Observlets: Empowering Analytical Observations on Web Observatory

Aastha Madaan
Web Science Lab
IIIT-Bangalore, India 560100
madaan.aastha@gmail.com

Tiropanis Thanassis
Web Science Institute
University of Southampton
United Kingdom SO17 1BJ
tt2@ecs.soton.ac.uk

Srinath Srinivasa
Web Science Lab
IIIT-Bangalore, India 560100
sri@iiitb.ac.in

Wendy Hall
Web Science Institute
University of Southampton
United Kingdom SO17 1BJ
wh@ecs.soton.ac.uk

ABSTRACT

The Web observatory is proposed as a global catalogue for sharing data-sets and analytic applications to support researchers from a variety of disciplines for analysing huge amount of research data for *Web Science* research. However, often these users fail to understand various transformations and consequences of complex data processing involved in a data analytic application. Therefore, there is a need to enable these users develop and re-use analytic applications on web observatory. In this study, we propose formal design patterns called "Observlets" for analytic applications to "observe" various web phenomena. The observlets provide abstract definitions for intermediate analysis required for a data analytic application. The users can share observlets across distributed web observatory nodes. The observlets are aimed to enhance end-users' awareness and engagement on web observatory and support programmers for *innovating* various data analytic applications.

Keywords

Web observatory, analytic applications, design patterns, infrastructure support, user-engagement

1. INTRODUCTION

Web is an infrastructure where end-users generate value and realize benefits through their activities which involve running various applications, consuming, generating and using content, and engaging in various socio-economic relations with other users [6]. These users actively engage in innovation and creation through commercial and non- commercial exchange [6]. They debate and comment on political and non-political resources, organize clubs and protests, build and sustain communities. These may be related to a number of real-world events and activities. These activities generate substantial social value for direct and indirect participants[6]. Various social media platforms such as Facebook, Twitter, open encyclopaedias such as Wikipedia, forums such as stack-overflow, Quora generate enormous volume of data about traces of various activities of people on the web. Moreover, various governments are increasingly publishing their data on the web. These and several other datasets are available through various nodes of *the web observatory* [13].

The web observatory is proposed as an infrastructural support for various interdisciplinary researchers to develop and share datasets, tools, research methodologies and analytic applications to observe various web phenomena [12]. The Web observatory is a global catalogue which engages user communities with datasets and analytic resources via dedicated portals for research and business purposes [12].

A web observatory node includes applications of computational social science, understanding of information flows, models of evolution of *social machines* and big data analytics. Datasets on a web observatory may include quantitative or qualitative data, real-time data, multimedia content, open-data, archives, and e-Science resources. End-users on a web observatory include individuals, public/private organizations and government agencies. In addition web observatory supports tools such as, harvesters and data mining software for development of analytic applications [13]. For instance, the web observatory node hosted at University of Southampton (SUWO)[3] catalogues a number of datasets about elections, natural disasters and encyclopaedic relevant articles from various web sources such as, Twitter and Wikipedia. It catalogues applications and visualizations built on these datasets. These include applications which characterize the social machines, such as the Wikipedia.

Linking resources hosted in remote locations at different web observatories can provide a global infrastructure to support observing and analyzing the web in different contexts and communities. As the web observatory is expected to span all continents, it should support sharing of data, tools and applications from another node which may be in a different *administrative domain*. Therefore, distributed management and sharing of resources is a critical challenge. This

WWW'16 Companion, April 11–15, 2016, Montréal, Québec, Canada.
ACM 978-1-4503-4144-8/16/04.
http://dx.doi.org/10.1145/2872518.2890593 .

becomes more complex considering a variety of users that engage with web observatory. To sustain such an infrastructure it is important to facilitate user-engagement in terms of complex analytic application development, sharing data aggregations and application modules through web observatory. Hence, in this study we propose to empower end-users on web observatory to share various resources ("by the people") owned by them, about people, their activities and applications which impact them on the web ("of the people") and develop analytics which can be distributed (selectively) to facilitate further analyses by other users ("for the people").

