Decision Support Systems 64 (2014) 90–99

Contents lists available at ScienceDirect

Decision Support Systems

journal homepage: www.elsevier.com/locate/dss

A provenance-based approach to semantic web service description and discovery

Tom Narock [a,*], Victoria Yoon [b], Sal March [c]

[a] Department of Information Technology and Management Science, Marymount University, Arlington, VA, USA
[b] Department of Information Systems, Virginia Commonwealth University, Richmond, VA, USA
[c] Owen Graduate School of Management, Vanderbilt University, Nashville, TN, USA

ARTICLE INFO

Article history:
Received 19 August 2013
Received in revised form 3 April 2014
Accepted 23 April 2014
Available online 4 May 2014

Keywords:
Web service discovery
Semantic web service
Ontology
Provenance

ABSTRACT

Web services have become common, if not essential, in the areas of business-to-business integration, distributed computing, and enterprise application integration. Yet the XML-based standards for web service descriptions encode only a syntactic representation of the service input and output. The actual meaning of these terms, their formal definitions, and their relationships to other concepts are not represented. This poses challenges for leveraging web services in the development of software capabilities. As the number of services grows and the specificity of users' needs increases, the ability to find an appropriate service for a specific application is strained. In order to overcome this challenge, semantic web services were proposed. For the discovery of web services, semantic web services use ontologies to find matches between user requirements and service capabilities. The computational reasoning afforded by ontologies enables users to find categorizations that weren't explicitly defined. However, there are a number of methodological variants on semantic web service discovery. Based on e-Science, an analog to e-Business, one methodology advocates deep and detailed semantic description of a web service's inputs and outputs. Yet, this methodology predates recent advances in semantic web and provenance research, and it is unclear the extent to which it applies outside of e-Science. We explore this question through a within-subjects experiment and we extend this methodology with current research in provenance, semantic web, and web service standards, developing and empirically evaluating an integrated approach to web service description and discovery. Implications for more advanced web service discovery algorithms and user interfaces are also presented.

1. Introduction

Service oriented architecture (SOA) defines a set of principles and methodologies for designing and developing software in the form of interoperable services. These services, referred to formally as *web services*, are applications that can be used automatically by a computer on behalf of a user. Frequently they are embedded within applications to enable rapid and reliable system development. Web services are described using the Web Service Description Language (WSDL) [1]. They are offered over the Web as functional software building blocks accessible via standard Internet protocols, independent of platforms and programming languages [2]. Web services have become common, if not essential, in the areas of business-to-business integration, distributed computing, and enterprise application integration due to their interoperability and extensibility [3]. Web services can be used together in a loosely coupled fashion and new services can be formed from the aggregation of existing services [2].

However, the XML-based standards for web service descriptions encode only a syntactic representation of what is expected by, and returned from, a service, i.e., its inputs and outputs. The actual meaning of the terms used, their formal definitions, and their relationships to other concepts are not represented. This poses several challenges for leveraging web services in the development of software capabilities. In particular, as the number of services grows and the specificity of users' needs increases, the ability to find an appropriate service for a specific application is strained [4]. Hendler [4] proposed using ontologies to provide a more powerful method for the discovery of web services. This work provided the foundations of what is now known as *semantic web services*. Semantic web services aim toward reducing the manual effort required for discovering and using web services [5].

One of the prime application areas for semantic web services is e-Science. An analog to e-business, e-Science [6] began as the application of computing to traditional science research. However, in recent years computers have become vital to scientific research and e-Science has transformed the way in which scientific research is performed [7]. Increasing data volumes within many scientific domains makes it no longer practical to copy data and perform local analysis. Instead, hypotheses are tested through online tools that combine and mine pools of data [8]. Web services are enabling such efforts, termed “industrial scale science” [7], by making data and algorithms programmatically accessible on the Web. Service oriented architectures have become the common distributed technology in e-Science [7,9] and web services allow an increasing volume of scientific analysis to take place on the web.

* Corresponding author at: Marymount University, 1000 Glebe Road, Arlington, VA, 22201.
E-mail address: tnarock@marymount.edu (T. Narock).

http://dx.doi.org/10.1016/j.dss.2014.04.007
0167-9236/© 2014 Elsevier B.V. All rights reserved.

The most prominent type of web service discovery is known as *interface matching* [10]. It consists of services being discovered by matching descriptions of input and output semantics of the web service to the application requirements. However, e-Science requirements have challenged this notion. Semantic web service discovery in e-Science requires much more conceptual description than is present in a service's interface [11]. Domain information including descriptions of tasks the service will perform, additional information resources consulted in the performance of those tasks, and software algorithms utilized are frequently required for scientists to adequately match requirements to web service [11]. The "black-box" interpretation of web services, which is used in interface matching, assumes that each output depends on all inputs provided, and fails to model the internal state and execution processes of the service. Such coarse-grained approximations are rarely true and can be misleading in understanding and interpreting a service's output, particularly in e-Science applications [12].

In reality, an output may depend only on a small subset of the input and on the internal state of the web service and its processing algorithm. For example, output from a service utilizing a learning algorithm depends on both current and historical inputs and on the specific learning algorithm itself [12]. In addition, documenting the assumptions or decisions made during web service execution give a context in which the results can be reused and enables proper crediting of the scientists involved [13]. Describing a web service at a granularity that includes algorithms and assumptions should ensure proper interpretation and validation of the related scientific claims made. These notions led Wroe et al. [11] to create semantic descriptions of web service processes and incorporate them in the discovery process. While a seminal work in this area, the proposed approach has several limitations on today's semantic web:

1. It is based on technology that is no longer utilized in the semantic web.
2. Current research in provenance and provenance ontologies can be leveraged to better model web service execution processes, and
3. it has never been evaluated for generality and impact on end-users.

Thus, it is unclear the extent to which this approach can be used outside of e-Science. Furthermore, it is not clear how end-users perceive similarity when presented with the additional information provided by this approach.

We address these limitations by providing an experiment to assess the generality of web service discovery based on the semantics of web service processes. Further, we provide an update to Wroe's methodology [11] through the creation of a new ontology based in provenance research and current semantic web standards. We show enhancements to the methodology enabled by this ontology and describe how this ontology can be linked to the emerging W3C web service discovery standard Semantic Annotations for WSDL (SAWSDL) [14]. Finally, we present implications for semantic web service discovery interfaces based on web service processes.

The remainder of this paper is organized as follows. First, we present a brief motivating example of two e-Science web services that appear identical, but function quite differently and thus would be appropriate for different applications. We then discuss related work on web service discovery. Next, we introduce our notion of service provenance, the role it can play in web service discovery, and our provenance-based web services ontology. This is followed by a description of our empirical study and experimental design. We then present an analysis of our results and conclude with a discussion of our contributions and a presentation of an agenda for future research.

2. Motivating example from e-Science

The Jena Geography Dataset[1] is a collection of 200 geographic services that have been collected from around the web. These services focus primarily on geocoding — the process of finding geographic coordinates from other geographic data, such as street addresses and zip codes. Consider the two Jena services shown in Table 1. Both services have input *Location* and output *Distance*. Within the Jena dataset from which these services originated, there are 18 services that have input *Location* and output *Distance*. These 18 services would all be considered to be similar by current web service discovery techniques. Interface matching techniques provide no means of distinguishing services by execution details and the process of matching task to service is left to the user. In this particular example, the results differ in precision, which limits the applicability of each web service. Inadvertently choosing a service that is not applicable can lead to incorrect conclusions and erroneous decisions, which can have far reaching consequences [15].

