DEMO: Managing the Provenance of Crowdsourced Disruption Reports*

Milan Markovic, Peter Edwards, David Corsar, and Jeff Z. Pan

Computing Science & dot.rural Digital Economy Hub, University of Aberdeen,
Aberdeen, AB24 5UA
{m.markovic,p.edwards,dcorsar,j.z.pan} @ abdn.ac.uk

Abstract. Human computation systems that outsource tasks to the crowd often have to address issues associated with the quality of contributions. We are exploring the potential role of provenance to facilitate processes such as quality assessment within such systems. In this demo we present an application for managing traffic disruption reports generated by the crowd, and outline the technologies used to integrate provenance, linked data, and streams.

1 Introduction

Part of the original vision for the World Wide Web described by Berners-Lee and Fischetti in *Weaving the Web* [2] was the creation of a human network that would make it possible to create abstract *social machines* on the Web. These machines are described as: "processes in which the people do the creative work and the machine does the administration...". This is very similar to the *human-based computation* concept [5], where certain steps of a computational process are outsourced to humans. Both these visions of the web create a need for an infrastructure to handle the incorporation of human elements within a larger social computing ecosystem. Hendler [3] noted that early social machines already exist on the Web in the form of interactive applications (e.g. Wikipedia[1]). Hendler also highlighted that these applications are limited as their functions are largely isolated from one another (e.g. they are unable to easily share data). We argue that this limitation could be addressed by emerging practices such as linked data [1] - a set of principles for consuming and publishing machine-readable data on the web.

One of the ways of obtaining human input is through harvesting of so-called *collective intelligence* via crowdsourcing methods. Systems using crowdsourcing typically rely on large, diverse crowds, where the number of error generating individuals is small, resulting in minimal effect to overall system performance. However, in situations where the crowd size is small (perhaps because of the

* The research described here is supported by the award made by the RCUK Digital Economy programme to the dot.rural Digital Economy Hub; award reference: EP/G066051/1

[1] http://wikipedia.org

nature of the task or a limited population of potential participants) the potential for adverse effects caused by unreliable individuals is significant. Within such systems it is therefore critical to reason about the quality of contributions. We propose a solution to facilitate such reasoning operations based on the maintenance of a *provenance* record within the crowdsourcing system. In this context provenance would mean a record of the data generated/maintained by the crowd and the process(es) involved. Another important characteristic of such applications is their dynamic nature, with participants creating, maintaining or validating data continuously over time. We argue that participant interactions should therefore be modelled as a continuous stream of data elements published in compliance with the Linked Data Principles [4]. This necessitates a provenance solution able to interoperate with such streams.

2 Application Scenario

Travel disruption is not easy to predict and even monitoring of disruption poses some challenges (e.g. how to obtain information from the site of an incident). A crowdsourcing application able to gather, manage, and assess disruption reports would provide an obvious solution. For example, consider a system that allows participants to report travel disruption events (e.g. an accident on a particular route) from their mobile device. In addition, they are able to perform other tasks such as the creation of links between disruption reports or validation of data provided by others (evaluation). By linking here we mean the identification of relationships between disruption reports (e.g. queuing traffic caused by an incident five miles ahead). However, this data alone does not provide important contextual detail such as who created it, who performed a maintenance operation, when and how it was performed - all of which are useful when assessing the credibility of participants and the data they contribute. We argue that the provenance record should be able to provide this context, by capturing information about participants and their activities. For example, user John linked two disruption reports as related, but in the past links created by him have always been subsequently reported as incorrect by others.

A disruption event report is likely to trigger a stream of data relating to this event (such as other disruption reports, or validation reports). It is therefore entirely natural to represent these data as a stream of elements, with participants contributing to a stream about a particular event (e.g. an incident on route A90). A system utilising the crowd to manage travel disruption would thus need to be built around a set of such streams. Capturing the provenance of a stream object (e.g. the disruption report that initiated the stream) and the provenance of stream elements (e.g. who created a specific data element, or created a link between elements) would provide additional context to support reasoning about the quality of the data on the stream.

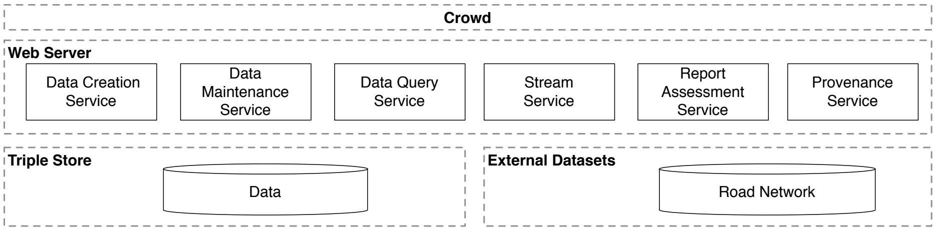
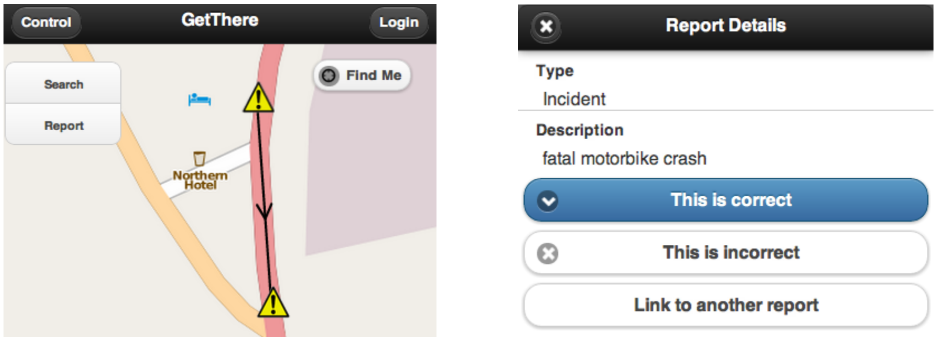


Fig. 1. Provenance-enabled travel disruption system architecture.

