A Multi-Functional Architecture Addressing Workflow and Service Challenges Using Provenance Data

Mahsa Naseri
University of Saskatchewan
110 Science Place
Saskatoon, SK S7N 5C9
Canada
naseri@cs.usask.ca

Simone A. Ludwig
University of Saskatchewan
110 Science Place
Saskatoon, SK S7N 5C9
Canada
ludwig@cs.usask.ca

ABSTRACT
In service-oriented environments, keeping track of the composition process along with the data transformations and services provides a rich amount of information for later reasoning. Current exploitation and application of this information, which is referred to as provenance data, is very limited as provenance systems started being developed for specific applications. Therefore, there is a need for a multi-functional architecture, which would be application-independent and could be deployed in any area. In this paper, we present an architecture, which exploits provenance information to target the current challenges of workflows. These challenges include workflow composition, abstract workflow selection, refinement, evaluation, and graph model extraction.

Categories and Subject Descriptors
H.2 [**Database Management**]: General

General Terms
Algorithms, Design, Reliability, Experimentation, Management.

Keywords
Provenance, Workflow, Bayesian Networks, Hidden Markov Model, Partially Observable Markov Decision Process.

1. INTRODUCTION
In service-oriented environments, services with different functionalities are combined in a specific order to provide higher-level functionalities. The composition of services is usually referred to as workflows. In such environments, great numbers of workflows are executed to perform mostly scientific and rarely business experiments. The workflow activities are run repeatedly by one or more users and large numbers of result data sets in the form of data files and data parameters are produced. As the number of such datasets increases, it becomes difficult to identify and keep track of them. Besides, in these large-scale scientific computations how a result dataset is derived is of great importance as it specifies the amount of reliability that can be placed on the results. Thus, information on data collection, data usage and computational outcome of these workflows provide a rich source of information.

The execution details of a workflow, referred to as provenance information, is usually traced automatically and stored in provenance stores. Provenance data contains the data recorded by a workflow engine during a workflow execution. It identifies what data is passed between services, which services are involved, and how results are eventually generated for particular sets of input values. Data associated with a particular service, recorded by the service itself or its provider, is also stored as provenance information. Such data may relate to the accuracy of results a service produces, the number of times a given service has been invoked, or the types of other services that have made use of it.

The exploitation of provenance data is so limited in comparison to the efforts accomplished and the costs paid for gathering and storing this data [7]. The major applications of provenance can be summarized into trust assessment, workflow re-execution and validation, and workflow reduction. In the following, a brief introduction of the most common applications of provenance is provided:

- Assessing trust measurements and believability for data, the confidence on the workflow steps executed, and the trust of each individual service can be determined by using the information regarding the past data or previous executions of services and workflow processes.
- Validating the data is possible by doing reasoning on provenance data, and to check for example whether the services still produce the same results and the workflow is valid yet.
- The workflow can be reduced by checking the provenance data and finding tasks that have been run previously and their results are still available and valid.

Although the mentioned applications provide rich and valuable usages of provenance data, more can be done to take advantage of the stored history of the previous executions. The research done in the area of provenance focuses mostly on the phases a provenance component goes through, such as the capturing mechanisms as well as data retrieval, querying and visualization. Little effort has been invested in discovering general applications for provenance.

One of the unexplored applications of provenance is exploiting it for the purpose of learning. A large store of the previous executions of services and workflows, as well as their specifications, provide an appropriate data set for learning and knowledge discovery. Applying learning and knowledge discovery methods to provenance data can provide rich and useful information on workflows and services. Therefore, the challenges with workflows and services are studied to discover the possibilities and benefits of providing solutions by using provenance data.

In this paper, an architecture is presented which addresses the discussed workflow and service issues by exploiting provenance data. The specific contribution of the proposed architecture is its novelty in providing a solid basis for taking advantage of the previous executions of services and workflows along with artificial intelligence and knowledge management techniques to resolve the major challenges regarding workflows. The following sections of the paper are organized as follows: in Section 2, the motivation and requirements for such an architecture is discussed; in Section 3, the architecture is presented along with explanation of its components; Section 4 provides the implementation details of the architecture; and in the final section the conclusion is presented.