The data processing on the web observatory is particularly challenging because the data is generated from diverse sources and is in a variety of formats. Moreover, often this data belongs to different administrative domains. This raises concerns to harmonise different datasets at remote web observatory nodes with respect to analytic applications. Most of the data in the archives, from social media networks and that published by government agencies describe an event or activity in a given geo-location at a given instance of time. Therefore, for any data analyses the users need to refine the data with respect to its spatio-temporal characteristics. Moreover, complex statistical aggregations are required to study these datasets. Further for any analysis the users will need to integrate these aggregations with visualizations. The above steps for developing analytic applications becomes challenging for interdisciplinary experts who are limited by their technical skills. Even the technical users may duplicate efforts for building similar analysis for different datasets which hinders them from building richer and insightful applications.

This study proposes to simplify development and sharing of data analytics on web observatory. For this it uses conventions of design patterns. These design patterns are termed as "observlets" as they can be combined to define analytic applications which observe various dimensions of a web phenomena. The basic set of observlets define components for data harmonization, spatio-temporal analysis, statistical aggregation and visualization. These observlets form a conceptual layer between the datasets and analytic applications catalogued on web observatory.

Roadmap In section 2 we describe the related literature for the study. Section 3 defines observlets and their role in an analytic application. A case study on using observlets for *disaster management* is described in section 4. Section 5 presents summary, conclusions and future work.

2. BACKGROUND STUDIES

Web observatory infrastructure is especially useful to understand the impact of emergence of software and applications on web and their influence on business and society by building on existing analytic tools, visualisation frameworks and research methodologies [7]. In this study, we draw our related work from existing literature on "web observatory". We discuss existing software design approaches and their applicability to design of analytic applications on web observatory. Further, we discuss existing work on distributed management of web observatory resources and modelling tools in context of supporting collaboration between different communities.

Tiropanis et.al. [13] state that a critical challenge for "the web observatory" with more nodes emerging is to have standardisation of meta-data for sharing web observatory resources. To address this concern a W3C community group proposed new classes which were extension to schema.org for web observatory, web observatory project, dataset, and tool [2]. These terminologies attempt to support web observatory as a decentralized and distributed infrastructure for sharing data and analysis using standard terminologies [1]. But pursuing variety of web observatory users to follow these is a complex task. Moreover, these terminologies are too generic for the web observatory resources and do not illustrate the semantics of how they can be combined for developing or re-using existing resources on web observatory.

Existing "software pattern language" is a structured collection of patterns that build on each other to transform needs and constraints into an architecture independent of the programming language. Each pattern works in a given context but may transform the system to a new system in a new context. If new problems are encountered in new context, next "layer" of patterns is added to the system [5]. Another study [8] describes design patterns for visualizations independent of the specific programming languages. It proposes a set of twelve design patterns that describe how visualizations can be rendered from a dataset. These approaches if applied in context of web observatory applications, will be of limited use as a majority of end-users do not understand data structures and programming languages.

Popov et.al [10] use "Linked Data" and "Web browsers" for *mashpoint framework* proposed in their work which considers data-centric applications as high-level lenses (views over graphs of) data on the Web. The underlying assumption of the approach is that end-users may view data within various contexts. It proposes the data from a repository to be pivoted into another application creating an interconnected lens (graph)[10]. The framework uses RDF and linked data ontologies for selecting applications on a given dataset. Such methods are useful for small datasets and are not scalable for *big* datasets catalogued on the web observatory.

An earlier work by Brown et. al.[4], considers curated content from the Web Observatory to comprise of several *facets* and *aspects*. These aspects are either organized with respect to *perspectives* such as academic, business, personal and government, or with respect to *purpose* such as, research, insight and profit [4]. The authors propose a taxonomy of the components of the Web Observatory including data, services, interfaces, platforms and actors [4]. While the proposed models are very attractive, we argue that it may still be too restrictive to capture vagaries of phenomena on the web. While aspects and taxonomies are important, it may be infeasible to impose any specific model for the aspects or concept hierarchy on the different datasets curated from the web. In contrast, the proposed study models different components of analytic applications on the web observatory as schematic entities called "observlets". These can be implemented to develop and share application code, intermediate aggregations across web observatory nodes.

3. OBSERVLETS ON WEB OBSERVATORY

As described earlier, the *observlets* are aimed to facilitate user engagement, collaboration and innovation on web observatory for data analyses. They are proposed to enable users to understand how their data can be processed, ways in which they can share data, tool-kits and develop applications on web observatory itself. In this section we describe

observlet inventory of a web observatory and semantics of different observlets.

