The International Journal of Digital Curation
Issue 2, Volume 6 | 2011

DataStaR: Using the Semantic Web approach for Data Curation

Huda Khan, Brian Caruso, Jon Corson-Rikert, Dianne Dietrich, Brian Lowe,
& Gail Steinhart,
Albert R. Mann Library,
Cornell University

Abstract

In disciplines as varied as medicine, social sciences, and economics, data and their analyses are essential parts of researchers' contributions to their respective fields. While sharing research data for review and analysis presents new opportunities for furthering research, capturing these data in digital forms and providing the digital infrastructure for sharing data and metadata pose several challenges. This paper reviews the motivations behind and design of the Data Staging Repository (DataStaR) platform that targets specific portions of the research data curation lifecycle (Higgins, 2008): data and metadata capture and sharing prior to publication, and publication to permanent archival repositories. The goal of DataStaR is to support both the sharing and publishing of data while at the same time enabling metadata creation without imposing additional overheads for researchers and librarians (Steinhart, 2010). Furthermore, DataStaR is intended to provide cross-disciplinary support by being able to integrate different domain-specific metadata schemas according to researchers' needs. DataStaR's strategy of a usable interface coupled with metadata flexibility allows for a more scaleable solution for data sharing, publication, and metadata reuse.[1]

[1] This paper is based on the paper given by the authors at the 6th International Digital Curation Conference, December 2010; received December 2010, published July 2011.
The International Journal of Digital Curation is an international journal committed to scholarly excellence and dedicated to the advancement of digital curation across a wide range of sectors. ISSN: 1746-8256 The IJDC is published by UKOLN at the University of Bath and is a publication of the Digital Curation Centre

Overview

Researchers rely on data as scientific evidence of their claims and as the basis for the knowledge that they generate (Arms, 2008). Descriptive metadata allow researchers to define the context needed for future data analysis and further review by themselves and other researchers, and thus adequate metadata are needed for effective data discovery, analysis, and reuse. At the same time, the process of metadata creation can require researchers to learn a particular metadata schema or to use specialized tools. Researchers may perceive metadata creation to be too time-consuming and tangential to the overall process of their research and may not learn and use a particular metadata schema unless metadata use is critical or necessary for research (Pritchard, Anand & Carver, 2005). Pritchard et al. (2005) suggest the use of metadata-agnostic repositories and interfaces that automate metadata creation as a means to support metadata use.

Librarians with metadata and/or subject area expertise are in a good position to assist researchers with metadata creation, but, as Steinhart and Lowe found in their efforts to support research data curation at Cornell University's Albert R. Mann Library, tasking librarians with metadata creation without appropriate tools is not a sustainable approach. Prior to developing the Data Staging Repository (DataStaR), Mann Library was engaged in several data curation initiatives, working with faculty and research teams to prepare, describe, and archive scientific datasets (Steinhart & Lowe, 2007). One such initiative involved working with a research group that was studying nutrient and sediment cycling in the Upper Susquehanna River basin. The members of this research group were from multiple institutions and expressed an interest in sharing documents and data within the group prior to publication (Steinhart & Lowe, 2007) as well as sharing their results publicly (Steinhart, 2010). In the process of supporting and training the group to document and publish their datasets using domain-specific metadata, Steinhart and Lowe realized that the strategy of shifting the bulk of metadata creation to librarians does not scale well with an increasing number of researchers and research groups. In order for more researchers to be able to create metadata without placing unsustainable demands on library staff time, researchers need tools that enable them to do most or all of data documentation themselves with occasional assistance from librarians as needed (Steinhart & Lowe 2007).

Ann Green and Myron Gutmann's (2007) description of the possibilities for partnerships between institutional and domain repositories further helped crystallize the need for a local, institutionally-based staging repository which enables domain-specific metadata definition before and up to publication (Steinhart, 2010). DataStaR seeks to provide such a service, scaffolding the process of eventual data publication to both institutional and domain-specific repositories (Dietrich, in press). DataStaR, as a staging repository, is not intended to serve as a permanent repository and thus does not itself need to conduct preservation planning as per Higgins' digital curation lifecycle model (Higgins, 2008), but the system does address curation of data at different stages of the research process and is designed to support best practices for preservation (Steinhart, Dietrich, & Green, 2009).

