Supporting Provenance in Service-oriented Computing Using the Semantic Web Technologies

Liming Chen and Zhuoan Jiao

***Abstract*—The Web is evolving from a global information space to a collaborative problem solving environment in which services (resources) are dynamically discovered and composed into workflows for problem solving, and later disbanded. This gives rise to an increasing demand for provenance, which enables users to trace how a particular result has been arrived at by identifying the resources, configurations and execution settings. In this paper we analyse the nature of service-oriented computing and define a new conception called augmented provenance. Augmented provenance enhances conventional provenance data with extensive metadata and semantics, thus enabling large scale resource sharing and deep reuse. A Semantic Web Service (SWS) based, hybrid approach is proposed for the creation and management of augmented provenance in which semantic annotation is used to generate semantic provenance data and the database management system is used for execution data management. We present a general architecture for the approach and discuss mechanisms for modeling, capturing, recording and querying augmented provenance data. The approach has been applied to a real world application in which tools and GUIs are developed to facilitate provenance management and exploitation.**

I. Introduction

PROVENANCE is defined, in the Oxford English Dictionary, as (i) the fact of coming from some particular source, origin, derivation; (ii) the history or pedigree of a work of art, manuscript, rare book, etc. This definition regards provenance as the derivation from a particular source to a specific state of an item, which particularly refers to physical objects. For example, in museum and archive management, a collection is required to have archival history regarding its acquisition, ownership and custody.

Provenance is an important requirement in many practical fields. For instance, the American Food and Drug Administration requires that the record of a drug's discovery be kept as long as the drug is in use. In aerospace engineering, simulation records that lead up to the design of an aircraft are required to be kept up to 99 years after the design is completed. In museum and archive management a collection is required to have archival history regarding its acquisition, ownership and custody.

In computer-based information systems, research on provenance has traditionally been undertaken in the arena of database systems under different banners such as audit trail, lineage, dataset dependence and execution trace [2] [3]. For example, the Chimera Virtual Data System [4] addresses data linage with the Chimera virtual data schema. Similar works were also described in [5] [6]. The common feature of these systems is that they try to trace the movement of data between data sources and obtain information on the "where" and "why" of a data item of interest as a result of a database operation. A separate thread of research, i.e. the so-called knowledge provenance, concentrated on explaining information provenance for Web applications [7] [8]. The research placed special emphasis on source meta-information and knowledge process information, in particular, the reasoning process used to generate the answer.

Recently, research on data provenance in service oriented computing has received growing attention [9] [10] [11] as the enabling Web/Grid service technologies and the infrastructure for Service Oriented Architecture (SOA), such as the Open Grid Service Architecture (OGSA), become mature and available. In a SOA, resources on the Web/Grid, including hardware, software code, application systems and knowledge, are regarded as services; and such services are brought together to solve a given problem typically via a workflow that specifies their composition. The running of an application programmed in a SOA style requires the enactment and execution of the workflow, which is referred to as a process. Web/Grid services are dynamic and distributed in nature, i.e. they can be published and withdrawn to/from the Web/Grid arbitrarily. This means a solution (a workflow) to a problem may not be always available or consists of the same set of services at different time of problem solving. Thus, recording and archiving how a result is derived becomes critical in order to validate, repeat and analyse the obtained results.

Data provenance in a SOA/OGSA is concerned with the entire execution history of a service workflow that leads to the particular result, i.e. evolving from traditional "data-centered" provenance towards "process-centered" provenance. An initial attempt has been made in myGrid project (www.mygrid.org.uk) where log files have been annotated and recorded for experiment validation and recreation [12]. A systematic research is conducted in the EU PROVENANCE project

Liming Chen is with the School of Computing and Mathematics, University of Ulster, Co. Antrim BT37 0QB, U.K. (e-mail: l.chen@ulster.ac.uk).

Zhuoan Jiao is with School of Engineering Sciences, University of Southampton, Southampton SO17 1BJ, UK. (e-mail: z.jiao@soton.ac.uk).

(twiki.gridprovenance.org/bin/view/Provenance) aiming to develop a generic architecture for capturing, recording and reasoning provenance data [13]. The project also intends to propose protocols and standards to formally standardize *provenance computing* in SOA/OGSA.

At the time of writing, most provenance systems focus on capturing and recording execution data passed between services within a workflow. Metadata about services, such as the quality of services, their parameters (functional and non-functional), and workflows are scarce and informal. There are no formal representation and common semantics. This imposes severe limitations on the interoperability, searchability, automatic processing capability and reasoning of provenance data, and ultimately the use and reuse of services.

