processes may arise. Because of possible complexities in workflows and time constraints, it might not be efficient for users to rerun the processes following all the steps in the workflow. Reproduction without making any changes and editing workflows are supported features in workflow tools but there is no support for what-if analysis.

At this point, we argue that what-if analysis support becomes a crucial feature for workflow tools to support. If workflow tools support what-if analysis, it should be possible to carry out what-if analysis in an efficient manner with only the changed sub-graphs being recomposed. Fast and extensive what-if analysis being done with little manual effort using the built-in wizards, users will see how the tasks will turn out with the modifications. By the help of what-if analysis toolkit the user will not go into burden of running the experiment with the changed parameters and painful modification process will be prevented.

It is important for workflows to support modifications because changes should be captured and understood in order to run a previous computation in a new environment [18]. If we extend the ideas to allow a user modify the process provenance, the system will be more powerful allowing the researchers to do what-if analysis. What-if analysis will give the researcher the opportunity to assess potential changes before actually making them. This will be advantageous for them because before doing the tasks in real life settings which will be time and energy consuming for them, they can simulate the run with no cost and they can safely explore the varying input assumptions and scenarios.

We would like to list a few possible situations where what-if analysis support will ease user's life significantly. To begin, a user may want to try a slightly different way of doing the same procedure. When changing the workflow, without a what-if analysis module, each time the user have to run the workflow and see the output. In scientific experiments, running workflows take a lot of time so it is very time inefficient. Besides most of the workflows lack a powerful debugging mechanism, even the users go through burden of changing and running the workflow multiple times, they might not really understand the bugs. With the framework we are proposing, in addition to the traditional debugging; the system has the ability of making an educated guess of which parts might be failing by mining the historical workflows. In workflow tools without what-if analysis support any change in data and processes force users to rebuild and rerun the workflow.

What-if analysis functions will also serve as a background for debugging ability of a workflow. The prediction based on historical workflows serves both as a debugging and what-if analysis module. Once the user starts the what-if process, the result informs the user whether this workflow will work or not. This serves also as valuable debugging information besides the traditional debugging messages that tell the user at which step the workflow stopped working. During modification of processes and data of the original work, if the workflow execution does not give the desired result, users will have the chance to debug and correct their workflows instead of rebuilding the experiment workflow. They will have an idea of faulty steps that ruin the workflow execution and will consider only changing these steps instead of going over all steps. This will save time, which is valuable and irreplaceable. It could be analyzed what will happen if the faulty sequences marked bad are replaced by good patterns. If the system starts acting normal when the bad sequences are exchanged with good sequences then the what-if analysis had saved a lot of time to the user.

V. Case Study

Workflows are the records of steps taken to do a complex task, they become a solid reference later in repeating, sharing, modifying, controlling the same task. Therefore it is important to keep the workflow original and error free. A workflow generating meaningful results can start showing faulty behaviors. At this point we argue that there should be a way in workflow systems to debug and find out the responsible data or process nodes for erroneous behavior in workflow systems.

In the scenario of one of our previous work, when workflows are executed historical workflows are created and according to the results they are marked as bad or good historical workflows [19]. Once labeling is done, the tool processes the provenance graphs of labeled results to generate discriminatory features. Branches and nodes of good and bad workflows are compared [20]. The problematic nodes and edges are found. However most of the time, the cause of anomaly is not local to a node or edge, it is due to unexpected sequence of nodes/edges. Doing sequence/subgraph mining, the frequent culprit sequences or subgraphs are found under the assumption that frequent means at least three times and these sequences are labeled as bad patterns. A discriminative feature, in this context, refers to a subgraph of nodes that is correlated with the occurrence of bad labels. As a result of identifying bad patterns, it could be concluded that data fusion graphs including good patterns lacking bad patterns will display correct results [21]. We evaluated the tool DustDoctor using workflow provenance graphs and presented a real life case study to show that such problems exist. The workflow of the real life case study is given in Fig. 1. To see the whole picture in detail, reader is advised to visit the link, as it is a big detailed picture it cannot be captured in this paper. In Fig. 2, an example of a correct workflow graph is given. The tool takes correct workflows as a reference for finding out the malfunctioning components of faulty workflows as illustrated in Fig. 3. Fig. 4 illustrates a faulty workflow.

