Principles of High Quality Documentation for Provenance: A Philosophical Discussion

Paul Groth, Simon Miles, and Steve Munroe

School of Electronics and Computer Science
University of Southampton
Highfield, Southampton SO17 1BJ, United Kingdom
{pg03r, sm, sjm}@ecs.soton.ac.uk

Abstract. Computer technology enables the creation of detailed documentation about the processes that create or affect entities (data, objects, etc.). Such documentation of the past can be used to answer various kinds of questions regarding the processes that led to the creation or modification of a particular entity. The answer to such questions are known as an entity's provenance. In this paper, we derive a number of principles for documenting the past, grounded in work from philosophy and history, which allow for provenance questions to be answered within a computational context. These principles lead us to argue that an interaction-based model is particularly suited for representing high quality documentation of the past.

History is important: in order to make progress in the future, it is important to learn the lessons of the past. History, therefore, becomes a vital resource for progressive societies. History is based on evidence or documentation of events that occurred in the past. Computer technology enables documentation to be produced that is more accurate and comprehensive than previously possible. Given the quantity of documentation that can be generated by computer systems (for example those running e-Science applications), what kind of documentation should be created that enables historians and users to most effectively answer questions they have about the past. To answer these questions, we argue for two principles for the creation of, what we term, *high quality documentation* of the past. One principle guarantees that users of documentation of the past have a precise semantics for it. The other principle guarantees that there exists a link between documentation and its creators. Together these principles are the first contribution of this paper.

A question that is often posed by historians and users regards the *provenance* of an entity (object, data item, etc..): what was the process that led to the entity in question? To enable the answering of provenance questions in a manner that conforms with the aforementioned principles, we introduce a model of documentation based on the exchange of messages between actors. The justification of this model is the second contribution of this paper.

The rest of this paper is organised as follows. We begin with a more detailed motivation of our work. After which, three assumptions about the world are presented. We then present two principles of high quality documentation. Each principle is grounded in work from philosophy and history. Finally, we argue for the adoption of a model, based on interactions between computational programs, as a representation of past documentation that fits both provenance and the principles outlined.

1 Motivation

Historical records are typically produced by historians weaving together scattered pieces of documentation to tell a story of past events and their relationships with one another [8]. Today, human activity is becoming increasingly automated with the aid of computational systems, which perform part or, in some cases, all of processes previously undertaken solely by humans. Interestingly, this means that activities in scientific, commercial and industrial settings are increasingly represented in computational systems, which presents an opportunity: it becomes possible to capture what happens in these settings both accurately and comprehensively.

The large amount of documentation that can be generated by computer systems changes the role of the historian, from someone who deduces history from paltry evidence to one who sifts through detailed documentation in order to make sense of history for others. If historians are to play this role, how can we ensure their task is simplified and facilitated? More specifically, how can computer scientists provide the correct kinds of documentation to facilitate the description and analysis of processes that have occurred and are captured within computational systems?

In this paper, we answer these questions for documentation about the functioning of computer systems. Furthermore, we take into account a specific type of historical question: *provenance*. Answering provenance questions about entities places requirements on the documentation of the past. In order to refer to the history of an entity as it existed at a point in time, or in the context of a specific event, documentation of the processes the entity goes through must exist up to and including that point in time. For instance, to track the provenance of a Renaissance painting, the owner needs to know both its owner in 1990 as well as its owner in 1800 and, preferably, all of the owners in between. The documentation must also be able to include information from several sources. Referring to our painting example, one source may give the ownership information for a painting in 1800, while another may provide it for 1990. Considering these requirements within computational systems, we derive a number of principles of documentation of the past that make it possible to find the provenance of data items. We term documentation of the past that conforms to these principles *high quality documentation*, which is the kind of documentation that we believe historians and users should be sifting through as they answer questions about the provenance of data items.

We now argue for our principles of high quality documentation of the past.

2 Three basic assumptions

The set of principles we intend to outline are based on a number of assumptions about the world. We believe these assumptions to be reasonable, but given that they are philosophical in nature, they are open to debate. We abstain from such debate here and instead encourage the reader to consult the philosophical literature (such as [9]).

