
The Point of View Axis: Varying the Levels of Explanation Within a Generic RDF Data Browsing Environment

Oshani Seneviratne
Tim Berners-Lee

Decentralized Information Group,
MIT Computer Science and Artificial Intelligence Laboratory

OSHANI@CSAIL.MIT.EDU
TIMBL@W3.ORG

1. Introduction

RDF is at the heart of the Semantic Web as it is the primary means by which applications can share data and interoperate. Tabulator is a generic data browser and editor for linked RDF data on the web. It was developed with the motivation of providing a natural and a seamless experience for browsing and editing data (Tim Berners-Lee, 2008). This paper describes how Tabulator can be used to develop custom applications which consume RDF data, in addition to providing a generic data browsing and editing environment. The goal is to make sure that the end-user has the ability to view the RDF data in a visualization that is most suitable given the nature of the data.

The paper is structured as follows. We begin by describing some related work in Section 2. Section 3 gives an overview of the Pane System in Tabulator, and then in Section 4, we give an example where Tabulator can be used to provide varying levels of explanations through *The Justification User Interface*. We then give a short overview of our future work in Section 5, and conclude the paper with a discussion of our results in Section 6.

2. Related Work

Tabulator builds on a long tradition of data presentation applications. There are many popular domain-specific semantic web browsers. Most of them are front-ends tailored to particular triple stores. MSpace (Gibbins et al., 2004) provides a front-end for a database of classical music detailing music pieces and composers. CS AKtive (Shadbolt et al., 2004) provides an interface to query from a database of people with information about their research fields and projects. Disco Hyperdata Browser (Bizer & Gaub, 2007) and OpenLink (Idehan, 2005) can navigate the Semantic Web as an unbound set of data sources. The Inference Web (Mcguinness & Pinheiro, 2004) is a comparable system to the *Justification User Interface*, which provides a Semantic Web based knowledge provenance infrastructure that supports interoperable explanations of sources, assumptions, learned information, and answers.

3. Panes in Tabulator

Tabulator is capable of generic data browsing, but goes one step further by allowing users to exploit the RDF data browsing and editing capabilities to build custom applications through a 'Pane'.

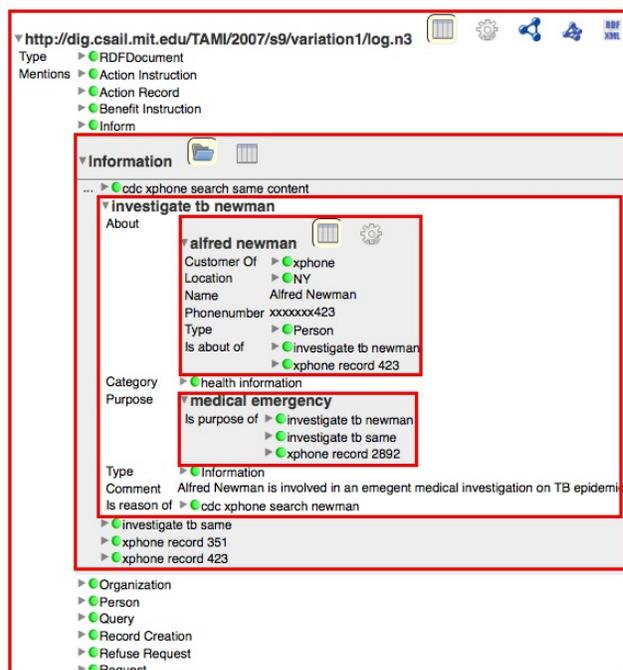


Figure 1. Some of the Expanded Panes in the Data View

A Pane is basically a view for the data fetched in the *Outline* of Tabulator. A system of Views and Panes allows the user to see the data in different perspectives. If the document fetched is an image, the image itself will be displayed in the default pane. If it is a web page, the web page itself will be displayed. However, if an RDF document is fetched, there are more options for visualizing the data. For example, suppose a server is serving a document with MIME type "application/rdf+xml". With a normal web browser which is capable of parsing RDF/XML, one would only be able to see the document in the RDF/XML format only. However with Tabulator, one can see the metadata associated with the document, serialization of the same data in RDF/N3 and the



Figure 2. Different Types of Panes/Views in Tabulator. These are **1:** About the Document View, **2:** Under the Hood View, **3:** Data View, **4:** RDF/N3 View, **5:** RDF/XML View, **6:** Explanations View, **7:** Lawyer’s View, and **8:** Friend-Of-A-Friend View

things described in the document in a class and property explorative manner.

