ourSpaces - Design and Deployment of a Semantic Virtual Research Environment

Peter Edwards, Edoardo Pignotti, Alan Eckhardt, Kapila Ponnamperuma, Chris Mellish, and Thomas Bouttaz

Computing Science, University of Aberdeen, Aberdeen AB24 5UA, UK,
{p.edwards, e.pignotti, a.eckhardt, k.ponnamperuma, c.mellish, t.bouttaz}@abdn.ac.uk

Abstract. In this paper we discuss our experience with the design, development and deployment of the *ourSpaces* Virtual Research Environment. *ourSpaces* makes use of Semantic Web technologies to create a platform to support multi-disciplinary research groups. This paper introduces the main semantic components of the system: a framework to capture the provenance of the research process, a collection of services to create and visualise metadata and a policy reasoning service. We also describe different approaches to support interaction between users and metadata within the VRE. We discuss the lessons learnt during the deployment process with three case study groups. Finally, we present our conclusions and future directions for exploration in terms of developing *ourSpaces* further.

Keywords: Provenance, Virtual Research Environment, Policies, NLG

1 Introduction

Research challenges are becoming increasingly complex requiring researchers from different institutions and different disciplines to work together. At the same time, a range of information technologies have gradually been adopted by researchers to support the transfer of ideas, knowledge and resources, leading to the emergence of Web-based Virtual Research Environments (VREs) [1]. These have been proposed as one way to help researchers in all disciplines to manage the increasingly complex range of tasks involved in carrying out research. In the UK, the Joint Information Systems Committee (JISC) VRE programme[1] explored the virtual research environment collaborative landscape. Results from this programme concluded that one of the most important tasks for the academic community is to provide general frameworks that can be used to develop and host different VREs. Such frameworks should provide core services (such as authentication and rights management; repositories; project planning, collaboration and communication tools) and allow the development or easy integration of modules for specific uses. JISC also recognised that a major shift in research practices will occur through the formation of common taxonomies, data standards and metadata as researchers collaborate with others across disciplinary, institutional and national boundaries [1]. Semantic web technologies [2] are seen as crucial in this context in order to

[1] http://www.jisc.ac.uk/whatwedo/programmes/vre.aspx

provide a common framework to allow the creation of intelligent applications and services which can be integrated with data resources, people and other objects in a VRE.

Some of the issues highlighted above have been explored by the PolicyGrid[2] project, a collaboration between human geographers and computer scientists as part of the UK Digital Social Research initiative. As part of this project we have developed *ourSpaces*[3], a semantic VRE which aims to provide a collaborative on-line environment for interdisciplinary academic research communities. Groups using *ourSpaces* work in socio-environmental and health-related domains and there are currently 183 registered users. A screenshot of the *ourSpaces* web interface in presented in Figure 1.

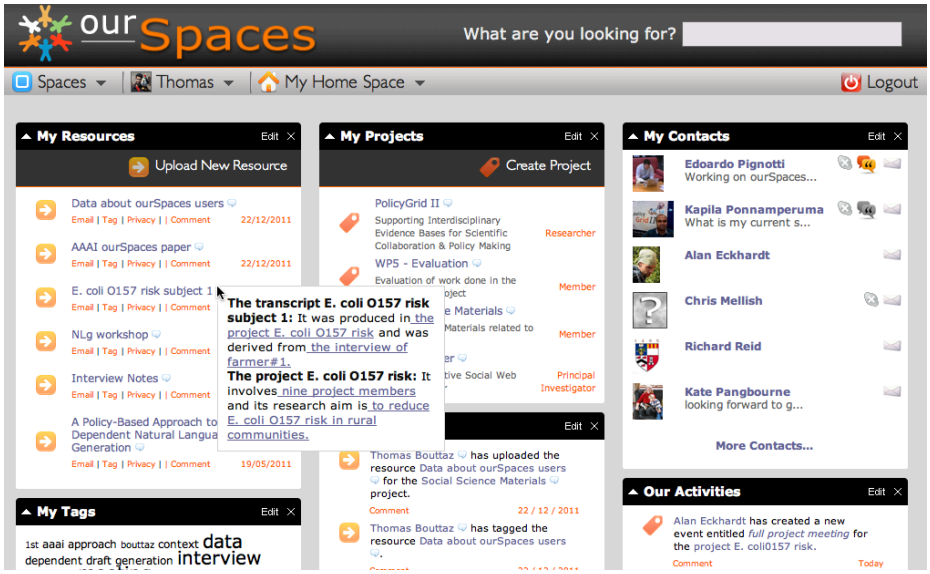

Fig. 1. A screenshot of the *ourSpaces* VRE.

Managing provenance is one of the main aspects of the *ourSpaces* VRE which is required in order to make the context surrounding research artefacts more transparent. Understanding the provenance of scientific data is crucial [3] in order to understand and verify its authenticity and completeness. Provenance (also referred to as lineage or audit trail) captures the derivation history of an artefact, including the original sources, the intermediate products and the steps that were applied to produce that artefact. Within the environment, there is also a need to manage users and their behaviours so that they comply with certain policies. For example, a user may impose certain access constraints on digital artefacts that he/she owns, such as restricting an artefact to people within their social network. Semantic web technologies play an important role in supporting

[2] http://www.policygrid.org
[3] http://www.ourspaces.net

the management of provenance and policies. At the heart of *ourSpaces* is an extensible ontological framework describing different aspects of the provenance of the research process [4] and a reasoning service for provenance and policies [5].