2. MOTIVATION AND REQUIREMENTS

In this section, we are going to discuss the knowledge requirements of each problem, and will argue how provenance data satisfies these requirements and provides a suitable platform for improving as well as optimizing the quality of the solutions to these problems. Workflow composition and selection methods require an expressive language that supports flexible descriptions of models and data to facilitate reasoning and automatic discovery and composition. Therefore, they mostly exploit the semantic descriptions of services as well as their QoS specifications from service repositories or service providers to perform the composition or selection. In [5], the authors discuss the requirements for workflow composition. These requirements can be summarized as follows:

- Workflows must be described at different levels of abstraction that support varying degrees of reuse and adaptation.
- Expressive descriptions of workflow components are needed to enable workflow systems to reason about how alternative components are related, the data requirements and products for each component, and any interacting constraints among them.

The requirements mentioned can be satisfied through provenance data. In a robust provenance system, provenance creation is performed by following a layered approach which fulfills the requirements of the workflow composition process. The first layer of such architectures represents an abstract description of the workflow which consists of abstract activities with the relationships that exist among them. The second layer provides an instance of the abstract model by presenting bindings and instances of the activities. The third layer captures provenance of the execution of the workflow including specification of services and run-time parameters. The final level captures execution time specific parameters including information about internal state of the activities, machines used for running, status and execution time of the activities. As the execution time specific parameters are also gathered in provenance stores, provenance data also includes the QoS specifications of services. Thus, service selection solutions can be applied to this data in order to automatically select appropriate services that provide some QoS requirements. Service providers may not be trustworthy enough to deliver the services based on the agreed-on QoS. On the other hand, the "validity period" of the agreement might have come to an end and no agreement updates might have been made afterwards. The ontological QoS specification of service providers are updated periodically while there might be lots of requests in each period. In case the QoS guarantees change during a period, the providers will not be able to satisfy the agreed-on thresholds. Or the service provider might not be able to provide the specifications at all. Using the history of previous executions, the provided QoS overcomes the inconsistencies between the guaranteed and delivered QoS values of services to some extent by providing an estimate of the QoS parameters of the services with regard to time.

As the provenance information maintains the records of previous execution details of workflows, it provides the facility to analyze, assess, and evaluate the behavior of a workflow as well as its performance. The performance of a workflow, its believability, improvements, and its future trend, etc. can be analyzed and evaluated through provenance data.

The workflow mining methods use the event-logs for discovering the patterns and mining the workflows, which keep track of a very small amount of information. The information provided in event logs is not enough for mining workflows with regard to all the mentioned workflow perspectives while much stronger reasoning and mining can be done over the data presented in workflow provenance.

To improve the efficiency of the composition and selection processes, previous executions of workflows and services can be used to augment these processes with more intelligence during the composition or selection. The feedback learned through previous runs secure the composition (or selection) from services that either do not have available resources, or do not satisfy the promised trust levels at a particular time. In case of the composition, the feedback of previous runs of the composed process will also be analyzed later to discover the possible deficiencies that might exist in the composed model. As more provenance information is gathered, the extracted workflow process models are refined over time and the structure is geared to improve the efficiency with regard to changes in data. These variations might include updates of the most frequently chosen paths, or assigning/changing the weights of the links in the model with regard to the rate of usage in time. These types of augmentations in the model also facilitate the process of refining or repairing a workflow model.

As mentioned earlier, the history of previous executions of workflows and services satisfies the requirements of addressing the discussed challenges. Apart from the requirements, it was discussed that the provenance data augments the challenges with more intelligence, efficiency, and reliability. Thus, there is a need for an architecture that facilitates addressing and solving all these issues by exploiting the provenance data.

3. ARCHITECTURE

In this section, the multi-functional architecture discussed earlier is presented along with its components.

Figure 1 outlines the overview of the architecture. The components include:

1. **Workflow Model Extraction and Discovery Component**: This component is responsible for extracting the workflow pattern and associations that exist among the relevant workflows previously run and executed. Two workflows are considered relevant if they are in the same area of interest. The extraction component discovers the hidden connections that might exist among services and were not known beforehand. It generates a policy graph of the relevant

services including all possible associations and paths that could exist between the services of similar functionality.

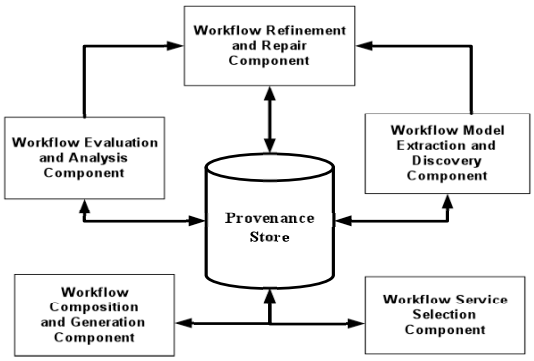

Fig. 1. The overview of the proposed multi-functional architecture.

