Provenance and XSLT

Technical Report for DCEP-H

10-1-2012

Ashley M. Clark, Research Assistant
Center for Research in Science and Scholarship
Graduate School of Library and Information Science
University of Illinois at Urbana-Champaign

---

1. Abstract

Today much information is transformed with XSLT, but there are no methods of documenting the provenance of the processed data, though there are a variety of methods to piece together such a history. This paper describes a "meta-stylesheet approach with the potential to generate provenance by transforming an ordinary stylesheet into a documenting stylesheet, which in turn produces a queryable set of RDF statements. Based on the Open Provenance Model, the RDF record describes the transformations which produced output data.

2. Introduction

When researchers publish their findings, they demonstrate the authenticity of their work, crediting others whose work impacted them. By writing bibliographies and providing samples of their data in the form of figures, these researchers show the basis upon which their work was founded. In science, by providing the methods which led to the researcher's findings, one might imagine that the description could be used as a reference for someone else to reproduce the experiment, giving new results which might complicate or improve the understanding of science (Chao-fan et al., 2010). In the humanities, reproducibility of results is not considered nearly as important, since every research paper could be considered a personal mixing of different sources and methods. However, the humanities do follow stringent rules regarding citation of others' work, acknowledgement of the works which came before and influenced the project at hand.

Traditionally, provenance documentation provides evidence that an object is what it is, as well as proving the history of the object. In the art and museum world, provenance of an artifact is historical documentation tracing the object's ownership and where it lived in the world (Sweeney, 2008). In a similar fashion, data provenance traces the changes to data as it is gathered and transformed for researcher analysis. Because much data is often digitally processed, computers can generate provenance documentation as the processes are happening; allowing researchers to easily piece together provenance information. Recent demand in the e-sciences has shown the value of sharing one's workflow, perhaps even making it actionable. In this way, processes can be refined, reused, and repurposed (Deelman et al., 2010). Unfortunately, while the e-sciences have

seen a growing awareness of data provenance and its benefits, the e-humanities have not yet taken up digital provenance (Svensson, 2009).

And yet an increasing number of documents are becoming data for humanists' use. XSLT is one way to transform these documents to make them more amenable to digital humanists' use. To my knowledge, little work has been done on provenance in the digital humanities, much less to compile the provenance of XSLTs. However, Zhao and Tennison created a module of the Open Provenance Model Vocabulary (OPMV) which can be used to mark up the transformations of XSLTs (2010). My own work makes use of theirs, with one difference - I have attempted to collect the provenance of XSL transformations automatically. Collecting provenance is a time-intensive process, and there is no reason that a machine-ruled process like an XSL transformation cannot capture provenance. A solution to generating the provenance of XSLTs should be automatic, to save on researchers' time.

The objectives of this project were:

- to explore the provenance information already available in e-research workflow software, and for XSLTs;
- to determine what kinds of provenance information could be automatically drawn out;
- to implement provenance generation for XSLTs;
- and to compare the provenance capabilities of XSLTs to the work that could be done manually.

3. Literature review

In digital humanities, documents and other digital objects often are data, and are sometimes even born as such, but digital humanities tools and software have largely ignored data provenance. Part of this may be due to a difference in terminology - humanists do not usually think of themselves as working with data, and this perception likely persists even when humanists are working with large collections of information processed into meaningful chunks by computers. As Borgman (2011) notes, "Data may exist only in the eye of the beholder: the recognition that an observation, artifact, or record constitutes data is itself a scholarly act."

Still, like scientists, humanists are taught to record a kind of provenance - to cite their sources, to give credit to the minds which influenced them, to acknowledge others' intellectual property. The citations cannot prove the reproducibility of the results, or map the process that led to the publication of a paper. But, just by including references and citations, humanists provide a pedigree for their own intellectual property, allowing a reader to infer the provenance of their work. The problem comes when humanists begin work with digital objects - data - and use tools to manipulate that data (Svensson, 2009). As Rockwell (2010) argues,

> *Tools are not used to extract meaning according to objective principles. In the humanities we reinvent ways of making meaning within traditions. We are in the maintenance by reinvention and reinterpretation business and we don't want our methods and tools to become invisible as they are part of the research.*

Besides the necessity of citing their software tools, humanists must also realize that, unlike the mental workflows involved in writing a paper, digital processes can be documented by computers, their provenance obtained automatically.

