Improving curated web-data quality with structured harvesting and assessment

Kevin Chekov Feeney
Trinity College Dublin, Ireland

Declan O'Sullivan
Trinity College Dublin, Ireland

Wei Tai
Trinity College Dublin, Ireland

Rob Brennan
Trinity College Dublin, Ireland

ABSTRACT

This paper describes a semi-automated process, framework and tools for harvesting, assessing, improving and maintaining high-quality linked-data. The framework, known as DaCura[1], provides dataset curators, who may not be knowledge engineers, with tools to collect and curate evolving linked data datasets that maintain quality over time. The framework encompasses a novel process, workflow and architecture. A working implementation has been produced and applied firstly to the publication of an existing social-sciences dataset, then to the harvesting and curation of a related dataset from an unstructured data-source. The framework's performance is evaluated using data quality measures that have been developed to measure existing published datasets. An analysis of the framework against these dimensions demonstrates that it addresses a broad range of real-world data quality concerns. Experimental results quantify the impact of the DaCura process and tools on data quality through an assessment framework and methodology which combines automated and human data quality controls.

Keywords: Linked Data, Data Curation, Web Data, Data Quality, Digital Humanities

INTRODUCTION

The 'publish first, refine later' philosophy associated with Linked Data (LD) has resulted in an unprecedented volume of structured, semantic data being published in the Web of Data. However, it has also led to widespread quality problems in the underlying data: some of the data is incomplete, inconsistent, or simply wrong. These problems affect every application domain and serve as significant impediments in building real-world LD applications [Zaver13a]: it is inherently difficult to write robust programs which depend upon incomplete and inconsistently structured data. In order for real-world applications to emerge which fully leverage the Web of

[1] Documentation for the DaCura system, along with demonstrations and examples, can be found at http://dacura.cs.tcd.ie

Data, there is a need for higher-quality datasets – more complete, more consistent and more correct - and this implies the need for tools, processes and methodologies which can, monitor, assess, improve and maintain data quality over time.

The focus of the research described in this paper is the maintenance of data quality in a locally managed dataset. This includes the maintenance of inter-links to remote datasets but does not address the quality of those linked datasets directly. The emphasis is on curating a Linked Dataset in such a way that it can be easily and reliably consumed by third parties and linked to from other datasets. Within these limits, we are concerned with managing the full life-cycle of the local dataset. The life-cycle starts with the creation or collection of the data, which could be fully automated, scraped from existing web sources, or require manual user input, or anything in between. It continues through review, assessment and compilation to the publication of the dataset for consumption by third parties [Auer12]. Our goal is to design a process, and a technical framework to support that process, which maximizes dataset quality over time, even as both the dataset and the underlying schema changes.

The basic research question that we are addressing is how one can build a linked data platform which allows people to harvest, curate and publish datasets in such a way that maximizes data quality over time. In order to answer this question, we need to address several sub-problems:

- How can data quality be evaluated and measured? Our aim is to produce a system to harvest and curate datasets which maintain a sufficiently high quality that they can serve as a basis for reliable application development. Thus a broad, multi-dimensional view of data quality is required to capture the variety of factors that are important to achieve this goal.
- What is a suitable workflow to integrate data quality checks into a linked data management life-cycle? Our goal is to produce a platform that can be used by non-knowledge engineers to harvest and curate datasets and our workflow must reflect this.
- What is a suitable architecture to allow us to efficiently instantiate this workflow?
- What tools and user interfaces can assist dataset managers, contributors and users in improving dataset quality? Once again, the answer to this question is influenced by our goal of supporting non-knowledge engineers.
- How can we evaluate the effectiveness of our process, architecture and tools in improving data quality?

We assume that our linked dataset is encoded as an RDF graph and stored in a triplestore. RDF is, by itself, a very flexible data model and RDF datasets can be self-describing. RDFS and OWL are languages, built on top of RDF, which provide support for the definition of vocabularies and ontologies. By defining schemata in these languages, automated validation algorithms can be applied to ensure that instance data is consistent with the schema defined for the relevant class. However, even with exhaustively specified RDFS / OWL schemata, there are still many degrees of freedom which present individuals with opportunities for making different,

incompatible choices [Hogan10]. In practice, Linked Data datasets tend to employ various conventions which are amenable to automated validation but are not easily expressed as semantic constraints [Hyland13]. For example, RDF specifies that every resource is identified by a URI, and Linked Data URIs should dereference to resource representations [Bizer09] but such constraints are beyond the scope of OWL and RDFS.

Data quality depends upon more than fidelity to a schema – in most domains it is also important that data be correct, that it is an accurate model of whatever real world entities it purports to represent [Zaver13b]. Validation of data correctness with respect to the real world entities it purports to model is often beyond the reach of automated mechanisms, as it depends upon domain expertise. For example, a dataset which represents a historical era may contain dates of events and the accuracy of such information cannot be properly ascertained without a deep knowledge of the specific domain. Thus, in order for a dataset to ensure high-quality of published data, there must be provision for domain experts to review, assess and correct harvested data. This is particularly challenging since domain experts tend to have limited expertise in knowledge engineering and semantic modeling languages [Dimitrova08]. Thus, if it is to be practically useful, a data quality framework must provide them with suitable interfaces which allow them to detect and correct errors and inconsistencies without requiring a deep knowledge of the underlying semantic representations. One of the major design challenges, in the construction of a framework for producing high-quality data, is the appropriate orchestration of automated and manual processes. In general, we can consider that human time is a scarce resource, compared to automated processing, especially when it comes to domain experts.

The principle contribution of this paper is firstly a novel workflow and process, based on the integration of a variety of automated and manual processes into an efficient improvement cycle for producing high-quality datasets. A second contribution is a novel architecture and the description of working implementation which instantiates that architecture and the underlying process. Further contributions include the concept of using data quality measures that have been developed to measure existing published datasets as requirements for generating such datasets; a novel means of measuring harvested data quality; a demonstration dataset that we have published with our framework and basic usability evaluation of our tools.

The remainder of this paper is organized as follows: section 2 looks at related work in the area of LD data quality, with particular focus on the variety of dimensions by which data quality can be measured; section 3 describes our process, architecture and framework in detail; section 4 describes how we have evaluated our work; section 5 presents our conclusions and a brief guide to future research directions.

Background

Widely varying data quality has been observed in Linked Open Data [Zaver13b] and data quality problems come in different forms, including duplicate triples, conflicting, inaccurate, untrustworthy or outdated information, inconsistencies and invalidities [Flouris12, Hogan10]. The issue of measuring data quality is relevant across the full range of datasets from carefully curated datasets to automatically extracted and crowdsourced datasets. Although data quality is

generally considered as fitness for use [Juran74] within a certain domain, it is possible to have a general framework for assessing data quality. In [Zaveri13a] the state of the art in the area has been surveyed and the authors distil a comprehensive list of dimensions that can be used to support data quality assessment. Six categories of dimensions (and associated metrics classified as subjective or objective) are identified - contextual, trust, intrinsic, accessibility, representational and dataset dynamicity. These categories and the quality dimensions that belong to them were largely developed with the intention of applying them to datasets that have been published by third parties, in order to efficiently gain a reasonably objective picture as to their quality. However they can equally serve as requirements for frameworks which aim to produce high-quality data. Thus, in the design of our system, we have adopted the above taxonomy from [Zaveri13a]. This is especially useful due to the fact that a significant number of metrics and measures for evaluating the various quality dimensions have been developed. Because our focus is on the production, rather than consumption, of high-quality datasets, some of the metrics do not apply. Furthermore, in some cases they are extraneous as the quality dimension can be engineered into the system. Nevertheless, they provide an excellent structure for identifying and defining requirements and for assessment and we thus adopt them.

Data quality assessment frameworks have been developed to be applied in a manual [Mendes12, Kontokostas13a], semi-automatic [Hogan10] or automatic [Gueret12] manner or in the case of [Zaveri13b] supporting both semi-automatic and manual approaches. A variety of techniques are used to test for data quality issues. For example, [Furber10] defines a set of data quality requirements through SPARQL/SPIN queries that are then used to identify data quality issues. [Kontokostas13b] takes this further by compiling a comprehensive library of quality tests expressed using SPARQL templates. Once issues are identified repair actions are required. For example, [Flouris12] uses automated repair methods initially based on provenance information. There is, however, no consensus yet as to in which part of a linked data dataset life-cycle should quality analysis be applied. In the W3C technical note for "best practices in publishing linked data" [W3C13a] three life-cycles are presented. The first life-cycle has (1) Identify, (2) Model, (3) Name, (4) Describe, (5) Convert, (6) Publish, and (7) Maintain. The second life-cycle incorporates (1) data awareness, (2) modeling, (3) publishing, (4) discovery, (5) integration, and (6) use cases. The third life-cycle presented uses the following steps: (1) Specify, (2) Model, (3) Generate, (4) Publish, and (5) Exploit. None of the life-cycles presented explicitly includes a data quality analysis step and we assume that this is because it may be considered implicitly in some of the steps. The life-cycle presented in [Auer12] is more explicit about when quality analysis and correction might occur. The life-cycle consists of the following steps: (1) Extraction; (2) Storage/Querying; (3) Manual Revision/Authoring; (4) Interlinking / Fusing; (5) Classification / Enrichment; (6) Quality Analysis; (7) Evolution / Repair; (8) Search / Browsing / Exploration. In the context of this life-cycle, analysis is undertaken as part of the authoring process in order to ensure that only quality data is provided.

Through analysis of changes in content and interlinking in linked data sources (such as the analysis undertaken in [Umbrich10]) it is clear that supporting the dynamics of datasets is a

critical requirement for dataset management platforms. The majority of the research in the area to date has focused on incorporation of approaches into the platforms themselves. KAON [Bozsak02] ensures through its API that all access to data is monitored and if necessary change rules (such as those needed for ontology evolution) are applied. More recently however there have been attempts to coordinate platform-independent efforts [W3C13b] and to introduce approaches to support dynamic monitoring through dataset change description vocabularies [Talis, Popitsch11] and change notification protocols [Passant10].