Semantic Web technologies have already been applied to virtual research environments. For example, *myExperiment* [6] enables people to share digital objects associated with their research. The notion of *research objects* is used in *myExperiment* to provide a container for semantic aggregation of resources produced and consumed by common services. *myExperiment* uses ontologies to support the publication of such objects as RDF so they can be shared within and across organisational boundaries.

Semantic Web approaches have also been used in enterprise knowledge management tools [7]. For example, the IBM *WebSphere* Portal [8] uses ontologies to support different aspects of document management such as tagging and searching. All three approaches (including *ourSpaces*) employ Semantic Web technologies to provide a representational framework that can be used across different domain applications. However, the main difference between *ourSpaces* and other semantic environments is that it utilises policy reasoning to control the behaviour of users and services. This allows us to adapt our environment to meet domain-specific requirements without changing the logic behind services. We have also developed a number of general-purpose services for creating and visualising Semantic Web data.

In the remainder of this paper we discuss the role of Semantic Web technologies in the design, development and deployment of the *ourSpaces* system. In section 2 we describe the VRE architecture and its underlying components. In section 3 we present the user interfaces that we have developed in order to support interaction with semantic metadata. In section 4 we discuss the lessons learnt during the deployment of *ourSpaces* with our case-study communities. Finally, we present our conclusions and future directions for exploration in terms of developing *ourSpaces* further.

2 The *ourSpaces* Semantic Architecture

Based upon interactions with case study groups and communities, initial requirements for the *ourSpaces* VRE were identified. Even though the requirements for the VRE were clear in broad terms, the finer grain requirements relating to the system architecture and user interfaces were much less well defined due to the diversity of the case study groups. We therefore decided to continuously involve the users in the development of the VRE. This is discussed in more detail in section 4. The initial requirements are summarised below:

- It should be possible to describe and uniquely identify a range of entities: artefacts (digital and physical); processes (both computational services and human activities); people; organisational structures and membership; social networks.
- The system should incorporate online communication (e.g. instant messaging, blog entries, email) into the provenance record.
- It should be possible to define relationships (e.g. causal, social, organisational) between entities.
- It should be possible to define access control and documentation policies.

Following these requirements we have developed an architecture which is summarised in Figure 2. Underpinning *ourSpaces* are a number of repositories and services. Each activity within the environment is enabled by a rich and pervasive RDF metadata infrastructure built upon a series of OWL ontologies (which are described more fully in section 2.1). Information is stored in metadata repositories (using the Sesame[4] triple store), databases (using MySql) and digital artefact repositories (using an NFS filesystem).

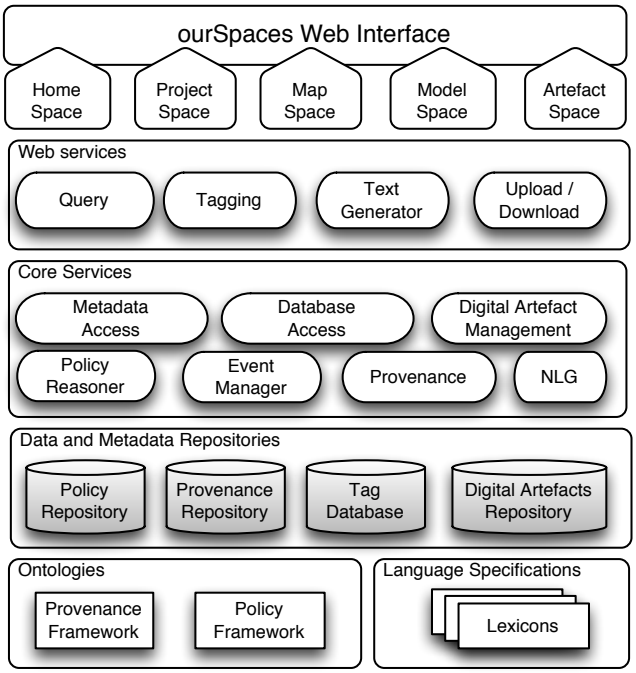

Fig. 2. The *ourSpaces* architecture.

The *ourSpaces* architecture implements a number of core and Web services for creating, editing and querying data, metadata and digital artefacts. These include a natural language service to support browsing and querying data, a policy reasoning service and a service used to download and upload digital artefacts.

The *ourSpaces* user interface was developed using Web technologies such as Java Server Pages[5], JavaScript[6] and jQuery[7]. The interface is structured around the concept

[4] http://www.openrdf.org/
[5] http://www.oracle.com/technetwork/java/javaee/jsp/index.html
[6] http://www.ecma-international.org/publications/standards/Ecma-262.htm
[7] http://jquery.com/

of a "space" (shown in Figure 2 - top), designed as a means to link, browse and share specific categories of data resources with other users. For example, the project space is used to manage a research project, the model space for handling simulation models, and the map space for browsing geospatial information.

In the remainder of this section we focus on two main components of the system where Semantic Web technologies were used: the provenance framework and the policy reasoning service.

2.1 Provenance Framework

We have designed and developed an extensible ontological framework for capturing the provenance of the research process based on the requirements highlighted in section 2. In order to describe and uniquely identify entities (such as artefacts, people, locations) and to make explicit relations between entities we followed the linked data principles [9].