3. Related work

The majority of semantic web service discovery algorithms operate on the assumption that explicitly defined service semantics can be exploited to match available services with user requests [16]. This type of discovery is known as interface matching [10] and consists of services being discovered by matching input and output semantics. Interface matching techniques require users to specify desired web service inputs and outputs using concepts from an ontology. SAWSDL [14], the current state of the art technology for interface matching, provides a means of encoding the linkage between ontology concepts and a service's inputs and outputs.

Discovery algorithms then identify exact, more general (superclass concept), and more specific (subclass concept) matches using the ontology, user input, and SAWSDL annotations. This classification scheme was developed by Zaremski and Wing [17] and is widely used in practice. However, interface matching has three main drawbacks [18]. First is low recall due to the rigid hierarchy that is required. Matches are missed due to discovery being limited to only exact, subclass, and superclass relationships. Although additional matches could be determined through relationships created by the ontology developers, the usage of such relationships in discovery algorithms is just now beginning to be studied. Second, depending on the context, more specific and more general results may not be suitable replacements, leading to false positives. Third, discovery is often a manual and iterative process in which a user progressively narrows down the set of candidate services. Users need to take into account the functions the service carry out and the resources it uses to accomplish its goals [11]. There is often insufficient information to make an informed decision leading to significant manual effort to identify an appropriate web service.

So-called hybrid approaches have been devised to overcome some of these limitations. These approaches combine interface matching with information retrieval techniques. Research has shown that hybrid approaches outperform ontology-based approaches by increasing precision and recall during service discovery [19,20]. However, regardless of whether it is hybrid or ontology-based, interface matching techniques lack information describing how services operate.

These deficiencies led Martin-Recuerda and Robertson [21] to request finer-grained classifications of web services. Chen and Jiao [22] met this challenge in a UK e-Science project [22] in which they developed a number of insights regarding service discovery. In particular, they note the need for *provenance* in service discovery. Historically, provenance referred to the history and lineage of an object. In this context, provenance refers to the creation and specification of the web service and includes such information as runtime environment, settings, and algorithms used. Chen and Jiao [22] highlight the need to find services with specific or equivalent algorithms for specific types of applications. Such information is not readily available in most web service descriptions and thus is not included in many discovery applications. They conclude that service discovery needs to include provenance information at multiple levels of abstraction and over multiple facets. While they offer a potential solution to this problem, their solution was

[1] http://fusion.cs.uni-jena.de/professur/jgd.

Table 1
Example service pair from Jena Geography Dataset.

Service 1	Service 2
Inputs:	*Inputs*:
• Location — longitude and latitude of a point on the surface of the Earth • Location — longitude and latitude of a point on the surface of the Earth	• Location — longitude and latitude of a point on the surface of the Earth • Location — longitude and latitude of a point on the surface of the Earth
Output:	*Output*:
• Distance — the distance in miles between the given locations	• Distance — the distance in miles between the given locations
Service details:	*Service details*:
Description: This services takes two locations on the Earth, given by longitude and latitude, and computes the distance between those points	*Description*: This services takes two locations on the Earth, given by longitude and latitude, and computes the distance between those points
How: The Spherical Law of Cosines is an application from trigonometry for computing distances on a sphere. The Spherical Law of Cosines equations are applied to the input locations to compute a distance. This distance is then converted to miles and returned to the user.	*How*: This service uses the Haversine formula to compute the distance between the input locations
Algorithms/Methodologies used:	*Algorithms/Methodologies used*:
• Spherical Law of Cosines	• Haversine Equations
Assumptions/Limitations:	*Assumptions/Limitations*:
The error in the distance calculation increases the further apart the input locations. For example, the distance between two locations in the same country is fairly accurate. However, distance estimates across continents is less certain.	Mathematically, the Haversine formula is equivalent to the Spherical Law of Cosines. However, computationally the Haversine formula is more precise. The Haversine formula addresses computational limitations of working with angles. As a result, software implementing the Haversine formula can more precisely compute distances than software using the Spherical Law of Cosines

developed in the context of the UK *Grid-enabled Optimisation and Design Search in Engineering* (GEODISE) project. The resulting ontology and provenance elements are specific to that project.

More recently, Gunay and Yolum [10] have identified additional use cases that require knowledge of service execution processes during discovery. These authors have generalized the e-Science problem by highlighting the need to know the order of operations carried out by web services configured to address tasks within the travel services domain. They introduce a method of service discovery based on Linear Temporal Logic (LTL). Using LTL, they model each execution step of a service as a state transition. A discovery algorithm is then proposed that discovers services via their allowable states.

Other researchers have echoed this emphasis on service processes. For example, Klein and Bernstein [23] use a workflow language to model web services and Wombacher et al. [24] use a finite state machine to model web service execution. While we agree with the overarching methodology of modeling web service processes, these recent approaches tend toward non-traditional logics and away from semantic web standards. Moreover, while these approaches model *what* happens within a web service, they do not model *why* it happens or what the potential implications might be. In this work, we provide additional evidence in support of the need to include web service process descriptions in web service discovery. We show the effects that such information has on discovery and develop a new ontology to capture web service processes. This new ontology is based on the Ontology Web Language (OWL) [25], a fundamental part of the semantic web technology family whose usage in many tools and environments is well-understood [26]. Further, this new ontology extends the capabilities of previous ontologies by including a justification component.

4. Service provenance ontology

4.1. Theoretical basis

The theoretical underpinning of the W7 model, and also our service provenance ontology, is Bunge's ontological theory [27,28]. In the application of that theory in the information systems domain [29,30] the data stored in an information system is viewed as a state-history, that is, the results of sequences of events that result in state changes. Provenance can be defined by recording the effects of all events that happen to data during its lifetime [31]. Yet, simply recording *what* events are not sufficient to meaningfully represent the provenance of data [27,28,31,32]. According to Bunge [27] causal relationships are also needed. Extending this existing philosophical foundation [27,28,31,32] into the area of web services leads us to recognize additional important concepts for web service provenance.

4.2. Foundation of service provenance ontology

The W3C has recently released PROV-O, a standard ontology for provenance capture [33]. PROV-O uses OWL to express a common provenance model. The ontology provides a set of classes, properties, and restrictions to represent and interchange provenance information generated from different systems and under different contexts. However, PROV-O is a high level ontology and needs to be specialized to create new classes and properties to model provenance information in specific applications and domains [33]. As a result, PROV-O can be used as the basis for web service descriptions; but by itself PROV-O cannot capture all of the required provenance information.

The W3C defines provenance[2] as

> *information about entities, activities, and people involved in producing a piece of data or thing, which can be used to form assessments about its quality, reliability or trustworthiness.*

Building upon this definition and the work of Chen and Jiao [22], we define *service provenance* as

> *information about a web service and its execution, which can be used to form assessments about its quality, reliability or trustworthiness. This includes applications invoked, methodologies used, actions and settings invoked, and any assumptions and hypothesis involved.*

We created a *service provenance ontology*[3] by integrating two existing projects in data provenance with additional concepts we identify as enhancing service discovery. Specifically, the W3C Provenance Ontology (PROV-O) [33] is integrated with the W7 model [31]. This combination is then augmented with our own concepts, described in Section 4.3 below, to provide a unique encapsulation of a service's components and their associated justification. We do not argue that this represents the "best" or the "true" point of view. Rather, it is a means for service creators to supply supplemental information to justify the rationale of their service. In fact, a critical component of our implementation

[2] http://www.w3.org/TR/prov-overview/.
[3] http://purl.org/provenance/service-provenance.

is the ability to discover a service's implementation details regardless of their acceptance within a broader user community.