Adoption of the semantic web has been slow, partially due to difficulty lay users have in using it [Heath06]. These issues have persisted into the linked data era, but have been more prominent and urgent due to increased adoption and due to the volume of data involved. Existing tools however make it difficult for users to interact with a linked data dataset, most of them require technical skills and in most cases the results are not very usable [Garcia 11]. Key requirements to support interaction with a Linked Data dataset have been identified in [Dadzie11], and in addition an analysis of prominent tools in the state of the art, where they succeed and their inherent limitations. From the survey it is clear that support for lay-users is considerably lower than for tech-users, and restricted largely to custom applications and visualizations built to suit specific end goals. To address this current limitation for supporting lay-users, two recent research directions appear to hold promise. The first argues that user interaction issues should be addressed during the software development process itself (and thus compliments nicely with those efforts on data quality assessments mentioned earlier). In this direction for example, [Shahzad11] presents an approach for mapping formal ontologies to GUI automatically, so that the GUI preserves the domain specific properties of the underlying domain ontology. The second direction is in the area of applying well known approaches from other application domains. For example, associating common purpose widgets (small UI components for visualization, exploration and interaction that can be declaratively embedded) with RDF resources so that interfaces can be dynamically composed [Haase11].

THE DACURA FRAMEWORK

The DaCura framework is designed to support the harvesting, assessment, management and publication of high-quality Linked Open Data. It is based upon a simple workflow where *data harvesters* generate reports (purported facts) from unstructured sources, *domain experts* interpret reports and generate facts, which are published for *consumers*, who can generate corrections and suggestions, while *data architects* define how those facts are represented. Figure 1 illustrates this basic workflow and shows how this functional division facilitates the construction of a process for building and maintaining a Linked Data dataset which addresses the various categories of data quality requirements at different stages in the information processing workflow.

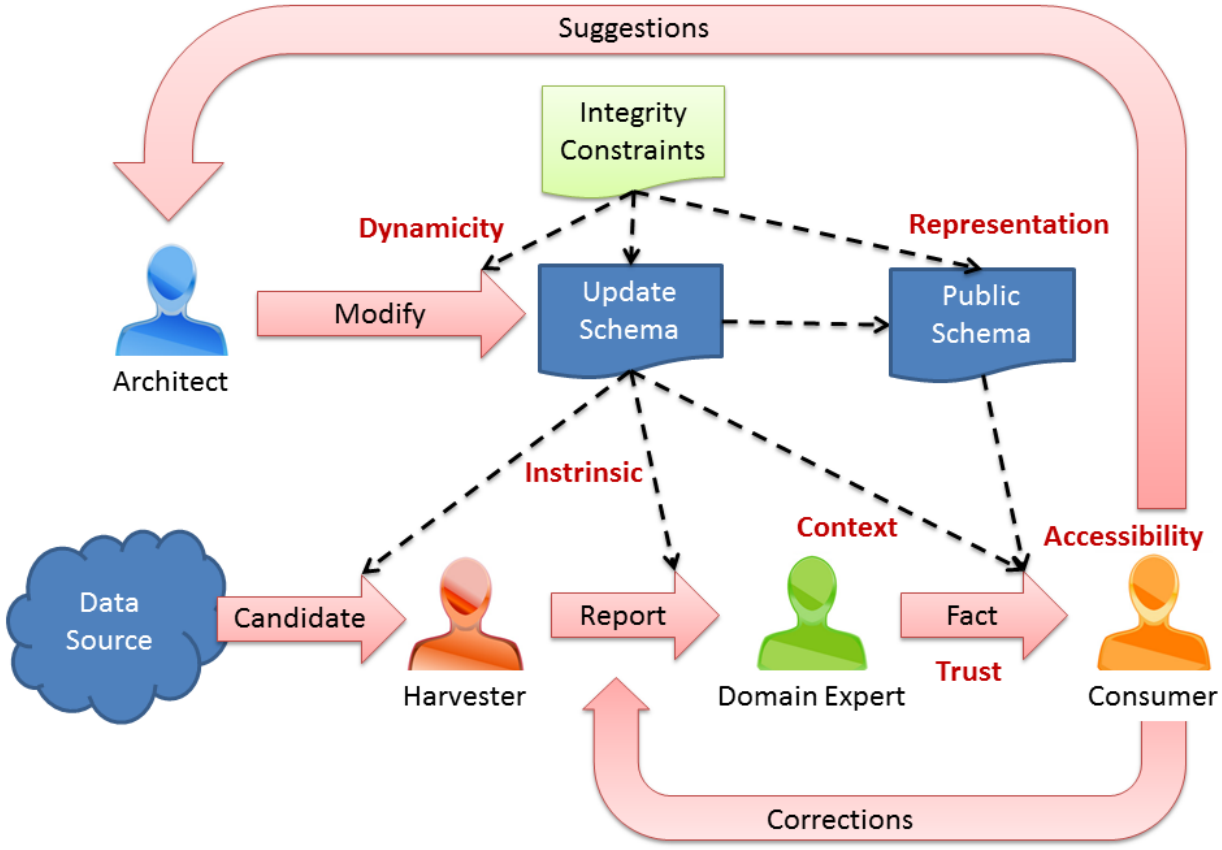

Figure 1 DaCura workflow & roles. Dashed arrows represent application of constraints to improve data quality dimensions

Workflow, Process & Roles

The framework defines four basic roles which collaborate together in maintaining a high-quality dataset. They are the data-architect, domain expert, data harvester and consumer. Each of these roles has a specific function in the DaCura workflow. The separation of roles is premised on the assumption that domain experts and data architects are relatively rare and should only be allocated functions which require their particular expertise. It enables the application of different data quality improvements at appropriate stages of the process. It is not intended to enforce a security-focused separation of duties approach: individuals can take on multiple roles if desired. The roles, and a description of the data quality services that the DaCura framework provides them, can be summarized as follows:

Data Architect: responsible for defining an *update schema* to constrain the entities that may be modeled in the dataset and making whatever changes to the schema that may be required as the dataset grows and evolves. The framework aims to help data architects to minimize schema errors by applying integrity constraints on all changes to the update schema to ensure that the overall dataset remains backward compatible over time.

Data Harvester: responsible for inputting information into the system and populating the dataset by creating reports (defined as purported facts), from candidates identified in some external data-source. The framework provides intrinsic data quality support by ensuring that they conform to the schema specified by the data architect and only accepting those that are compliant.

Domain Expert: responsible for transforming incoming reports, which may be unreliable, conflicting, duplicates or simply erroneous, into facts and a public schema, which represent the expert's best-effort compilation of reports into an authoritative dataset for public consumption. The framework provides tools to help experts improve quality in the trust, contextuality and representational categories of the published data.

Consumer: responsible for reading the published data in a variety of formats and feeding suggestions for corrections of erroneous data and schema-modifications back into the system, via tools designed to improve general data-accessibility.

Table 1 - DaCura Tasks Responsibility Assignment Matrix [PMI]

Tasks	Architect	Expert	Harvester	Consumer
Managing Schema	Accountable	Consulted	Informed	Informed
Filtering Candidates		Accountable	Responsible	Informed
Interpreting Reports		Accountable	Informed	Informed
Consuming Facts	Informed	Accountable		Responsible

Technical Architecture

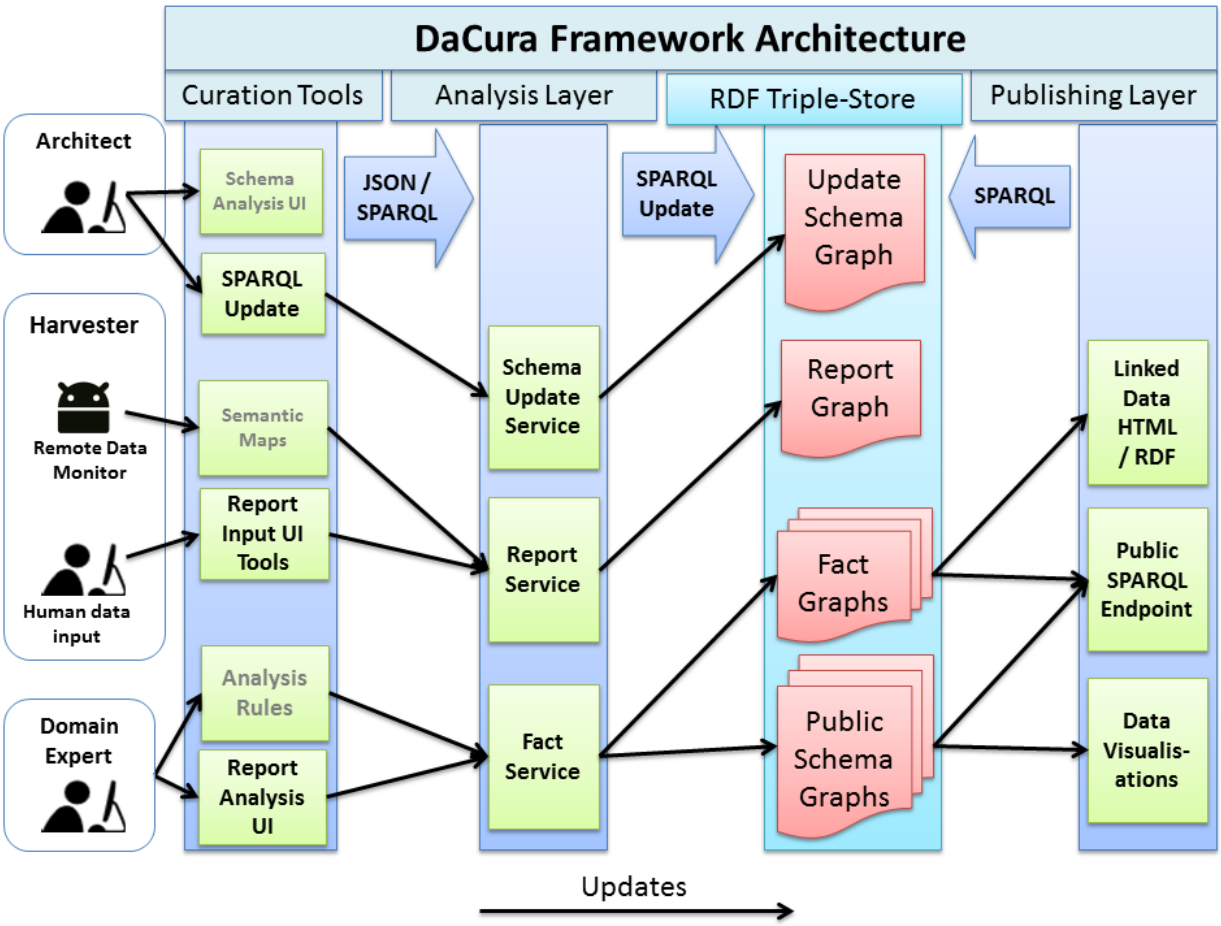

Figure 2: DaCura architecture and update dependencies

Figure 2 presents the high-level functional architecture of the DaCura framework. It consists of 4 layers. At the centre is a standard, generic triplestore containing data segmented into a number of named graphs. The publishing layer provides a suite of tools to provide the system with the ability to publish accessible, context-sensitive datasets and visualisations. The analysis layer validates all updates to the triplestore by applying a set of constraints against them, designed to provide data quality guarantees within the overall workflow. Finally, the curation tools layer provides support for the generation of end-user tools which leverage the validation services provided by the analysis layer and provide rich contextual information for further stages in the workflow.