DataStaR and the Semantic Web

Semantic Web technologies aim to define and interconnect data in a manner similar to that of traditional web technologies which define and interconnect pages of the World Wide Web (web pages). In the case of the traditional web, each web page can be considered a unit of information or entity, and pages are explicitly linked using html links. The Semantic Web also allows data to be shared using *linked data* support where entities can be referenced and their information can be accessed on the web as part of a linked network of data.[2,3] Entities are identified using Unique Resource Identifier (URIs), similar to URLs, and are described using Resource Description Framework (RDF) statements.[4] These statements describe entities using "`<subject> <predicate> <object>`" triples where "subject" is the entity, "predicate" refers to a property or relationship for the entity, and "object" can be either a literal value such as text or another URI referencing another entity . Semantic web applications can thus retrieve and integrate this web of statements describing a given entity.

DataStaR's use of semantic web technologies attempts to support more efficient creation of metadata by treating the metadata associated with a particular dataset as a collection of statements about that dataset, rather than a single, static document. This approach enables the reuse of statements for other datasets, potentially decreasing the effort involved in creating metadata, particularly as a researcher's "collection" of metadata statements in DataStaR grows.

This semantic web approach also enables DataStaR to support metadata creation across multiple discipline-specific metadata schemas. Different metadata schemas are integrated into DataStaR as needed by being converted into semantic ontologies using RDF statements and Web Ontology Language (OWL) classes.[5] DataStaR can thus be extended to describe datasets from various disciplines. In addition, DataStaR's use of the semantic web approach enables the reuse of metadata across different metadata schemas through the inclusion of mappings between ontology elements. For example, when DataStaR defines the mapping between Ecological Metadata Langauge (EML) and the Federal Geographic Data Committee's Content Standard for Digital Geospatial Metadata (FGDC-CSDGM) geographic coverage statements, information entered by the user for a geographic coverage element for an EML dataset can be reused for FGDC-CSDGM statements describing the dataset.[6,7] Researchers can thus use DataStaR to create, share, and publish datasets described by different schemas as required. Furthermore, DataStaR can describe a single dataset using multiple metadata schemas when needed.

[2] Linked data: http://www.linkeddata.org.
[3] Linked data (review of design issues, by Tim Berners Lee): http://www.w3.org/DesignIssues/LinkedData.html.
[4] RDF: http://www.w3.org/RDF/.
[5] OWL: http://www.w3.org/TR/owl-features/.
[6] EML: http://knb.ecoinformatics.org/software/eml/.
[7] FGDC-CSDGM: http://www.fgdc.gov/metadata/geospatial-metadata-standards#csdgm.

DataStaR Application: Architecture and Metadata Creation

Figure 1 provides an overview of DataStaR architecture. DataStaR extends the Vitro software developed by Mann Library at Cornell University and that "combines a Web-based ontology and instance editor with a public display interface" (Lowe, 2009).[8] Vitro is best known as the software underlying the VIVO research networking tool, also developed at Cornell and now expanding to a number of other universities under the sponsorship of the National Institutes of Health.[9]

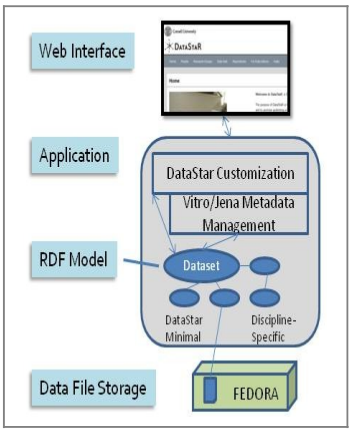

Figure 1. DataStaR architecture overview. Jena: A Semantic Web Framework for Java.[10]

DataStaR customizes Vitro to define and specify the relationships between datasets, individuals, and organizations. OWL ontologies are used to define the types of entities and what properties or predicates can be used to describe these entities. A dataset's metadata input forms are generated based on the associated ontologies. Files uploaded to a dataset are stored in the Flexible Extensible Digital Object Repository Architecture or Fedora repository.[11] DataStaR generates RDF statements to define this file as an entity with a URI and to store file-specific information such as size, content type, checksum, and the unique, persistent Fedora identifier (PID) for the file.