3.1 Observlet Inventory

Each web observatory node has a *observlet inventory*. The inventory catalogues observlets imported from other web observatory nodes, and those contributed by users registered at a web observatory. Each observlet is uniquely identifiable by its URI. The observlets can be registered at any web observatory node and can be discovered at other nodes through APIs. Formally, observlet inventory on a web observatory node can be described as:

$$inventory(wo) = (O, I) \quad (1)$$

where O O is a set of observlets contributed by users registered at *wo*. While I is a set of observlets imported from other web observatory nodes. A user can implement and modify any observlet available on the inventory. A modified observlet can be added to the observlet inventory with a different label. A user can also catalogue results of implementation of an observlet.

3.2 Defining Observlets

Here we define observlets, their purpose in a data analytic application and their schema. The study defines an initial set of observlets for harmonizing datasets, spatio-temporal refinement, statistical aggregations and visualization of results. Further observlets can be added to the observlet inventory based on requirements for newer analyses.

3.2.1 Data Harmonization

A number of databases and repositories are catalogued on web observatory which include HBase, EPrints and other proprietary databases [13]. The catalogued datasets include relational, NoSQL or RDF formats. However, different applications require different data formats as input. Therefore, re-using an analytic applications on a given dataset may require harmonising the dataset into another format.

For example, consider two demographical datasets about a geographical region. One of these is in *Mongodb* and another is in *relational* format. An application which determines "average number of people affected in floods per km^2 " requires input only relational format. In such a situation, a user will need to transform the MongoDB dataset into relational format. The **data harmonize observlet** performs such transformation to harmonize the datasets with respect to the applications. It requires a user to specify dataset to be transformed, any meta-data if available and the desired output format.

Description- The data harmonize observlet, homogenizes a dataset with respect to another dataset or application/ visualization to facilitate interoperable datasets on a web observatory. Also, it helps users understand structure and semantics of data, which applications from the inventory he or she can use for processing his or her dataset. Formally, data harmonize observlet can be defined as,

$$DHarmonise(dataSetId, ipType, metaData, opType) \quad (2)$$

where, *dataSetId* is the dataset identifier, *ipType* specifies the input dataset format, *opType* is the desired data format and *metaData* specifies any meta data about the dataset catalogued on a observatory.

Requirements- Any dataset registered on a web observatory can be input to the data harmonize observlet.

Applicability- At present data harmonize observlet supports MongoDB, RDF and SQL datasets.

Results- An equivalent dataset in the output format specified by a user.

3.2.2 Spatio-Temporal Analysis

Most of the datasets catalogued on web observatory are thematic. They describe an activity or an event in given location and time of occurrence. For more focused and in-depth analyses, the **location and temporal observlets** allow users to refine datasets with respect to their geo-location and temporal dimension. For e.g. consider a catalogued Twitter dataset on "floods in India" for the year 2014-15. For an analysis on role of social media in helping aid reach people in the state of "Kashmir in "2014" v/s state of "Tamil Nadu" in "2015", a user should refine the dataset into subsets focusing distinctly on both these states respectively. For this the users can use the location and temporal observlets.

1. **Location Observlet** queries a dataset with respect to location specified by a user. Each record of the dataset is evaluated with a boolean function which returns true if the location attribute in the dataset matches the location value given by the user. Such records are considered for further processing.

   Description- The observlet defines an evaluation function f which queries each record r of a dataset with user specified *geo_loc* location.

   $$loc\_dataset = f(r_1, r_2, \ldots, r_n) \quad (3)$$

   Here, loc_dataset is a subset of original dataset satisfying the user-defined evaluation function *f*. The evaluation function f can be defined as,

   $$f = Cond_1 \quad [AND \mid OR] \quad Cond_2 \quad [AND \mid OR] \quad \ldots \quad (4)$$

   and each Cond$_i$ can be defined as,

   $$loc\_attr \quad [IN|MATCHES] \quad geo\_loc \quad (5)$$

   where *geo_loc* can be name of a city, state, country or region. The current definition of the location observlet supports 'in', 'matches' operators. For example, a function for refining a demographical dataset about a given city in a given state can be defined as, "IN Tamil Nadu AND MATCHES Chennai". This returns a subset of dataset about the city of "Chennai" in the state of "Tamil Nadu". The observlet currently defines basic operators which can be revised and new operators can be added to it. Similar to geo-filter operators in the database world, a user can concatenate multiple query conditions using AND and OR.