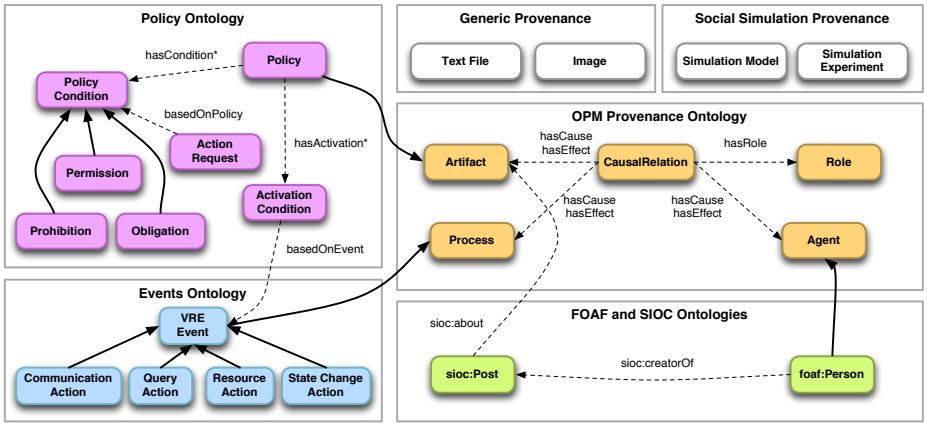

Fig. 3. *ourSpaces* ontological framework.

At the heart of *ourSpaces* (and thus, our provenance framework) is an OWL representation of the Open Provenance Model [10]. This ontology defines the primary entities of OPM as well as the causal relationships that link them (see Figure 3, OPM Provenance Ontology). OPM is a generic solution and as a result, our framework supports additional domain-specific provenance ontologies that are created by extending the concepts defined in the OPM ontology with domain-specific classes. For example, in a social simulation domain ontology (see Figure 3, Social Simulation Provenance), one might have a `Simulation Model` as a type of artefact and a `Simulation Experiment` as a type of process. To date we have developed a number of domain-specific provenance ontologies describing aspects of Human Geography and Social Simulation. Using these ontologies it is possible, for example, to describe a physical

research activity (e.g. an interview) as an `opm:Process`, and how such an activity causes an `opm:Artifact` to be generated (the interview notes).

Based on the requirements from our case study groups, the provenance framework should not only capture information regarding artefacts and processes, but must be able to situate these alongside people and their associated organisational structures. Friend-of-a-Friend[8] (FOAF) is an established RDF vocabulary for describing people and their social networks and we have opted to utilise this within our framework; a `foaf:Profile` is thus a subclass of `opm:Agent`. Several FOAF profiles are visible in Figure 1, as contacts of the user (My Contacts). Organisational structures such as projects or employer institutions can also be defined, and users within *ourSpaces* may belong to several projects or groups. An event ontology (Figure 3 - lower left corner) was introduced to describe events taking place in the system and to allow the policy framework to operate (see section 2.2 for more details).

Another requirement was to capture the provenance of online communication within the social network. However, the current OPM specifications support limited information about the relationship between a person (`opm:Agent`) and the research process (`opm:Process`). As a result, we have integrated the social networking vocabulary SIOC[9] (Semantically-Interlinked Online Communities) within our provenance framework. The SIOC ontology is designed to enable the integration of online community information by providing a model to express user-generated content such as posting a message in a blog or posting a comment. Using this vocabulary, traditional artefact and process-driven provenance can be extended to incorporate social data. For example, a collaborator (defined with `foaf:worksWith`) could post a comment (`sioc:Post`) about some artefact (e.g. `prov:Paper`) uploaded by a colleague asking for some clarification about the method used to generate the data.

2.2 Policy Reasoning

Projects or individual users in *ourSpaces* may have different metadata requirements and access restrictions associated with their data. For example, a user may impose certain access constraints on digital artefacts that he/she owns, e.g. an artefact may only be accessible to users who are members of a particular project and who contributed towards creation of the artefact (i.e. were named as a co-author). As a result, there is a need for a framework to support reasoning about policies within the VRE. We have thus extended our provenance framework to define such policies as a combination of obligations, prohibitions or permissions.

We have combined the existing OWL binding of the Open Provenance Model with an OWL ontology (inspired by the work of Sensoy et al. [11]) defining the concepts introduced above. An extract of the provenance policy ontology is shown in Figure 3 (Policy Ontology). Moreover, we make use of the SPIN ontology[10] to support the use of the SPARQL query language to specify rules and logical constraints necessary to reason about policies. The SPIN ontology allows SPARQL queries to be represented in

[8] http://www.foaf-project.org/
[9] http://sioc-project.org/
[10] http://spinrdf.org/spin.html

RDF and associated to classes in an ontology using two pre-defined description properties. `spin:constraint` can be used to define conditions that all members of a class must fulfil; `spin:rule` can be used to specify inference rules using SPARQL CONSTRUCT, DELETE and INSERT statements. An example of a `spin:rule` is presented in Figure 7. In our ontology a policy is a combination of `PolicyCondition` instances described by the property `hasCondition`*. Each condition can be defined as an Obligation, Prohibition or Permission depending on the nature of the policy. We define a condition as a `spin:Construct` query describing its logic in the form of an *if-then* statement where *if* is represented by the WHERE block of the query and *then* by the CONSTRUCT block of the query (see Figure 7). Once processed by the SPIN reasoner a *spin:Construct* can assert a new `ActionRequest` instance which is constructed as part of the query, such as the `NLGRectractionRequest` in Figure 7. A policy in our ontology also has one or more `ActivationCondition` instances describing the activation condition of the policy via a `spin:Construct` query. As a result of an activation, the `spin:Construct` query asserts a new `PolicyActivation` instance. A `PolicyActivation` links a specific policy instance to the event that activated the policy, e.g. a resource action `UploadResource`.

