Modelling Context and Provenance in a Sea of Data

Simone I. Conte[a], Alan Dearle[a], Graham N. C. Kirby[a], Adrian O'Lenskie[b], Ian Paterson[b]

[a]*University of St Andrews, Scotland, UK*
{sic2, alan.dearle, graham.kirby}@st-andrews.ac.uk

[b]*Adobe System, Inc*
{aolenski,ipaterso}@adobe.com

The problem we are addressing is how to better manage data on many different devices such as tablets, laptops, and various cloud providers and storage providers. We are also interested in the *provenance* of data in such a system; in particular provenance over how data is originated, how, when and by whom it was changed, and what processes have been applied to it.

The model we have adopted is that all data exists within a *sea of stuff* – a potentially infinite amalgam of storage repositories with potentially different storage costs, access latencies, etc. We are not concerned with the naming schemes in use in each of these repositories and treat names in them as flat. Data can be addressed using a pair (repository name, data path name). Thus the *sea of stuff* supports addressable persistent data. We do not over-write data in the sea of stuff to facilitate the management of provenance.

We term individual datums in the sea of stuff as *assets*, each of which is nothing more than a sequence of bytes. Assets may contain *intrinsic metadata*; an example of this is EXIF metadata contained within a JPEG. *Intrinsic metadata* is an inherent part of data and embedded within assets. By contrast *extrinsic metadata* may be associated with asset and is external to the asset. It might be provided *implicitly* by the storage system (*e.g.* creation times) or *explicitly*. Our model currently incorporates three forms of explicit (extrinsic) metadata: creating mappings between tags and assets; arbitrarily mapping tags and assets over relations (RDF); and defining predicates over assets. Each of these has different computational costs that increase with the amount of expressibility permitted.

The last abstraction in our model is *context*. In [1] context is defined as: *"any information that can be used to characterise the situation of an entity".* Contexts are defined by the metadata associated with assets and thus can be *implicit* or *explicit*. Implicit contexts include visibility, time, user, device and location. Explicit contexts are formed using the explicit metadata relations described above and by composition of contexts using set algebra, evaluated lazily over streams.

This model provides the ability to find data and how it has evolved in ways that are richer than what is currently possible. For example, a context for a scientific paper might allow a researcher to understand the provenance of reported results by locating the original NMR data and documentation relating to the processes that have been applied to it. As a second example, consider historical changes made to a policy document. The context for the owning organisation might permit staff to see what changes were made, by whom, the reasons for each change and what other documents influenced those changes. Whilst particular tools provide partial solutions to these problems, we seek simple, general solutions.

[1]A. K. Dey, "Understanding and Using Context," *Pers. Ubiquitous Comput. J.*, vol. 1, no. 5, pp. 4–7, 2001.