To capture provenance is to tell a story of the changes wrought to make refined data what it is today; to capture information about the events, people, and other digital 'objects' related to the data (Sahoo, 2010). Data provenance is often optimally built for reproducibility of results; however, it can also be used as a way to share and reuse data processing workflows. In e-science, arguments for provenance often focus heavily on reproducibility - the ability for an independent researcher to perform the experiment again and to obtain data to the original's conclusions (Chao-fan et al., 2010). In 2009, De Roure discussed what he called the "six Rs of the e-Research record": replayability, repeatability, reproducibility, reusability, repurposability, and reliability. Küster et al. (2011) responded to these qualities from an e-humanities standpoint, arguing that reusability and repurposability are not needed in the humanities. From this point of view, the research - and thus its workflow - is specific enough that it will not be of use to other researchers. Küster et al. do see reproducibility as useful, so that the finalized data can be reused.

However, Borgman (2011) argues that reproducibility in e-science, while much-coveted, is a "problematic" goal. Even in cases where data can be obtained again with some expectation of an duplicate result, too much documentation would be needed in order to account for significant variables. For digital experiments or data cleaning, software may be copyrighted, with opaque processes, or versioned, so that the same process with updated software might yield a different result. Further, in the humanities, the argument can be made that few researchers are really interested enough in proving someone else's results when they could be arguing their own. Instead of reproducibility, Goble (2012) touts reusability as a better way of selling provenance to e-scientists - by documenting and modularizing one's digital workflow, one might share the provenance of their own data and prompt others who want to make use of the same techniques to do so.

Provenance includes documentation about artifacts (data, files), agents (people, organizations, software), and processes (file copying, an XSL transformation, the addition of a clarifying sentence). These entities are connected by relationships. For example, an agent might catalyze a process, which in turn uses an input artifact and creates an output artifact. All of this information together forms a comprehensive history for data. Of course, it is not enough just to name the entities involved. Enough information should be gathered in order to protect IP rights (the organization which created a software tool), to identify the correct entity ("Bob Everyman" of UIUC versus just "Bob"; data version 2 as opposed to 3), to promote good curation (the MIME type of a file; the version of software used to run a process; dependencies which may cause the workflow to fail if reproduced), and "to explain the status quo of data" (Chao-fan et al., 2010).

Provenance can be constructed by hand, perhaps using one of the handful of provenance ontologies to structure the documentation. However, this can be laborious, as providing details about events and people can require lots of painstaking knowledge, often pieced together from many sources. A much better solution, when dealing with digital workflows, is to build automatic provenance capture into those systems. This way, acquisition doesn't require extra effort by a human user, and the documentation is there when needed.

a) Provenance ontologies

There have been a handful of attempts at standardizing provenance documentation in the form of metadata. Along with ontologies built specifically for provenance (such as Provenir and the Provenance Vocabulary), other established schemas (such as Dublin Core and OAI-PMH) have also included capabilities for provenance mark-up, generally to explain how the metadata record was generated, when information was documented, or how it was obtained (Sahoo et al., 2010). Of these standardization efforts, W3C PROV and the Open Provenance Model (OPM) are arguably the most notable.

The Open Provenance Model was created in 2010 to document provenance for scientific workflows, and refined over a series of Provenance Challenges. It has bindings in XML and RDF (Moreau et al., 2010). An implementation called the Open Provenance Model Vocabulary attempts to streamline OPM and provides modules for collecting provenance related to specific scenarios, such as XSLT (Zhao, 2010). OPM is well-used as far as provenance ontologies go; its provenance markup capabilities built into e-science workflow tools such as Taverna. Because I have used the OPMV module for XSLT provenance, I use the OPM/OPMV terms "Artifact," "Agent," and "Process" here to describe the entities involved in provenance.