When an activity is detected in the system, the event manager initiates a *policy session*. The *PolicyReasoner* checks if any of the policies stored in the policy repository can be activated by running the SPIN reasoner against the `spin:rule` instances associated with the policies and stores the outcome of the activation in the *policy session*. In order to reason about obligation, permission or prohibition conditions we require a reasoning mechanism able to check conditions over a provenance graph. This can be seen as a semantic matchmaking problem where a functional description of a condition is matched to a subset of a provenance graph. This is done by evaluating each condition defined as a `spin:rule`. For an obligation, conditions have to be met; for a prohibition, the condition cannot be met; and for a permission, the condition might (or might not) be met.

Using this approach in *ourSpaces* we were able to implement a policy for use by one of the project teams using the system. The policy specifies the kind of metadata required for artefacts that will eventually be archived to the UK social science data archive UKDA[11]. More specifically, the policy is created by the PI of the project and it is addressed to its members. The policy is activated when a person uploads an artefact. The required metadata vary for each artefact type, e.g. the policy activated by upload of a paper consists of three obligation conditions specifying that the title, author and date of publication are required.

3 Interaction with Semantic Web Data

Publishing data according to the linked data principles typically involves three main steps: choosing URIs and vocabularies, generating links and creating associated metadata [9]. Smith [12] states that it is challenging for non technical users to create and consume semantic metadata and this is considered to be a major issue while creating

[11] http://www.data-archive.ac.uk/

user interfaces. In this section we describe solutions integrated within the *ourSpaces* VRE to support the creation and visualisation of metadata.

3.1 Creating Semantic Web Data

We have developed a web interface to make creation of metadata by the users of the VRE as intuitive as possible, by allowing them to utilise a traditional web form and by automatically generating metadata were possible.

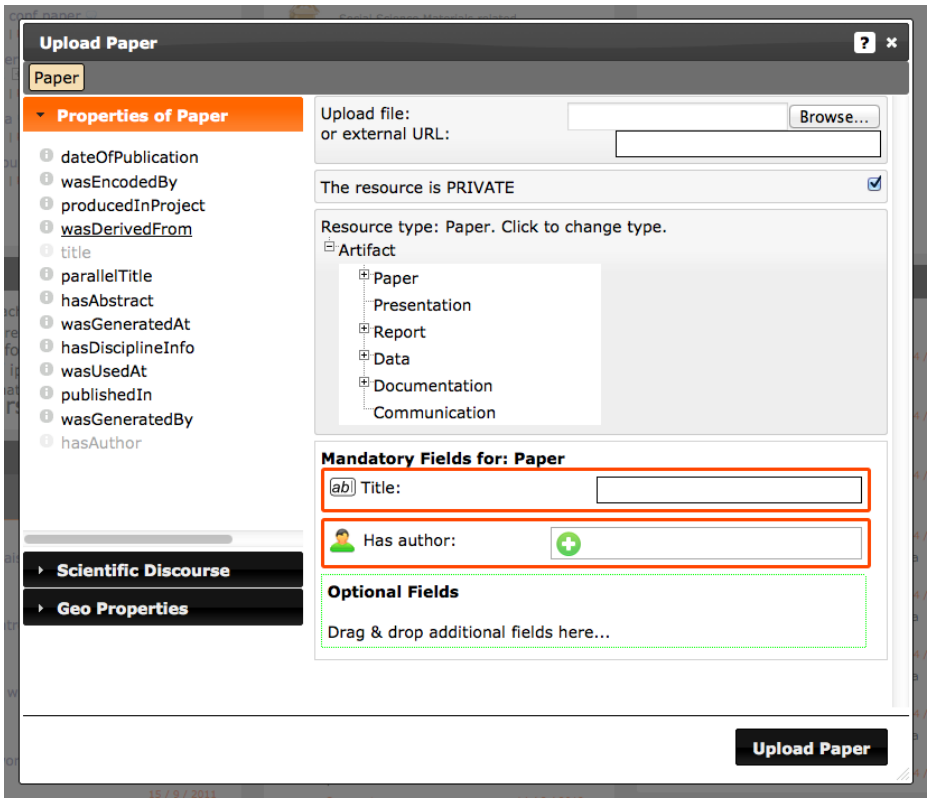

Fig. 4. A screenshot of the metadata creation form in *ourSpaces*.