Consider a scenario where a hypothetical environmental scientist named Sara creates a dataset using DataStaR. After logging into her account, Sara selects the option to create a new dataset and indicates that her intended submission repository is the Knowledge Network for Biocomplexity (KNB) which requires EML.[12] As Figure 2 below shows, the initial dataset creation page requires only a few metadata fields, such as title, destination repository, and owner, to be specified by the user, while the remaining fields such as dataset originator are automatically generated. This core set of DataStaR metadata fields is common to all datasets.

[8] Vitro: http://vitro.mannlib.cornell.edu.
[9] VIVO: http://vivoweb.org.
[10] Jena: http://jena.sourceforge.net/index.html.
[11] Fedora: http://fedora-commons.org/.
[12] KNB: http://knb.ecoinformatics.org/.

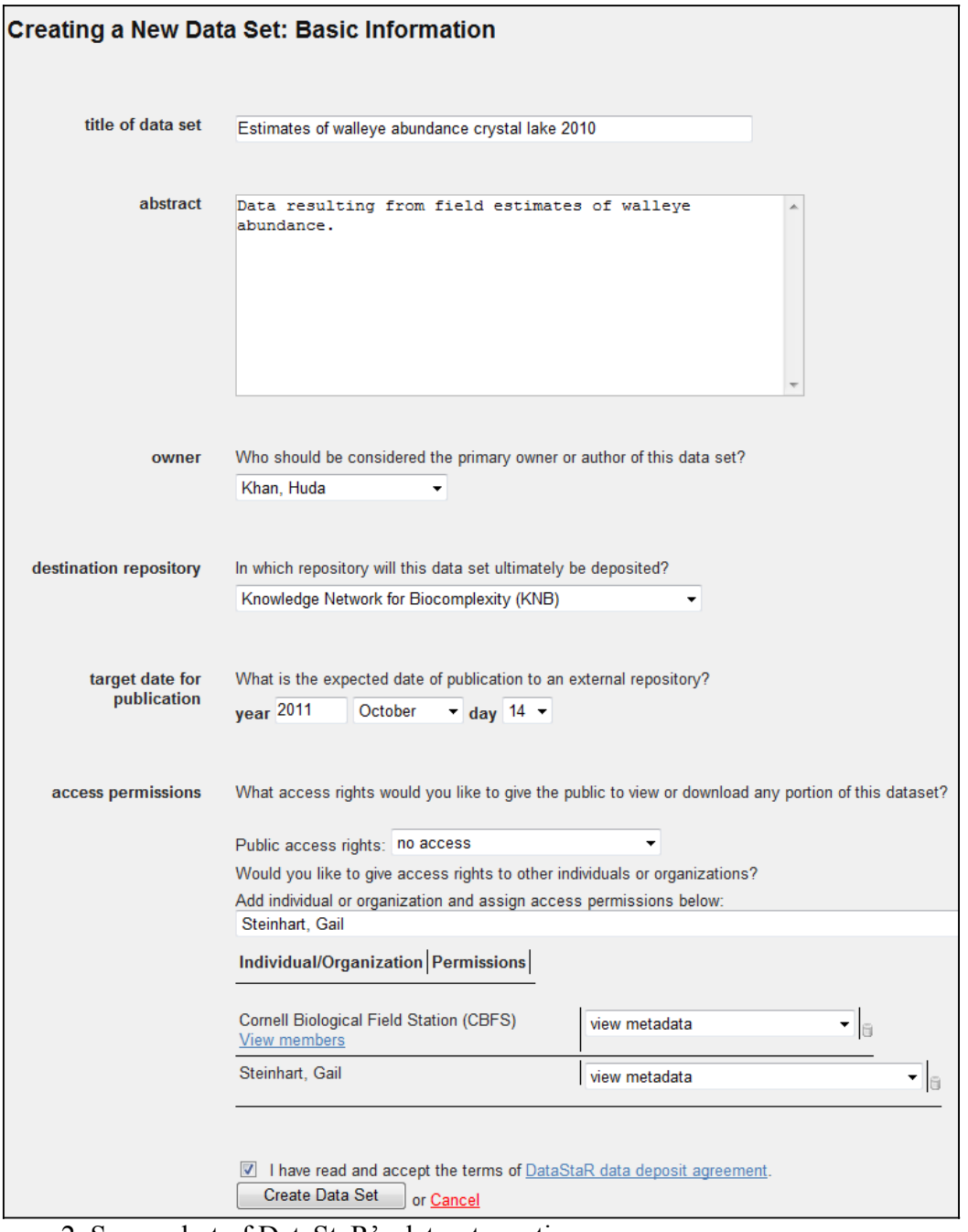

Figure 2. Screenshot of DataStaR's dataset creation page.

Sara could have selected "to be determined" as the destination repository if she was unsure of her publication plans or if she is only using as DataStaR as a means to share data with authorized colleagues. Intent to publish is not a requirement for researchers to use DataStaR. If no expected publication date is indicated at the time of dataset creation, a date one year in the future is included by default. When this date is reached and if the dataset has not yet been published, the DataStaR staff may contact the owner or originator of the dataset to request an update on the status of the dataset. Sara can also define access and modification permissions for different individuals and research groups.