   Requirements- User will need to formulate the evaluation function using the available operators or can define a new operator.

   Applicability - The location observlet can be implemented on any geo-tagged dataset.

Results- The output of implementing location observlet is a subset of the dataset satisfying the user defined evaluation function.

2. **Temporal Observlet**- It supports the user to refine a dataset by specifying a temporal window to obtain data during a given time period. For example, to analyse the role of social media with respect to criticality of floods in an area, the data specific to different time-intervals needs to be analysed. For such situations, a user can implement a temporal observlet to analyse subsets of this dataset based on their time-stamps.

   Description- User can specify a time window to select a subset of records for further processing. A time window can be set with start and end time. If the start time is not specified, all records before the end time are considered and if the end-time is not specified then all the records preceding the start time are considered from the dataset.

$$TWindow = SET \quad beginTime \quad \textbf{\textit{op}} \quad timestamp \quad AND \quad endTime \quad \textbf{\textit{op}} \quad timestamp \tag{6}$$

   The operator *op* can take values '$\leq$, $\geq$, $<$, $>$, $=$.

   Requirements - The user should specify the attribute which defines the time-stamp attribute from the dataset.

   Applicability -Temporal observlet is applicable to any time-stamped dataset.

   Result- A subset of original dataset with records from a given time window.

3.2.3 Statistical Aggregations

Due to the large volume of heterogeneous data it is impossible build data models by examining every data point. As a result, most of data analytics on web data use observational units (aggregates of smaller units). Therefore, statistical aggregations is a critical intermediate step of most data analytic applications. The database world provides basic aggregations, but complex aggregations such as ANOVA and regression require complex pieces of code.

The **aggregation observlet** supports complex data aggregations. It helps users define aggregation formulae and provides them with the pseudo-code which can directly be implemented. To address the discrepancies between aggregation type and input dataset, user can use data harmonize observlet.

Description- It gives schematic definition of a statistic. It can also cache the results of application of a aggregation on a dataset. However, this is a challenging task considering the number of stakeholders, datasets and possible aggregations. Therefore, in this study we only consider sharing of schematic definitions of aggregations.

Requirements- A user should specify the attribute(s) and the statistic to perform aggregation.

Applicability- The signature of the aggregation observlet defines attributes of dataset, their type, on which the statistic θ is to be applied. The owner of an aggregation observlet can define access rules to limit sharing of aggregations and their results. Formally, λ is the expected value of the aggregation, θ is the statistic used for aggregating values of an attribute.

$$\lambda = \theta(attr\_value_1, attr\_value_2, \ldots, attr\_value_n) \tag{7}$$

Results - It generates aggregated snapshots of the underlying dataset. These results may be integrated with available visualization libraries and tools to characterize various web phenomena.

3.2.4 Visualizations

The phenomenon observed through the datasets on the web observatory occur on a global scale and are diverse in nature. The visualizations help researchers to zoom into specific dimensions of a phenomenon. Visualization libraries such as *d3.js* are frequently imported by data analytic applications on web observatory. Several visualizations are also contributed by the computer scientists engaging with a web observatory in form of **visualization observlets**.

Description- The visualization observlet is schematic definition of the visualizations that can be created using different types of data. It also describes various features a visualization may have such as, zoom-in and change focus to a particular point or region in the visualization.

Requirements- The user can project data and aggregations to multiple visualization observlets to obtain the most appropriate depiction of the results.

Applicability - Visualization code can be integrated with a compatible set of aggregations or dataset.

Results - A complete visualization of a phenomenon is observed through analyzing a given dataset.

4. DISASTER MANAGEMENT STUDIES ON WEB OBSERVATORY

Disaster management is a global problem impacted by various social, cultural and linguistic differences [11]. It needs myriad of analytics for effective preparedness, mitigation, response and recovery processes. Recently, research communities from computer science, environmental sciences, and health sciences have come together to contribute towards effective disaster management [9]. These communities require tools, systems to report on current situation and ways to share infrastructure and resource data, and exchange emergency communications. Today web is acting as a space for various social interactions, real-time updates, warnings and rescue requests during floods and earthquakes. Moreover, data from various departments, such as, meteorology, seismology are published on the web.