Consider the example where a user would like to upload into the VRE a new journal article he has published in order to share it with other researchers. By opening the upload form (see Figure 4), the user will be asked to specify the type of artefact he is uploading by selecting from a tree-like structure (see Figure 4-centre: *Resource type*). The tree is dynamically generated by processing the ontologies described in section 2. Once the user has selected an artefact type from the tree, the properties associated with

it are displayed in the form (see Figure 4-left: *Properties of Paper*), by processing the appropriate ontology. Mandatory properties are automatically added depending on the cardinality defined in the ontology. Other properties might also be required depending on relevant policy activations. This can occur while opening the form or while selecting properties and entering values. The user is also able to select additional properties from a list. When inputting the values of a property, the upload form also guides the user as to what type of information should be entered by looking at the *rdf:range* axioms associated with the property. If the range of a property is an object (e.g. a `Person` for the `hasAuthor` property), the interface will use an autocomplete search functionality to help the user find an existing object from the repository.

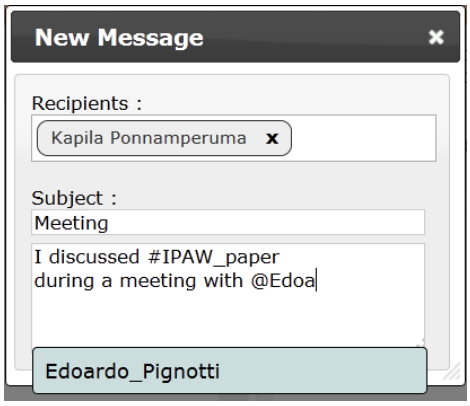

Fig. 5. Using # and @ tags.

We have also developed methods to support the creation of semantic links within communication items such as messages, comments and posts. For instance, when writing a message to a colleague, a user can refer to a person or an artefact in the system, by using @ (for people) or # (for artefacts) tags in combination with an autocomplete search function which returns instances from the repository. Examples of such links are illustrated in Figure 5.

3.2 Metadata Visualisation

In order to allow users to browse metadata, we have incorporated different metadata visualisation modalities including a natural language generation interface, a space-based interface, and a graph-based interface. The behaviour of these interfaces can be controlled by the policy framework in order to adapt to the specific preferences of a user or a project. We will now describe each of these visualisation modalities in turn.

Natural Language Generation Interface
We have developed a service enabling users to generate short textual descriptions of the

resources stored in the *ourSpaces* repository. This service translates RDF statements into English sentences, based on the approach described by [13]. To generate the description of a particular RDF resource, this service queries the metadata repository with the ID of that resource to retrieve all related statements. A local model is then built from that list of statements, representing the information about the resource (see Figure 6 for an example of such a model).

This model is then used by the NLG service that converts its axioms into plain text, using the appropriate *language specification* files. These files (encoded in XML) describe how axioms should be translated in English. Each file represents a particular property in the ontology and contains linguistic information about how to structure the sentence corresponding to that property (e.g. syntactic category, verb, source and target). In the example shown in Figure 6, the *Transcript* artefact has a *producedInProject* property with a value of *E. coli O157 risk*. The language specification of *producedInProject* will indicate that this information must be rendered as: "The transcript was produced in the project E. coli O157 risk". Those files use a dependency tree structure to represent the relationships between the different syntactic units of the sentence. This allows properties with similar syntactical structures to be aggregated together in the text. The final stage of linguistic realisation is carried out using the SimpleNLG realiser [14], which converts abstract representations of sentences into actual text using rules of grammar (morphology and syntax).

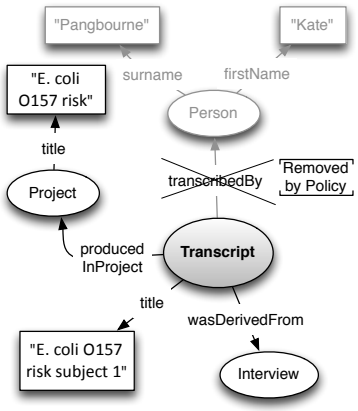

Fig. 6. Internal representation of a resource described by the NLG service.

```
CONSTRUCT {
   _:b0 a pol:NLGRetractionRequest .
   _:b0 pol:requestAboutResource ?this .
   _:b0 pol:requireObject ?rmObject .
   _:b0 pol:requirePredicate pggen:transcribedBy .
   _:b0 pol:requireSubject ?this .
}
WHERE {
   ?this a pggen:Transcript .
   ?this pggen:transcribedBy ?rmObject .
   ?policy pol:basedOnEvent ?nlgAction .
   ?policy pol:activePolicy :NLGTranscriptPolicy .
   ?nlgAction a vre:NLGAction .
   ?nlgAction  opmv:Used ?this .
   ?nlgAction opmv:WasControlledBy ?person .
   NOT EXISTS {
      ?this pggen:producedInProject ?project .
      ?project project:hasMemberRole ?role .
      ?role project:roleOf ?person .
   } .
}
```

Fig. 7. Example of policy condition.