This paper aims to tackle the aforementioned problems by exploiting the Semantic Web technologies, and our research contributions are: (1) introducing the conception of augmented provenance based on the characteristics of service-oriented computing, which enhances conventional provenance with rich metadata and formal semantics; (2) proposing a Semantic Web Service (SWS) based hybrid approach to supporting augmented provenance; (3) desiging and prototype implementing a system architecture for the proposed approach. Our work is motivated by the realisation that SOA/OGSA-based applications require extensive rich metadata in multiple facets, at multiple levels of granularities in order to make effective use of previous problem solving expertise. The central idea of the approach is to capture provenance data from the semantic descriptions of the web services, thus enabling the use of the Semantic Web technologies for provenance data representation and storage. We place special emphasis on semantics, particularly the ontological relationships among diverse metadata, which enables deep use of provenance by reasoning.

The remainder of the paper is organized as follows: Section 2 analyzes the characteristics of service-oriented computing from which we draw the conception of augmented provenance. Section 3 describes the proposed approach and its system architecture for managing augmented provenance. We give an application example in Section 4 and discuss our experiences and lessons in Section 5. Section 6 concludes the paper and points out some future work.

II. AUGMENTED PROVENANCE FOR SERVICE-ORIENTED COMPUTING

We have defined the concept of *augmented provenance*, after analyzing the key characteristics of provenance data in a SOA. We believe this is more instructive than trying to produce an all embracing conceptual definition. To help clarify our conception of augmented provenance and justify our proposed approach, we present below a motivating scenario that captures what we believe are the requirements of provenance in a SOA/OGSA..

A. A motivating scenario

This scenario is based on the UK e-Science project *Grid-enabled Optimisation and Design Search in Engineering* (GEODISE). Engineering Design Search and Optimisation (EDSO) is a computationally and data intensive process whereby existing engineering modeling and analysis capabilities are exploited to yield improved designs. An EDSO process usually comprises many different tasks. Consider the design optimization of a typical aero-engine or wing, it is necessary to (1) specify the wing geometry in a parametric form, (2) generate a mesh for the design, (3) decide which analysis code to use and carry out the analysis, (4) decide the optimisation schedule, and finally (5) execute the optimisation run coupled to the analysis code. Apparently a problem solving process in EDSO is a process of constructing and executing a workflow.

GEODISE aims to aid engineers in the EDSO process by providing a range of Internet-accessible Web/Grid services comprising a suite of design optimization and search tools, computation packages, data management, analysis and knowledge resources. In the GEODISE problem solving environment services are composed into a workflow which is subsequently enacted and executed. The executed workflow is described by a XML file which is stored in the database together with limited metadata such as the file's size, location, etc [14].

After the system was introduced to engineers, a number of questions have been raised regarding to the service and workflow reuse. For instance, engineers may want to find a workflow that uses a particular service *S1*; to find workflows that use a service with the similar algorithm to the algorithm used by *S1*, or to find a similar service to replace service *S1* used in the current workflow and re-run the workflow. To answer these questions, we identify a number of requirements for provenance data, as described below.

Firstly, provenance should include metadata at multiple levels of abstraction, namely process level, service level and parameter level. For example, a workflow instance with all its parameter settings and values is a provenance record for the data derived from it, but the workflow itself also needs provenance information, i.e. which workflow specification was it instantiated from, who enacted it, etc.

Secondly, provenance should include metadata in multiple facets. These may include knowledge provenance, e.g. what knowledge is involved and used; and the decision provenance, e.g. how a decision was arrived at, etc. Each facet of provenance has its roles and uses, and different applications have different emphases and requirements for provenance.

Finally, provenance is not only used to validate, repeat and analyze previous executions but, more importantly, to further advance investigation and exploration based on the previous results. In EDSO an optimisation can be performed using different services (algorithms), and each of them can generate different qualities of results. Engineers, particular novices, usually start a new design by looking at previous best design practices (workflows), and perform design search and optimization by changing constituent services and/or tuning control parameters of the previous workflows. This requires knowledge and decision trails become an indispensable part of the provenance.

B. *Provenance analysis and augmented provenance*

The essence of service-oriented computing is the sharing and reuse of distributed, heterogeneous resources for coordinated problem solving in dynamic, multi-institutional virtual organizations (VO). Service-oriented computing has the characteristics of dynamic service provisioning and across-institutional sharing, i.e. VOs are formed or disbanded on-demand. In such environments a workflow consists of services from multiple organizations in a dynamic VO. The success of workflow execution depends on domain knowledge for service selection and configuration, and a mutual understanding of service functionalities and execution between the service providers and consumers. The complexity of a problem solving process requires not only the execution data of a workflow (e.g. the inputs and outputs of services, the configuration of service control parameters), but also rich metadata about the services themselves (e.g. their usages, the runtime environment setting, etc.), in order to validate, repeat and further investigate the problem solving process at a later stage.