The service provenance ontology is thus defined as:

$$\text{Service Provenance Ontology} = \text{W3C PROV-O} + \text{W7 Model} + \text{additional concepts}.$$

We define the service provenance of a given web service as an instantiation of the service provenance ontology. That is, service provenance is the set of related instances (the RDF/OWL graph) that uniquely defines a web service. More specifically, service provenance is the set of related instances that define the terms under which a web service can be fully understood. This includes applications invoked, methodologies used, actions invoked, settings invoked, and the assumptions and hypotheses involved.

4.3. Creation of the service provenance ontology

Historically, provenance research has focused on data provenance. The creation, processing, and proper usage of data have been the focus (e.g. [31,34,35]). In our ontology we take a new look at provenance. We focus on the history and rationale behind a web service. In this regard, we reuse many ideas from data provenance; however, we apply them in a new way.

Our service provenance ontology is more than a merging of existing provenance research. Service provenance requires two components — a historical lineage (e.g. the steps that occurred) and the ability to describe how the service operates (e.g., applications, methodologies, and settings). We argue that neither PROV-O nor the W7 model, nor any existing provenance ontology, contains the necessary concepts to accomplish this. Independently, PROV-O provides the historical lineage while W7 provides some of the concepts needed to describe how a service operates. By combining these two ontologies we have many of the capabilities required to represent service provenance. However, additional concepts were created to complete the service provenance ontology.

The service provenance of a web service provides justification for why a service returns the results that it does. Justification then allows web applications to provide explanations [36] to users. These explanations can lead users to judgments about reliability [36] and trust [37]. Moreover, our work in this area allows justification to be taken one step further. Current web justification systems focus primarily on the source and quality of data [36,37]. Service provenance allows for the citation of applications, methodologies, and settings that operate on the data.

When creating ontology concepts we adopted the notion of *Necessary Feature* [32]. A *Necessary Feature* is a component of a process that helps to uniquely define what happened during that process and why. For example, consider the following pseudo-code (taken from [32]):

```
MERGE_FUNCTION (a,b)
    If (a.value = b.value) then merge (a,b)
END MERGE FUNCTION
```

For a user to understand why *a* and *b* where merged (or not merged) it must be recorded that a merge function takes place with *a* and *b* being merged if *a.value* is equivalent to *b.value*. Thus, the *Necessary Features* of this simple service are:

1.) A merge function will be executed
2.) The inputs *a* and *b* will be merged if *a.value* is equivalent to *b.value*.

In reality there will always be some limit to the amount of provenance that can be captured [32]. It is not practical or feasible to present users with line-by-line code for a service's execution. Thus, we utilize the *Necessary Feature* principle in the creation of our service provenance ontology. We seek to develop an ontology that captures the minimum amount of service execution details to uniquely describe what happened. Following Ram and Liu's notion [31] that complete provenance extends beyond *what* and *where*, we enable the encoding of hypotheses, limitations, and references as *Necessary Feature*s. Exposing provenance at this level of abstraction has proven beneficial [32] despite the fact that one can never provide complete provenance.

Our service provenance ontology can be used in two ways. First, we can encode the algorithms used by a service, their limitations and assumptions, their inputs and preconditions, as well as where the service and processes will execute. Moreover, we do so as instances of a standardized provenance ontology, capturing the relationships between types of algorithms and other service related information. This information can then be exploited during service discovery.

PROV-O, and thus our ontology, also offers notions of time. A second use of our service provenance ontology is to capture the exact execution of a web service as it is executing. In doing so, we can capture the temporal order of steps taken and the rationale for each step. We note that a web service may have multiple execution paths that it could follow depending on inputs supplied. During discovery, service provenance can tell users what a service *can* do. During service execution, service provenance can tell users what a service *did* do. There are many applications for this type of real-time provenance capture [12,22,32].

4.4. Ontology components

4.4.1. PROV-O

PROV-O uses OWL [25] to express a common provenance model. It is an ontological representation of “a core data model for provenance for building representations of the entities, people and processes involved in producing a piece of data or thing in the world” [33]. This core data model is an attempt by the W3C to create a standard domain-agnostic provenance model with “well-defined extensibility points allowing further domain-specific and application-specific extensions to be defined” [33].

As a result, PROV-O can be used as the basis for web service descriptions; yet, by itself PROV-O cannot capture all of the required execution semantics. The descriptions below define key PROV-O concepts and their proposed usage within our service provenance ontology (see Fig. 1 for an illustration of the integration of PROV-O and W7 with our extensions).

Agent “An agent is an entity that takes an action and bears some form of responsibility for an activity taking place” [33]. We define web services as agents and model their execution steps as activities.

Software Agent a subclass of Agent and defined simply as “running software” [33].

Plan “A plan is an entity that represents a set of actions or steps intended by one or more agents to achieve some goals” [33]. Within service provenance, *Plans* are high-level descriptions of the algorithms, settings, preconditions, hypothesis, and beliefs of a web service. *Plans* indicate what could happen within a web service and are used for discovery. *Plans* are a type of *Entity*.

Entity “An entity is a thing one wants to provide provenance for. For the purpose of this specification, things can be physical, digital, conceptual, or otherwise; things may be real or imaginary” [33]. Since entities are what one provides provenance for, and *Plans* are entities, the focus of our service provenance ontology is on capturing the execution actions of a web service and their associated rationale.

Bundle “A bundle is a named set of provenance descriptions, and is itself an Entity” [33]. Within our specific use, a *Bundle* contains a step-by-step description of how the service executes. *Bundle*s consist of instances of the PROV-O *Activity* concept. The difference between Plan and Bundle/Activity is that a Plan describes what *could* happen while Activities describe what *did* happen. A Plan is intended for the discovery phase when a user would like to examine the capabilities of a web service. In this manner, a Plan describes all the possible algorithms that could be executed. However, a given service

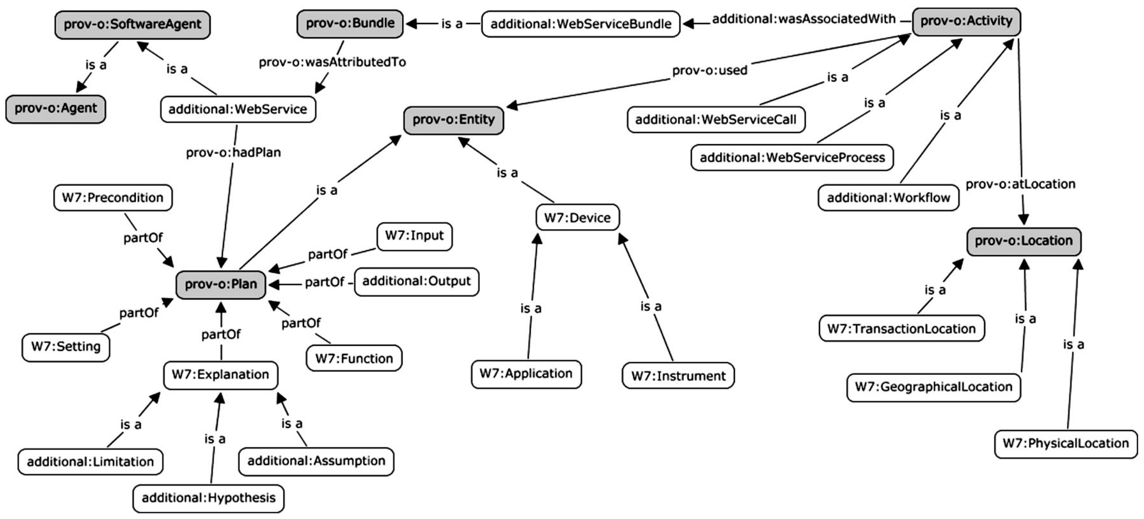

Fig. 1. A graphical depiction of our service provenance ontology.

may choose a particular execution path based on how it is called (e.g. if input1 > x choose path A else choose path B). Thus, not all of the described algorithms will be executed for each invocation of the service. The Activity concept allows for the capturing of specific execution steps for each service invocation. *Bundle* provides a container for this information and relates the set of steps to the web service. Plans are useful for the discovery phase and Bundle/Activity is useful for the execution phase and subsequent interpretation of the results.