Within the RDF model, the dataset is now defined as an entity with a URI with related RDF statements. Figure 3 below provides a simplified sample of these statements as an RDF graph. `<dataset>` designates the dataset URI and `<Sara>` indicates the owner URI in DataStaR. The "rdf" and "dsr" prefixes designate the RDF and DataStaR-specific namespaces respectively.

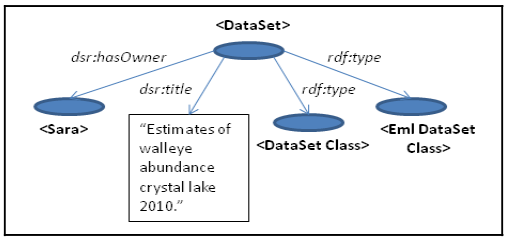

Figure 3. RDF graph representation of statements describing an EML dataset.

Once Sara has completed and submitted this form successfully through the interface, she can now view and edit the fields. Because Sara indicated the KNB repository as the destination repository, the system generated a statement defining the dataset as having an "EML dataset" type in addition to the regular dataset type. The EML type triggers the dataset view form to include fields and properties that are from the EML ontology. For example, Sara can add geographic coverage information which maps to the EML geographic coverage elements.

Figure 4 shows a high-level overview of how these EML statements integrate with the minimal and EML-specific ontologies in DataStaR. The statements shown in the figure are not RDF but simplified versions that show the kinds of information encoded into RDF statements, both for statements generated when the dataset is created and edited, and the ontologies for the core DataStaR and integrated EML schemas.

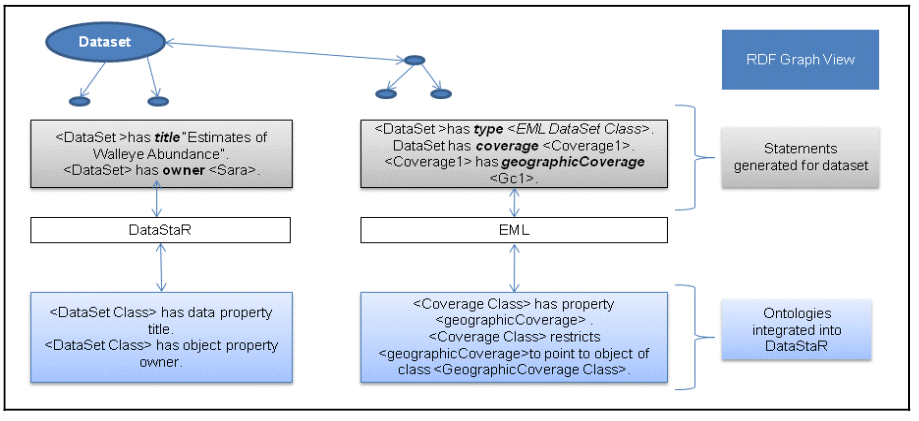

Figure 4. An overview of how an EML dataset in DataStaR has both DataStaR core ontology statements as well as EML statements.