While specific domains or applications determine the actual levels of abstraction and interested facets of provenance, we can identify some common characteristics of provenance data in a SOA. First, SOA oriented provenance data contain both execution data and execution independent metadata. The metadata are centered on the key SOA entities, namely workflows, services and parameters.

Second, rich relationships exist among multiple levels and facets of metadata in SOA/OGSA applications. For instance, a workflow consists of services that in turn contain various parameters. Furthermore, services within a workflow, as well as the parameters of a service, may be organized in various ways. The relationships actually form a kind of knowledge model, which can be used to encode domain knowledge. Appropriate modeling of the metadata can facilitate the data retrieval and the discovery of new knowledge through reasoning. For example, a hierarchical tree structure could be used to model the "is part of" relation between workflows, services and parameters; ontological links could be used to denote semantic relations between services, parameters and commonly accepted types.

Third, not all provenance data can be captured automatically, especially those pertaining to knowledge and decision provenance. Annotation and commenting are therefore an important aspect of provenance. For example, in an EDSO experiment, engineers may annotate why a specific service or algorithm or a value for a parameter is selected. They may wish to annotate the performance of a particular service or the quality of overall results so that future designs can be improved based on the annotations.

Text comments and tagging have been traditionally used to add metadata, but they suffer limitations such as the lack of interoperability, the inability of automation, etc. It is obvious that formal modeling and representation of provenance data with explicit semantics are required in order to facilitate automatic, seamless access and sharing of the provenance data.

To differentiate from traditional provenance understanding, we introduce the concept of *augmented provenance*, defined as: the augmented provenance of a piece of data is the process that leads to the data, and the related semantic metadata of the process.

Although our motivating scenario and analysis are based on EDSO, it is not intended to be domain-specific. The scenario depicts the general features of and requirements for provenance in service-oriented computing. Therefore, the augmented provenance conception and the proposed SWS-based approach are broadly applicable to a range of service-based applications.

III. A SWS-BASED HYBRID ARCHITECTURE FOR AUGMENTED PROVENANCE

We propose a SWS-based hybrid architecture for creating and managing augmented provenance as shown in Figure 1. Central to the architecture is the use of SWSs for managing execution-independent metadata and a hybrid mechanism for handling the execution data. The architecture consists of a set of components, namely the Web/Grid Services (WGS), Semantic Web Service Repositories (SWSR), Workflow Construction Environment (WCE), Workflow Enactment Engine (WEE) and Augmented Provenance Management Services (APMS). These components communicate and interact with each other to enable effective and efficient management of augmented provenance, which we discuss in the rest of this section.

A. *A SWS-based perspectives*

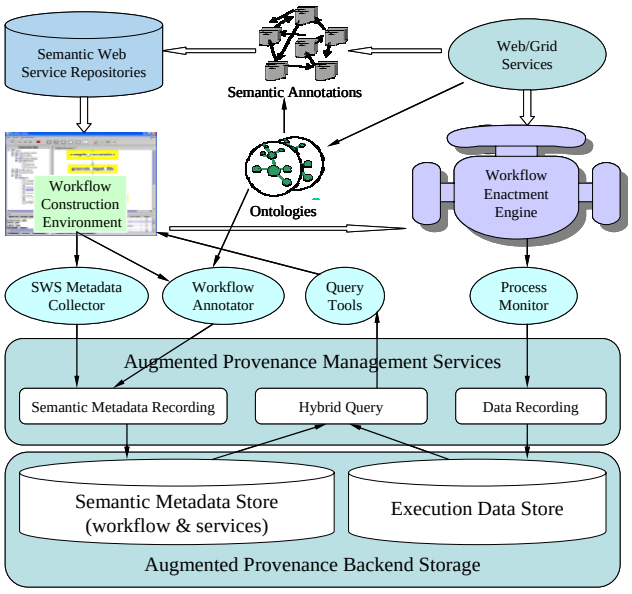

Fig. 1. The augmented provenance architecture

In service-oriented computing, distributed Internet-accessible services such as those contained in the WGS component, serve as the basic computing blocks in SOA/OGSA. As Web/Grid services are described in WSDL [1], published in UDDI (www.uddi.org) and invoked by SOAP, all these technologies provide limited support for service metadata and semantics,

[1] WSDL along with SOAP, RDF, and OWL are W3C standards, please refer to www.w3.org.

thus unable to produce directly augmented provenance.The SWS-based approach uses ontologies and semantic annotation for the acquisition, modeling, representation and reuse of provenance data. The rationales behind this approach are that (1) ontologies can model both provenance data and their contexts in an unambiguous way; (2) provenance data generated via semantic annotation are accessible, shareable and machine processable in a SOA/OGSA; and (3) the Semantic Web technologies and infrastructure can be exploited to facilitate provenance data acquisition, representation, storage and reasoning. More specifically, it will make use of semantic descriptions of Semantic Web Services to generate augmented provenance directly and other Semantic Web technologies such as ontology languages, semantic repository and reasoning for provenance data representation, storage and querying.