Activity "An activity is something that occurs over a period of time and acts upon or with entities. This action can take multiple forms: consuming, processing, transforming, modifying, relocating, using, generating, or being associated with entities. Activities that operate on digital entities may for example move, copy, or duplicate them" [33]. Within our service provenance ontology an 'Activity' is used to capture the set of processes that execute when a web service is called.

The notion of where an action happened has also been identified as a vital element of provenance [38]. Ram and Liu [31] have extended this notion to include both physical and transactional locations. A physical location is a geographical location while a transactional location is a location within a database or information system. We see these concepts as vital in meeting the requests for additional insights into web service execution details [12,22,32]. Thus, our service provenance ontology utilizes the PROV-O generic notion of *Location* described below along with the Ram and Liu [31] extensions of *Location* as described below.

Location "A location can be an identifiable geographic place (ISO 19112), but it can also be a non-geographic place such as a directory, row, or column. As such, there are numerous ways in which location can be expressed, such as by a coordinate, address, landmark, and so forth" [33]. In our usage, *Location* can be used to identify the geographical location of the service as well as the non-geographical location (e.g. server name or database name) of algorithms and utilities.

4.4.2. W7 model

Ram and Liu [31] proposed the W7 model as a means of dealing with data provenance. The authors intended the model to capture all the interrelated elements of data history — its creation, processing, modification, and storage. W7 garners its name from the "What", "When", "Why", "Where", "hoW", "Who", and "Which" elements that make up the model components. These components have historically been used to describe the lineage of data, but we have found that they work equally well in describing the rationale for the way in which a web service was implemented. To this end, we have utilized the W7 model as the basis of encoding service provenance rationale. The W7 components are used to encode what a service is doing and how it is doing it. The W7 model components are used to extend PROV-O and uniquely define the methodology and rationale of the service. Their definitions are augmented as follows:

The W7 model defines the concept of *Device*, which describes applications and instruments. We have defined *Application* and *Instrument* as subclasses of *Device*, which in turn is made a subclass of PROV-O *Entity*. These classes have the following definitions:

Application A software tool or methodology used in data collection or analysis
Instrument A physical piece of hardware used to collect or analyze data.

Ram and Liu [31] have noted that complete provenance extends beyond notions of *what* and *where*. In many domains, provenance includes literature references, experimental procedures, and the sequence of ideas leading to a procedure. We agree with this notion and believe that the capturing of beliefs and settings is also vital to complete service provenance. As a result, we have included the W7 notions of *Precondition*, *Input*, *Function*, *Setting*, and *Belief* as part-of the description of a PROV-O Plan. We have, however, renamed *Belief* to *Explanation* in an attempt to remove any association with fuzzy logic. Our intent is not to capture belief statements that may vary from individual to individual, but rather to capture statements that can be universally evaluated true or false. This makes *Plan* more specific to web services. The W7 concepts are defined as:

Precondition Conditions that must hold prior to enactment of an action
Input Data objects that are manipulated by an action
Function The ways in which a device can operate or be operated
Setting The way in which a device can be configured
Explanation A reason, justification, or clarifying statement.

We have also added the specific W7 notions of location (Geographical, Transactional, and Physical) as subclasses of PROV-O Location. Specifically, they are defined as:

Geographical Location A location specified by geographical boundaries such as a State or country

Physical Location A location specified by coordinates within a coordinate system
Transactional Location A location within a database or server, often specified by a URI.

As a result, we have repurposed the W7 model as a means of extending the W3C Provenance Ontology, which was designed as a high-level data provenance ontology and does not include specifics needed to describe web services.

4.4.3. Additional concepts

Integrating W7 model concepts with PROV-O enables a more specialized representation of web services; however, it does not address all of the required specialization. To achieve this we have defined additional concepts and further extended PROV-O. In PROV-O, the *Agent* class has subclass *SoftwareAgent*. We extended the *SoftwareAgent* class to have subclass *WebService* creating a specialized *Agent* for our purposes. This is analogous to the already existing Agent subclasses of *Person* and *Organization*. This specialization provides provenance aware applications the needed semantics to infer that a web service was used.

Further, we have added three subclasses to Activity, which describe specific web service activities.

WebServiceCall A call to execute a web service
WebServiceProcess An execution step within a web service
Workflow A scientific workflow that consists of the chaining of multiple web services

We found that the W7 model lacked the notion of *Output* as a compliment to *Input* and *Precondition*. We have added *Output* as part of a *Plan*. Finally, we have extended the *Explanation* concept to include *Assumption*, *Limitation*, and *Hypothesis*. This results from the notion that the sequence of ideas leading to a procedure is important [31] and provides further semantics for quantifying those ideas.

Assumption A belief lacking any evidence of support
Limitation A defect, failing, or condition that limits ability
Hypothesis A proposed explanation based on limited evidence

Finally, we have subclassed PROV-O *Bundle* with *WebServiceBundle* to provide semantics for the specific bundles we are dealing with.

A graphical depiction of the integrated and extended web services ontology is shown in Fig. 1. Within Fig. 1 gray rectangles depict the W3C Provenance Ontology concepts that are the foundation of the ontology. White rectangles are concepts added from the W7 model or from our ontology evaluation. The namespaces within each rectangle designate the concept as from either PROV-O (prov-o:), W7 model (w7:), or added by the authors (additional:).

5. Connection to existing standards

Semantic Annotations for Web Service Description Language (SAWSDL) [14] is the W3C standard for web service annotations. It allows the syntactic Web Service Description Language (WSDL) to be annotated with references to semantic information. Within WSDL the notion of *Interface* is defined as the set of operations a web service carries out. We argue that it is most appropriate to link WSDL web service descriptions to service provenance via SAWSDL at this *Interface* level. We propose using SAWSDL to annotate WSDL documents at the *Interface* level with references to the appropriate service provenance document. As a result, a web service's description file (WSDL file) will provide complete information for service discovery and usage. We note that this approach is entirely W3C standards compliant. Additionally, this approach extends initial research [11], which did not link semantic service descriptions to web service standards.

We also advocate exposing service provenance as Linked Open Data (LOD) [39] enabling both human and machine dereferencing of web service execution details. LOD is a data publication methodology that utilizes the semantic web to make data publicly accessible on the Web. LOD allows data from multiple sources to be combined and queried over by exposing RDF databases on the Internet. This allows service execution details to be linked to domain specific semantic information published by other parties — a linkage that has been shown to enable the answering of more detailed provenance questions [40,41].

6. Evaluation

6.1. Quality of the service provenance ontology

The features of an ontology depend on the purpose for which it was created. The major goals of ontology-dependent projects have included natural language understanding, information retrieval, theoretical investigation, knowledge sharing and reuse, simulation, and modeling [42]. As a result, several ontology creation methodologies have been developed. These methodologies range from the purely philosophical (e.g. [43]) to the extremely practical (e.g. [44,45]). In the latter approach, the ontology is designed around a use case and undergoes an iterative design. The ontology and associated application are revised based on user feedback and changing requirements.