Sara continues to edit and share these datasets with colleages or research groups. When her colleagues download the dataset, DataStaR returns a zipped file containing the files uploaded as well as separate XML files corresponding to the different schemas with which the dataset was associated. In this case, they would receive the uploaded data files, a metadata XML file corresponding to minimal DataStaR metadata, and an EML file mapping to the EML statements for the dataset. DataStaR creates the EML record using the Gloze application's transformation of the dataset's RDF statements to XML (Battle, 2006). DataStaR may provide additional changes to the output from Gloze's transformation for better alignment between the resulting XML and EML specifications. DataStaR uses a similar process to create an EML record when Sara wishes to publish the dataset to the KNB repository.

Challenges and Questions

Named Graphs: Information Integrity and Controlled Access

One of the appeals of semantic web technologies lies in the potential for linking and integrating data from multiple sources and then being able to query and retrieve information across these different sources. In spite of the desirability of linking data in this manner, a concern that arose during the development of DataStaR was how to maintain information integrity through controlled access while still supporting metadata reuse. If all information in the system is available to all users, it is possible for an individual to edit an entity created and used by someone else (the originator) in such a way that introduces changes or errors into the description of one or more of the originator's datasets. An example we have already encountered has to do with changing roles of research participants. A researcher may be described accurately as the director of a research facility at the time a dataset is created, but may later retire, with another individual being promoted to that role. The information in the system is changed to reflect the changes in roles, but it is not necessarily appropriate to change that information for a dataset created earlier. We realized it would be necessary to stabilize information about a particular dataset to avoid propagating later changes unintentionally.

At the same time, a researcher may wish to give different levels of access to different individuals for the same dataset. For example, our example scientist Sara may wish to restrict her dataset's public visibility but share her dataset with a group of colleagues. She may wish to allow a researcher working on the same project to be able to modify the metadata and she may decide at a certain point in the future that she would like to make the dataset visible to the public.

In order to address these scenarios, DataStaR employs *private named graphs* which are a collection of statements referenceable by a URI. A given dataset's information is stored in an associated named private graph. Certain information, such as the title or the graph URI itself, is stored in the *public* layer of RDF statements while the remaining set of statements for that dataset are included in that dataset's named graph. Every user can see the publicly accessible RDF statements, but access to the named graphs is based on whether or not the user has explicit permissions to view a particular dataset.

Consider again the dataset created by Sara which actually consists of two sets of statements, one set which consists of basic identifying information and another set which is comprised of all other information stored within a named private graph. When Sara created this dataset, she specified additional users or groups who could have access to the data or metadata. In accordance with this information, DataStaR created RDF statements defining permissions related to this dataset and automatically gave full permissions to the owner Sara. When Sara herself logs in, DataStaR checks for which datasets she has permissions and then adds the corresponding private graphs to the main or public graph which is visible to Sara.

If Sara then sees, for example, a set of geographic coordinates (perhaps for a common sampling location) in another dataset which she would like to reuse in her dataset, she can select it from the list of previously defined coordinates. These coordinates are then copied over into the private graph for the dataset she is editing. This copying process also occurs when a dataset is first created. The system searches for the object references that are used to describe the dataset and then copies information about these objects into the dataset's private graph. For example, our example dataset's owner is defined using a statement which declares that the dataset has the owner Sara (where Sara is identified by a URI). The system searches for additional statements in the public model describing the URI representing Sara, such as statements describing the label or name associated with the URI, and then copies those statements to the dataset's private graph. This copying process allows for the user to see the owner name when they are editing the private graph, whereas without the copying process they would only see the owner URI.