The foundation of the architecture is the SWSR component, which contains semantic descriptions of Web/Grid services. SWSR is based on SWS technology that complements current web service standards by providing a conceptual model and language for semantic markup. While the original goal of SWS is to enable the (total or partial) automation of service discovery, selection, composition, mediation, execution and monitoring in service computing, SWS does provide a mechanism for incorporating rich metadata, which can be utilised for provenance purpose. More concretely, SWSR consists of semantically enriched metadata describing the properties and capabilities of services in unambiguous, computer-interpretable form, which can serve as a source of a data item's augmented provenance.

The key enabling technology for SWS is service ontology that provides machine processable models of concepts, their interrelationships and constraints. Service ontology can be used to capture the background knowledge and vocabulary of a domain. For example, OWL-S (www.daml.org/services/owl-s) service ontology defines a number of terms and relationships to describe a service metadata. As an upper service ontology, OWL-S can be further extended based on domain characteristics and application requirements to accommodate domain-specific service description requirements. Semantic descriptions in SWSR are generated by applying service ontologies to services through an annotation tool provided by the Ontological Annotation component. SWSR provides the WCE with a pool of semantically described services through which the WCE can discover and select required services.

Critical to the success of our approach is the WCE component, which collects semantic metadata and records them in provenance stores. WCE allows users to discover and select required services from SWSR locally or on the Web/Grid to compose a service workflow for a given problem. The generated workflow will be passed onto WEE for binding and enactment.

With regards to the provenance, WCE can play three roles, i.e. extracting semantic metadata from service descriptions, generating workflow semantic metadata as part of augmented provenance and performing provenance queries. As WCE uses services from SWSR, the collection of selected services' metadata is straightforward. Each time a service is added into a workflow, the SWS Metadata Collector will retrieve the service's semantic metadata from SWSR and linked to the workflow. For a new workflow, semantic metadata has to be created on the fly because they do not exist in prior.

The Workflow Annotator component will operate in WCE and enable users to describe a workflow in terms of workflow ontology. Workflow's metadata could include a workflow identifier, its creator (i.e. individual or organization), problem solved, date, etc. In practice, an ontology-driven form can be generated automatically from the workflow ontology to help users capture relevant metadata. Some information may be collected directly from the workflow construction process such as date, time, and machine identifiers. Both workflow and service semantic metadata will be submitted to APMS for recording, and later be queried using the Query Tools.

Augmented provenance management services (APMS) are designed for managing augmented provenance data beyond the lifetime of a SOA/OGSA application. It provides recording (archiving) and querying interfaces for augmented provenance backend storage as well as additional administration functionalities such as authentication, authorization and housekeeping. In the context of a SOA/OGSA, provenance backend storage can be decentralized in multiple sites, and APMS are implemented as web services, thus facilitate web accessibility to provenance data and improve the scalability..

B. A hybrid mechanism

Augmented provenance contains execution data generated at the run-time, e.g. the values of inputs and outputs of services; as well as semantic metadata at the design time, e.g. the descriptive information about the workflows, services and parameters. The different nature of these two types of provenance data is reflected in the way they are captured, modeled, represented and stored. To support the heterogeneity of provenance data in a SOA/OGSA, a hybrid approach is adopted, i.e., the approach uses the Semantic Web technologies to handle a workflow's semantic metadata, and the database technologies to deal with execution-dependent process data, thus avoiding duplication and making maximum use of existing DBMS infrastructure. It also proposes a hybrid storage and retrieval mechanism to facilitate coordinated archiving and query of augmented provenance data.

The WEE is responsible for interpreting workflow scripts, binding individual constituent services with corresponding inputs, and invoking executions. A Process Monitor operating in the WEE will extract initial default or user-configured input variable names and values from the interpretation of a workflow script. It will then monitor the execution process of the workflow by querying the execution data repository periodically, thus intermediate and final output results from the workflow's execution could be captured.

As can be seen from the architecture, semantic metadata are collected from WCE and recorded to APMS's Semantic Metadata Store (SMS) via the Semantic Metadata Recording interface. Semantic metadata shall be represented in semantic web languages such as RDF or OWL. Semantic metadata backend store could be a semantic repositories such as 3Store [15] or instance store [16]. Normal workflow execution data will be collected from the WEE and recorded into APMS's Execution Data Store (EDS) via the Data Recording interface.

The execution data backend store could be any commercial database systems.