Web observatory can bring together diverse group of researchers to collaborate for research in urban and natural disasters to help society respond to these events. Consider two web observatory nodes, *wo1* located in UK and *wo2* located in India (figure 1). These catalogue datasets about "floods" from the respective regions and observlets for data aggregation and visualization. Rob is a climate change expert registered at *wo2*. After December 2015 floods in city of Chennai, India, he wishes to study the change in climate

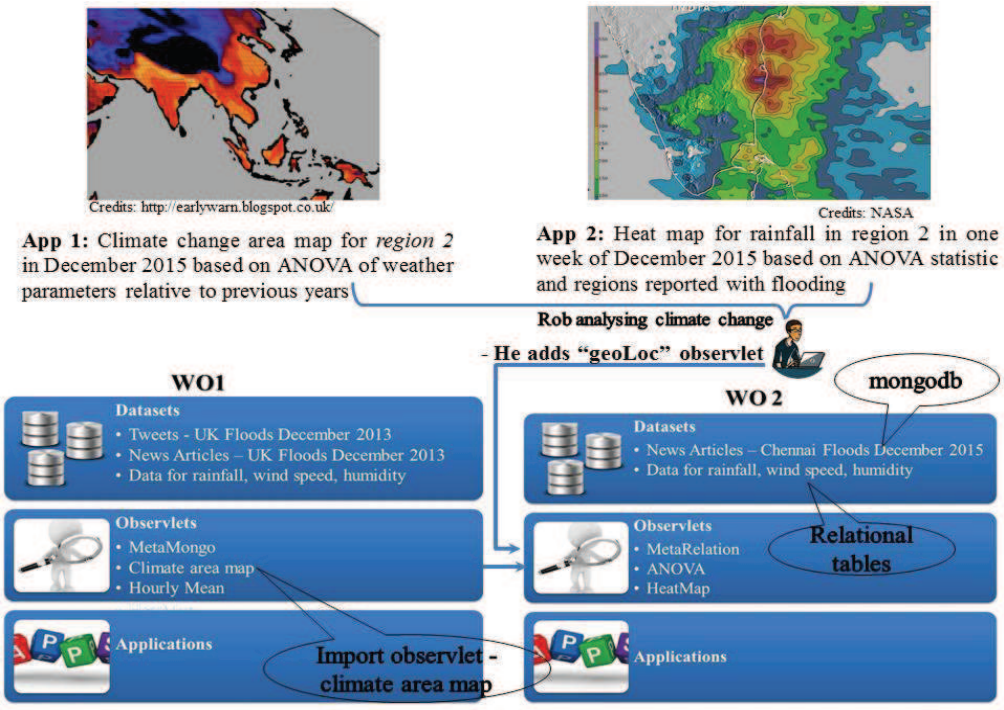

Figure 1: Sharing observlets for analyses of floods in Indian peninsula

in the region over the recent years and analyse the severely flooded areas. For his analysis he decides to use ANOVA statistic.

He logs on *wo2* and finds a *MongoDB* collection of news articles reporting the flooded areas, contributed by a regional university. Second dataset is about the periodic weather monitoring readings in form of SQL tables by the Indian meteorological department. Next, he explores observlets catalogued on *wo2*. He finds an *ANOVA aggregation* observlet, a *heat-map* observlet and also a *metaRelation* observlet (figure 1) with their signatures. To begin with his analysis, he wants to map the affected areas from the news articles (in MongoDB collection) to the corresponding weather data (in relational format). He invokes *metaRelation* observlet to harmonize the news articles datset to equivalent relational dataset. Then he refines meteorological data for the affected areas from the news articles during time window ($1/12/2015 \leq 15/12/2015$). Later, he adds the code as an observlet "geoLoc" for other users to *wo2*. The resultant dataset from mapping is *MetData(area, rainfall, humidity, windspeed, warning, temperature, timestamp)*. He then inputs the attributes of rainfall, humidity and wind speed to the ANOVA observlet for comparative analysis. Then he integrates the results to a heat-map observlet to re-create the picture of December 2015 floods in the Indian peninsula (figure 1).

He then analyses the weather data for last five years about these areas and wants to use a better visualization for the results. For this, he discovers observlets on other web observatory nodes and finds *wo1* has a visualization observlet for a *climate-change area map*. He reads its signature and imports it to *wo2* to integrate it with his analysis. He later studies each area closely and tweets his analyses on annual variation in climate in the region and impact of December floods in these areas. Alice who is a web scientist sees this tweet trending for several days and harvests the stream of tweets and catalogues it to web observatory *wo2*. She studies and summarizes public opinion voiced on-line about disaster preparedness and possible threats of changing climate. She shares her results with local authorities on the web. The authorities decide to draw a plan of action to improve disaster preparedness in the region. They also register on *wo2* to discover if any similar analyses are done on any other web observatory node for another region.