The use of private graphs, though helping to resolve the issue of maintaining information integrity, raises additional questions. When information is copied into the private graph from the public layer or from another dataset, that information is a snapshot of the content available at the time of the copy. The question then becomes when information, and what portions of the information, should be synchronized with the public layer, and under what circumstances? For example, dataset B may be related to dataset A, and dataset B's private graph would contain the copy of dataset A's title when this relationship was created by the user. If dataset A's title changes at some point prior to dataset A's publication, dataset B would still display the old title by virtue of the information stored in dataset B's private graph. This case suggests the need to include a synchronization feature which would allow certain properties to be updated with the information that is present in the public model, if desired, prior to publication or export to another repository.

Metadata: XML to RDF

In most data repositories, metadata are stored using XML files based on XML schemas which may allow complex, nested, and ordered elements. In order to be able to integrate different metadata schemas into DataStaR, the development team had to consider how to translate the XML Schema Document underlying a given XML metadata record into an OWL ontology and how to transform the XML record into a dataset described using RDF statements. In addition, the system then should be able to take the resulting dataset and transform it back into an XML file consistent with the publication repository's metadata requirements.

The Gloze application can help to convert a metadata specification's XML Schema Document into a set of OWL classes of objects and corresponding predicates (Battle, 2006). In the case of very complex metadata schemas, we may include selective portions of the schema, for example only those elements available in commonly used metadata creation or editing tools for that schema, or the most commonly used elements of a particular schema. The ontology resulting from Gloze can be refined or extended as needed. The DataStaR team has explored the integration of EML as well as the custom schema employed by the Virtual Center for Language Acquisition to store metadata for a linguistic study.[13] The integration of additional metadata schemas has exposed certain challenges in the conversion of XML to equivalent RDF statements and in the displaying of these statements in a way which makes sense to those editing the statements through the DataStaR interface. These challenges include (a) converting implicitly ordered XML elements in a parent element; (b) generating an interface for XML schema restrictions involving a "choice" element, where only one element out of a set of options should be included in a parent element.

Nested repeatable XML elements and implicit order.

In some cases, XML files have an implicit ordering that then needs to be correctly captured in the RDF statements. For example, an EML record can contain multiple method steps nested in the methods element. Although there is no explicit order number given to these elements, the elements are listed in a specific order. When configured to order these nested elements, Gloze generates an RDF sequence element which describes the order of nested elements using predicates such as "rdf:_1" and "rdf:_2". In order to be able to use these predicates, DataStaR's ontology would have to create a separate "rdf:_x" predicate, where "x" corresponds to a number, for an entire range of numbers, that is, a separate property for rdf:_1, rdf:_2, rdf:_3, and so on. This solution would either result in a very large number of "rdf:_" properties or the need to add a new "rdf:_" property every time a new order number was needed. DataStaR adopted a different solution, indicating order by specifying a set of intermediate entities that link the parent object to the child object while providing ordering information. As part of integrating EML into DataStaR, special "ordering" objects and properties were defined in the DataStaR ontology. These properties can be extended based on the type of objects being ordered. Figure 5 shows the mapping from the XML method steps to the generic RDF ordering relationship as well as the extended relationship "orderedMethodStep".

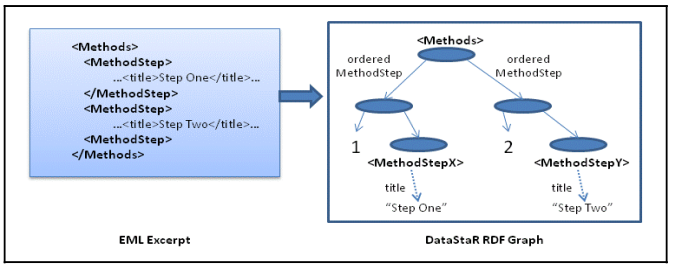

Figure 5. The EML excerpt for methods translates into RDF statements employing an intermediate ordering context. Ellipses in the EML excerpt and dotted arrows in the RDF graph representation indicate additional child elements or hierarchy of RDF statements respectively.