Data Storage Layer

All persistent information is stored in a triplestore. The prototype implementation uses Fuseki[2] as its triplestore, but DaCura supports any triplestore that supports the SPARQL 1.1 Update standard [W3C13c]. All communication between the triplestore and the rest of the

[2] http://jena.apache.org/documentation/serving_data/index.html

system uses only W3C SPARQL standards, to avoid any dependencies on any particular triplestore system. A simple convention is employed whereby data is divided between different named graphs according to function.

The **Update Schema Graph** contains the definition of the structure of the dataset and the constraints that data instances must conform to. This graph is used by the analysis layer to validate the format of incoming reports and to generate user-interface data-input elements. A named graph is used to separate this, schema-related, information from the rest of the triplestore. DaCura uses this convention to distinguish schema-related triples from instance related triples as this distinction cannot be cleanly inferred from inspection of the RDF and RDFS statements. For example, enumerated properties may be modeled as instances of a generic class – from a statement-level view, they are instances, whereas from a dataset point of view, they are components of the schema. The Update Schema Graph includes properties and classes which help automated processes analyze incoming data and enforce integrity rules and schema conformance. In general, the greater the precision with which the schema is defined, the easier it is to automate quality-control analysis. However, the philosophy of Linked Data tends to favor datasets that do not rely upon complex, tightly-defined, formal schemata, hence it may be configured how much of this schema is exposed through the publication layer.

The **Report Graph** contains reports submitted by data harvesters. Reports are considered to be purported facts which must be mapped into facts by domain experts and are maintained as a separate graph. DaCura uses a simple class model where reports consist of an instance of a fact type, as defined by the update schema, and an instance of a context class, describing the provenance of the report and the context in which it was generated.

The **Fact Graphs** contain the dataset, as it has been published for broad consumption. Facts are instances of the RDFS classes defined in the update schema graph. They are facts, not in the sense that they are objectively true, but in the sense that they are the highest quality approximation to the fact that the system can produce at any given time. As soon as data is harvested from multiple sources, the potential for conflicts arises. A fact graph contains a mapping from a potentially conflicting set of claimed facts (reports in our terminology) into a consistent, composite view that constitutes the best available approximation to fact. The mapping between the report graph and a fact graph is the interface which allows domain experts to interpret the evidence provided by the reports in order to produce a coherent, consistent view of the data.

There may be several, competing interpretative frameworks that experts might apply to the underlying evidence, in any given domain, to produce different, incompatible depictions of fact. Thus, there may be multiple fact graphs, each corresponding to a different interpretation of the reports. This supports data quality in two categories. Firstly, it contributes to contextual quality dimensions by providing great flexibility in how data is presented. Secondly, it supports the creation of trust relationships between domain experts and consumers by allowing individual experts to create their own personal datasets, with their own quality guarantees.

The **Public Schema Graphs** define the schema of the published data in RDFS. They are separated from the update schema graph to enable the publication, by domain experts, of structured data with different schemata. This supports situations where the publisher wishes to hide the complexity of the constraints that apply to the creation and modification of facts.

Analysis Layer

The analysis layer has an interface that consists of 3 services, each with a well-defined RESTful API: (i) a service to support architects carrying out schema updates; (ii) a service to support harvesters creating and modifying reports; and (iii) a service to support domain experts compiling facts from reports. These services are effectively front-ends which parameterize a general purpose analysis pipeline. The internal architecture of the analysis engine is shown in figure 2.

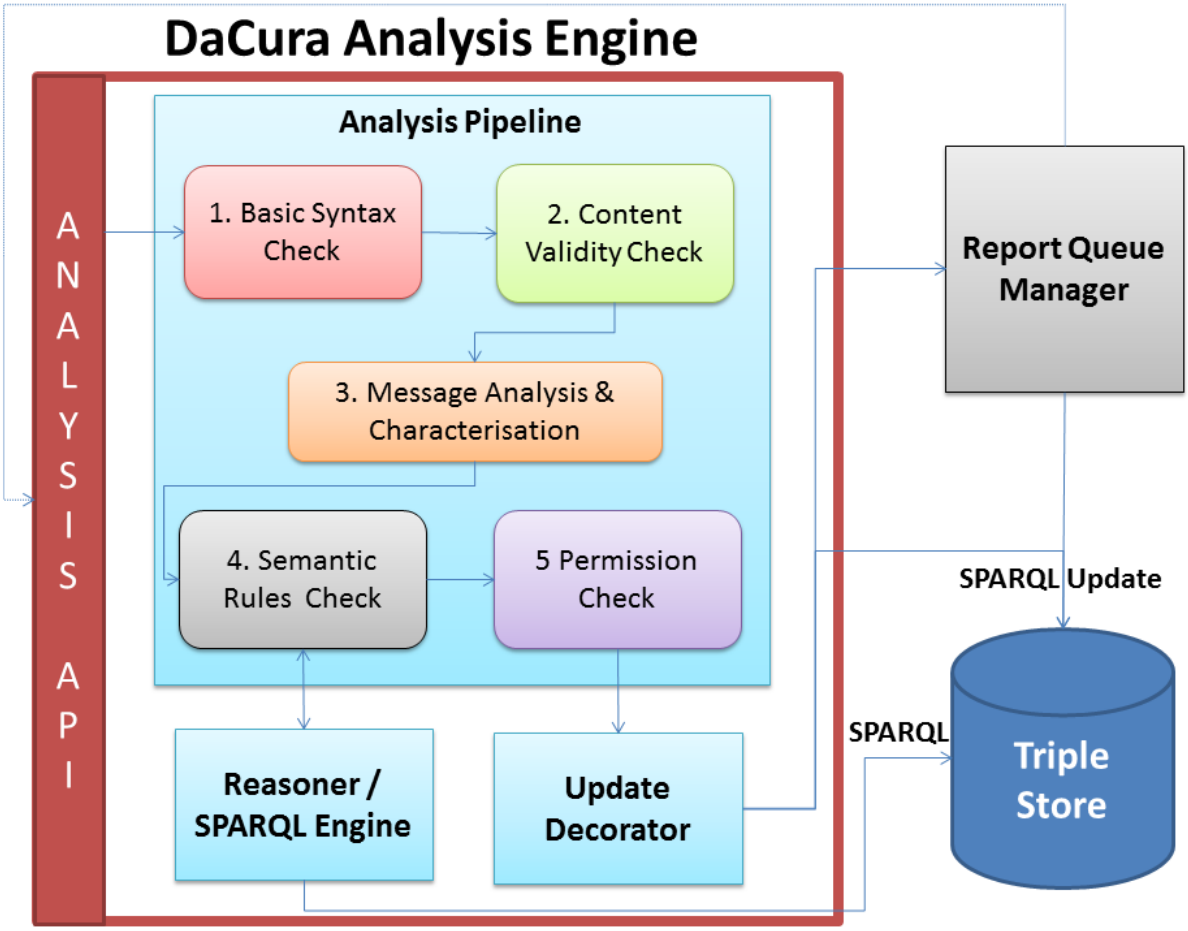

Figure 3 DaCura Update Analysis Engine

All updates to data managed by the DaCura framework pass through an analysis pipeline which ensures that messages transmitted via the API are correctly formatted and have valid content. The messages are then inspected and characterized, firstly as either an insert, delete or modify operation, secondly by the graph(s) that the statements modify. Then a set of semantic rules are tested against the updates, using a reasoner, in order to identify specific patterns which

impact upon data quality and backwards compatibility. Finally, the system checks to ensure that the update is permitted for the user's role and either executes the update against the triplestore, or passes it to the update queue manager for approval.

The *semantic rules check* component performs the most sophisticated role in the analysis pipeline. It first identifies the specific change type embodied in the update payload, according to pattern matching rules, shown in Table 2.

Table 2 Specific change types

Update type	**Payload pattern**	**Description**
Insert_Class	?c rdf:type rdfs:Class. ?c rdf:type owl:Class. ?c rdfs:label ?lb. ?c rdfs:comment ?cmt. ?c rdfs:seeAlso ?e. ?c rdfs:isDefinedBy ?e.	Define a class. Every statement in the maximum pattern is counted as part of the class definition.
Insert_Property	?p rdf:type rdf:Property. ?p rdf:type owl:DatatypeProperty. ?p rdf:type owl:ObjectProperty. ?p rdfs:range ?c. ?p rdfs:domain ?d. ?p rdfs:label ?lb. ?p rdfs:comment ?cmt. ?p rdfs:seeAlso ?e. ?p rdfs:isDefinedBy ?e.	Define a property. Every statement in the maximum pattern is counted as part of the property definition.
Insert_SubClassOf	?c1 rdfs:subClassOf ?c2.	Define a subclass relation
Insert_SubPropertyOf	?p1 rdfs:subPropertyOf ?p2.	Define a subproperty relation
Insert_InstanceOf	?a rdf:type ?c.	Define an instance statement.
Insert_PropertyStmt	?a ?p ?b.	Insert a property statement
Delete_Class	?c rdf:type rdfs:Class.	Delete the entire definition of a class. Simple execution of this update may cause instances with an unknown type and undefined class in hierarchy.
Delete_Property	?p rdf:type rdf:Property.	Delete the entire definition of a property. Simple execution of this update may cause an undefined property used in property links and undefined property in hierarchy.
Delete_Domain	?p rdfs:domain ?c.	Delete a property domain
Delete_SubClassOf	?c1 rdfs:subClassOf ?c2.	Delete a subclass relation.
Delete_SubPropertyOf	?p1 rdfs:subPropertyOf ?p2.	Delete a subproperty relation
Delete_InstanceOf	?a rdf:type ?c.	Delete an instanceOf statement. Simple execution of this update may cause orphaned instances in property statements.
Delete_PropertyStmt	?a ?p ?b.	Delete a property statement
Modify_Label	?tm rdfs:label ?lb.	Modify the label of a term.
Modify_Range	?p rdfs:range ?c.	Modify the range of a term.
Modify_Domain	?p rdfs:domain ?c.	Modify the domain of a property if there is one, and otherwise add the domain.