2. **Workflow and Service Evaluation Component**: Workflows need to be assessed and analyzed to discover how trustful the composition of services are, therefore, in case the trust given by a workflow is not satisfactory, the workflow sequence can be repaired and improved. Another responsibility of this component is to identify the points in time at which a significant variation in trust occurs. This information can help us in identifying the parts of the workflow that are not providing the promised or required trust. Similar to workflows, the services are evaluated by this component. Large fluctuations of the QoS values of services are investigated to predict when in the future the service will not support the promised QoS.

3. **Workflow Repair and Refinement Component**: The repairment/refinement component takes advantage of the extracted policy graph of the workflow along with the assessment results of the evaluation component. The policy graph is traced to find a path that can replace the defective part of the workflow. The defective path is either inefficient due to lack of trust provision, or can not be executed any longer because of unavailable services. In case a service is predicted to not provide the promised non-functional requirements, the service is replaced by another service or services to provide a similar functionality.

4. **Workflow Composition and Generation Component:** The stored specifications of services and their states provide the facility of composing the services automatically. On the other hand, having the previous history of executions, provides the data, which is essential for learning, therefore, the composition will be done in a more intelligent way by exploiting the provenance data. This component receives the requirements and composes a workflow dynamically by taking advantage of the service specifications provided in the store.

5. **Workflow Service Selection Component:** In order to find the set of concrete services that match a single abstract service, service registries are looked at and matchmaking algorithms are applied to discover the matching services. The service discovery phase is much simpler if provenance data is used. Previous executions of workflows along with the workflow templates simplify the process of service discovery for a simple query. The set of suitable concrete services for the abstract workflow can then be selected more optimally by using the selection mechanisms along with the evaluations of previous executions.

4. IMPLEMENTATION

The implementation of the architecture is mostly based on artificial intelligence and statistical methods. The techniques being applied for discovering models of processes, and mining sequences, can be used for the case of workflow model extraction. Data mining algorithms, including the Generalized Sequential Pattern (GSP) algorithm, and the Apriori algorithm to discover sequential patterns are used. Methods used for event-data analysis are a set of techniques which are used for process discovery. Some of the methods used for discovering sequences and processes were previously exploited by the research done in the area of workflow pattern discovery from event logs. Techniques were developed for discovering workflow models from timed logs [4]. Our process discovery method to workflow model extraction is based on Bayesian reasoning. The method used by this component exploits the Bayesian structure discovery technique to learn the model and build the workflow policy graph. In order to model the problem as Bayesian structure discovery, the services serve as the nodes of the Bayesian graph, each having values representing different states a service provide. The links in the Bayesian graph represent the causal relationships that exist among the services. Therefore, the graph extracted from the provenance data depicts the workflow policy graph.

The evaluation component is based on statistical approaches such as Hidden Markov Models (HMMs) [2] and multivariate time series methods. These solutions are used for analyzing and evaluating the trust of the workflow, or to discover the trend of trust in workflow or services over time. This component evaluates the trust of a workflow using a Hidden Markov Model and specifies the trends of changes in workflow trust over time. Therefore, the time series evaluation method is applied on the data to provide assessments for trust and QoS values of workflows and services.

Many research efforts address the problem of workflow composition. One of the most studied areas of workflow composition is solving the problem via AI planning techniques. The state change produced by the execution of the service is specified through the precondition and effect properties which are provided in the semantic service descriptions. Our solution to the composition problem is based on probabilistic planning. As it is important to have a composition that is efficient in the sense that QoS specifications of services do not outweigh the benefits of reaching the goal. Therefore, probabilistic planning provides a reasonable solution as it generates the workflow by maximizing the probability of reaching the goals considering the QoS requirements.

Hierarchical Partially Observable Markov Decision Process (POMDP) [1] planning techniques [10] provide a suitable approach for composing services. Discrete POMDP models the relationship between an agent and its environment. Hierarchical POMDPs break a problem into many related POMDPS based on

action hierarchy. The original action set is partitioned such that it spans a collection of hierarchically-related smaller POMDPs which are referred to as subtasks. Each action is assigned to one or more subtasks and each subtask learns the policy over its subset actions using POMDP solving methods. The parameters of the POMDP, which include conditional transition probabilities, conditional observation probabilities, and rewards, are learned through the data available in the provenance store. The planning process is augmented with learning methods to make the composition as intelligent as possible.