[13] Virtual Center for Language Acquisition: http://vcla.clal.cornell.edu/.

The DataStaR interface must then recognize these ordering constructs in addition to the base ontology for the schema. We modified the dataset view page to order content with respect to the order values for these intermediate objects, and updated the metadata field editing page to allow for the addition of new elements while, in the back end, creating new intermediate objects to define their order as the last in the sequence. For example, when Sara edits the "methods" field for an EML dataset and adds a new method step when two method steps already exist, the new method step will be interpreted and displayed as third in the sequence of method steps. We expect to keep updating the interface to allow for a more seamless way of ordering these elements on the same page without having to submit or refresh the page itself.

XML choice: Which options to display?

XML schemas use the choice element to specify that an element can contain one and only one of multiple kinds of nested elements. As an example, consider that in an EML record, a "TextType" element, which is used to contain text, may consist of either a "section" element or a "para" element (short for paragraph). OWL, while capable of expressing the minimum or maximum number of section and para elements allowed, does not have a direct equivalent to XML's choice element. If Sara, our example scientist, were to use DataStaR using this ontology, she would see that, where a TextType entity is included such as in the case of a MethodStep, she can edit two text areas, one entitled "section" and one entitled "para". The interface would not indicate that she only needs to fill out one input. Currently, DataStaR reviews these situations on a case by case basis, updating the integrated ontology to include choices that are consistent with EML but that don't include more options on the interface than necessary. For example, in the case of method steps for an EML method field, we restricted the ontology to include only a section as being part of a method step, allowing the interface to display just the inputs for section. Future work will explore an ontology-based approach to resolve this issue, such as the use of additional annotation properties to describe which field out of different choices should be selected given which element is being edited.

Current Status and Ongoing Work

The first production version of DataStaR is intended to be ready for use in early 2011. We have developed several partnerships with research teams that intend to use DataStaR to store and publish datasets and that include: Agriculture, Energy and the Environment Program, Cayuga Lake Watershed Network, Cornell Biological Field Station, Cornell Plantations Natural Areas Program, the Loon Project, the Virtual Center for Language Acquisition, and the Data Conservancy project (Steinhart, 2010). Table 1 below shows the publication repositories and their metadata specifications that DataStaR will support.

Repository	Metadata Specification
Cornell University Geospatial Information Repository (CUGIR)[14]	Federal Geographic Data Committee Content Standard for Digital Geospatial Metadata (FGDC-CSDGM)
eCommons@cornell[15]	Modified Dublin Core[16]
Knowledge Network for BioComplexity (KNB)	Ecological Metadata Language (EML)
Data Conservancy[17]	TBD

Table 1. Publication repositories for datasets in DataStaR and corresponding metadata specifications.TBD: To be determined.

The DataStaR development team has identified several important interface usability issues such as the ability to add multiple elements on a page without having to refresh or reload the page. For example, the user should be able to add multiple method steps and order them for an EML methods section and they should be able to add multiple keywords while editing the keyword set for a dataset. The development team is exploring future opportunities for conducting usability testing on the interface employing faculty or graduate students that are representative of the researchers who would use DataStaR. Furthermore, we will also need to explore how researchers from different domains may have different requirements or workflows and what subset of these requirements we will be able to support using a single system.

Another area for ongoing development is supporting the different workflows for different repositories. In addition to the specific metadata standards mentioned in Table 1, DataStaR will integrate support for dataset publication to repositories, such as the Data Conservancy, which support or plan to support the Simple Web-service Offering Repository Deposit protocol.[18] For some repositories such as KNB, the Data Conservancy, or eCommons, a direct submission from DataStaR on publication may be possible. For other repositories with unique architecture or submission procedures, DataStaR may have to create a submission package that would then need to be submitted manually. In addition, the current interface only allows end users to select a single publication repository and, in the case of KNB, generate EML fields in the resulting dataset. DataStaR will also need to support cases allowing for more flexibility, for example if a person wishes to submit a dataset to eCommons, a repository that requires modified Dublin Core metadata, along with a discipline-specific metadata record (as a supplementary document).