It is this latter design science approach that we adopt [46]. The service provenance ontology was iteratively created through interactions with the Earth science community and evaluations of existing ontologies. Initially, we conducted a literature review to identify applicable ontologies — finding the PROV-O and W7 ontologies. The ontology was subsequently refined through discussions with Earth science experts at various conferences and meetings. Specifically, we spent a year as part of the Federation of Earth Science Information Partners (ESIP). ESIP is an open community of Earth scientists, educators, and information technology practitioners. In addition to an online presence, the group meets face-to-face twice a year with the goal of enhancing Earth science research and outreach. The first author participated in monthly teleconferences and presented the ontology for discussion at ESIP meetings. During this time the ontology was evaluated for completeness, consistency, conciseness, and coverage, as suggested by [47]. The final ontology resulted after several iterations with ESIP and subsequent follow-up meetings with Earth scientists at NASA's Goddard Space Flight Center. The ontology was designed as a general-purpose service provenance ontology, despite being created within the Earth science community.

We expect additional subclasses and instances of the service provenance ontology to be necessary throughout the lifecycle of its usage. There are several ways in which this can be accomplished and we highlight one methodology here as an example. The On-To-Knowledge methodology [44] is an application oriented ontology development methodology. Within this methodology, a group of knowledge engineers is responsible for monitoring environmental and application changes that result in the need to update an ontology. Through a series of evaluation–maintenance–refinement loops the ontology is continually and incrementally updated to meet new application requirements. Similar approaches have been taken in previous research [11] with success.

6.2. Utility of provenance based discovery

We conjecture that the use of the service provenance ontology will have a positive effect on the discovery process. Specifically, when users must choose among similar web services we hypothesize that service provenance will lead to a measureable difference in three areas: the assessment of Dissimilarity, Relative Advantage, and Decision

Confidence. Assessment of dissimilarity enables users to more effectively choose the most appropriate service. Further, we hypothesize that service provenance will increase participants confidence with their decisions.

We also hypothesize that participants will more effectively be able to make their decisions in the presence of service provenance thus leading to an increase in "Relative Advantage." Relative Advantage is defined as "the degree to which an innovation is perceived as being better than the idea it supersedes" [48, p.15]. Innovation theory suggests that Relative Advantage is related to innovation adoption and diffusion [49]. Our third construct, Decision Confidence, is one's belief in the quality of their decision [50]. Overconfidence can lead to low quality decisions as the user may ignore other sources of information [51]. Under confidence can result in users not taking action [52].

Specifically, we have the following three hypotheses:

H1. Service provenance will give subjects an increased accuracy in detecting service dissimilarity as compared to interface matching.

H2. Service provenance will give subjects an increased Relative Advantage in determining service dissimilarity as compared to interface matching.

H3. Service provenance will give subjects an increased Decision Confidence in determining service dissimilarity as compared to interface matching.

Support for our three hypotheses would indicate the utility of service provenance and quantify how it affects users' notion of similarity. These results would then substantiate non-interface matching techniques as well as impact design choices.

6.2.1. Experimental materials

We utilized web services from two different domains: the Jena Geography Dataset[4] and the Programmable Web.[5] The Jena dataset is a collection of about 200 geography services that have been collected from around the web. These services focus primarily on geocoding — the process of finding geographic coordinates from other geographic data, such as street addresses and zip codes. Programmable Web is an online portal offering comprehensive access to services on the web. The web site offers 6000[6] services as well as links to applications built by combining existing services.

We initially identified six pairs of web service descriptions, three from each dataset, as shown in Table 2. Our choices from Programmable Web were of straightforward easy to understand web services. We felt that this complimented the more mathematically oriented Jena services. We believe that these services offer a sampling of multiple domains and allow us to test the generality of our hypotheses.

Each pair of web services is highly similar in terms of service inputs and outputs. Thus, each pair of services resembled what might be returned to users from an interface matching algorithm. However, the pairs differ in terms of operations and internal functionality. Using information external to the web services (e.g. web pages and customer support contact information), we assembled and encoded the service provenance of the 12 web services for this experiment. This external information was used to create instances of our service provenance ontology, which were subsequently presented to participants in formatted web-based tables.

These initial six pairs of web service descriptions were used in a preliminary study to evaluate our hypotheses. Upon receiving positive results from our initial survey, we expanded the evaluation experiment to further demonstrate the applicability of our proposed service provenance ontology to real-world problems. The extended study includes ten pairs of web services — five from Programmable Web and five from Jena.

6.2.2. Data collection

Fifty-two university students participated in the initial study with six pairs of web services. The participants were primarily senior undergraduates and graduate students. The majority of the students had some familiarity with web services. Forty-nine subjects participated in the larger follow-up study. Table 3 lists demographic data for each of the studies.

6.2.3. Experiments

Both studies used a within-subjects design where each participant was shown all pairs of web services with and without service provenance (order randomized and services unnamed). In the latter scenario, participants saw only the service inputs, outputs, and a brief description. This scenario represents interface matching, the predominant method of service discovery discussed above. As mentioned by Stollberg [53], with this approach users are often left with the manual task of trying to determine which service to use. Hence, in the service provenance scenario we sought to measure participant's notions of similarity, their confidence in differentiating services, and the advantage (if any) they felt when additional provenance information was provided. In this scenario, participants where again shown the same pairs of web services (randomized). However, in this scenario the inputs, outputs, and description were accompanied by service provenance information. Participants were again asked about Similarity, Decision Confidence, and Relative Advantage. Again, all pairs of web services were randomized for each participant and all questions were given on a 7-point Likert scale. One question was given for dissimilarity and three questions each for Decision Confidence and Relative Advantage. The seven questions asked for each pair of web services are listed in Appendix A.

An example service pair with service provenance was shown in Table 1. Service pairs without service provenance did not include the How, Algorithms/Methodologies, and Assumptions/Limitations sections. The two services in Table 1 are identical in terms of inputs and outputs; yet, the precision of the results differs due to the difference in computational algorithms. This could affect which service is most appropriate for a particular task.

Cronbach's alpha [54] was used to measure the internal consistency of the Decision Confidence and Relative Advantage questions, respectively. We obtained Cronbach's alpha values of 0.93 for Decision Confidence and 0.96 for Relative Advantage in the first study. Values of 0.90 and 0.94 were obtained for Decision Confidence and Relative Advantage, respectively, in the follow up study. These values are in the "Excellent" range of commonly accepted values [55]. Having found internal consistency, Decision Confidence and Relative Advantage questions were each combined into one respective construct as is common practice.

Table 2
Web services used in experiment.

Pair ID	Web service source	Description
1	Jena Geography	Compute distance between two locations using different data sources
2	Jena Geography	Compute distance between two locations using different algorithms
3	Jena Geography	Determine height above sea level for a location
4	Programmable Web	Return song information for song title given
5	Programmable Web	Currency exchange rate services
6	Programmable Web	Return a word that rhymes with input word
7	Jena Geography	Verify the accuracy of a street address for delivery purposes
8	Jena Geography	Returns the geographic location of an IP address
9	Programmable Web	Return the definition of an input word
10	Programmable Web	Given a musical performer returns all upcoming events for that performer

[4] http://fusion.cs.uni-jena.de/professur/jgd.

[5] http://www.programmableweb.com/.

[6] Statistics as of May 2012.

Table 3
Demographics of participants.

		Initial study	Follow-up study
Gender	Male	36	35
	Female	16	14
Age	19–20	3	0
	21–22	6	6
	23–24	7	2
	> 24	36	41
Education	College sophomore	2	0
	College junior	3	0
	College senior	15	11
	Master student	22	28
	Doctoral student	7	9
	Post-school professional	3	1
Web service familiarity (as determined by Likert scale responses)	Little to no familiarity	7	2
	Some familiarity	14	17
	Very familiar	31	30

There are considerable advantages to using non-parametric tests with Likert scale data [56,57] despite many parametric tests being robust to violations of normality and homogeneity of variance. Following this rationale, we analyzed our data using the non-parametric Mann–Whitney significance tests [58] as well as the non-parametric Cliff delta [59,60] determination of effect size.