3 System Architecture

We have constructed a system that is able to gather and manage disruption reports from the crowd, and to capture the provenance of these activities; the architecture of this system is shown in Figure 1. A mobile client application (Figure 2) was built using the jQuery Mobile[2] and OpenLayers[3] library. The client collects information from the crowd and communicates crowdsourced results back to users. It is optimised for use on touch screen mobile devices and supports the following functionality: creation of disruption reports, creation of validation reports, creation of reports about relationships (links) between two disruption reports, visualisation of other reports and their links.



Type
Incident
Description
fatal motorbike crash

Fig. 2. A mobile client application.

The server-side framework was built as a set of RESTful web services. The data within the system is stored in a TDB[4] triple store, and accessed via a

[2] http://jquerymobile.com/
[3] http://openlayers.org/
[4] http://jena.apache.org/documentation/tdb/

Fuseki[5] SPARQL[6] endpoint. The framework is responsible for managing reports created by the mobile client application, managing the provenance of operations within the system, and the storage of this data (both the reports and their provenance). The stream service manages the creation of streams (as an ordered sequence of elements) and is responsible for handling stream operations such as: registering/unregistering queries; inserting reports (with associated provenance); and closing/deleting streams. Stream elements consist of the Unique Resource Identifier that refers to the data stored in a triple store. The provenance service generates annotations and is described further in section 3.2. The report assessment service (described in section 3.3) assesses and annotates reports generated by the crowd. Disruption reports are represented using a travel disruption ontology, describing a set of concepts from the domain of transport and travel disruption; this ontology was developed following a review of a number of UK travel information services.

3.1 Data Integration

To provide additional contextual information about a report, the system automatically records a timestamp, user location (from the phone's GPS receiver), the error associated with the location, and the result of reverse geocoding the location. Each report is thus a combination of this data and the data directly contributed by the participant. When the system receives a report, if a stream already exists for reports in that location (e.g. the street/road), then the report is added to that stream; otherwise, a new stream is created. Some reports (e.g. validation) explicitly state the relationship to an existing report, and are therefore added to the same stream as the report to which they refer.

3.2 Provenance Information

Three components within the system either manage or use provenance information: the *provenance service*, the *stream service*, and the *report assessment service*. Two types of provenance annotations are generated by the provenance service and are stored in a triple store: data provenance and stream provenance. Data provenance is generated in response to a number of events: when data is created; when data arrives from the client; or when links between disruption reports are created. The data provenance record[7] then contains information such as the agent that created the report, the activities involved in creating the report (e.g. acquiring the agent's location, uploading from the client application, and subsequent processing by the web service), and the entities used/generated by these activities. Stream provenance is generated in response to: creation of a new stream; closing a stream; and data being added to a stream. The stream

[5] http://jena.apache.org/documentation/serving_data/

[6] http://www.w3.org/TR/rdf-sparql-query/

[7] Expressed using terms from the provenance model being developed by the W3C Provenance Working Group (http://www.w3.org/2011/prov)

provenance record then contains information such as the activities that triggered the creation/closing of a stream, the activities that added elements to a stream, and the entities used by those activities (e.g. the report that was received).

3.3 Report Assessment

The report assessment service performs evaluation of the submitted reports using the provenance record and other contextual information (other reports, and links between reports). Currently we have implemented two prototype metrics within the service. The first metric is based on the distance between the reported disruption and the participant providing the report: the greater the distance between the location of the incident being reported and the location of the user reporting it, the lower the reliability of that report. The second metric uses a simple reputation model for the report creator, based on how previous reports generated by that user have been validated by others. For example, if a user creates a disruption report, which is later validated as correct by others, that user's reputation for creating disruption reports will increase. However, if others claim the report is incorrect, then the reputation decreases. As every disruption event has a limited lifespan, only validation reports received within that time period should be used when building the reputation of a participant. However, as each disruption event has a different duration, we are investigating how to incorporate crowdsourcing the end of disruption events into the client application.

During the demonstration the following features of the client application and travel disruption framework will be presented: observation (disruption report) contribution via the client application to highlight our ontology-driven solution for creation of the reports and generation of associated provenance data; validation of disruption reports via the client application; creation of links between related disruption reports and provenance associated with this process; reasoning with provenance within the system to identify unreliable disruption reports; and visualisation of the crowdsourced results via the client application.

References

1. T. Berners-Lee. Linked data. *http://www.w3.org/DesignIssues/LinkedData.html*, accessed:10/03/2012.
2. T. Berners-Lee and M. Fischetti. *Weaving the Web: The original design and ultimate destiny of the World Wide Web.* Harper Collins, NY, 1999.
3. J. Hendler and T. Berners-Lee. From the semantic web to social machines: A research challenge for AI on the world wide web. *Artificial Intelligence*, 174(2):156–161, 2009.
4. J. F. Sequeda and O. Corcho. Linked stream data: A position paper. In *2nd International Workshop on Semantic Sensor Networks (SSN)*, Washington DC, US, 2009.
5. L. von Ahn. *Human Computation.* PhD thesis, Carnegie Mellon University, 2005.