The APMS operates as follows: each time a workflow is built in WCE, the WCE will store a workflow template in SMS. This template will contain the overall semantic descriptions about the workflow; the semantic metadata for each of the constituent services, including each service's profile metadata and input/output metadata, and an auto-generated unique workflow template ID (UUID, Universally Unique Identifier, www.ietf.org/rfc/rfc4122.txt) as a handle for later reference. An executable workflow based on the workflow template is instantiated by providing values for the required input parameters, and the WEE will store the workflow instance in EDS and associate it with the workflow template ID. If a user reuses a previous workflow template to perform another run without changing the services and the sequence of service execution, the WCE will not record a new workflow template but the WEE will record another workflow instance under the same workflow template ID.

Based on the hybrid storage mechanism, querying augmented provenance data becomes flexible and efficient. A user can use ontologies to frame semantic queries, e.g. in terms of a service profile metadata or a workflow's metadata or a parameter's metadata or any combination of them. Once a workflow template is discovered, all its execution instances can be found from EDS based on the workflow template ID. Further search can be performed to find the set of executed workflows matching other search criteria (e.g. its creator, creation-date, input parameter-values, etc) using the database query mechanism.

The separation of semantic metadata and execution data has many advantages: Firstly, metadata can be formally modeled using ontologies and represented using expressive web ontology languages. This helps capture domain knowledge and enhance interoperability. Secondly, workflow execution usually produces large amount of data that have little added value for reasoning, and the traditional database systems are optimal for handling them. Finally, the hybrid query mechanism provides flexibility and alternatives – users can perform semantics based query or direct database query or a combination to meet application needs.

IV. Application Example

The proposed approach has been applied in GEODISE to manage augmented provenance for grid-enabled service-based EDSO, and in turn the provenance data are used to aid engineers in the design process by answering provenance-related questions. Figure 2 shows the provenance management system in GEODISE, which is described in detail below.

A. Creating semantic metadata

To manage augmented provenance in GEODISE we have built a number of EDSO ontologies, including domain ontology and service ontology, through extensive knowledge acquisition and modelling [17]. Figure 3 shows a fragment of the service ontology developed using Protégé OWL plugin (protege.stanford.edu/plugins/owl). The left column displays ontological concepts while the right column lists ontological properties. We regard a workflow as a composite service.

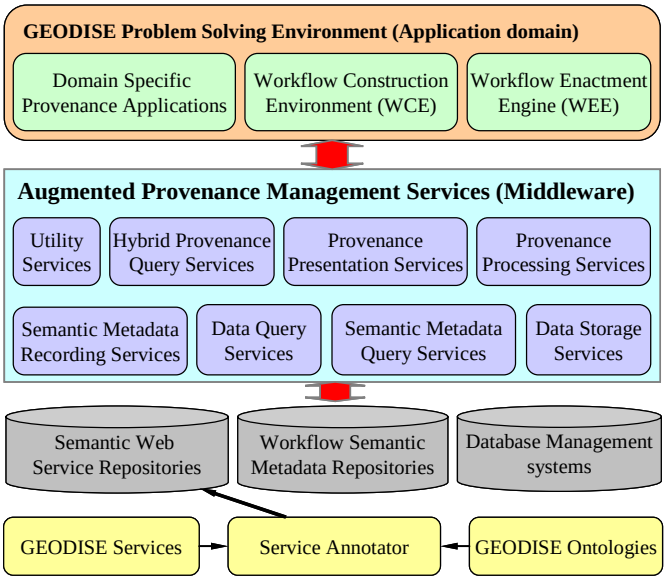

Fig. 2. The provenance management system

Therefore, the service ontology can be used to model semantic metadata for both services and workflows. EDSO service ontology is based on OWL-S upper service ontology. It further extends OWL-S to incorporate EDSO specific metadata such as algorithmUsed, dataPhysicalMeaning, dataUnitType, previousService, followingService, derivedFrom, etc.

We have developed semantic metadata annotation interfaces for capturing semantic metadata. A front-end GUI, known as Service Annotator [19], was developed to help users extract automatically service's metadata, which are then enriched using EDSO domain and service ontologies. The annotation API is also used to implement the Workflow Annotator wizard in WCE to capture and annotate workflow metadata during workflow construction process. The generated semantic metadata for both services and workflows are represented in OWL and stored in the Semantic Web Service Repositories and

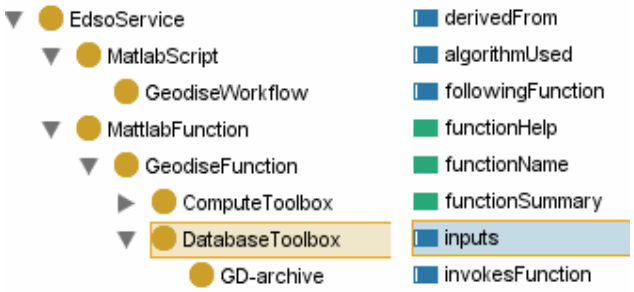

Fig. 3. GEODISE service ontology

Workflow Semantic Metadata Repositories respectively. Both repositories were implemented using the Instance Store technology [16], which provides recording and query interfaces for manipulating semantic metadata. The interfaces use the description logic based reasoning engine Racer [18] to reason over semantic metadata [19].