Once the specific change type has been identified, integrity rules are applied to the change, according to the schedule defined in Table 3 below, in order to identify, and prevent, updates which might degrade the dataset's integrity. There are currently 20 integrity rules, 15 covering structural integrity and 5 covering domain integrity. They have been selected, based on the

literature review and the authors' own experience, to represent a set of constraints which can be applied together to prevent integrity issues developing over time. They represent a set of constraints above and beyond those supported by RDFS: such as conventions which dictate aspects of the dataset such as URL naming conventions, the presence of labels and so on.

Table 3 Integrity rules used in update control

Update type	**Structural integrity rule**	**Reason**
Insert_Class	1. New classes must not already exist	Avoid multiple definitions of a class.
Insert_Property	2. New properties must not already exist	Avoid multiple definitions of a property.
Insert_SubClassOf	3. Parent class must exist	Avoid undefined classes in the hierarchy.
Insert_SubPropertyOf	4. Parent class must exist	Avoid undefined properties in the property hierarchy
Insert_PropertyStmt	5. Property statement can only be used for linking existent instances.	Avoid untyped instances in property statements
Insert_InstanceOf	6. An instanceOf statement can only apply to an existing class.	Avoid an undefined class as a type of an instance
Delete_Class	7. The class to be deleted must not be the range of any property.	Avoid an undefined class as the range of a property
	8. The class to be deleted must not be the domain of any property.	Avoid an undefined class as the domain of a property
	9. The class to be deleted must not have instances.	Avoid instances having an undefined class as a type
	10. The class to be deleted must not be in a class hierarchy.	Avoid an undefined class in a class hierarchy.
Insert_Property, Set_Range	11. The range of a property must be an existent class.	Avoid an undefined class as the range of a property
Insert_Property, Set_Domain	12. The domain of a property must be an existent class.	Avoid an undefined class as the domain of a property
Delete_Property	13. The property to be deleted must not be used by any property statement.	Avoid instances being linked by an undefined property
Insert_PropertyStmt	14. The property in the property statement must exist.	Avoid instances being linked by a undefined property
Delete_InstanceOf	15. Ensure the instanceOf statement is the only type of the instance (the instance is not used in any property statement)	Avoid deleting an instance that is in use in property statements
Update Type	**Domain integrity rule**	**Reason**

All changes	16. No blank nodes	Ensure the dataset can be easily shared with others by disallowing blank nodes.
All changes	17. All URIs are compliant with naming schema	Avoid arbitrarily introducing terms or instances.
Insert_Class, Insert_Property	18. A label statement must be included for vocabulary terms.	Ensure the vocabularies terms are understandable by human.
Insert_Property	19. Properties have one and only one range statement.	Avoid arbitrarily introducing a property without a concrete idea of how it is used.
Insert_Class, Insert_Property, Set_Label	20. The label is unique in the dataset.	Avoid humans naming two different terms with the same label.

The system can be configured to respond to breaches of the integrity rules in different ways – they can be simply rejected, or can raise simple warnings or be sent to the update queue for approval by a moderator. Calls to the analysis services via the schema update API, the report API or the fact API are all passed through the analysis pipeline. Each service corresponds to a different stage in the data-curation process and different integrity rules are applied, depending on the type of update. The goal of the system is to progressively provide stronger data quality assurances across the six quality-dimensions as reports and facts are generated.

- The Schema Update Service: ensures that updates will not break dataset conventions, and will be backwards compatible
- The Report Service: ensures that reports will be well formatted and compliant with data schema.
- Fact Service: ensures that published data is domain experts' best guess at what is a fact, from the received reports

Curation Tools

The DaCura Analysis API is used by the **Curation Tool Layer** in order to validate and perform updates on the managed dataset. In the work described in this paper, the architect used a simple web form to manage changes to the dataset. The form produces SPARQL Update statements, sends them to the analysis API, and reports back any broken integrity rules or other problems. Future work will describe the Schema analysis User Interface that we are developing.

There are two types of tools which support data harvesting. A **semantic mapping layer** describes mappings to remote data-sources which can be consumed by a remote data monitor to automatically map that data into instances of the local schema. The Update schema graph can be used to ensure that the remote data-source is mapped to a conformant instance in the local dataset. The mapping tool also gathers information about the provenance of the updates and appends them to the reports, in order to provide context to domain experts. This tool will be described fully in a further paper, it was not used in the experiments described in this work.

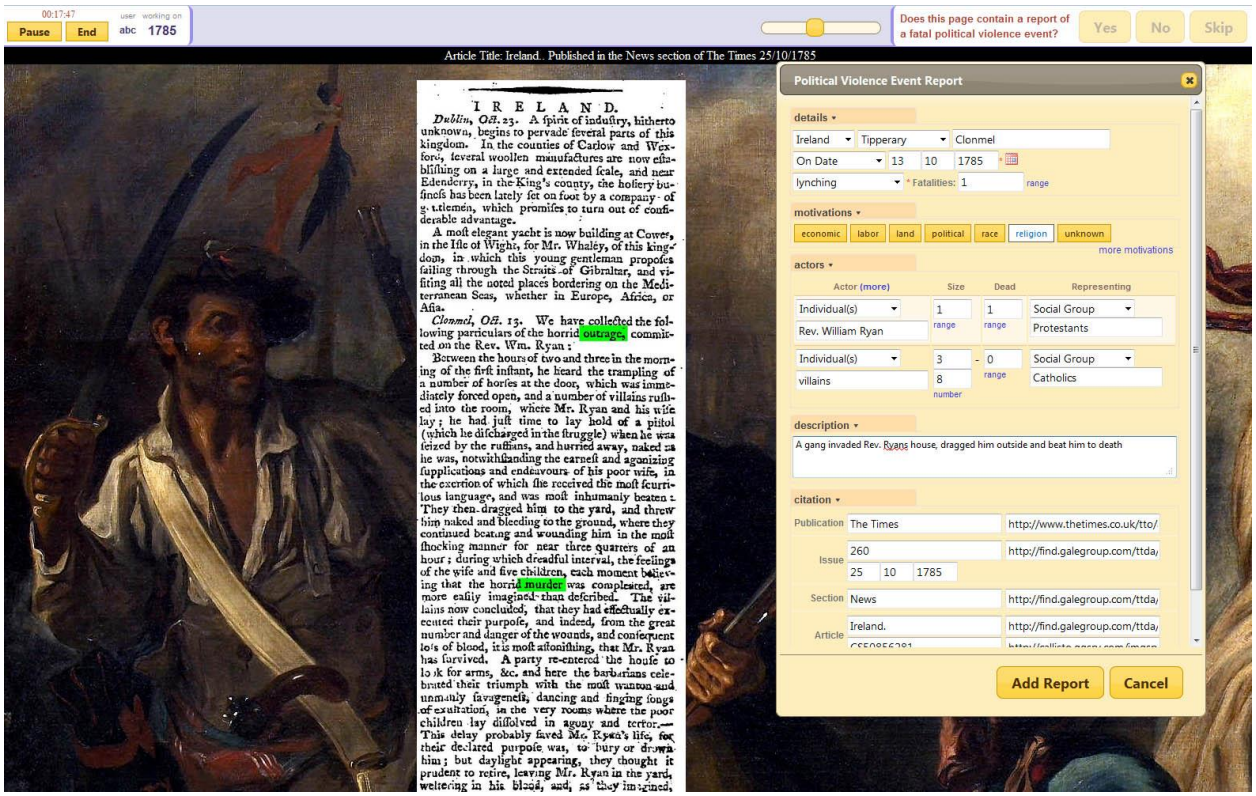

Figure 4 Report User Interface Tool Screenshot – the widget is generated from RDFS Update Schema graph and supplemented with script snippets

The Curation Tool Layer also provides human users with simple web-forms which allow them to access and modify the curated datasets in controlled ways via a web-browser. A **Report Input UI Tool** has been developed to support users acting in the harvester role. It is designed as a dialog which can be summoned to appear on any web-page through invocation of a bookmarklet. An example of such a tool, designed to capture information from newspaper reports about political violence events, is presented in figure 4. The Report Input UI Tool performs an important data quality function. If the Update Schema is defined in such a way that every property has a defined RDFS domain and range, it is possible to generate an input form from the schema and also to automatically validate user-input through the form against the schema. This supports intrinsic data quality in two important ways: the form only displays options that conform to the update schema and the report service only accepts reports that conform to the schema. The combined effect is that the system can provide a guarantee that all accepted reports will conform to the structure and content-constraints of the schema.

The second major function of the Report Input UI Tool is to generate contextual meta-data and to attach it to the generated reports. This meta-data is designed to identify the provenance of the report and whatever information is deemed most useful by a domain expert in interpreting a report's trustworthiness.

It is difficult, in practice, to map directly from an RDFS schema to a graphical user interface form. With RDFS domains defined for each property in the schema, it is possible to generate

user-input elements for each of an entity's properties. However, a simple listing of properties is rarely sufficient for a modern user interface – questions such as which properties to display and which to hide by default, and any complex interactions between UI elements depend on specific context and are hard to capture in RDFS. Hence, rather than supporting very complex schemata definitions, DaCura provides support for script-based overrides of properties, whereby a simple JQuery script can over-ride the schema-generated UI element for any given property. This approach allows sub-input elements to be defined for specific vocabularies and combined together into sophisticated forms to capture complex entities across linked datasets.

Finally, the **Report Analysis UI Tool** provides a web-based environment which supports domain experts in interpreting incoming reports and generating facts, which will be published to consumers, from those reports. It has been implemented as a simple, proof-of-concept interface, which allows domain experts to identify related reports (duplicates and those describing the same purported facts) and to correct harvested reports. It also shows experts information about the provenance of reports and the performance of the harvester who generated hem. Its implementation has focused on ensuring that sufficient functionality is available in order to allow us to designate reports as being related to one another. Future work will see this interface expanded and evaluated. The experimental focus of this paper is the assessment of the quality metrics provided by the Report Input UI Tool, whose implementation is far more mature.

Publishing Layer

The publishing layer is not the primary focus of the work described in this paper: we attempt to improve the quality of the data before publication, in order to make it easier to develop applications which leverage the dataset. However, the publishing layer is a necessary component of a data quality platform. It serves as a test-bed for evaluating the compliance of the dataset with the requirements for representational, contextual and accessible quality dimensions. Secondly, the major focus of our early efforts has been to support social scientists in acquiring, publishing and analyzing datasets. The publishing layer provides visualizations which can be useful for analysis and also serves as a bridge into the general field of digital humanities. Figure 5 presents an example of one of the standard visualizations provided by the framework. A subsequent paper will describe the publishing layer and the data quality measures that it includes.