As shown in the Figure 1, Tabulator Outline has the *Data View*, which could be expanded and collapsed to browse data at several different levels. All the expanded panes are shown highlighted red in the figure. At each level of expansion, Tabulator shows the available *Panes* so that one could visualize the data in different views available. As shown in Figure 2, these are some of the icons displayed which signal the user of the available Panes once a document is loaded. Given the nature of the data viewed in a particular Pane, some of these icons may be displayed whereas others may not. For example, if one is looking at FOAF data, the ‘FOAF’ icon will be displayed, but the ‘Explanations View’ or the ‘Lawyer’s View’ icons may not be displayed. If a particular Pane is selected, the corresponding icon will be shown to be depressed as in icon 7.

The functionalities of each of the different types of Panes available in Tabulator are as follows: **About the Document** gives metadata associated with the document such as the *License, Title, Maker, Primary Topic, etc.* **Under the Hood** shows what happened when a thing was looked up on the web, such as the contents of the HTTP headers and the other URIs it referred to. The **Data View** gives the big picture of the things described in the document, i.e. ordered view of the classes and properties. **RDF/N3 View** gives Notation 3 (Tim Berners-Lee and Dan Connolly and Lalana Kagal and Jim Hendler and Yosi Scharf, 2008) serialization, and the **RDF/XML View** gives the XML serialization of the RDF data. The **Explanations View** allows the user to explore the justification proof trace from the AIR reasoner in a step by step manner, whereas the **Lawyer’s View** gives a summary of the scenario and the justification behind the reasoning for the same data. The **Friend-Of-A-Friend View** gives an embryonic open social network view based on a user’s FOAF data.

4. An Example Application

This application was built on Tabulator using the modular ‘*Pane system*’, and was intended to be used in accountable data systems. We have named it the ‘*Justification User Interface*’, because it provides the ‘justifications’ for a legal reasoning given the facts which are based on a log of a person’s actions, and a set of policies or rules. A reasoner built independent from the Tabulator infers whether or not the actions were appropriate, and derives a proof tree based on the facts and the rules (Lalana Kagal and Chris Hanson and

Daniel Weitzner, 2008). The challenge we describe here is that of explaining the inference and presenting the proof in essence, but in a way understandable by people with completely different backgrounds and needs.

There is a Firefox sidebar extension which will invoke the reasoner given the URIs of the relevant log file and the policy file. One could also invoke the reasoner by appending the proper parameters to the URI of the reasoner. The explanations are produced in the form of proof trees in RDF/N3, and indicates whether the given facts comply with the policy specified. It also gives the premises on which this decision was based. A part of a sample proof tree in the raw format is shown in Figure 3. The data visualization in this default outline view could be a bit overwhelming to read and understand, as one might only be interested in the final outcome of the policy reasoning. The Justification User Interface solves this problem by providing two alternative views as described below.

4.1 The Explanations View

In this Pane, the final policy decision from reasoning is filtered, and displayed first as in Figure 4. A non-compliant event will be highlighted in red, whereas a compliant event will be highlighted in green. Clicking on the “Why?” button which is located underneath this conclusion will initiate the traversal of the proof tree. Figure 5 shows a sample proof several steps down. The algorithm used here extracts the most appropriate reason for the final outcome based on the *air:descriptions* given. The top box in this view gives the description of the rule with the necessary variable bindings that generated the assertion, whereas the box on the bottom gives the premises or the matched graph of that rule. The “More Information” button provides a way for the user to step through the proof starting from the most relevant information working back to the top of the proof tree. The goal is to select the most appropriate high-level description from the proof tree.