Effect size can be understood as the magnitude of the impact that the manipulation of the independent variable causes on the dependent variable [61]. It is a quantification of the treatment importance [62]. The non-parametric Cliff delta effect size ranges from −1 to 1 and represents the degree of overlap between two sets of Likert scale responses. In this study, Cliff delta is calculated relative to the group with service provenance. Thus, a negative effect size indicates that when shown service provenance there is a shift toward lower Likert scale values and less Similarity, Decision Confidence, or Relative Advantage, respectively.

Mann–Whitney non-parametric tests were applied to our data to test for statistical significance in Dissimilarity, Decision Confidence, and Relative Advantage with and without service provenance. Table 4.a lists the Mann–Whitney results of hypothesis testing. We also tested the effects of web service complexity on Dissimilarity, Decision Confidence, and Relative Advantage. By complexity we mean the Jena services vs. the Programmable Web services. We were looking for differences in responses for the “simple” Programmable Web services as compared to the more complex mathematically oriented Jena services. Previous research has clearly indicated the need to capture service execution details within e-Science. However, nothing is known about the impacts of service provenance when applied to more simplistic services such as currency conversion. By examining the Programmable Web services separately we can see if the effects of service provenance translate to more simplistic services. These results can lead credence to the generality and scope of our research. Table 4.b presents the results. We caution, however, that complexity of web services is poorly understood and can be defined in a number of ways [63]. We make a simple, and arbitrary, distinction for complexity — mathematical e-Science services vs. common services used by the general public. No other distinction is made to define or discriminate on complexity.

6.2.4. Power analysis

The power of a statistical test is the probability that the null hypothesis will be rejected given that it is in fact false. Post hoc power analysis [64] is useful after a study has been conducted and computes power as a function of measured effect size and sample size. In post hoc analysis the effect size and sample size are used to assess whether enough participants were available to reliably accept statistical significance. The output of post hoc power analysis is the probability, given the measured effect size and number of participants, that the null hypothesis was correctly rejected. Intuitively, large effects can be seen with few participants while small effects require many participants to ensure that the null hypothesis is correctly rejected. Post hoc power analysis is a means of quantifying this intuitive notion.

We utilized the G*Power 3 software [65] to conduct our post-hoc power analysis. A generally accepted guideline for minimum power is 0.8 [64], meaning there is an 80% chance that the null hypothesis will be rejected if it is indeed false. Thus, a 20% false positive rate for an experiment is considered acceptable [64].

We found that we had reliable power to see medium and large effects. That is, we had enough participants in our study to reliably detect Cliff delta effect sizes of 0.3 or greater. In other words, the parameters of our study indicated that any statistically significant result with an effect size of 0.3 or greater could be accepted. Statistically significant results with an effect size below 0.3 should be rejected, as the false positive rate of those results was too high. Our power analysis implies that we did not have enough participants in our study to reliably detect small effects. However, we do not see this as a limitation. If the use of service provenance only produced small effects on end users then its utility is of little value. Thus, small effects are of little concern. Rather, we are interested in measuring noticeable effects of service provenance, which we see as corresponding to increased value, and our study has enough participants to do so.

Our data and power analysis indicate that Hypothesis 1 was supported. However, Hypotheses 2 and 3 were not supported. Additionally, we did not see any significant difference between Programmable Web and Jena services in terms of Dissimilarity, Decision Confidence, or Relative Advantage.

7. Discussion

There is a measureable difference in dissimilarity assessment with and without service provenance. Participant's views of dissimilarity were shifted toward lower similarity values when exposed to service provenance. We interpret this as service provenance enabling a better assessment of web service functionality and a clearer determination of dissimilarity. We had expected that service provenance would be sufficient to completely discern differences; yet, we saw that some participants still focused on the service interface. This interpretation is supported by participant comments acquired in our experimental data collection survey. This qualitative data revealed that similarity assessment became uncertain for some, as participants reported services were similar in terms of interface, but different in terms of functionality. When exposed to service provenance, some participants gave more weight to the interface while others valued functionality higher in

Table 4.a
Results of Mann–Whitney test and effect size (with/without service provenance).

	Initial study (6 pairs of services)		Follow-up study (10 pairs of services)	
	Means (with, without)	Cliff delta effect size	Means (with, without)	Cliff delta effect size
Dissimilarity	4.33, 5.41*	−0.386 medium	4.42, 5.18*	−0.3 medium
Decision Confidence	5.51, 5.88	No effect	5.61, 5.53	No effect
Relative Advantage	5.42, 5.66	No effect	5.53, 5.26	No effect

* Indicates statistical significance at 0.05 level.

Table 4.b
Results of Mann–Whitney test and effect size (Programmable Web vs. Jena).

	Initial study (6 pairs of services)		Follow-up study (10 pairs of services)	
	Means (PW, Jena)	Cliff delta effect size	Means (PW, Jena)	Cliff delta effect size
Dissimilarity	4.14, 4.59	No effect	5.61, 5.53	No effect
Decision Confidence	5.51, 5.51	No effect	5.53, 5.53	No effect
Relative Advantage	5.43, 5.43	No effect	4.37, 4.46	No effect

their similarity determinations. This may explain why Hypotheses 2 and 3 were not supported. We suspect that despite being able to clearly discern differences in services, the similar interface and varying functionalities led to no measureable difference in Decision Confidence and Relative Advantage.

We found no significant difference between results from the Jena and Programmable Web services. This is interesting given the simplicity of the Programmable Web service pairs. Our results indicate that even for simple tasks, such as currency conversion, the execution details of the web service can lead to changes in how similarity is assessed — and thus how a service is matched to a task. We find this a compelling result, as service provenance seems to be applicable for the spectrum of web services and not just mathematically oriented scientific web services.

Another interesting result emerged from inspecting the responses to pairs 2 and 5. Both of these service pairs are identical in terms of interface. Yet, in the case of no service provenance, not all of the participants rated the services as identical (Likert value 7). One participant remarked that "functionally they could be dissimilar — there is insufficient information." We believe these qualitative responses strengthen our argument and further validate the utility of service provenance. Similarly, when exposed to service provenance for pairs 2 and 5, the results were not all "no similarity" (Likert value 1). Participants saw importance in the interface and some valued a similar interface as more important than differing execution details.

8. Summary and conclusions

Utilizing a web service's provenance information has proven vital to efficient web service discovery in e-Science [11]. Yet, little was known about the generality of this methodology. It has been suggested [66] that provenance will play a central role in emerging digital infrastructures and important steps have been taken to research its theoretical and conceptual aspects. However, provenance remains incomplete, unreliable, and ill-defined [66]. Drawing upon Bunge's ontological theory [27,28], we have developed a comprehensive service provenance ontology by integrating PROV-O [33] and W7 [31] with our own concepts allowing for a unique encapsulation of a service's components and their associated justifications. Such an ontology enables service creators to supply all necessary provenance information of web services, thereby improving web service discovery.

Recent advances in semantic web technologies have rendered previous provenance-based discovery systems obsolete. Our proposed approach utilizes state-of-the art techniques and brings provenance-based discovery in-line with current standards in semantic provenance and semantic web services. Moreover, in validating our approach we have collected empirical data, which has been sorely lacking in provenance research. We have set out to provide empirical evidence on provenance to begin building a foundation for future research.

We have shown that web service provenance information can play a role in service discovery outside of e-Science. However, our results indicate that some users value interface similarities over provenance details while others have the opposite response. This notion can lead to more effective user interface design. Current service discovery applications [e.g. 10,11,17] rank results based solely on interface or process semantics. To the best of our knowledge, there exists no hybrid approach based on user weighting of the interface and execution details. This will be the subject of our future research.