We postulate that historians as well as others take a realist view of the world in everyday life. A general definition of realism is given in our first assumption:

Assumption 1 (Realism) *"a, b, and c and so on exist, and the fact that they exist and have properties such as F-ness, G-ness, and H-ness is independent of anyone's beliefs, linguistic practices, conceptual schemes, and so on." [3]*

That is to say, things like computers, paper and post-it notes as well as their attributes of being rectangular, made of wood, yellow in colour are independent of what anyone thinks of them. People tend to believe that cars, post-it notes, and magazines exist without having to be there to see them. Extending this concept, we introduce the assumption of verification.

Assumption 2 (Verification) *Observers can check the existence of all entities and their properties independently from other observers.*

The implication of this assumption is, for example, if an observer claims that there is a large red dinosaur outside their office window, another observer who is there at the same time can check whether this is the case or not. The ability to verify is dependent on two things: the independence of the environment from the observer (a consequence of realism) and the observers' senses not malfunctioning.

Our final assumption is as follows:

Assumption 3 (Truthful representation) *Users of documentation of the past, such as historians, want documentation to be an accurate or truthful representation of what occurred.*

In essence, the more detailed, accurate, and truthful portrait of the past given, the more users' analyses can be supported and buttressed. With these assumptions in mind, we now describe principles for high quality documentation. We begin with a discussion of truth.

3 Truth and factuality

As stated in Assumption 3, we assume that users want documentation of the past to be truthful. To establish that documentation does indeed truthfully represent the past, we must find a definition of truth that works with respect to both computational systems and the past. We begin by presenting a theory of truth that fits our assumptions. We then show how the assumption of verification is broken with respect to computer systems. The verification assumption is then discarded and we develop a new principle, based on the theory of truth we present, that provides documentation of the past for computational systems with a precise semantics.

From Assumption 1, the correspondence theory of truth seems to be the standard defensible theory of truth that we should adopt [7]. It is defined as follows:

Definition 1 (Correspondence theory of truth). *"A proposition is true just in case (if and only if) it corresponds to fact or the world." [7]*

That is to say, if a statement is expressed and it corresponds (or can be mapped) to the world, then it is true.[1] The correspondence theory means that statements can be verified by observing what they correspond to. Assumption 2 holds in the real world. Humans can check if statements are true by observing the world. They can observe whether or not a large red dinosaur is outside the office window.

In computer systems, however, the assumption of verification does not work. Software programs or components, which we call actors, do not have independent access to a common element (piece of hardware, software, component). There is no equivalent to the independently accessible environment present in the physical world. Instead, actors rely on information communicated between themselves. For example, to access the hard drive, an application program communicates with the operating system, which communicates with the driver program which accesses the physical drive and provides the data back through the chain. The application program's access to the hard drive is mediated by other programs. It does not have independent access to the hard drive. The only actor that can know directly the contents of the hard drive is the driver program. If it receives six different requests and provides the same data back for every request, no other actor would be able to detect if that was an incorrect or correct response.

In distributed computational systems, actors' dependence on communication is further emphasised by their spatial isolation. For an actor executing in computer A to know about the state of computer B, it must receive information from an actor executing in computer B. There is no other way for an actor in A to gain direct knowledge of computer B. Therefore, unlike a human, an actor cannot readily verify the state of the world independently of the information it receives from other actors. If an actor executing on computer A makes a proposition about the state of computer A, even if that proposition corresponds to the world, an actor on another computer cannot verify the truthfulness of the proposition. Hence, the assumption of verification does not hold in a computer system.

Given this problem, we make explicit what actors can verify to be truthful. Actors come to know the world via communication. They observe the world through the receipt of information. Therefore, an actor can determine whether a proposition is true with respect to what it observes. The statement "actor

[1] Throughout this paper, we will use "statement" and "proposition" interchangeably.

A received data item X" can be verified by actor A as being true or false. However, no other actor can verify that statement. In a computational system, the verification of a statement is dependent both on its correspondence to a fact and the actor that observed the fact.