B. *Collecting and recording execution data*

GEODISE uses Matlab (www.mathworks.com) as its workflow enactment and execution engine. Therefore, input and output variables and their values can be captured and collected from Matlab workspace memory. Acquired execution data are managed by the GEODISE database toolbox [14]. The database toolbox exposes its data management capabilities to the client applications through Java API, as well as a set of Matlab functions. The Java API has been used by the workflow construction environment to archive, query, and retrieve the workflow instances for reuse and sharing; and the Matlab function interfaces allow Matlab scripts to archive, query and retrieve data on the fly at the workflow execution time. Data related to a workflow instance are logically grouped together using the datagroup concept supported by the database toolbox.

C. *Querying augmented provenance data*

Augmented provenance contains rich metadata and semantic relations, which enable users to perform extensive manipulation of provenance data (instead of simple retrieval of data). Such manipulation could include, among other things,

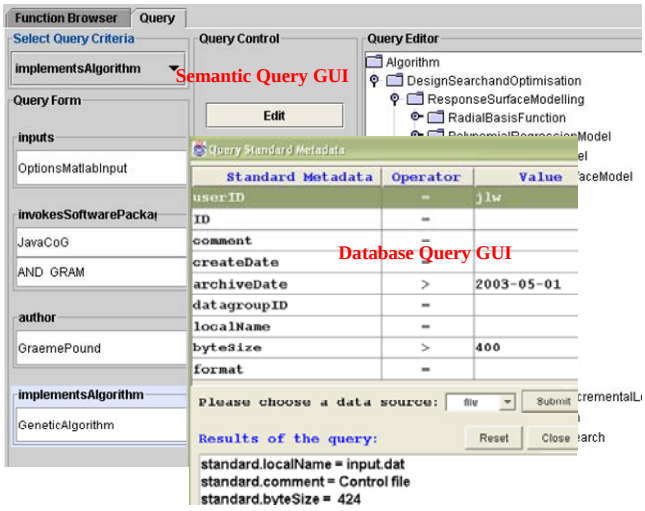

Fig. 4. The query GUIs

retrieving, matching, aggregating, filtering, deriving, inferring and reasoning provenance data in terms of ontological links. This gives rise to many choices and possibilities regarding resource reuse and provenance in addition to validation, repetition and verification. For example, a service of a workflow could be replaced by a semantically compatible service based on augmented provenance.

As an initial step, we have implemented two front-end query GUIs, see Figure 4, to provide dual query mechanisms for flexible and efficient provenance data search and retrieval. The semantic query GUI (i.e. the form) aims to get the high-level provenance data of different facets based on ontology-driven query criteria. The GUI is generated automatically from the EDSO service ontology, and query expression is constructed with support of ontological relations among a workflow, services and parameters. The database query GUI is based on the database schema and can perform keyword-based search and retrieval.

In terms of specific requirements of an application, a user can choose either query GUI accordingly. For example, if a user just wants to know the generic metadata about a workflow profile, its constituent services and types of parameter rather than concrete execution input/output values, a semantic query suffices. To retrieve the full augmented provenance, i.e. both semantic metadata and execution data, a joint query can be launched from either GUI. A workflow's semantic metadata and execution data is cross-referenced using workflow ID.

D. *Provenance services*

To manage augmented provenance, recording interfaces and APIs are needed to accumulate provenance data. A provenance store is not just a sink for provenance data: it must also support some query facility that allows, in its simplest form, browsing of its contents and, in its more complex form, search, analysis and reasoning over process documentation so as to support use cases. Therefore, query interfaces and APIs are an indispensable component in the architecture. Since provenance stores need to be configured and managed, an appropriate management interface is also required.

Apart from the aforementioned fundamental functionality, high-level processing and presentation user interfaces may be required to provide feature-rich functionality. For instance, processing services can offer auditing facilities, can analyse quality of service based on previous execution, can compare the processes used to produce several data items, can verify that a given execution was semantically valid, can identify points in the execution where results are no longer up-to-date in order to resume execution from these points, can re-construct a workflow from an execution trace, or can generate a textual description of an execution. Presentation user interfaces can, for instance, offer browsing facilities over provenance stores, visualise differences in different executions, illustrate execution from a more user-oriented viewpoint, visualise the performance of execution, and be used to construct provenance-based workflows. However, such interfaces typically are application specific and therefore cannot be characterised in a generic provenance architecture.

While interfaces could be implemented in different ways in view of application characteristics and use scenarios, in our example we have provided Web service interfaces for these basic provenance management interfaces. Figure 2 shows the proposed and partially implemented provenance services as system middleware upon which higher-level provenance system or provenance aware applications can be built.