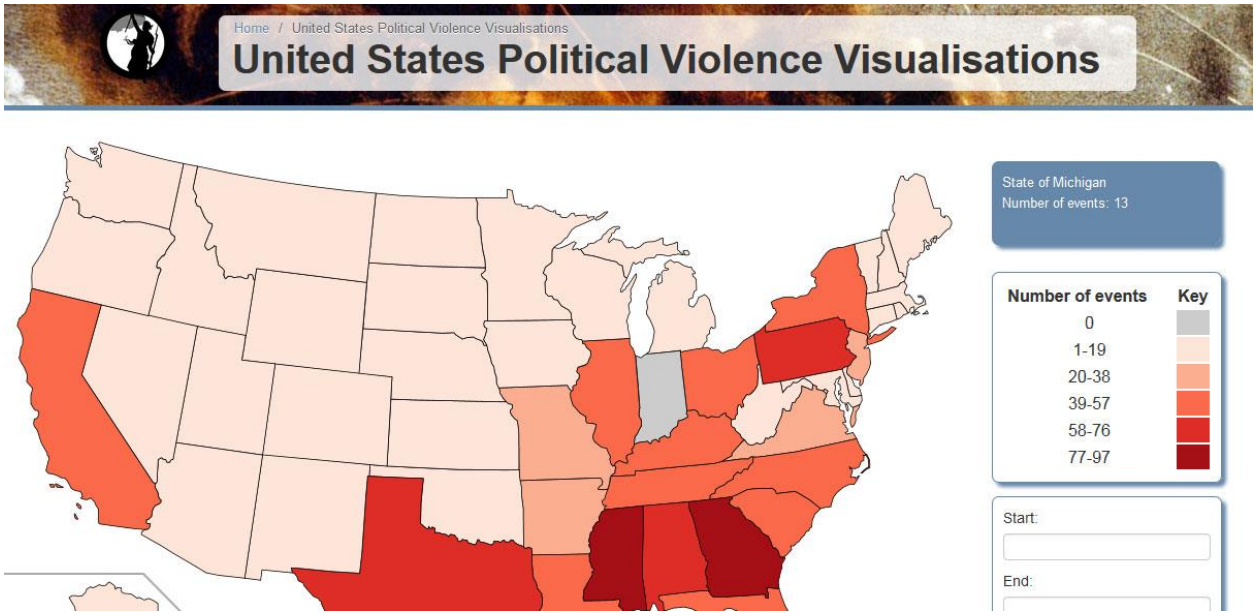

Figure 5 Screenshot of DaCura publication layer, showing visualization of US political Violence events since 1780, by state.

EVALUATION

There are two phases to our evaluation, first we examine the DaCura framework with respect to our system requirements as expressed by the 26 data quality dimensions detailed by Zaveri et al. [Zaveri13a]. For each quality dimension we identify specific quality metrics that are directly or indirectly supported by our framework and the degree of automation for each metric (see Zaveri et al. §4.4). In the second phase of the evaluation we present supporting evidence in the form of quality metric values derived from the application of the DaCura framework to a specific Linked Data dataset creation scenario – collection of political violence event reports for the United Kingdom and Ireland 1795-2006 for use by the Cliodynamics historical research community [Turchin08, Turchin12]. In addition to data quality metrics this study also evaluates the usability of the DaCura system, an important additional system requirement for the framework itself (see Zaveri et al. §4.5).

DaCura Support for Data Quality Dimensions

In this section we use the data quality dimensions and metrics defined in Zaveri et al. [Zaveri13a] to evaluate the DaCura platform's support for data quality. In determining which metrics are relevant we focus on objective metrics rather than subjective metrics to keep our analysis straightforward. It is important to note that this analysis differs from Table 8, "Consideration of data quality dimensions in each of the included approaches", of Zaveri et al. Rather than evaluating if the quality dimension or metric in question is calculated by the framework, Table 4 below specifies the quality classes, dimensions and metrics enforced by the DaCura framework, the degree of automation provided for each dimension/metric (automated,

semi-automated or manual) and the specific framework component where the quality dimension is enforced.

Table 4: DaCura framework support for data quality on dimension and metric basis

Category	Dimension	Metric Name	Component	Automation
Contextual	Completeness	property completeness	Report Service	Automated / Semi-Automated
Contextual	Completeness	population completeness	Report Service	Semi-Automated
Intrinsic	Accuracy	detection of outliers	Fact Service	Manual
Intrinsic	Accuracy	inaccurate facts	Fact Service	Manual
Intrinsic	Accuracy	inaccurate values	Report UI	Semi-Automated
Trust	Objectivity	objectivity of the source	Fact Service	Automated
Intrinsic	Accuracy	malformed datatype literals	Analysis Engine	Automated
Intrinsic	Accuracy	erroneous annotation / representation	Report Service, Analysis Engine	Semi-automated/ Automated
Intrinsic	Consistency	usage of homogeneous datatypes	Report UI	Semi-Automated
Intrinsic	Consistency	invalid usage of undefined classes and properties	Analysis Engine	Automated
Intrinsic	Consistency	misplaced classes or properties	Analysis Engine	Automated
Intrinsic	Consistency	use of members of owl:DeprecatedClass or owl:DeprecatedProperty	Analysis Engine	Automated
Intrinsic	Accuracy	literals incompatible with datatype range	Report UI, Analysis Engine	Automated
Representational	Representational-conciseness	keeping URIs short	Report UI , Schema Update UI	Automated
Representational	Understandability	human-readable labelling of classes, properties and entities by providing rdfs:label	Analysis Engine	Automated
Representational	Understandability	dereferenced representations: providing human readable metadata	Report UI, Report Analysis UI, Schema Update UI	Automated
Representational	Interpretability	Use of self-descriptive formats	Analysis Engine	Automated
Representational	Interpretability	interpretability of data	Report UI, Report Analysis UI, Schema Update UI	Automated
Representational	Interpretability	atypical use of collections, containers and reification	Analysis Engine	Automated
Dataset dynamicity	Currency	age of data	Report Service, Fact Service	Automated
Dataset dynamicity	Volatility	timestamp associated with the source	Report Service, Fact Service	Automated

Of the 23 quality dimensions identified by Zaveri et al., DaCura support for nine is shown in Table 4. The missing dimensions are mainly due to the focus of this paper on the harvesting and assessment aspects of DaCura while publication aspects are deferred to a later paper. Hence for example, quality dimensions in the accessibility class (availability, performance, security, licensing, interlinking) are not considered. Other dimensions such as versatility that focus on consumption are similarly omitted. Nine of the remaining eighteen quality dimensions are supported by DaCura. The supported dimensions are completeness (2 metrics), accuracy (6 metrics), objectivity (1 metric), consistency (4 metrics), representational conciseness (1 metric), understandability (2 metrics), interpretability (3 metrics), currency (1 metric) and volatility (1 metric). Hence the framework's strengths are currently in enforcing intrinsic, contextual and representational data quality dimensions. In a forthcoming paper we describe how many of the other dimensions are relatively simple extensions of the current framework. This is a function of our specification of a well-defined data quality-oriented data harvesting and curation process that separates reports, facts and data publication. Next we discuss how each of our system components contributes to the quality metrics described in table 4.

The fundamental DaCura component for enforcing intrinsic data quality dimensions is the Analysis Engine discussed above. The application of rules and schema information prevents low

quality data entering the system. One way in which the system differs from the standard metrics is that deprecated classes and properties are recorded through the use of the vs:term-status [vs-status] property rather than OWL deprecated properties. In future work this will be extended to the OWL case.

The Report Service component contributes to data quality in the completeness, accuracy and currency dimensions. In the next section of the evaluation we present experimental evidence for its effectiveness in terms of the property completeness and object completeness metrics. The data currency and volatility metrics are addressed by this service enforcing time-stamping of the reports when generated and the temporal information contained in the Report's citation of an original source article or document.

Our automatically-generated role-specific UIs mainly support accuracy, interpretability and usability of the dataset for data harvesters, domain experts and data architects by providing human-friendly renderings of the underlying graph data, for example see figure 4. Detection of semantically incorrect values, relevant for both consistency and accuracy, is supported by rules encoded in the data-harvesting widgets (Report UIs). However we only claim this as semi-automated support for these checks since it depends on the richness of the schema and script rules defined for a particular data collection task. Hence property values without script support in the UI may contain semantically incorrect values. For example in the political violence data collection widget, there is a description field which accepts free-text and there is no support for automated checks to ensure that what the user enters in this field is a reasonable summary of the fact that they are reporting. On the other side of the coin, there are multiple checks to ensure that the dates, places, actors and type of the event are consistent with one another and correspond to the source.

The Fact Service supports verifiability, accuracy, objectivity, currency and volatility dimensions. Verifiability is supported through the enforced requirement for Facts created by domain experts to be linked to Reports which in turn contain citations of original source material such as specific newspaper article. Accuracy is at the core of our harvester and domain expert review phases. While currently this is largely based on manual checks, it is planned that future work will add automated generation of context such as quality metrics and related material search results to support this manual process and perhaps semi-automate it in some circumstances. The objectivity dimension metric for defining the objectivity of the source is specified by Zaveri et al. as a subjective measure but the DaCura Report has a many-to-many relationship with Facts. This offers a way for us to build objective measures of Fact objectivity – are there many sources and how closely do they concur with the published version? Do many data harvesters or domain experts agree that these sources are corroborating the same Fact? These are promising avenues for the future development of new metrics.

In summary, the harvesting and assessment features of the DaCura framework currently addresses nine of the desirable Linked Data quality dimensions. Many of these dimensions are enforced by the framework so low quality data publication is not possible, for others such as the property completeness metric the platform can support local customization to a specific dataset.

In the next section we evaluate the platform in use and explore both the effectiveness of the system at ensuring data quality and the system usability.

DaCura Data Quality Case Study

In order to evaluate the impact of applying our methodology and tools on the quality of the dataset created, we performed a series of experiments based on the creation of a political violence dataset. Wherever possible the DaCura platform enforces constraints by construction. For example, the software which generates the widgets enforces constraints on the data that it inserts into the triplestore which effectively prevents data-consistency errors. Thus, several of the data quality metrics supported by DaCura are dependant only on the software conforming to its specifications. The experimental analysis provided here focuses on metrics where quality metrics cannot be provided by construction and the DaCura workflow and roles add value and produce notable results such as in the population completeness and the accuracy of the dataset generated.