Therefore, the correspondence theory of truth still holds in computational systems except that only an actor that has observed a particular event can know whether a proposition about it is true. This implies that other actors as well as users cannot know whether or not documentation about what happened in computer systems is true. Hence, we are led to the notion that documentation cannot be independently verified as truth. Instead, for every statement in documentation of the past, a user must make a judgement about its veracity. To enable this sort of judgement we introduce the following principle.

Principle 1 (Factuality) *As part of documentation of the past, actors must only record propositions that they can verify to be true, where truth is defined by the correspondence theory of truth.*

Actors, then, should only make statements about what they observe, but not about guesses or inferences about the world. If this principle is followed, users of documentation can know that statements represent reality at that time for the actors who made them. This is a powerful notion. Documentation created using Principle 1 can be interpreted as *representing the reality of the computer system* assuming that the users believe the actors that created the documentation. This notion is currently not enforced by most provenance systems. Scientific Annotation Middleware [5], for example, allows actors to record inferences about what other actors have done.

4 He said, she said

In the previous section, we stated that users should interpret documentation as statements by actors in computer systems about what they have observed. Documentation about the past then acts as *evidence* that the past occurred in some manner. Evidence plays a critical role in society. Historians, juries and others rely on evidence to make judgements about the past and predict what will happen in the future. In a legal setting, evidence is defined as follows:

Definition 2 (Evidence). *"Evidence is information, whether in the form of personal testimony, the language of documents, or the production of material objects that is given in a legal investigation, to establish the fact or point in question" (Oxford English Dictionary)*

We now draw a parallel between the statements that make up the documentation of the past for a computer system and a particular type of evidence in a legal setting, *testimony.* Coady (p. 33) gives a six point list of how testimony in a legal setting can be identified [1]. We enumerate the most pertinent here.

1. Testimony is a form of evidence.

2. Testimony is constituted by persons A offering their remarks as evidence so that we are invited to accept p because A says that p.
3. The person offering the remarks is in a position to do so, i.e. he has the relevant authority, competence, or credentials. Within English law and proceedings influenced by it, the testimony is normally required to be firsthand (i.e. not hearsay).

The statements made by actors are similar to testimony. They are evidence that something happened in a computer system. We are invited to accept a statement by an actor because the actor states it. Furthermore, the actor, from the principle of factuality, should have firsthand knowledge of what occurred. Just like eyewitnesses to a crime scene testifying in court, actors provide statements as to how things were at a particular time, hopefully without inference. However, just like testimony from people, statements made by actors may be incomplete, inaccurate or misconstrued.

Users of documentation of the past must then play a similar role to juries. Just as juries make a judgement about whether to believe the set of claims provided by an eyewitness, users must make the same judgement about documentation of the past. Such judgements are usually based on the source of the evidence. Users can interpret documentation based on a variety of factors: e.g. what other actors state about the source, the actor's past performance, the content of the documentation.[2] The key to making this judgement is to know the creator of the statement. Therefore, it is fundamental that documentation about the past in computer systems be attributable, which is our second principle.

Principle 2 (Attribution) *Each statement making up documentation of the past for a computer system must be attributable to a particular actor.*

We have argued that documentation of the past for computer systems should be both attributable and factual. We now endorse a particular model for representing documentation of the past.

5 Endorsing a model

We have developed mechanisms to record documentation about the processes executing in computational systems [2]. These mechanisms adopt a specific model centered on message exchanges, termed *interactions.* We now argue that an interaction model that follows the aforementioned principles is best suited for representing documentation. The interaction model is defined as follows.

Definition 3 (Interaction Model). *In the* interaction model*, a system is composed of* actors *that exchange information via* interactions*. An interaction*

[2] A user is determining whether to trust an actor. The concept of trust and related research are outside the scope of this paper. We refer the reader to [6] for more in-depth discussions of trust in computational systems

consists of one actor sending a message *and another receiving the same message. Actors receive no data other than via interactions, i.e. there is no external environment.*

Following Milner [4], we argue that any computational system can be described using the concepts of the interaction model. From this model, we can define a form of documentation that describes a system according to that model.