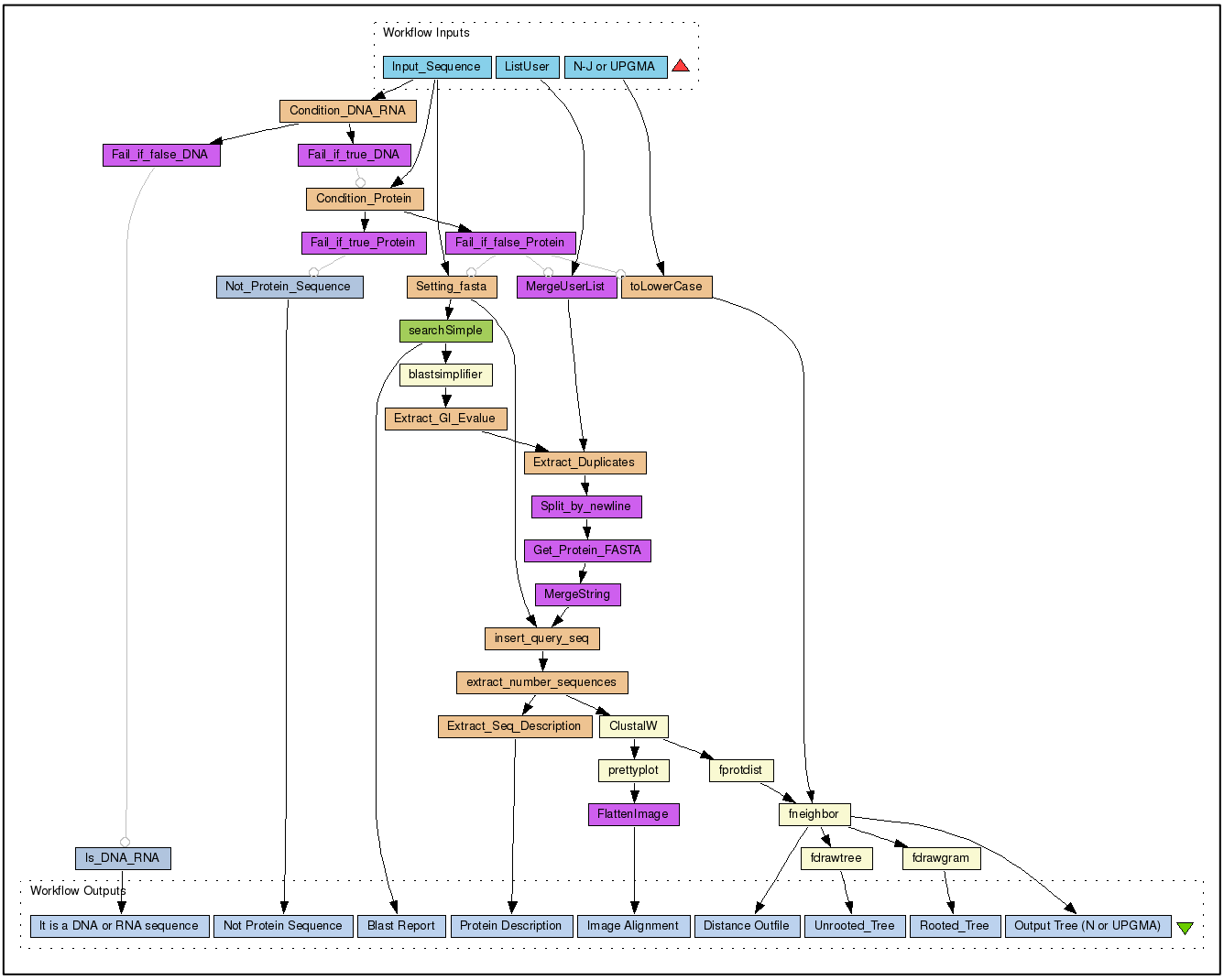

Fig. 1. Protein sequence analysis workflow [19].