To integrate this mechanism within *ourSpaces*, we developed a RESTful service that requests a textual description when a user hovers the mouse pointer on top of a resource name in the interface. The generated text is presented to the user in a popup window, who then has the option to further explore the repository by clicking anchors to related resources. Every time a user clicks on an anchor, the service is invoked with

a new resource ID and the generated text is inserted in the same popup window. For example in Figure 1, the user generated the description of *the transcript E. coli 0157 risk subject 1* and then requested more information about a related resource (*the project E. coli 0157 risk*).

ourSpaces is designed to support collaboration within multidisciplinary research groups. Members of our case study groups have often stressed that people from different backgrounds tend to have different information presentation preferences. Empirical evidence also suggests [1] that there is a need to adapt information interfaces to users and their context. This can include information about the users themselves, the task they are currently performing including information about the project they are working on. To address this issue, we used the policy framework described in section 2.2 to choose between data presentation strategies (e.g. graphical or textual visualisation to explore the provenance graph), as well as controlling the content of the text generated by the NLG service ([15]). The latter is a kind of rule-based content determination ([16]). For example, the principal investigator of a project might want to protect the identity of the person that transcribed an artefact from users that are not members of that project. This preference can be expressed by constructing a policy enforcing a rule similar to the one shown in Figure 7. This rule triggers an action request to remove the *transcribedBy* property, if the user visualising the description of a *Transcript* is not a member of the project where that artefact was produced. The corresponding nodes are removed from the model used by the NLG service, as shown in Figure 6. The use of this policy is illustrated in Figure 1, where the private information has been omitted from the textual description. The policy framework can also be used to assert information to the model used for NLG. By retrieving information deeper in the repository (i.e. metadata about related resources), a more complete description of an artefact can be generated. In this manner, the NLG service combined with the policy framework allows the system to generate descriptions aligned to the user's context and preferences.

Space and Graphical Visualisation Interfaces

As mentioned earlier in this paper, a "space" acts as a container to provide access to information about a specific resource or a category of resources. In order to generate a space a number of SPARQL queries are performed over the provenance repository, extracting relevant sections of the RDF graph. Different metadata visualisation modalities can be utilised within a space (table, graph-based and NLG-based) all of which may offer links to other resources. By exploring such links, the focus of the space changes thus providing the user with a tool to navigate the graph. To illustrate this, an example of an *Artefact Space* is presented in Figure 8 showing data about a *focus group* resource. The properties of the artefact are presented in a table (see Figure 8 - top left). The same information is also described using natural language (see Figure 8 - top right). Related artefacts are also presented. Figure 8 (bottom left) illustrates the graphical interface used to visualise metadata about the provenance of the artefact. In this example the *focus group* is highlighted and the immediate provenance properties and values are displayed. By clicking the + button, the user can expand the graph in order to explore additional provenance information. Moreover, by hovering the mouse pointer over pro-

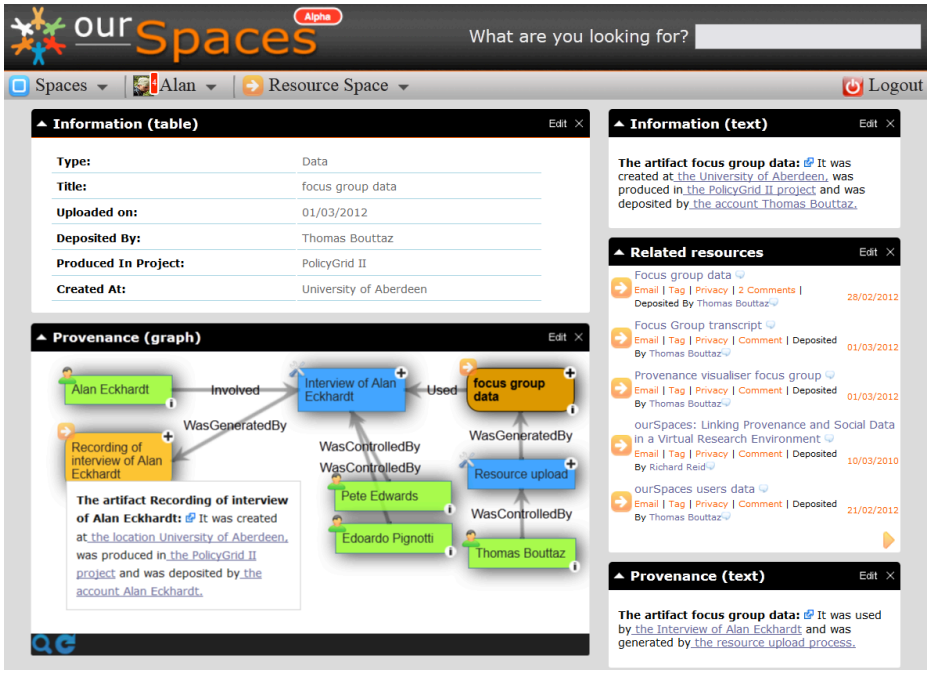

Fig. 8. A screenshot of the *Artefact Space*.

cesses or artefacts the user is presented with additional information which is rendered in plain text by the NLG service.

4 Lessons Learnt from Deploying the VRE

ourSpaces has been deployed for use with three interdisciplinary case study groups: a 'project' (all researchers working on the ESRC-NERC funded RELU programme's 'Reducing Escherichia coli O157 risk in rural communities' project); a research 'community' (all members and affiliates of the Aberdeen Centre for Environmental Sustainability - ACES) and a group of agent based social simulation modellers. Members of these groups have been encouraged to use *ourSpaces* and to contribute to its continued development by reflecting on how it has been used, and how they see that it might be developed to enable deeper research integration.

The VRE has been available on-line since September 2009. To date, there are 254 *foaf:profiles* defined in *ourSpaces* of which 183 are registered users. Non registered profiles have been created by the system while specifying authors of documents. The social network in the VRE is composed of 204 links (*foaf:knows*) between user accounts. Users created 49 projects and sub-projects and 92% of the accounts in *ourSpaces* are members of at least one project. Users have also uploaded 435 research artefacts. The *ourSpaces* metadata repository contains 14680 triples describing 4388 distinct entities.