Definition 4 (Interaction Model Documentation). *Documentation of the past that describes a system according to the interaction model is called* interaction model documentation.

From the principle of factuality, actors should only document what they observe. This means that the documentation produced by one actor will be created independently from that produced by any others. Given that determining the provenance of a data item, which can include events spanning time and space, may require examining documentation from multiple actors, we need to know when multiple actors are providing documentation of the same event. According to Section 2, the observations made by actors cannot, in themselves, be independently verified but this does not prevent a historian from *inferring* that multiple actors' observations are of the same or connected events from the content of the documentation.

There are two ways that a connection between actors' observations can be inferred from interaction model documentation. First, in some cases, something about where the content of an actor's observations came from can be inferred from that content. For example, the port which an actor received TCP/IP communication on may be apparent from the documentation of that communication. Moreover, in some cases the actual actor that the content came from can be inferred, e.g. a message may be digitally signed so the author can be determined. In this way, we can infer a connection between the data received by one actor and the observations of another actor (the sender).

Second, we can infer explicit connections between actors' communications. In a system of actors with no shared world to observe, connections between actors' observations can only be made if it is apparent when some data was *sent* by an actor over the communication medium. Since observation is the receipt of data, an actor will observe, and so can document, *receiving* a message from another actor but will not directly observe the sending of a message. Fortunately, some observations can be inferred as indirectly documenting information leaving an actor. For example, the content of documentation can imply that data is about to be sent by the actor over the wire, with the intention of reaching another actor. Where an actor receives *feedback* that it is sending or has sent data, it can document this feedback and a historian can infer that it sent the data. Given this, it can be inferred that where actor A recorded receiving data, actor B recorded (feedback on) sending data, the data has identical content and each message communicated can be safely assumed to be unique, then we can infer that the actors' observations are of the same message.

The interaction model connects what would be, in an event-based model, for example, isolated, disconnected and unrelated events into connected, related and meaningful observations that, taken together, allow descriptions of coherent processes.

6 Conclusion

In this paper, we have proposed two principles that documentation about the past for computational systems should abide by to be considered *high quality*. First, we argued for creating factual documentation so that users have a precise semantics in which to interpret documentation. The argument was grounded in a philosophical investigation into what would make for a truthful representation of the past. Second, based on the observation that statements made by computational actors are equivalent to testimony in a court of law, we derived the necessity for attributable documentation. Finally, an interaction model was endorsed as an effective computational model for representing the past, especially in comparison with event based models.

If historians or any users are to effectively use documentation of the past for provenance, they must understand the underlying principles that were used in the generation of it. Therefore, it is critical that the community enumerate and justify these principles. This paper is the first to state such principles explicitly.

Acknowledgements: This research is funded in part by EPSRC PASOA project (GR/S67623/01) and the European Community's Sixth Framework Program's EU Provenance project (IST511085).

References

1. C. Coady. *Testimony: A Philosophical Study.* Oxford University Press, 1992.
2. P. Groth, M. Luck, and L. Moreau. A protocol for recording provenance in service-oriented grids. In *Proceedings of the 8th International Conference on Principles of Distributed Systems (OPODIS'04)*, Grenoble, France, Dec. 2004.
3. A. Miller. Realism. In E. N. Zalta, editor, *The Stanford Encyclopedia of Philosophy.* Fall 2005.
4. R. Milner. Elements of interaction: Turing award lecture. *Communications of the ACM*, 36(1):78–89, 1993.
5. J. Myers, A. Chappell, M. Elder, A. Geist, and J. Schwidder. Re-integrating the research record. *IEEE Computing in Science & Engineering*, pages 44–50, 2003.
6. D. H. S. D. Ramchurn and N. R. Jennings. Trust in multiagent systems. *The Knowledge Engineering Review*, 19(1):1–25, 2004.
7. F. F. Schmitt. *Truth: A Primer.* Westview Press, 1995.
8. M. Stanford. *An Introduction to the Philosophy of History.* Blackwell Publishers, 1998.
9. E. N. Zalta. The Stanford Encyclopedia of Philosophy. http://plato.stanford.edu/, Spring 2006. ISSN 1095-5054.