Appendix A. Survey questions

Similarity (1 = Completely Dissimilar, 7 = Exactly the Same)

- How similar are the operations of these two web services?

Decision Confidence (1 = no confidence/not sure, 7 = completely confident/certain)

- How much confidence do you have in your similarity decision?
- How sure are you that you chose the best similarity answer?
- Would you make the same similarity decision again?

Relative Advantage (1 = completely disagree, 7 = completely agree)

- The provided information enables me to make my determination of similarity/dissimilarity more quickly.
- The provided information makes it easier to make my determination of similarity/dissimilarity.
- The provided information enhances my effectiveness in determining similarity/dissimilarity.

References

[1] R. Chinnici, J.-J. Moreau, A. Ryman, S. Weerawarana, Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language, W3C Recommendation, available at: http://www.w3.org/TR/wsdl20/ 2007 (last accessed April 30, 2011).

[2] M. Herold, State-of-the-Art Semantic Web Services: Evaluation and Advancement in Context of a Tourist Information System, (Masters Thesis) University of Applied Sciences, Wedel, Germany, 2008. (Available online at: http://tinyurl.com/bcw4dud last accessed April 24, 2011).

[3] J. Kopecky, E. Simperl, Semantic web service offer discovery for e-commerce, Proceedings of the 10th International Conference on Electronic Commerce 2008, Innsbruck, Austria, August 19–22, 2008.

[4] J. Hendler, Agents and the semantic web, IEEE Intelligent Systems 16 (2) (2001) 30–37.

[5] C. Pedrinaci, J. Domingue, R. Krummenacher, On the integration of services with the web of data, Technical Report kmi-09-06, Knowledge Media Institute, The Open University, 2010, (Available at http://kmi.open.ac.uk/publications/pdf/kmi-09-06.pdf , last accessed April 24, 2011).

[6] J. Gray, Jim Gray on eScience: a transformed scientific method, based on the transcript of a talk given by Jim Gray to the NRC-CSTB in Mountain View, CA, on January 11, 2007, in: Tony Hey, Stewart Tansley, Kristin Tolle (Eds.), The Fourth Paradigm: Data-Intensive Scientific Discovery, 2009.

[7] J. Zhoa, O. Corcho, P. Missier, K. Belhajjame, D. Newman, D. de Roure, C.A. Goble, eScience, in: J. Domingue, D. Fensel, J.A. Hendler (Eds.), Handbook of Semantic Web Technologies, ISBN: 978-3-540-92912-3, 2011, pp. 701–736.

[8] C. Goble, D. De Roure, The impact of workflow tools on data-centric research, in: Tony Hey, Stewart Tansley, Kristin Tolle (Eds.), The Fourth Paradigm: Data-Intensive Scientific Discovery, 2009.

[9] S. Miles, P. Groth, M. Branco, L. Moreau, The requirements of recording and using provenance in e-Science experiments, Technical Report, Electronics and Computer Science, University of Southampton, 2005.

[10] A. Gunay, P. Yolum, Service matchmaking revisited: an approach based on model checking, J. Web Semant. (2010) 292–309.

[11] C. Wroe, R. Stevens, C. Goble, A. Roberts, M. Greenwood, A suite of DAML + OIL ontologies to describe bioinformatics web services and data, International Journal of Cooperative Information Systems. Special issue on Bioinformatics and Biological Data Management 12 (2) (2003) 197–224.

[12] Y. Amsterdamer, S.B. Davidson, D. Deutch, T. Milo, J. Stoyanovich, Putting lipstick on pig: enabling database-style workflow provenance, Proceedings of the VLDB Endowment, Vol. 5, No. 4, August 27–31, Istanbul, Turkey, 2012.

[13] A. Shaon, S. Callaghan, B.N. Lawrence, B. Matthews, Opening up climate research: a linked data approach to publishing data provenance, The International Journal of Digital Curation 7 (1) (2012) 163–173.