In the system, the recording and query services are responsible for archiving and retrieving augmented provenance data. The Utility Services provide administration facilities such as authentication, authorisation and the lifetime management of provenance data. The processing services provide added-value to the query interfaces by further searching, analysing and reasoning over recorded provenance data. For instance, they can offer such facilities as auditing, comparison of different processes, and check up of semantic consistency and so on. Provenance presentation services provide mechanisms to present query results and processing services' outputs, they are prone to be application dependant. For instance, presentation services can offer browsing, navigation, visualization,

graphical illustration, etc. for provenance data and execution processes. At the time of writing we have developed recording and query APIs and wrapped them into core services, which underpin the implementation of Service/Workflow Annotator and the two query GUIs.

E. Provenance use cases in GEODISE

GEODISE augmented provenance management system enables a number of provenance use cases, some of them are described below.

1) Find the data derivation pathway for a given design result. A user first performs a direct query over the database to retrieve the instantiated workflow description and scripts for the result. Associated input data and generated output data can also be retrieved via the datagroup ID. This workflow script can be enacted in an enactment engine, i.e. Matlab environment, for a re-run.

2) Find information about the optimisation service in the workflow that generates the given result. From the above query, a user can get the workflow template ID through which users can find all involving services, and select the optimisation service to retrieve its associated metadata.

3) Find the similar optimisation algorithms to the one used in this workflow that produces the given result. Following the above query steps we can obtain metadata of an optimisation service, which will contain the type of the optimisation algorithm, e.g. a genetic algorithm (GA). Using the type information in conjunction with the service ontology we can then find out all optimisation services from the SWSR by performing a query based on the algorithmUsed property metadata of the service ontology.

Many other data and/or semantic queries can be framed. For example, find all instantiated workflows that are executed after a specific date; find all workflows that are built by the author who produce this design result.

V. DISCUSSIONS

Whilst provenance has been investigated in other contexts [9] [10] [11], our work concentrates on provenance related to service workflow in a SOA/OGSA. This process-centered view of provenance is motivated by the fact that most scientific and business activities are accomplished by a sequence of actions performed by multiple participants. The recently emerging service-oriented computing paradigm, in which problem solving amounts to composing services into a workflow, is a further motivating factor towards adopting this view.

We identify that augmented provenance in a SOA/OGSA consists of two types of provenance data: execution independent metadata and execution data. We have placed special emphasis on execution independent metadata as Web/Grid services are dynamically published, discovered, aggregated, configured, executed and disbanded in a virtual organisation. Further examination on the motivating scenario shows that execution independent metadata exist at multiple levels of abstraction and multiple facets, and rich relationships exist among them. If such rich metadata can be modeled and represented in a way that semantics and domain knowledge are captured and preserved, it will provide great flexibility and potential for deep processing of provenance data later. This leads to the conception of augmented provenance and further our decisions to use ontologies for metadata modeling and use SWS for capturing semantic metadata.

The employment of service-oriented paradigm for provenance management system is based on several considerations. Firstly, provenance can provide maximum added value for complex distributed applications that are increasingly adopting a service-oriented view for modeling and software engineering. Secondly, a service-oriented implementation of the provenance infrastructure simplifies its integration into a SOA/OGSA, thus promoting the adoption of the infrastructure in service-based applications. Finally, a service-oriented provenance infrastructure deploys easily into heterogeneous distributed environments, thus facilitating the access, sharing and reuse of provenance data.

The hybrid approach to provenance data collection, storage and query are flexible and pragmatic. Semantic metadata contain rich semantic and knowledgeable information by which users can perform reasoning or mining to derive added values or discover implicit knowledge. In contrary, execution data are usually raw data, containing little semantic information. Practically the hybrid approach is easy to be implemented by marrying the state of the art of the Semantic Web and database management technologies.

The benefits of developing a reference augmented provenance system in GEODISE are multiple. Firstly, it helps pin down the conception, modeling and representation of augmented provenance. Secondly, it helps capture user requirements for and characteristics of provenance in the context of service-based applications. Thirdly, it helps identify software requirements for a provenance system, i.e. what a provenance system has to do. Fourthly, the successful design, implementation and operation of the provenance system, though still preliminary, have demonstrated our conception of provenance, its design approaches and implementation rationale. Finally, it helps identify a number of problems and motivate the discovery of possible solutions.

We also learn lessons from the deployment: First, tools should be provided for end users in their familiar working environments. Second, easy-to-use tools should hide as much technical details as possible that are not relevant to the end users.