At the time of this writing, the Provenance Working Group is in the process of refining the W3C PROV ontology. Based on OPM, the W3C PROV attempts to be a provenance mark-up solution for everyone instead of just scientists, a general but flexible vocabulary for creating provenance records for the semantic web. Because the W3C PROV drafts are still constantly being revised, I have not used it extensively. Still: the Working Group contains a number of members who previously built other provenance ontologies; the older Provenance Vocabulary is being revised as an implementation of PROV; and Dublin Core provenance markup is being developed in tandem with the W3C efforts. These are foremost among many indications that the W3C PROV ontology will become the foremost provenance ontology. Most importantly, serious effort is being put into fixing older ontologies' technical problems, for example OPM's inability to distinguish versions of the same artifact (Belhajjame et al., 2012).

4. XSLT provenance and the meta-stylesheet

As previously discussed, XSLT is a powerful tool for transforming textual data-documents. However, there are few options to automatically generate documentation of an XSL transformation, and a complete provenance record would have to be written out manually. Despite the limited capabilities of XSLT processors, XSLT itself can be used to obtain information about stylesheet A and its templates, the input document, the XSLT processor, the XSLT version, and the transformations themselves. However, coders of XSLT have to be aware enough of provenance and interested enough to include XSLT for provenance documentation besides the XSLT already needed for their work. Knowing this, I sought to find - and, as it turned out, to create - a tool for XSLT provenance that was not reliant on the XSLT programmer, but rather could be used to automatically generate provenance documentation on command.

Wendell Piez of Mulberry Technologies suggested a meta-stylesheet method for annotating the provenance of XSL transformations, using XSLT to return XSLT provenance. The "meta-stylesheet" takes a stylesheet A as input, tranforming it into a stylesheet B, which contains enough of stylesheet A's structure so that when both are given the same XML input document,

the same transformation processes are triggered. However, instead of new or modified elements, stylesheet B outputs a file composed of RDF, describing the transformation process of Step 2. The files associated with the transformation are included, as are the processor agent, the templates of the stylesheet used, and the template-level transformation processes.

Because the meta-stylesheet essentially builds a new stylesheet, its code must be layered. Some provenance information is only available at run-time, and some of it can only be generated through the intercession of the meta-stylesheet. The following figure demonstrates the steps in the workflow for obtaining the provenance of Step 2 (stylesheetA.xsl transforming input.xml). Since XSLT stylesheets aren't self-reflective, the meta-stylesheet step is necessary for obtaining information about stylesheet A - its file name and location, the XSLT code within each of its templates, the XSLT version used, etc. It can then pass on this information to stylesheet B, which can also report on the transformations as they occur.

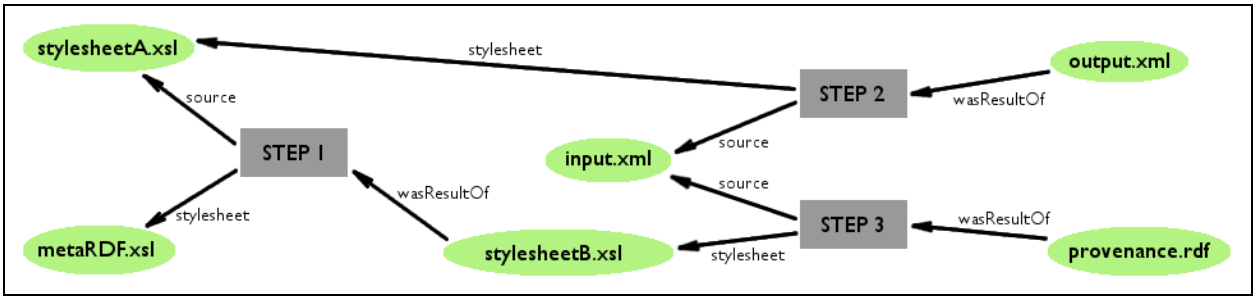

Figure 1: A workflow for applying the meta-stylesheet

Thankfully, xsl:element and xsl:attribute can be used as a placeholder for the XML output one wants, including other XSLT commands. The figures below show the kinds of code layering necessary to build a simple comment:

from output.rdf

```
<!-- Provenance for the transformation of
file:/C:/Users/Ashley/Desktop/global_001/input.xml using
stylesheetC.xsl: -->
```

from stylesheetB.xsl

```
<xsl:comment> Provenance for the transformation of <xsl:value-
of select="base-uri()"/> using stylesheetC.xsl: </xsl:comment>
```

from metaRDF.xsl

```
<xsl:element name="xsl:comment">
   <xsl:text> Provenance for the transformation of </xsl:text>
   <xsl:element name="xsl:value-of">
      <xsl:attribute name="select">base-uri()</xsl:attribute>
   </xsl:element>
   <xsl:text> using </xsl:text>
   <xsl:value-of select="$regularStylesheet"/>
   <xsl:text>: </xsl:text>
</xsl:element>
```