[14] J. Farrell, H. Lausen, Semantic Annotations for WSDL and XML Schema, W3C Recommendation, available online at: http://www.w3c.org/TR/sawsdl/ August 2007 (last accessed April 24, 2011).
[15] P. Fox, The rise of informatics as a research domain, Proceedings of WIRADA Science Symposium, Melbourne, Australia, 1–5 August 2011, 2012, pp. 125–131.
[16] M. Klusch, B. Fries, K. Sycara, Automated semantic web service discovery with OWLS-MX, Proceedings of AAMAS 2006, May 8–12, Hakodate, Hokkaido, Japan, 2006.
[17] A. Zaremski, J. Wing, Specification matching of software components, ACM Transactions on Software Engineering and Methodology 6 (4) (1997) 333–369.
[18] E. Toch, I. Reinhartz-Berger, D. Dori, Humans, semantic services and similarity: a user study of semantic web services matching and composition, Web Semantics: Science, Services and Agents on the World Wide Web 9 (2011) (2011) 16–28.
[19] M. Klusch, B. Fries, M. Khalid, K. Sycara, Owls-mx: hybrid semantic web service retrieval, Proceedings of 1st Intl. AAAI Fall Symposium on Agents and the Semantic Web, AAAI Press, 2005.
[20] R. Mikhaiel, E. Stroulia, Examining Usage Protocols for Service Discovery, ICSOC, 2006. 496–502.
[21] F. Martin-Recuerda, D. Robertson, Discovery and uncertainty in semantic web services, Proceedings of Uncertainty Reasoning for the Semantic Web, Galway, Ireland, November 2005, 2005.
[22] L. Chen, Z. Jiao, Supporting provenance in service-oriented computing using the semantic web technologies, IEEE Intelligent Informatics Bulletin 7 (1) (December 2006).
[23] M. Klein, A. Bernstein, Toward high-precision service retrieval, IEEE Internet Computing 8 (1) (2004) 30–36.
[24] A. Wombacher, P. Fankhauser, B. Mahleko, E. Neuhold, Matchmaking for business processes based on conjunctive finite state automata, International Journal of Business Process Integration and Management 1 (1) (2005) 3–11.
[25] D.L. McGuinness, F. van Harmelen, Web Ontology Language, W3C Recommendation, available at: http://www.w3.org/TR/owl-features/last February 10 2004 (accessed: June 30, 2011).
[26] M.L. Sbodio, D. Martin, C. Moulin, Discovering semantic web services using SPARQL and intelligent agents, Journal of Web Semantics 8 (2010) 310–328.
[27] M. Bunge, Treatise on basic philosophy, Ontology I: The Furniture of the World, vol. 3, Reidel, Boston, MA, 1977.
[28] M. Bunge, Treatise on basic philosophy, Ontology II: A World of Systems, vol. 4, Reidel, Boston, MA, 1979.
[29] Y. Wand, R. Weber, An ontological model of an information system, IEEE Transactions on Software Engineering 16 (11) (1990) 1282–1292.
[30] Y. Wand, R. Weber, On the deep structure of information systems, Information Systems Journal 5 (3) (1995) 203–223.
[31] K. Belhajjame, J. Cheney, D. Garijo, M. Lang, S. Soiland-Reyes, S. Zednik, J. Zhao, Prov-O: the PROV ontology, in: By T. Lebo, S. Sahoo, D. McGuinness (Eds.), W3C Working Draft, May 2012, 2012, (http://www.w3.org/TR/prov-o/).
[32] S. Ram, J. Liu, Understanding the Semantics of Data Provenance to Support Active Conceptual Modeling, LNCS 4512, Springer-Verlag, 2007. 17–29.
[33] S. Zednik, P. Fox, D.L. McGuinness, P. Pinheiro da Silva, C. Chang, Semantic provenance for science data products: application to image data processing, in: Juliana Freire, Paolo Missier, Satya Sanket Sahoo (Eds.), Proceedings of the First International Workshop on the role of Semantic Web in Provenance Management (SWPM 2009), Washington DC, USA, October 25, 2009, CEUR Workshop Proceedings, 526, 2009, (CEUR-WS.org).
[34] S. Zednik, P. Fox, D.L. McGuinness, System transparency, or how i learned to worry about meaning and love provenance! Proc. 3rd International Provenance and Annotation Workshop (IPAW2010), Troy, NY, USA (June 15–16, 2010), LNCS, 2010.
[35] D.L. McGuinness, P. Pinheiro da Silva, Explaining answers from the semantic web: the inference web approach, Journal of Web Semantics 1 (4) (October 2004) 397–413.
[36] X. Li, T. Lebo, D.L. McGuinness, Provenance-based strategies to develop trust in semantic web applications, Proceedings of the Third International Provenance and Annotation Workshop (IPAW 2010), 2010.
[37] A. Chapman, H.V. Jagadish, Understanding provenance black-boxes, Distributed and Parallel Databases 27 (2010) 139–167.
[38] P. Buneman, S. Khanna, W.C. Tan, Why and where: a characterization of data provenance, ICDT 2001 (2001) 316–330.
[39] C. Bizer, T. Heath, T. Berners-Lee, Linked data — the story so far, International Journal on Semantic Web and Information Systems 5 (3) (2009) 1–22.
[40] L. Ding, J. Michaelis, J. McCusker, D. McGuinness, Linked provenance data: a semantic web-based approach to interoperable workflow traces, Future Generation Computer Systems 27 (6) (June 2011) 797–805.
[41] J. Zhoa, S.S. Sahoo, P. Missier, A.P. Sheth, C.A. Goble, Extending semantic provenance into the web of data, IEEE Internet Computing 15 (1) (2011) 40–48.
[42] N.F. Noy, C.D. Hafner, The state of the art in ontology design — a survey and comparative review, AI Magazine 36 (3) (1997) 53–74.
[43] J.F. Sowa, Distinctions, combinations, and constraints, Proceedings of the Workshop on Basic Ontological Issues in Knowledge Sharing, 19–20 August, Montreal, Canada, 1995.
[44] S. Staab, H.P. Schnurr, R. Studer, Y. Sure, Knowledge processes and ontologies, IEEE Intelligent Systems 16 (1) (2001) 26–34.
[45] P. Fox, D.L. McGuinness, An open-world iterative methodology for the development of semantically-enabled applications, 27th AAAI Conference on Artificial Intelligence, Washington, 2013. http://videolectures.net/aaai2013_fox_semantically_enabled_applications/.
[46] A. Hevner, S.T. March, J. Park, S. Ram, Design science research in information systems, MIS Quarterly (28:1) (2004) 75–105.
[47] J. Yu, J.A. Thom, A. Tam, Requirements-oriented methodology for evaluating ontologies, Information Systems 34 (2009) 766–791.
[48] E.M. Rogers, Diffusion of Innovations, Free Press, New York, 1983.
[49] S.J. Harrington, C.P. Ruppel, Telecommuting: a test of trust, competing values, and relative advantage, IEEE Transactions on Professional Communication 42 (4) (1999) 223–239.
[50] G.M. Kasper, A theory of decision support system design for user calibration, Information Systems Research 7 (2) (1996) 215–232.
[51] I.M. Duhaime, C.R. Schwenk, Conjectures on cognitive simplification in acquisition and divestment decision making, The Academy of Management Review 10 (2) (1985) 287–295.
[52] J.A. Sniezek, Groups under uncertainty: an examination of confidence in group decision making, Organizational Behavior and Human Decision Processes 52 (1) (June 1992) 124–155.
[53] M. Stollberg, Scalable Semantic Web Service Discovery for Goal-driven Service-Oriented Architectures, (PhD Thesis) Leopold-Franzens Universität, Innsbruck, March 17 2008. (available at: http://www.michael-stollberg.de/phd/html-sources/thesis.html , last accessed May 1, 2011).
[54] L.J. Cronbach, Coefficient alpha and the internal structure of tests, Psychometrika 16 (3) (1951) 297–334.
[55] D. George, P. Mallery, SPSS for Windows Step by Step: A Simple Guide and Reference. 11.0 Update, 4th ed. Allyn & Bacon, Boston, 2003.
[56] M.J. Nanna, S.S. Sawilowsky, Analysis of Likert scale data in disability and medical rehabilitation research, Psychological Methods 3 (1998) 55–67.
[57] J. Romano, J.D. Kromrey, J. Coraggio, J. Skowronek, Appropriate Statistics for Ordinal Level Data: Should We Really be Using t-Test and Cohen's d for Evaluating Group Differences on the NSSE and Other Surveys? Florida Association of Institutional Research, Cocoa Beach, Florida, February 1–3 2006.
[58] H.B. Mann, D.R. Whitney, On a test of whether one of two random variables is stochastically larger than the other, Annals of Mathematical Statistics 18 (1) (1947) 50–60.
[59] N. Cliff, Dominance statistics: ordinal analyses to answer ordinal questions, Psychological Bulletin 114 (1993) 494–509.
[60] N. Cliff, Answering ordinal questions with ordinal data using ordinal statistics, Multivariate Behavioral Research 31 (1996) 331–350.
[61] G. Macbeth, E. Razumiejczyk, R.D. Ledesma, Cliff's Delta calculator: a non-parametric effect size program for two groups of observations, Universitas Psychologica 10 (2) (2011) 545–555.
[62] J.E. Hunter, F.L. Schmidt, Methods of Meta-Analysis. Correcting Error and Bias in Research Findings, Second edition Sage, Thousand Oaks, CA, 2004.
[63] Q.P. Thi, D.T. Quang, T.H. Quyet, A Complexity Measure for Web Service, Proceedings of the International Conference on Knowledge and Systems Engineering, October 13–17, 2009, 2009, pp. 226–231.
[64] J. Cohen, Statistical Power Analysis for the Behavioral Sciences, Academic Press, San Diego, CA, 1988.
[65] F. Faul, E.E. lder, A.-G. Lang, A. Buchner, G*Power 3: a flexible statistical power analysis program for the social, behavioral, and biomedical sciences, Behavior Research Methods 39 (2) (2007) 175–191.
[66] J. Cheney, S. Chong, N. Foster, M. Seltzer, S. Vansummeren, Provenance: a future history, Proceedings of OOPSLA 2009, October 25–29, Orlando, Florida, USA, 2009, pp. 957–964.

Tom Narock is an Assistant Professor in the Department of Information Technology and Management Science at Marymount University in Arlington, VA. Dr. Narock teaches and researches in the areas of Data Science and Semantic Web Technologies. He is particularly interested in applications to eScience.

Victoria Yoon is a Professor in the Department of Information Systems at the Virginia Commonwealth University. She received her M.S. from the University of Pittsburgh and her Ph.D. from the University of Texas at Arlington. She has published articles in such leading journals as MIS Quarterly, Decision Support Systems, Communications of the ACM, and Journal of Management Information Systems. Her primary research area has been the application of Artificial Intelligence to business decision-making in organizations and technical and social issues surrounding such applications.

Salvatore March is the David K. Wilson Professor of Management in the Owen School of Business at Vanderbilt University. He teaches courses on Information Technology and Internet Commerce, Database Management Systems, and Managerial Economics. His primary research interests are in the areas of Information System Development, Electronic Commerce, Logical and Physical Database Design, Distributed Database Design, and the Economics of Information. He is currently an Associate Editor for *Decision Sciences Journal*, *Information Systems Frontier*, and *Journal of Database Management* and a Senior Editor for *Information Systems Research*.