label: 1
Input Sequence => Condition DNA RNA
Input Sequence => Condition Protein
Input Sequence => Setting fasta
ListUser =>MergeUserList
N J or UPGMA =>toLowerCase
Condition DNA RNA =>False if false DNA
Condition DNA RNA => Fail if true DNA
Fail if true DNA => Condition Protein
Condition Protein => Fail if true Protein
Condition Protein => Fail if false Protein
Fail if true Protein => Not Protein Sequence
Not Protein Sequence =>WorkflowOutput
Fail if false DNA =>Is DNA RNA
Setting fasta =>searchSimple
searchSimple =>WorkflowOutput
searchSimple =>blastsimplifier
blastsimplifier => Extract GI Evaluate
Fail if false Protein => Setting fasta
Fail if false Protein =>MergeUserList
Fail if false Protein =>toLowerCase
MergeUserList => Extract Duplicates
Extract GI Evalue => Extract Duplicates
Extract Duplicates => Split by newline
Split by newline => Get Protein FASTA
Get Protein FASTA =>MergeString
MergeString =>insert query seq
insertqueryseq => extract number sequences
extract number sequences => Extract Seq Description
extract number sequences =>ClustalW
ClustalW =>prettyplot
Prettyplot =>FlattenImage
Extract Seq Description =>WorkflowOutput
FlattenImage =>WorkflowOutput
toLowerCase =>fneighbor
Is DNA RNA =>WorkflowOutput
Setting fasta => insert query seq
Fneighbor =>fdrawgram

Fig. 2. Correct workflow.

Clustalw =>fprotdist
toLowerCase =>fneighbor

Fig. 3. Malfunctioning components.

label: 0
Input Sequence => Condition DNA RNA
Input Sequence => Condition Protein
Input Sequence => Setting fasta
ListUser =>MergeUserList
N -J or UPGMA =>toLowerCase
Condition DNA RNA =>False if false DNA
Condition DNA RNA => Fail if true DNA
Fail if true DNA => Condition Protein
Setting fasta =>searchSimple
searchSimple =>WorkflowOutput
Setting fasta => insert query seq
searchSimple =>blastsimplifier
blastsimplifier => Extract GI Evaluate
Fail if false Protein => Setting fasta
Fail if false Protein =>MergeUserList
Split by newline => Get Protein FASTA
Get Protein FASTA =>MergeString
MergeString =>insert query seq
insert query seq => extract number sequences
extract number sequences => Extract Seq Description
extract number sequences =>ClustalW
ClustalW =>prettyplot
Fneighbor =>WorkflowOutput
Fneighbor =>fdrawtree
Fdrawgram =>WorkflowOutput
Extract Seq Description =>WorkflowOutput
toLowerCase =>fneighbor
ClustalW =>fprotdist

Fig. 4. Faulty workflow.

DustDoctor adapts algorithms borrowed from previous discriminative mining literature to analyze data fusion flow graphs; called provenance graphs, and isolates sources and conditions correlated with anomalous results. More specifically, the tool applies association rule mining [22] to identify all sequences of virtual nodes that most accurately correlate with bad results. This information is subsequently used to isolate malfunctioning components or filter out erroneous reports.

VI. Challenges

There are challenges that we foresee in implementing what-if analysis support. One challenge will be the standardization of what-if analysis scenarios. As workflows can belong to various domains such as scientific workflows, business workflows, the what-if analysis scenarios can be very diverse too. We believe that software engineers can overcome this challengeby designing a flexible framework after a careful requirement analysis in numerous domains. One other difficulty that might be faced will be the data mining without sufficient historical workflows. In case there are not enough good runs and most of the bad runs do not have common points, our tool might not be very successful. This is a common bottleneck of most of the data mining applications. We believe that this can be overcome easily in scientific domain. As in this domain same workflow is repeatedly used several times by multiple users and having enough good runs over time is a big possibility.

VII. Conclusion

In this paper we discussed what-if analysis issues in workflows, and briefly described the usage of what-if analysis tool, an implementation of a case study and how users of workflow systems might benefit from. We believe that scientific workflow systems should consider adding a what-if analysis framework to their tools as scientists will greatly benefit.

Acknowledgment

This research was sponsored by the Technological Research Council of Turkey and was accomplished under Project Number TUBITAK 2232 114C143. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official polices, either expressed or implied, of the Technological Research Council of Turkey or the Turkish Government.

References

[1] T. Abdelzaher, M. M. H. Khan, H. Ahmadi, and J. Han, "Dustminer: Finding symbolic bug patterns in sensor networks," *Distributed Computing in Sensor Systems*, pp. 131-144, 2009.

[2] T. Abdelzaher, M. M. H. Khan, L. Lou, and C. Huang, "Snts: Sensor network troubleshooting suite," in *Proc. 3rd IEEE Int. Conf. on Distributed Computing in Sensor Systems*, 2007, pp. 142-157.

[3] U. Acar, P. Buneman, J. Cheney, J. V. Bussche, N. Kwasnikowska, and S. Vansummeren, "A graph model of data and workflow provenance," in *Proc. Workshop on the Theory and Practice of Provenance*, 2010.