As evident above, the observlets helped non-technical domain expert to quickly analyse the data without writing complex code. They factored the process of building a data analytic application into intermediate analyses which could be shared across web observatory nodes. The user could simply select an observlet and implement on his or her dataset. The user could import observlet from other web observatory node by invoking the observlet API. The user also contributed the *geoLoc* observlet which can be used by other users for similar analyses in the future. Moreover, the user could combine his dataset with visualizations without worrying about the format of the dataset.

5. SUMMARY AND CONCLUSION

Web observatory provides infrastructure support to users for sharing their datasets, tools and analytics to study interdisciplinary nature of various web phenomena. The proposed study considers that the web observatory eco-system can evolve as more users engage with it. It further considers that a number of end-users of web observatory engage with it for data analytic applications. Hence, it proposes design patterns, called *observlets* to support users for developing, sharing and re-using data analytic applications on web observatory. The study factors the intermediate steps of a data

analytic applications into a basic set of observlets - data harmonization, spatio-temporal analysis, statistical aggregation and visualization. It describes how these observlets can be combined for various data analytic applications on the web observatory.

At present we are implementing described observlets as part of the web observatory architecture. As a future work, we wish to support development of data analytic application on web observatory itself using observlets. Further, we plan to test the application development using observlets with multi-disciplinary researchers by organizing *"datathons"*. This can help us to understand shortcomings of observlets and design a framework to enable the end-users define their own observlets. The authors also propose to extend observlets to help end-users understand possible risks and privacy concerns when they share their resources on the web observatory.

6. ACKNOWLEDGEMENTS

We thank the web observatory team at Web Science Institute, University of Southampton, UK for sharing their insights about web observatory architecture and vision.

7. REFERENCES

[1] W3c community group for web observatory. `www.w3.org/community/webobservatory`. Accessed: 2015-11-26.

[2] Web observatory schema. `https://www.w3.org/wiki/WebSchemas/WebObsSchema`. Accessed: 2015-11-26.

[3] Web observatory, university of southampton. `http://web-001.ecs.soton.ac.uk/`. Accessed: 2015-12-11.

[4] I. C. Brown, W. Hall, and L. Harris. Towards a taxonomy for web observatories. In *Proceedings of the 23rd International Conference on World Wide Web Companion*, WWW Companion '14, pages 1067–1072, Republic and Canton of Geneva, Switzerland, 2014. International World Wide Web Conferences Steering Committee.

[5] J. O. Coplien. Software design patterns: Common questions and answers. *The Patterns Handbook: Techniques, Strategies, and Applications. Cambridge University Press, NY*, pages 311–320, 1998.

[6] B. M. Frischmann. *Infrastructure: The social value of shared resources*. Oxford University Press, 2012.

[7] W. Hall and T. Tiropanis. Web evolution and web science. *Computer Networks*, 56(18):3859–3865, 2012.

[8] J. Heer and M. Agrawala. Software design patterns for information visualization. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):853–860, september 2006.

[9] V. Hristidis, S.-C. Chen, T. Li, S. Luis, and Y. Deng. Survey of data management and analysis in disaster situations. *J. Syst. Softw.*, 83(10):1701–1714, Oct. 2010.

[10] I. O. Popov, M. M. C. Schraefel, G. Correndo, W. Hall, and N. Shadbolt. Interacting with the web of data through a web of inter-connected lenses. In *WWW2012 Workshop on Linked Data on the Web, Lyon, France, 16 April*, 2012.

[11] C. Pu and M. Kitsuregawa. Big data and disaster management: a report from the jst/nsf joint workshop. *Georgia Institute of Technology, CERCS*, 2013.

[12] T. Tiropanis, W. Hall, N. Shadbolt, D. De Roure, N. Contractor, and J. Hendler. The web science observatory. *IEEE Intelligent Systems*, (2):100–104, 2013.

[13] T. Tiropanis, X. Wang, R. Tinati, and W. Hall. Building a connected web observatory: architecture and challenges. 2014.