4.2 The Lawyer’s View

From the perspective of a lawyer, the interesting part of the result from any legal reasoning is the finding of non-compliance or the compliance with the given policy and the associated facts (Waterman, 2008). Therefore, instead of enabling the user to explore the proof tree, this pane gives a summarized view, which should be very familiar for a legal professional. This includes the Issue, Rules, Facts, Analysis and the Conclusion as shown in Figure 6. **Issue** is the main problem investigated. In a typical scenario, this is described in the *transaction log*. The **Rule** is the policy with which the issue in the scenario is checked against. If the policy file has an *rdfs:comment* which describes what the intended rule of the policy is, that is taken to be the content for this section. **Facts** are the ground truths presented in the log file. However, only the facts relevant to the scenario are presented, which are basically *air:descriptions* of the outer *air:rules* in the proof tree (i.e. intermediate rules that were fired in the

http://mr-burns.w3.org/cgi-bin/server CGI.py?logFile=http://dig.csail.mit.edu/TAMI/2007/s9/variation1/log.n3&rulesFile=http://dig.csail.mit.edu/TAMI/2007/s9/variation1/demo-policy

Bettyrejectsbobsreq MA Disability Rule 1 MA Disability Rule 2	non compliant with justification description	MA Disability Discrimination Policy premise Bob Same's request, Bobsrequest, was refused because of xphone record 2892	antecedent expr sub expr	MA Disability Rule 1									
				<table border="1"> <tr> <td>Bettyrejectsbobsreq</td> <td>reason receiver</td> <td>xph</td> </tr> <tr> <td></td> <td>reply to type</td> <td>cus</td> </tr> <tr> <td>customer351</td> <td>name</td> <td>Ref Bot</td> </tr> </table>	Bettyrejectsbobsreq	reason receiver	xph		reply to type	cus	customer351	name	Ref Bot
Bettyrejectsbobsreq	reason receiver	xph											
	reply to type	cus											
customer351	name	Ref Bot											
MA Disability Rule 3	description	rule name The requester, Bob Same, resides in MA and is covered by the MA Disability Discrimination Law	type MA Disability Rule 1	And justification									
	justification	antecedent expr sub expr		<table border="1"> <tr> <td>Bobsrequest</td> <td>instruction type</td> <td>bs re</td> </tr> <tr> <td>bs request content</td> <td>intended beneficiary type</td> <td>Requ</td> </tr> <tr> <td>customer351</td> <td>location</td> <td>custc Bene MA</td> </tr> </table>	Bobsrequest	instruction type	bs re	bs request content	intended beneficiary type	Requ	customer351	location	custc Bene MA
Bobsrequest	instruction type	bs re											
bs request content	intended beneficiary type	Requ											
customer351	location	custc Bene MA											

Figure 3. Explanation in the Default Outline View

http://mr-burns.w3.org/cgi-bin/server CGI.py?logFile=http://dig.csail.mit.edu/TAMI/2007/s9/variation1/log.n3&rulesFile=http://dig.csail.mit.edu/TAMI/2007

/s9/variation1/demo-policy.n3

Bettyrejectsbobsreq is non compliant with MA Disability Discrimination Policy

Why?

Figure 4. Final Conclusion in the Explanations View

http://mr-burns.w3.org/cgi-bin/server CGI.py?logFile=http://dig.csail.mit.edu/TAMI/2007/s9/variation1/log.n3&rulesFile=http://dig.csail.mit.edu/TAMI/2007

/s9/variation1/demo-policy.n3

Bettyrejectsbobsreq is non compliant with MA Disability Discrimination Policy

More Information Start Over

Bobsrequest is denied based on health information contained in xphone record 2892. Under the MA Disability Discrimination Law it is illegal to use health information to deny a service request.

The requester, Bob Same, resides in MA and is covered by the MA Disability Discrimination Law

Bob Same's request, Bobsrequest, was refused because of xphone record 2892

Premises:

Bettyrejectsbobsreq	reason receiver	xphone record 2892
	reply to type	customer351
customer351	name	Bobsrequest
		Refuse Request
		Bob Same

Figure 5. Explanations after clicking the "why" button

▼ <http://mr-burns.w3.org/cgi-bin/server.cgi.py?logFile=http://dig.csail.mit.edu/TAMI/2007/s9/variation1/log.n3&rulesFile=http://dig.csail.mit.edu/TAMI/2007/s9/variation1/demo-policy.n3>

▼ Issue:

Whether the transactions in [log](#) comply with [MA Disability Discrimination Policy](#)

▼ Rule:

To be compliant, Denial of Service rule of MA Disability Discrimination Law requires reason of an event to be other than disability

▼ Facts:

- Bob Same's request, [Bobsrequest](#), was refused because of [xphone record 2892](#)
- The requester, Bob Same, resides in MA and is covered by the MA Disability Discrimination Law

▼ Analysis:

[Bobsrequest](#) is denied based on health information contained in [xphone record 2892](#). Under the MA Disability Discrimination Law it is illegal to use health information to deny a service request.