VI. CONCLUSIONS

In this paper we have analysed the nature of service-oriented computing and elicited the conception of augmented provenance from a real world application scenario. We have proposed a SWS-based hybrid approach for managing augmented provenance based on the latest technologies in the Semantic Web, ontologies, and SWS. We have described a system architecture that specifies the core components and functionalities for managing the lifecycle of augmented provenance. The proposed approach and architecture have been implemented in the context of GEODISE project, which produced a suite of generic APIs and front-end GUIs that are

applicable for the realisation of provenance systems for other application domains.

Although our work is still in its early stage, the conception of augmented provenance and SWS-based approach are innovative and inspiring: provenance will be an indispensable ingredient in the future Web; and reusing SWS's semantic descriptions for provenance is a good example of the Semantic Web applications. By the GEODISE example we have shown how provenance system can be designed and used for problem solving. Further investigation will focus on the granularity of provenance data, and its use to support trust and security.

ACKNOWLEDGMENT

This work is based on the UK EPSRC GEODISE e-Science pilot project (GR/R67705/01) and EU FP6 PROVENANCE project. The authors gratefully acknowledge the contributions of Dr. William Cheung for his insightful and inspiring comments and suggestions.

REFERENCES

[1] Foster, I., Kesselman, C., Nick, J., Tuecke, S. (2002), Grid Services for Distributed System Integration, Computer, 35(6), 37-46

[2] Cui, Y., Widom, J. and Wiener, J.L. (2000), Tracing the Lineage of View Data in a Warehousing Environment. ACM Trans. on Database Systems, 25(2):179–227

[3] Buneman, P., Khanna, S. and Tan, W.C. (2001), Why and Where: A Characterization of Data Provenance. In Proceedings of 8th International Conference on Database Theory, pp316–330

[4] Foster, I., Vockler, J., Wilde, M., Zhao, Y. (2002), Chimera: A virtual data system for representing, querying, and automating data derivation, Proceedings of the 14th International Conference on Scientific and Statistical Database Management, pp37-46

[5] Boss, R. (2002). A conceptual framework for composing and managing scientific data lineage, Proceedings of the 14th International Conference on Scientific and Statistical Database Management, pp47-55

[6] Buneman P., Chapman A. and Cheney J. (2006), Provenance management in curated databases. SIGMOD Conference 2006: 539-550

[7] da Silva, P.P., McGuinness, D.L. and McCool, R. (2003), Knowledge Provenance Infrastructure. IEEE Data Engineering Bulletin Vol.26 No.4, pp26-32

[8] McGuinness D.L. and da Silva P.P. (2004), Explaining Answers from the Semantic Web: The Inference Web Approach, Journal of Web Semantics, Vol.1, No.4, pp1-27

[9] Workshop on Data Derivation and Provenance, (2002), http://www-p.mcs.anl.gov/~foster/provenance/

[10] Workshop on Data Provenance and Annotation, (2003), http://www.nesc.ac.uk/esi/events/304/

[11] International Provenance and Annotation Workshop IPAW'06, (2006), http://www.ipaw.info/ipaw06/

[12] Zhao, J., Wroe, C., Goble, C., Stevens, R., Quan D., and Greenwood M. (2004). Using Semantic Web Technologies for Representing e-Science Provenance, LNCS, No.3298, pp92-106

[13] Moreau, L., Chen, L., Groth, P., Ibbotson, J., Luck, M., Miles, M., Rana, O., Tan, V., Willmott, S. and Xu, F. (2005). Logical architecture strawman for provenance systems, Technical report, University of Southampton.

[14] Jiao, Z., Wason, J.L., Song, W., Xu, F., Eres, H., Keane, A.J., and Cox, S.J. (2004). Databases, Workflows and the Grid in a Service Oriented Environment, Euro-Par2004, Parallel Processing, LNCS, No.3149, pp972-979.

[15] Harris, S., Gibbins, N. (2003). 3store: Efficient Bulk RDF Storage. Proceedings of 1st International Workshop on Practical and Scalable Semantic Systems, pp1-15.

[16] Horrocks, I., Li, L., Turi, D., Bechhofer, S. (2004). The instance store: DL reasoning with large numbers of individuals, Proceedings of the 2004 Description Logic Workshop, pp31-40

[17] Chen, L., S. J. Cox, C. Goble, A. J. Keane, A. Roberts, N. R. Shadbolt, P. Smart, and F. Tao (2002). Engineering knowledge for engineering grid applications. In Proceedings of Euroweb 2002 Conference, The Web and the GRID: From e-science to e-business, pp12–25.

[18] Haarslev, V., Möller, R. (2003). Racer: A Core Inference Engine for the Semantic Web, Proceedings of the 2nd International Workshop on Evaluation of Ontology-based Tools (EON2003), pp27-36.

[19] Chen, L., Shadbolt N.R., Tao F. and Goble C. (2006). Managing Semantic Metadata for Web/Grid Services, International Journal of Web Service Research, in press.