The collected dataset collates historical political violence events causing fatalities in the UK and Ireland between 1785 and the present as reported by the Times newspaper, available in the London Times digital archive [TimesArchive]. Previously we have published a Linked Data vocabulary [Brennan13] for expressing these events based on earlier work by Shaw et al. on linking open events [Shaw09].

The experiments focused on the completeness and validity of the harvested data. These dimensions are particularly important for this dataset because it is intended to be used as input for a cliodynamic analysis of long term societal trends [Turchin08, Turchin12]. Such analyses require a time series of data about instability events recording properties such as the number of actors involved, the timespan of the event and domain expert-based event classifications. These time series are then subjected to statistical spectral analysis. This requires a consistent sample of events rather than a complete, authoritative record – the important thing is the dynamics not the absolute number. Given that, the more complete the dataset, the more accurately it will capture the underlying dynamics.

Hypotheses

The following hypotheses were evaluated through these experiments.

H1: Population Completeness. The framework assists users to generate complete datasets in terms of real-world reports and events.

> *Experimental Metrics*: degree to which real-world objects are not missing in both user-generated data versus gold standard and manual gold standard versus automated gold standard.

H2: Property Completeness. The framework's tool support ensures high levels of property completeness.

> *Experimental Metrics*: degree to which values for a Report or Event property are not missing based on meta-information categorizing properties as mandatory, desirable or optional.

H3: Accuracy: The framework will support users generating less erroneous reports

Experimental metrics: erroneous annotation / representation of irrelevant events in the system.

H4: Provenance: The framework will support determination of data trust values.

Experimental metrics: determining trust value for data based on the frequency of user annotations of properties as "dubious", post-task questionnaire subjective responses on data confidence.

H5: Usability: Users will find the tool easy to use and will have greater confidence in the datasets that they collect.

Experimental metrics: Standard usability scale (SUS) scores from post-task questionnaires.

Experimental Approach

In order to evaluate the quality of the dataset created by users using the tools in the experiment, a Gold Standard was created for a portion of the archive. This gold standard contains a complete set of political violence event reports for the years 1831 and 1982. These two years were chosen for a number of reasons. Firstly they both represent historical periods of considerable political violence in the UK and Ireland and hence are guaranteed to have a relatively high number of relevant articles within a single calendar year. Two reference years were picked to ensure a large temporal spread of the dataset as initial, manual investigations by a historian had indicated significant differences in the corpus in terms of the structure of articles, use of language, difficulty of interpretation by users and the optimal search terms used to query for candidate articles. Thus by spreading our users' effort in both parts of the archive it would give them different challenges, perhaps reducing boredom and enabling us to compare their performance with different usage scenarios.

The creation of each gold standard, for 1831 and 1982, followed a slightly different process. For 1831 the gold standard was created in three stages by a historian employed in our group. First they manually went through the Times digital archive for 1831 with no tool support except the basic search interface provided by the archive. This involved detailed reading of articles on an exploratory basis and developing a set of appropriate search terms to locate candidate articles. A canonical set of search terms for the early 19th century was identified by the historian as a part of this process and later incorporated into our tools as the basis for creation of lists of candidate articles to be processed by users. Articles identified as suitable for creation of political violence event reports were recorded in a spreadsheet. This set of events was designated our "manual 1831 gold standard". This manual 1831 gold standard represents the results of an independent, exploration-based approach to the archive. This manual 1831 gold standard took two weeks (10 days of effort) to produce. Then the historian used our enhanced tool to search through the archive again, using only the fixed search terms. This second gold standard for 1831 was designated our "automated gold standard". It represents the maximum set of events that a user in

the experiment could be expected to find (since only fixed search candidates are presented to users). It took the historian two days of effort to produce. Finally a composite 1831 gold standard was created, it represents the best possible gold standard and includes all events identified in either the manual or automated gold standard. Table 5 shows the composition of the 1831 gold standards: rejected reports are those which turn out, upon analysis, to be out of scope of the dataset, although this is not detectable from the individual article. The automatic column is the standard against which experimental subjects were tested. Each gold standard consisted of between 8000 and 10,000 triples and between 400 and 500 interlinks.

Table 5 breakdown of instances in 1832 political violence gold standard

1831 Gold Standards	Manual	Automatic	Composite
Reports	105	101	109
Relevant*	93	91	97
Rejected*	12	10	12
Events	44	42	44

For 1982 a different approach was used. Malcolm Sutton's database of Northern Irish Conflict deaths (1969.07.14-2001.12.31), hosted at the Conflict Archive on the Internet (CAIN) [Sutton94], was used as an independent record of political violence events. Each death recorded in the CAIN chronological list for 1982 was used as the basis of a search on the dates from their given date of death to one week after using the Times Digital Archive "Advanced Search" tool. A record of articles recording their death forms the basis for an independent gold standard for 1982 political violence events in Northern Ireland. Then a search of the archive for 1982 was performed to generate an overall 1982 gold standard.

These gold standards linked articles with event reports, which is sufficient to provide an estimate of the coverage of the datasets in terms of their ability to capture all relevant events (as each event may have multiple event reports). However in order to validate the full meta-data fields present and their values, for example for consistency it was necessary to go back through all of the event records and double-check the values filled in for each field. Due to the amount of effort involved in this manual data-cleaning process this was only performed for the first three months of 1831 and 1982, corresponding to the tasks performed by most volunteers.

DaCura Usage Scenario

Users were presented with a data extraction task – collecting structured data from an online newspaper archive. No training was given, instead two self-training documents were provided – step by step instructions for using the current data harvesting tool and a set of event classification guidelines (corresponding to the event data vocabulary description). Users were asked to use the harvesting tool identify historical newspaper articles in the archive relating to political violence within a given set of dates (typically 3 months) on a specific year according to the event classification guidelines. A set of candidate articles was pre- selected (ensuring that all users saw

the same sub-set of the corpus) and users were instructed to check each candidate to evaluate it for relevancy. For each relevant article they were instructed to enter event report data about the article into the data entry tool provided. The tools were designed so that users can break each task into smaller segments. After completing a task with a tool each user was asked to complete an online questionnaire.

User Recruitment

Users were recruited via internal computer science student and staff mailing lists in Trinity College Dublin and through personal contacts with local historical researchers interested in Cliodynamics. All users were asked to self-profile in terms of applicable historical research or computer science skills. The experimental process, including recruitment was subject to ethical approval under the Trinity School of Computer of Computer Science and Statistics research ethics protocol.

Experimental Results

Over a period of 6 weeks, 8 users used the framework to capture a total of 178 event reports – each user was allocated a 3 month time period of the archive to extract report meta-data from. Each period contained approximately 25 valid reports and 100 report candidates. Processing scanned newspaper articles from 1831 proved particularly time consuming: just over 80 hours in total was spent on the task by volunteers, whereas a similar volume of reports from 1982 took just under half of that time.

Population Completeness

The first experimental result is demonstrated in the development of our gold standard – the manual gold standard (table 5) was created by a historian over the course of 2 weeks with a relevant report completeness measure of 95.4% compared to our composite gold standard and an event completeness measure of 100% compared to our composite gold standard. In contrast the automated gold standard created by a historian using our tool in less than 2 days and had a relevant report completeness measure of 93.1% and event completeness measure of 95.5%. Table 6 shows the user-study results with respect to population completeness, the percentage of reports that were entered by a user that refer to events that appear in the composite gold standard in the time period that they extracted data from.

Table 6 Per-user results of report population completeness compared to composite gold standard

	User1	User2	User3	User4	User5	User6	User7	User8	Mean
Report Completeness (%)	22	66	17	75	40	76	70	40	49

Property Completeness

In order to assess property completeness we use the approach of defining additional meta-schema knowledge about the structure of reports and events. Under the political violence event vocabulary[3], events have between 4 and 47 properties depending on the complexity of the event and the number of referenced concepts. We classify 4 properties as mandatory (date, country, fatalities and type/classification) and a further 28 properties as desirable with the remaining 11 properties as optional. In figure 6 we plot the fraction of all event instances that contain a value for each specific property.

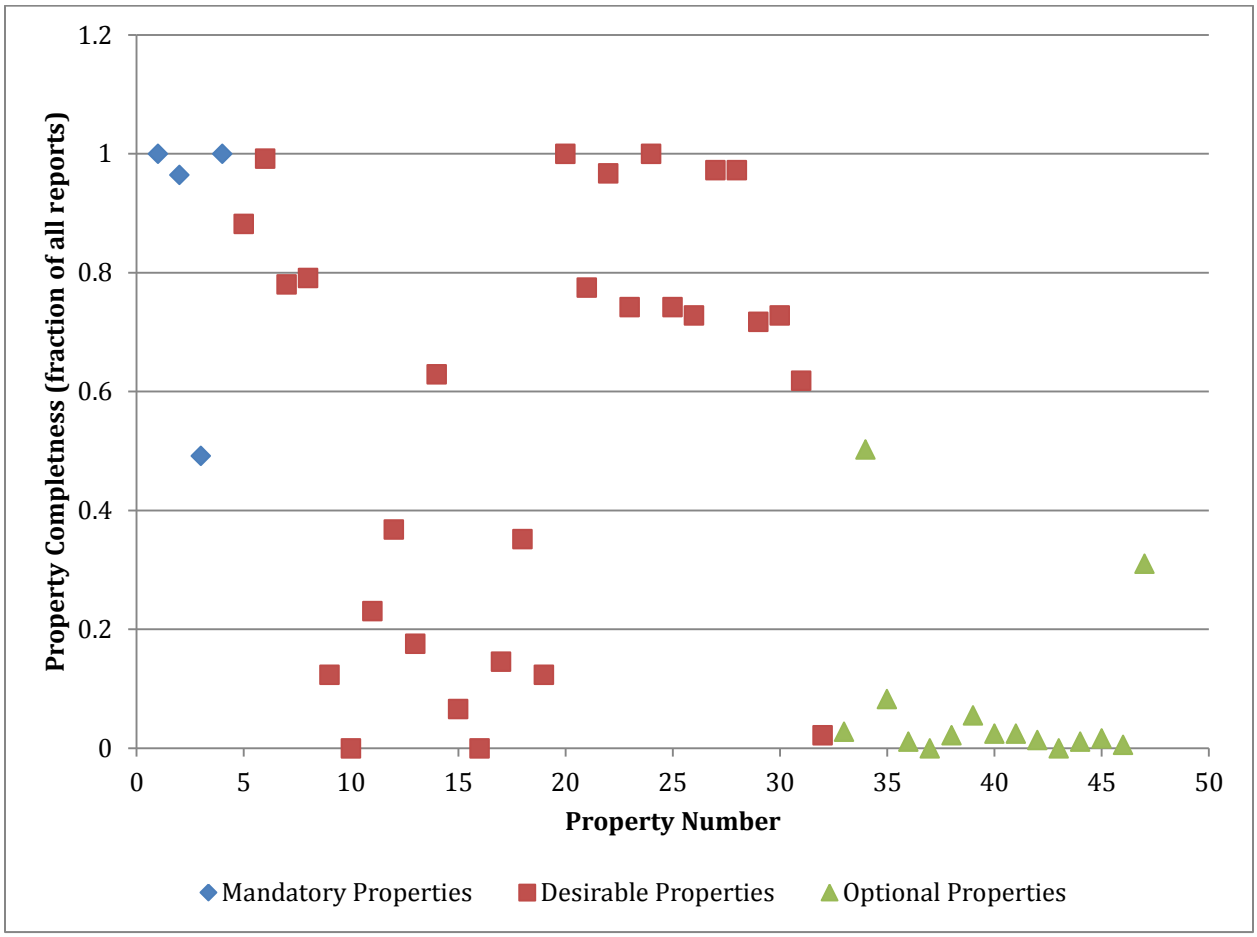

Figure 6: Property completeness measure of user-generated data on a per-property basis – illustrating how DaCura property designations and sequencing affects property completeness