[4] R. Agrawal, R. Srikant *et al*., "Fast algorithms for mining association rules," in *Proc. 20th Int. Conf. Very Large Data Bases*, VLDB, vol. 1215, 1994, pp. 487-499.

[5] I. Altintas, O. Barney, and E. Jaeger-Frank, "Provenance collection support in the kepler scientific workflow system," *Provenance and Annotation of Data*, Springer, pp. 118-132, 2006.

[6] S. Bowers, T. McPhillips, B. Ludascher, S. Cohen, and S. B. Davidson, "A model for user-oriented data provenance in pipelined scientific workflows," Springer-Verlag, LNCS 4145, pp. 133-147, 2006.

[7] A. Chebotko, X. Fei, C. Lin, S. Lu, and F. Fotouhi, "Storing and querying scientific workflow provenance metadata using an rdbms," in *Proc. Second IEEE Int'l Workshop Scientific Workflows and Business Workflow Standards in e-Science*, 2007, pp. 611-618.

[8] J. Cheney, "Causality and semantics of provenance," *ArXiv preprint arXiv*, 2010.

[9] E. Deelman and A. Chervenak, "Data management challenges of data-intensive scientific workflows," in *Proc. 8th IEEE International Symposium on Cluster Computing and the Grid*, 2008, pp. 687-692.

[10] I. Foster, J. Vckler, M. Wilde, and Y. Zhao, "A virtual data system for representing, querying and automating data derivation," in *Proc. 14th Intl. Conf. on Scientific and Statistical Database Management*, Edinburgh, 2002, pp. 37-46.

[11] J. Freire, C. T. Silva, E. S. Steven P. Callahan, C. E. Scheidegger, and H. T. Vo, "Managing rapidly-evolving scientific workflows," *Provenance and Annotation of Data*, Heidelberg, Berlin: Springer, vol. 4145, 2006.

[12] R. Ikeda, J. Cho, C. Fang, S. Salihoglu, S. Torikai, and J. Widom, "Provenance-based debugging and drill-down in data-oriented workflows," in *Proc. 2012 IEEE 28th International Conference on Data Engineering*, 2012, pp. 1249-1252.

[13] M. M. H. Khan, H. Ahmadi, G. Dogan, K. Govindan, R. Ganti, T. Brown, J. Han, P. Mohapatra, and T. Abdelzaher, "Dustdoctor: A self-healing sensor data collection system," in *Proc. 10th International Conference on Information Processing in Sensor Networks*, 2011, pp. 127-128.

[14] D.Koop, C. E. Scheidegger, J. Freire, and C. T. Silva, "The provenanceof workflow upgrades,"in *Proc. of International Provenance and Annotation Workshop*, vol. 6378, 2010.

[15] L. Moreau, B. Lud äscher *et al*., "The first provenance challenge," *Concurrency and Computation: Practice and Experience*, 2007.

[16] B. Lud äscher, 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*, vol. 18, no. 10, pp. 1039-1065, 2006.

[17] J. Mesirov, "Accessible reproducible research," *Science*, pp. 415-416, 2010.

[18] S. Miles, M. Branco, and L. Moreau, "The requirements of recording and using provenance in e-science experiments," Technical Report, Electronics andComputer Science, University of Southampton, 2005.

[19] M. Monteiro. (2008). Workflow for protein sequence analysis. [Online]. Available: http://www.myexperiment.org/workows/124.html.

[20] T. Oinn, M. Addis, J. Ferris, D. Marvin, M. Greenwood, T. Carver, M. R. Pocock, A. Wipat, and P. Li, "Taverna: A tool for the composition and enactment of bioinformatics workflows," *Bioinformatics*, vol. 20, no. 17, 2003.

[21] K. Project, (2010). Kepler analytical repository. [Online]. Available: http://library.kepler-project.org/kepler/.

[22] S. B. Davidson and J. Freire, "Provenance and scientific workflows: Challenges and opportunities," in *Proc. the ACM SIGMOD International Conference on Management of Data*, Vancouver, BC, Canada, 2008, pp. 345-1350.

Gulustan Dogan is currently working at Yildiz Technical University, Istanbul, Turkey as an assistant professor. She worked at NetApp and Intel as a software engineer in Silicon Valley. She received her PhD degree in computer science from City University of New York. She received her B.Sc degree in computer engineering from Middle East Technical University, Turkey. She is one of the founding members of Turkish Women in Computing (TWIC), a systers community affiliated with Anita Borg Institute.