As work proceeds with DataStaR, we have seen an increased interest in the application. Some researchers with whom we have worked previously intend to use DataStaR as part of their data dissemination plans. Other institutions and projects are exploring the use and adaptation of Vitro, a core component of DataStaR, alone or in combination with the VIVO research networking ontology. One such project is the Australian National Data Service which is developing a national data registry and has funded enhancements to Vitro as a metadata acquisition and submission tool at several participating Australian universities including Queensland University of Technology,

[14] Cornell University Geospatial Information Repository: http://cugir.mannlib.cornell.edu.
[15] eCommons@Cornell: http://eCommons.cornell.edu.
[16] Dublin Core Metadata Initiative: http://dublincore.org/.
[17] The Data Conservancy: http://dataconservancy.org/.
[18] Simple Web-service Offering Repository Deposit: http://swordapp.org.

Griffith, and the University of Melbourne.[19] We continue to explore the integration of additional metadata standards and the improvements to the design of the interface to support researchers in their metadata creation for research data.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. III-0712989.[20] Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation.

References

Arms, W.Y. (2008). Cyberscholarship: High performance computing meets digital libraries. *The Journal of Electronic Publishing 11*(1). Retrieved August 6, 2010, from http://quod.lib.umich.edu/cgi/t/text/text-idx?c=jep;view=text;rgn=main;idno=3336451.0011.103.

Battle, S. (2006). Gloze: XML to RDF and back again. In *Proceedings of First Jena User Conference, Bristol, UK.* Retrieved May 30, 2011, from http://web.me.com/stevebattle/CurriculumVitae/Publications_files/GlozeJUC.pdf.

Dietrich, D. (in press). Metadata management in a data staging repository. *Journal of Library Metadata*.

Green, A. G., & Gutmann, M. P. (2007). Building partnerships among social science researchers, institution-based repositories and domain specific data archives. *OCLC Systems & Services, 23*(1). Retrieved June 5, 2011, from http://deepblue.lib.umich.edu/bitstream/2027.42/41214/1/dig-repositories-green-gutmann-072006.pdf.

Higgins, S. (2008). The DCC Curation Lifecycle Model. *International Journal of Digital Curation, 3*(1), 134-140. Retrieved August 6, 2010 from http://www.ijdc.net/index.php/ijdc/article/download/69/48.

Lowe, B. (2009). DataStaR: Bridging XML and OWL in Science Metadata Management. In *Metadata and Semantic Research Third International Conference, MTSR 2009. Milan, Italy*. Berlin: Springer.

Pritchard, SM., Anand, S. & Carver, L. (2005). *Informatics and knowledge management for faculty research data.* Retrieved August 6, 2010 from http://net.educause.edu/ir/library/pdf/ERB0502.pdf.

Steinhart, G. & Lowe, B. (2007). Data curation and distribution in support of Cornell University's upper Susquehanna agricultural ecology program. In *DigCCurr2007, an International Symposium on Digital Curation. Chapel Hill, NC*. Retrieved 05 June, 2011, from http://ecommons.cornell.edu/bitstream/1813/7517/1/Steinhart_and_Lowe2007.pdf.

[19] Australian National Data Service: http://www.ands.org.au/funded/eif-fast-start.html.
[20] National Science Foundation: http://www.nsf.gov/.

Steinhart, G., Dietrich, D. & Green, A. (2009). Establishing trust in a chain of preservation: The TRAC checklist applied to a data staging repository (DataStaR). *D-Lib Magazine Magazine, 15*(9/10). Retrieved June 5, 2011, from http://www.dlib.org/dlib/september09/steinhart/09steinhart.html.

Steinhart, G. (2010). DataStaR: a data staging repository to support the sharing and publication of research data. In *International Association of Scientific and Technological University Libraries, 31st Annual Conference. West Lafayette, IN*. Retrieved June 5, 2011, from http://docs.lib.purdue.edu/cgi/viewcontent.cgi?article=1012&context=iatul2010.