In the case of abstract workflow selection, several works have addressed this issue proposing exact algorithms or heuristics to determine the appropriate concrete services for each individual component invocation or over the complete composite request. Our solution toward this problem relies on sensor scheduling, which is the problem of optimally choosing which single sensor to use at each time instance to minimize a cost function. Past observations together with past choices of sensors affect which sensor to choose at present. This problem perfectly matches the abstract workflow service selection. The sensors to be chosen at each time represent the concrete service that should be chosen at that time instance. The solution to the sensor scheduling problem selects an appropriate concrete service at each time instance while keeping the total values of QoS specifications as low as requested.

5. EVALUATION

In order to evaluate the architecture, different provenance systems were studied to investigate the one which best satisfies the data requirements for the components. Taverna [3], Triana [11], and Provenance Aware Service Oriented Architecture (PASOA) [9] are the provenance systems studied. Triana does not provide a separate provenance system; instead it has a rudimentary history tracking system that allows workflows to be stored with the interim states of the components in the workflow. Taverna is a workflow workbench that has a provenance model which captures both internal provenance locally generated in Taverna and external provenance gathered from data providers. The provenance data gathered in Triana is very limited in comparison to Taverna and does not support annotations. Although the PASOA project presents an architecture, which addresses issues such as provenance generation, representation and reasoning, its implementation is not complete and is just intended as a technology preview. In order to perform real world and valuable experiments with the architecture, Taverna was selected as a practical provenance system and will be expanded to incorporate the additional features of the proposed architecture.

The evaluations include assessing the accuracy and performance of the workflow model extraction component with regard to the graph provided. The refinement component will be assessed to observe the rate of improvements of the workflow. The behaviors of the components will be assessed in terms of scalability to observe the effect of different number of services on the model. The results of the components will be compared with on the fly solutions to investigate the influence of learning as well as the feedback fed into the components from previous executions.

6. CONCLUSION AND FUTURE WORKS

In this paper, a multi-functional architecture was proposed which addresses the current issues of workflows and services using provenance data. The components of the architecture, and its implementation details were discussed. The different techniques applied to the same problem will be compared with each other in terms of their execution time, support, and scalability. The proposed architecture will be augmented with other services to provide more functionality, robustness, and reliability. Components will return feedback to the provenance store to feed the provenance data with the information learned about the data. Thus, the stored data will get trained dynamically through time and the components will operate more intelligently.

7. REFERENCES

[1] Murphy, K. P., A Survey of POMDP Solution Techniques: Theory, Models, and algorithms, *management science*, 28 (1982).

[2] Rabiner, L. and Juang, B. An introduction to Hidden Markov Models. *IEEE Acoustics, Speech & Signal Processing*, (1986).

[3] Taverna, last retrieved from http://www.taverna.org.uk/, (2010).

[4] Aalst, W. M. and Dongen, B. F. 2002. Discovering Workflow Performance Models from Timed Logs. In *Proceedings of the First international Conference on Engineering and Deployment of Cooperative information Systems*, September 17 – 20, (2002).

[5] Gil, Y., Workflow Composition: Semantic Representations for Flexible Automation, Book Chapter, (2005).

[6] Wu, D., Sirin, E., Hendler, J., Nau, D., and Parsia, B. Automatic Web services composition using SHOP2. In Workshop on Planning for Web Services, Italy (2003).

[7] Sannella, M. J., *Constraint Satisfaction and Debugging for Interactive User Interfaces*. Doctoral Thesis. UMI Order Number: UMI Order No. GAX95-09398. University of Washington, (1994).

[8] Forman, G., An extensive empirical study of feature selection metrics for text classification. *J. Mach. Learn. Res.* 3, 1289-1305, (2003).

[9] PASOA Project, last retrieved from http://www.pasoa.org/index.html, (2004).

[10] Pineau, J., Roy , N., Thrun, S., A Hierarchical Approach to POMDP Planning and Execution, Workshop on Hierarchy and Memory in Reinforcement Learning (ICML), (2001).

[11] The Triana Project, last retrieved from http://www.trianacode.org/collaborations/, (2003).