To reiterate, stylesheet B is an attempt to grab the provenance of Step 2, not Step 3. Two kinds of provenance are collected - file-level provenance (citing the stylesheets and input documents used) and template-level provenance (tracking which of stylesheet A's templates are fired given an input element). The diagrams below contain representations of each provenance type, paired with the output of the example case. A full example of the output can be found in Appendix A. The output captures provenance at two levels of granularity - the file-level and the template-level.

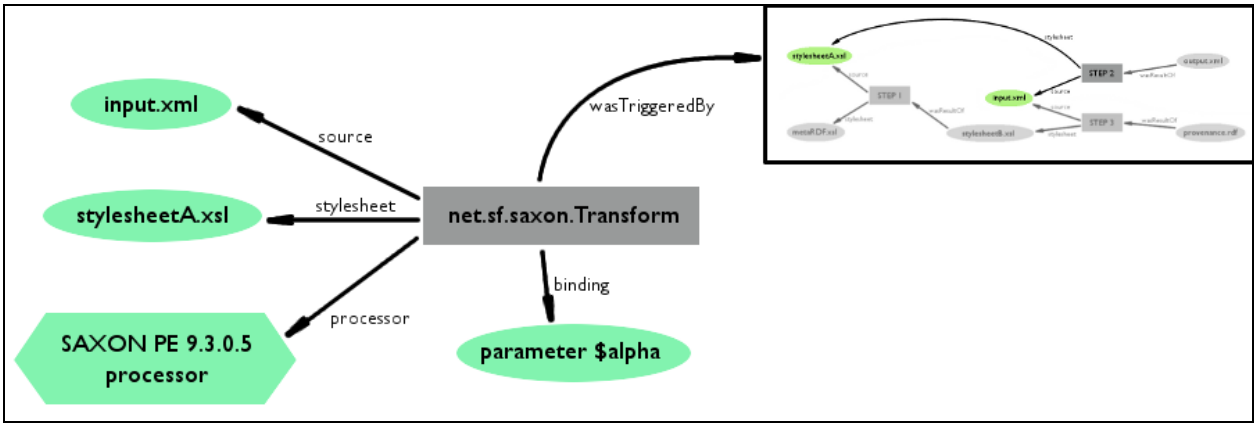

Figure 2: File-level provenance

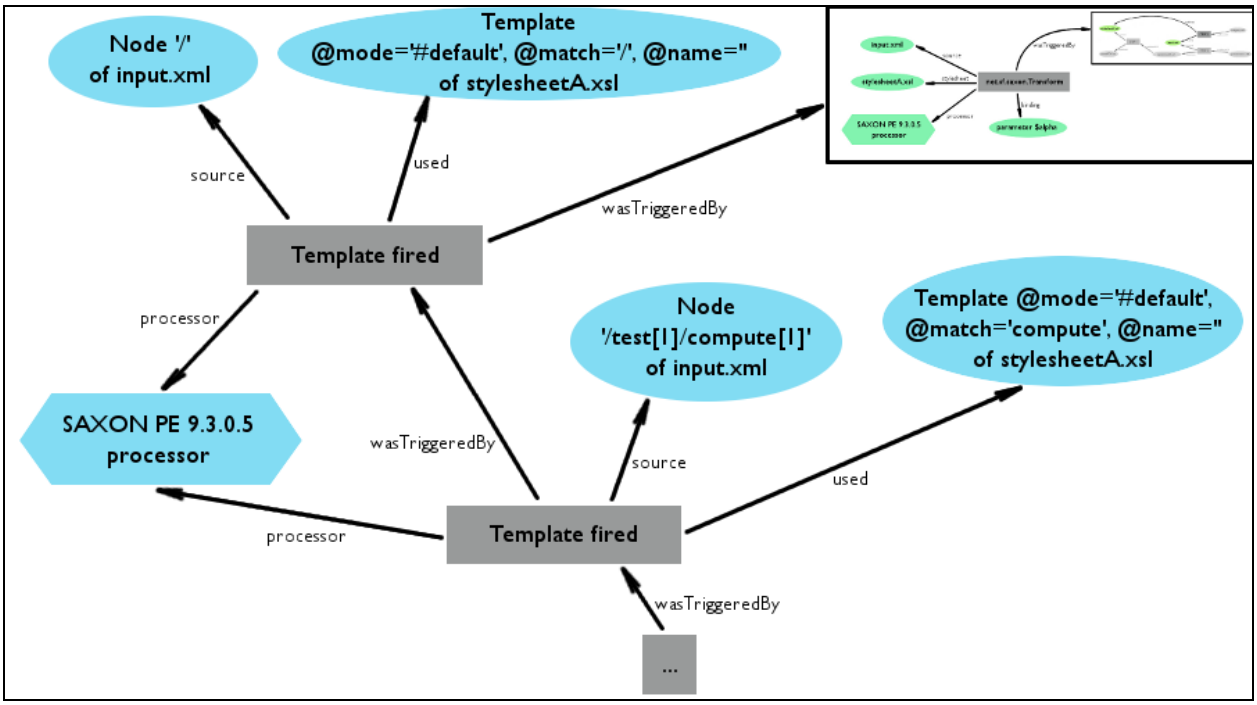

Figure 3: Template-level provenance

5. Discussion

The meta-stylesheet method demonstrates that XSLT by itself allows the capture of a lot of provenance-related information. As much information as possible is given in order to fully describe the significant agents, processes, and artifacts. However, the approach suffers in a few notable ways. Firstly, not all entities can be represented in the output. An archivist's provenance documentation is tied to the actual object itself - the provenance is based around the object, with information included about the surrounding entities. This provenance information is instead tied to the transformation and its surrounding entities, and cannot look forward and tie itself to the output object of stylesheet A. If one wanted to determine the traditional 'object-oriented' provenance of an object, one would have to document the connection between stylesheet A's output and the new provenance output.

Further, the provenance documentation does not capture any information about the human agent who set the transformation in motion, since the XSLT processor contains no information about who runs a transformation. One could argue that the immediate agent is the XSLT processor, but the fact remains that important provenance information might be helpful, but cannot be captured. Similarly, the author of a stylesheet or XML document cannot be reliably drawn out of those documents; manual efforts must be made to record authorship and ownership rights of the files in question.