▼ Conclusion:

The transaction - [Bettyrejectsbobsreq](#) is non compliant with [MA Disability Discrimination Policy](#)

Figure 6. Lawyer's View

reasoning). **Analysis** is the summary of how the facts relate to the rules given in the policy. This usually comprises of the the *air:description* of the final *air:rule* in the proof tree. Finally, the **Conclusion** is the final outcome of the policy reasoning which specifies whether the log is compliant with the policy or not.

5. Future Work

The Justification User Interface is still undergoing development to make it more intuitive to end-users. At the moment it is only using some specific things in the AIR ontology to filter out the policy decision and display the relevant panes in the outline view. But we intend to make it more flexible by allowing other types of filters as well. We also wish to incorporate transaction log specific queries. For instance, it is useful to know how many violations there are in a given set of transaction log entries, how many violations are due to some particular policy, which policies would a set of transactions violate, and so on. It would be nice to have a what-if-analysis with the Justification User Interface as well: i.e. it would be interesting to know the consequences of changing one rule in the policy and see how many more or fewer violations would have occurred in a given set of transactions in the log.

Also, in addition to providing a retrospective approach for policy reasoning with the use of the Pane System as described in this paper, we also intend to integrate policy aware capabilities into Tabulator. It will then be able to do reasoning in real-time as the actions are performed. We hope that this will make the browser more intelligent and advice users against potential policy violations by keeping track of the provenance information for the data, and the user interactions on the browser.

6. Conclusion

Tabulator provides a browsing environment for RDF data where not only can the user explore the data space, but can also vary the level of explanation which they are given at any point. The Justification User Interface with the integrated Explanations View and the Lawyer's View have demonstrated this fact. Although this particular application is centered around the legal reasoning domain, the same principle could be applied to develop other such applications. It has allowed a new dimension to browsing linked data and the user can easily slip into a more sophisticated level wherever in the graph of data. Therefore, we believe we have demonstrated that it is possible to provide very different levels of user-friendliness to different types of users, and these specialized applications can be integrated with a generic data browsing environment such as the Tabulator very easily.

7. Acknowledgements

Research described here was supported by from the National Science Foundation Cybertrust Grant (award 04281) and IARPA (award FA8750-07-2-0031). Also, many thanks to Daniel Weitzner, Lalana Kagal and K. Krasnow Waterman for their valuable feedback on the project, UI design advice and comments on the paper.

References

- Bizer, C., & Gaub, T. (2007). Disco. <http://www4.wiwiwss.fu-berlin.de/bizer/ng4j/disco/>.
- Gibbins, N., Harris, S., & mc schraefel (2004). mSpace: Applying mspace interfaces to the Semantic Web. *World Wide Web Conference, 2004*.
- Idehan, K. (2005). Open Link Software's Data Browser. <http://ode.openlinksw.com/>.

- Lalana Kagal and Chris Hanson and Daniel Weitzner (2008). Using Dependency Tracking to Provide Explanations for Policy Management. *IEEE Policy*.
- Mcguinness, D. L., & Pinheiro, P. (2004). Explaining Answers from the Semantic Web: the Inference Web Approach. *Web Semantics: Science, Services and Agents on the World Wide Web, 1*, 397–413.
- Shadbolt, N., Gibbins, N., Glaser, H., & Harris, S. (2004). CS AKTive Space, or How We Learned to Stop Worrying and Love the Semantic Web. *IEEE INTELLIGENT SYSTEMS, 2004*.
- Tim Berners-Lee (2008). Tabulator Redux: Browning and Writing Linked Data . *Linked Data on the Web Workshop at WWW08*.
- Tim Berners-Lee and Dan Connolly and Lalana Kagal and Jim Hendler and Yosi Scharf (2008). N3Logic: A Logical Framework for the World Wide Web. *Journal of Theory and Practice of Logic Programming (TPLP), Special Issue on Logic Programming and the Web*.
- Waterman, K. (2008). Accountability Appliances: What Lawyers Expect to See. <http://dig.csail.mit.edu/breadcrumbs/node/225>.