To date, 63 distinct classes and 105 distinct properties are used to describe entities in the repository, utilising 33% of the classes and 40% of the properties defined in the supporting ontologies.

The two primary sources of data for our ongoing evaluation of the system are: a) the repository containing metadata about resources, people, events, projects, etc. and b) the mysql database containing user account information and system logs. We have also conducted interviews and focus groups with our case study groups in order to gather evidence on user perspective and feedback. In the remainder of this section we will use the outcome of these data gathering exercises in order to illustrate some of the lessons learnt during the deployment of the VRE.

4.1 Deployment Issues and Lessons Learnt

One of the most significant difficulties was the unwillingness of users to provide metadata about artefacts. In the early stages of *ourSpaces*, users were required to provide a great deal of metadata about research artefacts in order to guarantee a detailed metadata record. The result was that few users went through the effort of providing such metadata and only a small number of artefacts were uploaded. Following feedback from users we adopted a more relaxed approach, where very few mandatory fields were required and the users themselves had the option to choose which metadata to add to the artefact. This resulted in more artefacts being uploaded at the cost of producing a much sparser metadata record. To illustrate this issue we now present some summary data collected from our metadata repository. As illustrated in section 3.1, the user is required to select the type of artefact that he/she is uploading and mandatory fields are displayed in the form depending on the class selected. Types of artefact are shown on a tree-like structure, where *Artefact* is the root class and more specialised types are presented up to two levels down the class hierarchy. From our analysis, 20% of the artefacts in *ourSpaces* have been associated with the root class, 71% with subclasses of *Artefact* and 9% with classes at the next level in the hierarchy.

We aimed to solve the problem of sparse metadata by allowing people (with the right authority) to define policies in *ourSpaces* to specify the mandatory information required when uploading a research artefact. In this way, the request for additional information originated with a person rather than the system, e.g. the principal investigator of a project. After introducing policies into one of the projects in *ourSpaces*, the average number of RDF triples used to describe each research artefact increased from 9 to 32. However, policies and particularly the SPIN reasoner require additional computational resources, resulting in a delay when a user is using a web form. The time taken by SPIN to process each policy depends on the precise nature of that policy. Some, such as logging in, do not require much data to be evaluated. Others, such as uploading an artefact, require a large provenance graph to be evaluated.

We have analysed the logs from the policy reasoning service in order to determine the performance of the reasoner. The hardware used for the deployment of the *ourSpaces* services and repositories consists of three Sun Fire X4100 M2 with two dual-core AMD Opertron 2218 processors and 32 Gb of memory. Based on 2895 runs of the reasoner logged by the system the average time to run a policy was 1993ms. Miller [17] and Card [18] argue that system response times of less than ten seconds

do not compromise the user's attention on the current task. However for delays greater that one second, it is necessary to indicate to the user that the system is busy. Based on the result from the analysis of the logs we determined that time taken to reason about a policy was acceptable without compromising the overall performance of the system. However, we had to make use of the AJAX spinning wheel widget to inform the user that the system was performing a task. A similar analysis has also been conducted for the use of policies by the NLG service. In spite of the overhead associated with the use of the policy reasoner, text is generated and appears within 200ms.

Policy reasoning also presents a cost in terms of data storage. The inferences generated by policy reasoning could be stored in the RDF repository in order to provide additional provenance about user actions. However, it is not always necessary to do this in practice as recording very fine grained provenance may not always be useful. For example, policy reasoning is triggered every time the user makes a change in a field during the upload of an artefact but it is not necessary to record the provenance of each individual action. On the other hand, knowing that a user had twenty failed log-in attempts with an incorrect password could indicate a potential security issue and is something worth registering for later inspection.

Based on the feedback from our users we have discovered that the graphical interface for visualising metadata (section 3.2) served a useful function as a means to validate the metadata uploaded via the form based interface. This was especially useful as the UKDA documentation policy required them to provide detailed information about the methods used for generating a research artefact. The graphical interface was also used by representatives of the UKDA to review the data (and metadata) uploaded by project members. Screen-grabs of the provenance graphs generated by our system have been used by the UKDA as part of their internal documentation describing the project archive.

Maintaining the *ourSpaces* VRE and introduction of new user requirements often requires changes to the underlying data structures. From the beginning, *ourSpaces* has used both an SQL database and an RDF store, with the SQL database used to log user activities for monitoring and security purposes. When the policy reasoning facilities were subsequently introduced, it was necessary to change the representation of user activities so that the reasoner could infer policy activations. As a result, the data structure describing activities had to be changed from a relational database to RDF. This has allowed us to describe system activities in terms of artefacts, processes and agents and to include them as part of the wider provenance graph.

A crucial part of maintenance of the system is to take into account the requirements of new user groups. When a new group joins *ourSpaces*, it is normal to expect that they might have their own way to describe research artefacts and processes. The *ourSpaces* provenance framework can be easily extended in order to accommodate new domain-specific provenance concepts. This issue was detected very early during the development of *ourSpaces* and we therefore designed the system in such a way that new domain ontologies could be integrated into the system without the need to change the source code. Our implementation of the policy framework also allows new policies to be integrated into the system without the need for alterations to the underlying code.