Also unusual about the results is the fact that there are no timestamps. The chain of template calls should reliably show a relativistic progression of events, but placing the transformation events in time is important to maintaining an accurate history of such events. The timestamp on changed files is yet one more way of addressing this lack. Still, leaving timestamps out of the meta-stylesheet approach means that there must be one more step for the person who wishes to

compile all possible provenance information. Later versions of the meta-stylesheet should include this functionality.

There is also the matter of context. Effort has been made to allow the provenance documentation to stand on its own as much as possible, from the inclusion of the original template code, to the use of the entire filepath where applicable. The context of a single file-level transformation is given in one provenance file. What happens if there is more than one XSLT transformation, a workflow of events? Should all provenance of a series of XSLTs be contained in one file? How should workflow provenance be represented, and collected? What about the files referenced in the provenance documentation? Is it necessary to access them as well as the provenance documentation? How much access is too much to ask? A number of researchers have noted a fear of sharing data common to both e-scientists and e-humanists. What can be done to alleviate their fears?

6. Conclusions

Even on its own, XSLT provides the ability to capture a fair amount of provenance information, but it is not ideal for a macro view of data provenance as many provenance ontologies explain it – there are people (agents) who catalyze a transformation (process), which makes use of and results in artifacts. Still, some documentation is better than nothing, especially in cases where the data artifacts may be hidden due to rights issues, and the only information that can be given is about the transformation itself. Just this provenance documentation enables replayability, and the reusability of transformations. Also, XSLT programmers wishing to include provenance with their code can do so by using the techniques listed here – but, most importantly, through their own efforts by documenting their coding decisions, assigning credit within their code, and providing a versioning history of the stylesheet. XSLT is flexible enough to allow provenance capture, so long as there are those who are interested in capturing such information.