Accuracy

An important measure of accuracy is the volume of erroneous data collected, corresponding to the metric "number of inaccurate facts" recorded in the system. In this user study we characterize this aspect of the dataset as the percentage of false positive event reports generated by users. Once again the evidence provided by the creation of our gold standard by a professional historian (table 5) is a useful cross-check on these false positive rates. Table 5 shows that the manual gold standard has a false positive percentage of 11.4% and the automatic gold standard has a false

[3] http://dacura.cs.tcd.ie/pv/doc/Political%20Violence%20Ontology.htm

positive percentage of 9.9% compared to the gold standard. Table 7 shows the false positive percentage on a per-user basis and the overall mean.

Table 7 per-user results of validity of documents metric – erroneous representation of reports in system

	User1	User2	User3	User4	User5	User6	User7	User8	Mean
% False Positives	78	44	83	25	60	24	30	60	51

Provenance

In our capture tool, we included check boxes which allowed users to specify that the various fields of the record were 'dubious'. This allows data-harvesters to highlight particular pieces of information to the domain expert for special attention. As part of our experiment, we measured how frequently it was employed by users. Table 8 summarises those results. The figure is the ratio of number of dubious distinctions to reports produced (so on average, one report in every 5 contains a field that is marked dubious).

Table 8 user utilisation of dubious distinction

	User1	User2	User3	User4	User5	User6	User7	User8	Mean
Frequency of 'dubious' marker	0	0.52	0.14	0.5	0.06	0.38	0	0	0.2
Subjective self-assessment for confidence in data collected (questionnaire)	Moderate	High	Very low	Moderate	Moderate	High	High	Moderate	n/a

Analysis

We analyze the experimental results against the hypotheses in turn:

H1 Population Completeness: the principle evidence in support of the hypothesis relates to the difference between the time taken for the historian to compile a manual gold standard (2 weeks) and the time taken to compile the same standard using the tool (2 days) – a five-fold increase in efficiency. This did come at a slight cost: 2 events were omitted from the automatic standard as they were not covered by the pre-canned search terms. It is also worth noting that the volunteer users, who received no training in historical research or tool usage were able to achieve a mean coverage of 49% of the relevant events with the tool, which is far higher than one might expect if they were to attempt the task unaided. Finally, the variability between individual scores for completeness – between 76% and 22% illustrates how important this measure is in terms of evaluating population completeness. Based on these results, we are incorporating an automated evaluation step into our framework, which will associate population completeness confidence scores with each individual harvester, to support domain experts in evaluating dataset completeness.

H2 Property Completeness: the framework supports property completeness quality by allowing data architects to specify properties as mandatory, optional or desirable. The capture tool ensures that all mandatory values are filled in before accepting data and warns users when desirable properties are missing. Figure 6 illustrates the levels of property completeness achieved

in the experiment. From the plot it is clear that property designation has a dramatic effect on completeness. There are a couple of outliers in the data – the 'fatalities' property of events was originally designated as optional (a missing value was interpreted as 'unknown') and later changed to 'mandatory' half way through the experiments with an explicit 'unknown' supplied as an option. The other mandatory property that achieved less than 100% was a result of a bug that was fixed early in the experimental cycle.

*H3 Accuracy***:** on average 50% of the reports generated through the experiments were false positives. By contrast less than 10% of the articles in the archive were actual positives. The data-harvesting process therefore achieved an increase by a factor of 5 in the document validity over the raw search results used to compile the archive. As the low-skilled data-harvesters output forms the input to the domain expert's role – this reduction in search space is likely to be a good trade off in many domains.

H4 Provenance: The dubious distinction was used by data-harvesters on average once in every 5 reports – manual analysis shows that this figure should have been higher, as many of the extracted details were uncertain in the source material. Nevertheless, where it was used, it was almost always valid and we can thus confirm that it provides a useful way of highlighting untrustworthy information in reports for review by experts.

H5 Usability Analysis: According to ISO 9241-11 [ISO 9241-11] usability should address user effectiveness, efficiency and satisfaction. We have already dealt with the first in the section above and here we deal with the latter leaving efficiency for future work. As part of our experiment we conducted post-task questionnaires on all of our volunteers. This custom questionnaire incorporated the 10 questions of the Standard Usability Scale (SUS) [SUS] and a further 7 custom usability questions targeted to the system and tasks under study. Our mean SUS score was 73.4 compared to the mean of 68 achieved over 10 years of previous SUS usability studies [SUS-dimensions]. This places the current DaCura system as above average but significantly short of top 10% of the distribution required to have an A-class SUS user satisfaction and rating. This also suggests that the system has above average learnability, given SUS's use as a learnability metric. This is significant because our users received no face to face training but instead received their task instructions through a simple step by step user guide. This suggests that the Report candidate processing interface could be further developed and has the potential to become a crowdsourced data platform.

CONCLUSION & FUTURE WORK

This paper has described the novel DaCura workflow and process, an efficient improvement cycle for producing high-quality datasets, its architecture and an implementation. A broad multi-dimensional concept of data quality, borrowed from work on assessing third party datasets, has been used to directly specify curation system requirements. The data produced by the curation system has then been evaluated in terms of data quality. Our analysis demonstrates how the metrics associated with data quality dimensions could be applied to our framework. The experiments have further shown that there are real and measurable benefits to dataset managers

in using our framework. Usability analysis has demonstrated that our tools are usable by non-experts and that they can generate useful results in challenging tasks.

In this paper we focused on 13 of the 26 data quality dimensions that were most closely related to dataset creation and demonstrated how the framework helps to achieve higher quality in all of these dimensions through a mix of manual, automated and semi-automated processes. A forthcoming paper will focus on the data-publishing supports offered by the framework and will deal with some of the missing dimensions, as illustrated in table 9 below. Some of these are relatively trivial to implement, for example producing publisher information to accompany published datasets, yet they are important parts of a full end-to-end data quality platform, which is the goal of our research efforts. We will be publishing full, publicly accessible versions of all of our data-sets, including the gold-standards mentioned in this paper, as linked data through the DaCura platform alongside that paper which is scheduled to be completed by the end of 2013.

Finally, we will continue to enhance our framework with greater support for high quality data-set production. Immediate plans include the integration of automated tests for data accuracy and the production of a fully featured UI to support data architects in managing schema changes over time.

Table 9 DaCura Publishing Layer Data quality Features

Category	Dimension	Metric Name	DaCura Support
Contextual	Amount-of-data	appropriate volume of data for a particular task	Linked Data API with support for multiple views at different granularities.
Contextual	Verifiability	verifying publisher information	Automatic publication of author and contributors, publisher of the data, along with links to sources / reports as described here.
Accessibility	Licensing	machine-readable indication of a license	Automatic, mandatory publication of licensing information along with UI tool to allow selection of common licences (e.g. Creative Commons)
Accessibility	Licensing	human-readable indication of a license	As above
Accessibility	Licensing	permissions to use the data-set	As above
Accessibility	Licensing	indication of attribution	As above
Accessibility	Interlinking	existence of links to external data providers	Semi-automated generation of links to external URIs (e.g. DBpedia) using owl:sameAs links.
Representational	Understandibility	indication of metadata about a dataset	Automatic, mandatory publication of title, content and URI of the dataset along with other meta-data
Representational	Understandibility	indication of the vocabularies used in the dataset	Publication of a manifest of vocabularies used along with dataset.
Representational	Versatility	provision of the data in different serialization formats	Publication of all datasets in RDF/XML, Turtle, HTML and CSV formats.
Representational	Versatility	provision of the data in various languages	Support for internationalisation / localisation of datasets

Representational	Versatility	accessing of data in different ways	Access to datasets through SPARQL endpoint, Linked Data HTML / HTTP site, RESTful URI and download dumps in CSV.
Dataset dynamicity	Timeliness	stating the recency and frequency of data validation	Publication of timeliness meta-data as part of the dataset.
Dataset dynamicity	Timeliness	no inclusion of outdated data	Automated purging of outdated data in response to schema changes and temporal rules.

REFERENCES

[Auer12] Auer, S., Bühmann, L., Dirschl, C., Erling, O., Hausenblas, M., Isele, R., & Williams, H. (2012). Managing the life-cycle of Linked Data with the LOD2 Stack. *In The Semantic Web–ISWC 2012* (pp. 1-16). Springer Berlin Heidelberg.

[Brennan13] Brennan, R., Feeney, K., Gavin, O. (2013), Publishing social sciences datasets as linked data - a political violence case study. In S. Lawless, M. Agosti, P. Clough, O. Conlan (Eds.), *Exploration, Navigation and Retrieval of Information in Cultural Heritage (ENRICH) workshop at 36th Annual ACM SIGIR conference*. Dublin.

[Bizer09] Bizer, C., Heath, T., and Berners-Lee, T. (2009). Linked Data - The Story So Far, *International Journal on Semantic Web and Information Systems* (IJSWIS), *5* (3). 1–22.

[Bozsak02] Bozsak, E., Ehrig, M., Handschuh, S., Hotho, A., Maedche, A., Motik, B., Zacharias, V. (2002). KAON—Towards a large scale semantic web. In *E-Commerce and Web Technologies* (pp. 304-313). Springer Berlin Heidelberg

[Dadzie11] Dadzie, A. S., & Rowe, M. (2011). Approaches to visualising linked data: A survey. *Semantic Web*, 2(2), 89-124.