Although we don't yet have a specific tool for designing policies, a standard ontology editor can be used to design the individual policy ontologies.

5 Conclusions

In this paper we have introduced the *ourSpaces* virtual research environment focusing on three elements which make use of Semantic Web technologies - a provenance representation framework, a collection of services for creating and visualising metadata and a policy reasoning service.

The ontological support in *ourSpaces* allows us to capture entities such as artefacts, people and processes and to include links between them. This 'linked data' makes certain aspects of information discovery and presentation possible within the *ourSpaces* environment. For example, the concept of a "space" (home, project, etc.) would not be effective unless it was possible to identify the entities linked to the resource featured in the space such as related resources, people involved, communication activities, etc. Linked data also allows components such as the natural language visualisation service to exploit this model to allow users to explore the provenance graph.

The policy and provenance framework provides a real benefit in terms of adaptability of the system. As was discussed in section 4, new policies and ontologies can be introduced by changing the configuration files without having to change the Java code. Moreover, the declarative nature of policies allows the introduction of new logic into the system even by users that are not familiar with the underlying VRE source code.

In future, we plan to implement a model space, which would enable users to run their own simulation experiments using standard simulation environments such as Repast[12] or Netlogo[13]. This space will enable users to explore the provenance associated with simulation models and to support the reproducibility of simulation experiments.

Future plans also include a personalised thesaurus service, which would accommodate the vocabulary differences of users and disciplines. This service could be used by the existing NLG and search services to deal with user or discipline specific terminology differences.

Acknowledgments

This work is supported by the UK Economic & Social Research Council (ESRC) under the Digital Social Research programme; award RES-149-25-1075.

References

1. Butterworth, A., Reimer, T.: Virtual research environment collaborative landscape study. Technical report, JISC (2010)
2. Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web. Scientific American **284**(5) (2001) 34–43

[12] http://repast.sourceforge.net/
[13] http://ccl.northwestern.edu/netlogo/

3. Heinis, T., Alonso, G.: Efficient Lineage Tracking for Scientific Workflows. In: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data (SIGMOD'08), ACM (2008) 1007–1018
4. Pignotti, E., Edwards, P., Reid, R.: A multi-faceted provenance solution for science on the web. In: 4th International Provenance and Annotation Workshop, Troy, USA (2010)
5. Pignotti, E., Edwards, P.: Using web services and policies within a social platform to support collaborative research. In: Working Notes of AAAI 2012 Stanford Spring Symposium on Intelligent Web Services Meet Social Computing. (March 2012)
6. Roure, D.D., Goble, C., Stevens, R.: The design and realisation of the virtual research environment for social sharing of workflows. Future Generation Computer Systems **25**(5) (2009) 561 – 567
7. Aastrand, G., Celebi, R., Sauermann, L.: Using linked open data to bootstrap corporate knowledge management in the organik project. In: Proceedings of the 6th International Conference on Semantic Systems. I-SEMANTICS '10, New York, NY, USA, ACM (2010) 18:1–18:8
8. Kreiser, A., Naurex, A., Bakalov, F.: A web 3.0 approach for improving tagging systems. In: Proceedings of the International Workshop on Web 3.0: Merging Semantic Web and Social Web (in conjunction with the 20th International Conference on Hypertext and Hypermedia 2009). Volume 467., Torino, Italy (June 2009)
9. Bizer, C., Heath, T., Berners-Lee, T.: Linked data - the story so far. International Journal on Semantic Web and Information Systems (IJSWIS) (2009)
10. Moreau, L., Freire, J., Futrelle, J., McGrath, R.E., Myers, J., Paulson, P.: The open provenance model: An overview. In Freire, J., Koop, D., Moreau, L., eds.: IPAW. Volume 5272 of Lecture Notes in Computer Science., Springer (2008) 323–326
11. Sensoy, M., Norman, T.J., Vasconcelos, W., Sycara, K.: Owl-polar: Semantic policies for agent reasoning. In: International Semantic Web Conference. (2010)
12. Smith, D.A., Popov, I., mc schraefel: Data picking linked data: Enabling users to create faceted browsers. In: Web Science Conference 2010. (March 2010) Event Dates: 26-27.
13. Hielkema, F.: Using Natural Language Generation to Provide Access to Semantic Metadata. PhD thesis, University of Aberdeen (2010)
14. Gatt, A., Reiter, E.: Simplenlg: a realisation engine for practical applications. In: Proceedings of the 12th European Workshop on Natural Language Generation. ENLG '09, Stroudsburg, PA, USA, Association for Computational Linguistics (2009) 90–93
15. Bouttaz, T., Pignotti, E., Mellish, C., Edwards, P.: A policy-based approach to context dependent natural language generation. In: Proceedings of the 13th European Workshop on Natural Language Generation, Nancy, France, Association for Computational Linguistics (September 2011) 151–157
16. Reiter, E., Dale, R.: Building natural language generation systems. Cambridge University Press, New York, NY, USA (2000)
17. Miller, R.B.: Response time in man-computer conversational transactions. In: Proceedings of the joint computer conference 1968, New York, NY, USA, ACM (1968) 267–277
18. Card, S.K., Robertson, G.G., Mackinlay, J.D.: The information visualizer, an information workspace. In: Proceedings of the SIGCHI conference on Human factors in computing systems: Reaching through technology. CHI '91, New York, NY, USA, ACM (1991) 181–186