7. Acknowledgements

Many, many thanks to Wendell Piez for coming up with the concept of a meta-stylesheet, as well as providing guidance and technical help along the way. Thanks also to Allen Renear, Megan Senseney, and my colleagues at CIRSS for their advice and unflagging support.

A version of this report was presented at Balisage 2012, and at the e-Research Roundtable at the Graduate School of Library and Information Science (GSLIS) at the University of Illinois at Urbana-Champaign.

This project was supported by DCEP-H, an initiative to extend the GSLIS Data Curation Education Program to the humanities. Funded by IMLS Grant RE-05-08-0062-08, DCEP-H is based at the Center for Informatics Research in Science and Scholarship at UIUC.

8. Bibliography

Babeu, A. (2011). "Rome wasn't digitized in a day": Building a cyberinfrastructure for digital classics. Washington, D.C.: Council on Library and Information Resources. Accessed at http://www.clir.org/pubs/reports/pub150/pub150.pdf

Belhajjame, K., Deus, H., Garijo, D., Klyne, G., Missier, P., Soiland-Reyes, S., Zednik, S. (2012). PROV Model Primer, W3C Working Draft 03 May 2012. Accessed at http://www.w3.org/TR/2012/WD-prov-primer-20120503/

Borgman, C.L. (2011). The conundrum of sharing research data. Journal of the American Society for Information Science and Technology. Accessed at http://ssrn.com/abstract=1869155

Chao-fan, D., Tao, W., Peng-cheng, Z., Yang-He, F. (2010). A comparation of data provenance systems based on processing. IEEE International Conference on Intelligent Computing and Intelligent Systems 3, 374-379. Institute of Electrical and Electronics Engineers. doi:10.1109/ICICISYS.2010.5658641

Deelman, E., Berriman, B., Chervenak, A., Corche, O., Groth, P., Moreau, L. (2010). Metadata and provenance management. In A. Shoshani & D. Rotom (Eds.), Scientific Data Management: Challenges, Technology, and Deployment (433-466). Boca Baton, FL: Taylor & Francis Group. Accessed at http://arxiv.org/abs/1005.2643/

De Roure, D. (2009). Replacing the paper: the six Rs of the e-research record. Accessed at http://blog.openwetware.org/deroure/?p=56

Goble, C. (2012). The reality of reproducibility for in silico. George Washington University, ACM/IEEE Joint Conference on Digital Libraries 2012. Washington, D.C. 13 Jun. 2012.

Küster, M., Ludwig, C., Al-Hajj, Y. & Selig, T. (2011). TextGrid provenance tools for digital humanities ecosystems. Proceedings of the 5th IEEE International Conference on Digital Ecosystems and Technologies 2011. (pp. 317-323). Daejeon, Korea: IEEE. doi:10.1109/DEST.2011.5936615

Moreau, L., Clifford, B., Freire, J., Futrelle, J., Gil, Y., Groth, P., ..., Van den Bussche, J. (2010). The Open Provenance Model core specification (v1.1). Accessed at http://eprints.ecs.soton.ac.uk/21449/

Rockwell, G. (2010). As transparent as infrastructure: on the research of cyberinfrastructure in the humanities. Accessed at http://cnx.org/content/m34315/1.2/

Sahoo, S., Groth, P., Hartig, O., Miles, S., Coppens, S., Myers, J., ..., Garijo, D. (2010). Provenance Vocabulary Mappings. Accessed at http://www.w3.org/2005/Incubator/prov/wiki/Provenance_Vocabulary_Mappings

Svensson, P. (2009). Humanities computing as digital humanities. Digital Humanities Quarterly, 3(3). Accessed at http://digitalhumanities.org/dhq/vol/3/3/000065/000065.html

Sweeney, S. (2008). The ambiguous origins of the archival principle of "provenance". Libraries & the Cultural Record 43(2), 193-213. University of Texas Press. doi:10.1353/lac.0.0017

Zhao, Jun. (2010). Open Provenance Model Vocabulary specification. Accessed at http://purl.org/net/opmv/ns-20101006