[Dimitrova08] Dimitrova, V., Denaux, R., Hart, G., Dolbear, C., Holt, I., Cohn, A.G. Involving Domain Experts in Authoring OWL Ontologies, 7th International Semantic Web Conference, ISWC 2008, Karlsruhe, Germany, October 26-30, 2008. Lecture Notes in Computer Science Volume 5318, 2008, pp 1-16

[Flemming10] Flemming, A. (2010) Quality characteristics of linked data publishing datasources. Master's thesis, Humboldt-University of Berlin, 2010.

[Flouris12] Flouris, G., Roussakis, Y., Poveda-Villalón, M., Mendes, P. N., & Fundulaki, I. (2012). Using provenance for quality assessment and repair in linked open data. *1st International Workshop on the role of Semantic Web in Provenance Management* October 25, 2009, Washington D.C., USA.

[Furber10] Fürber, C., & Hepp, M. Using SPARQL and SPIN for data quality management on the Semantic Web. In Business Information Systems (pp. 35-46). Springer Berlin Heidelberg.

[Garcia11] García, R., Brunetti, J. M., López-Muzás, A., Gimeno, J. M., & Gil, R. (2011, May). Publishing and interacting with linked data. In Proceedings of the International Conference on Web Intelligence, Mining and Semantics (p. 18). ACM.

[Gueret12] C. Gueret, P. T. Groth, C. Stadler, and J. Lehmann. Assessing linked data mappings using network measures. *ESWC, volume 7295 of LNCS*, pages 87–102. Springer, 2012.

[Haase11] Haase, P., Schmidt, M., & Schwarte, A. (2011, October). The Information Workbench as a Self-Service Platform for Linked Data Applications. In 2nd Intl. Workshop on Consuming Linked Data (COLD), 2011, Bonn.

[Hogan10] Hogan, A., Harth, A., Passant, A., Decker, S., & Polleres, A. (2010). Weaving the pedantic web. *Linked Data on the Web (LDOW 2010)*, WWW2010 Workshop, Raleigh, North Carolina, USA, 27 April, 2010

[Hyland13] Hyland, B., Villazón-Terrazas, B. and Atemezing, G. (eds.). (2013). *Best Practices for Publishing Linked Data*. W3C Note, 6 June 2013. Retrieved June 6, 2013, from W3C: https://dvcs.w3.org/hg/gld/raw-file/default/bp/index.html

[ISO 9241-11] ISO 9241-11 (2008), *Ergonomic requirements for office work with visual display terminals (VDTs) -- Part 11: Guidance on usability*, ISO.

[Juran74] J. Juran. *The Quality Control Handbook*. McGraw Hill, New York, 3rd edition, 1974.

[Kontokostas13a] Kontokostas, D., Zaveri, A., Auer, S., & Lehmann, J. (2013). TripleCheckMate: A Tool for Crowdsourcing the Quality Assessment of Linked Data. *Knowledge Engineering and the Semantic Web* (pp. 265-272). Springer Berlin Heidelberg.

[Kontokostas13b] D. Kontokostas, S. Auer, S. Hellmann, J. Lehmann, P. Westphal, R. Cornelissen, and A. Zaveri. Test-driven data quality evaluation for sparql endpoints. Submitted to *12th International Semantic Web Conference*, 21-25 October 2013, Sydney, Australia, 2013.

[Mendes12] B. C. Mendes P.N., Muhleisen H. Sieve: Linked data quality assessment and fusion. In 2nd International workshop on linked web data management (*LWDM)* 2012.

[Passant10] Passant, A., Mendes, P. N. (2010, May). sparqlPuSH: Proactive Notification of Data Updates in RDF Stores Using PubSubHubbub. In SFSW.

[PMI] Guide to the Project Management Body of Knowledge (PMBOK Guide). PMI Standards Committee, Project Management Institute. 2010

[Popitsch11] Popitsch, N., & Haslhofer, B. (2011). DSNotify–A solution for event detection and link maintenance in dynamic datasets. *Web Semantics: Science, Services and Agents on the World Wide Web*, 9(3), 266-283.

[Shahzad11] Shahzad, S. K., Granitzer, M., & Helic, D. (2011, November). Ontological model driven GUI development: User Interface Ontology approach. In *6th International Conference on Computer Sciences and Convergence Information Technology* (ICCIT), 2011 (pp. 214-218). IEEE.

[Shaw09] Shaw, R., Troncy R., and Hardman L. 2009. LODE: Linking Open Descriptions of Events. In Gómez-Pérez A., Yong, Y., and Ying, D. (eds.), *Proceedings of the 4th Asian Conference on The Semantic Web (ASWC '09)*, Springer-Verlag, Berlin, Heidelberg, 153-167. DOI=http://dx.doi.org/10.1007/978-3-642-10871-6_11

[SUS] Brooke, J. (1996). SUS: a "quick and dirty" usability scale. In P. W. Jordan, B. Thomas, B. A. Weerdmeester, & A. L. McClelland. *Usability Evaluation in Industry*. London: Taylor and Francis.

[SUS-dimensions] Bangor, A., Kortum, P. T., Miller, J. T. (2008). An Empirical Evaluation of the System Usability Scale. *International Journal of Human-Computer Interaction* 24 (6): 574–594.

[Sutton94] Sutton, M. (1994). *Bear in mind these dead ... An Index of Deaths from the Conflict in Ireland 1969-1993*. Belfast, Northern Ireland: Beyond the Pale Publications. Retrieved September 23, 2013, from http://cain.ulst.ac.uk/sutton/index.html

[Talis] *Talis Changeset Vocabulary*, http://docs.api.talis.com/getting-started/changesets, Retrieved September 2013.

[TimesArchive] The Times Digital Archive. Thomson Gale. Retrieved September 10, 2013, from http://gdc.gale.com/products/the-times-digital-archive-1785-1985/

[Turchin08] Turchin, P. (2008). Arise 'cliodynamics'. *Nature*, 454, 34-35.

[Turchin12] Turchin, P. 2012. Dynamics of political instability in the United States, 1780–2010. *Journal of Peace Research*, 49(4). 577-591. DOI:10.1177/0022343312442078

[Umbrich10] J. Umbrich, M. Hausenblas, A. Hogan, A. Polleres and S. Decker Towards Dataset Dynamics: Change Frequency of Linked Open Data Sources", *Proc. of the Linked Data on the Web Workshop* (LDOW2010), Raleigh, North Carolina, USA, April 27, 2010.

[vs-status] Brickley, D., Dodds, L. and Miller, L. (2009). *Term-centric Semantic Web Vocabulary Annotations*, (Editor's Draft of a potential) W3C Interest Group Note 31 December 2009.Retrieved October 3, 2013, from http://www.w3.org/2003/06/sw-vocab-status/note

[W3C13a] *Best Practices for Publishing Linked Data*, W3C Note, 06 June 2013 http://www.w3.org/TR/2013/NOTE-gld-bp-20130606/, Retrieved September 2013.

[W3C13b] *W3C Dataset Dynamics*, http://www.w3.org/wiki/DatasetDynamics, Retrieved September 2013

[W3C13c] *SPARQL 1.1 Update*. W3C Recommendation, 21 March 2013 Retrieved October 2013

[Zaveri13a] Zaveri, A., Rula, A., Maurino, A., Pietrobon, R., Lehmann, J., Auer, S., & Hitzler, P. Quality Assessment Methodologies for Linked Open Data. Submitted to *Semantic Web Journal*.

[Zaver13b] Zaveri, D. Kontokostas, M. A. Sherif, L. Bu¨hmann,. Morsey, S. Auer, and J. Lehmann. User-driven quality evaluation of dbpedia. *Proceedings of 9th International Conference on Semantic Systems, I-SEMANTICS '13*, Graz, Austria, September 4-6, 2013. ACM, 2013.

Appendix 1: Participant Questionnaire

This appendix lists the contents of the online questionnaire completed by all experiment particpants with the exception of questions 5-14 which were the standard SUS questions [SUS] and so are not included here.

Q.1 Please characterise your own experience with historical research (pick all that apply):
- Work professionally in this area for less than 5 years
- Work professionally in this area for 5 or more years
- Have an academic qualification in History
- Have another relevant academic qualification
- Have general knowledge of UK and Irish modern history (e.g. from school or reading)
- Have specialised knowledge of UK and Irish modern history (e.g. from academic study or personal research)
- Interested amateur
- No prior experience
- Comment (Optional):

Q.2 If you indicated in the last question that you have an academic qualification in History or another relevant subject, please indicate the level: (Pick all that apply):
- Diploma/Certificate
- Bachelor (BA)
- Masters (MA/MPhil)
- Doctorate (PhD)
- Subject (Optional):

Q.3 Please characterise your own experience with information technology (pick any that apply) Work professionally in this area
- Have an academic qualification in Computer Science or related discipline
- Some experience with semantic web technology
- 5+ years experience with semantic web technology
- Some experience with UML or ERD modelling or database design
- 5+ years experience with UML or ERD modelling or database design
- Some experience with web development (JS, AJAX, HTML/CSS, PHP, etc)
- 5+ years experience with web development (JS, AJAX, HTML/CSS, PHP, etc)
- Basic IT familiarity
- Comment (Optional):

Q.4 If you indicated in the last question that you have an academic qualification in Computer Science or another relevant subject, please indicate the level: (Pick all that apply):

Diploma/Certificate
Bachelor (BA)
Masters (MA/MPhil)
Doctorate (PhD)
Subject (Optional):

Q.5 – Q14 Omitted as standard SUS questions

Q.15 What level of confidence do you have in the data you collected about the newspaper articles?
Very low confidence, Low confidence, Moderate confidence, High confidence, Very high confidence

Q.16 What percentage, of the total number of relevant articles for the time period you examined, do you think you discovered during the experiment?
Percentage (%):

Q.17. In your opinion, what percentage of the articles you entered data about will be included in the final political violence dataset?
Percentage (%):

Q.18. In your opinion, did the tool ensure that you collected more complete data about the articles?
Yes
No
Unsure
Add any additional comments you have

Q.19. In your opinion, did the tool help you to eliminate articles that were not relevant to political violence?
Yes
No
Unsure
Add any additional comments you have:

Q.20 Overall, would you prefer to use this tool instead of the raw Times archive and your own notes or a spreadsheet?
Yes
No
Unsure
Add any additional comments you have:

Q.21 Do you agree or disagree with this statement: I found more relevant articles in a shorter time by using the tools provided.
Yes
No
Unsure
Add any additional comments you have:

Q.22 Please add any further comments you have:

End of questionnaire.