Exploring Heterogeneous Datasets from Different Searcher Perspectives

Max L. Wilson[1] and m.c. schraefel[2]

[1]Future Interaction Technologies Lab
Department of Computer Science
Swansea University, UK
m.l.wilson@swansea.ac.uk
[2]Intelligence, Agents, and Multimedia Research Group
School of Electronics and Computer Science
Southampton University, UK
mc+swui@ecs.soton.ac.uk

Abstract. This paper demonstrates how a recently developed analytical usability evaluation method, the Sii framework, can by used to inspect semantic search interfaces for how they support people working with the large heterogeneous datasets afforded by Linked Data. To enrich the discussion, an analysis of the Tabulator browser for Linked Data is presented and discussed in terms of the workshop's case study surrounding archivists. The analysis shows that while the Tabulator provides some strong support for sense-making, it would struggle to support such archivists in first defining their needs. In analyzing the Tabulator from the perspectives of the archivists, this paper demonstrates how the new Sii method can provide rigor and reason to the assessment of future design decisions made for Semantic Web user interactions.

1. Introduction

While semantically Linked Data [1] can enhance large diverse, unorganized, and heterogeneous datasets, the unique affordances also challenge our assumptions about how we access information [2]. As the links between data can be numerous, endless, and of any granularity, the assumptions about carefully structured classifications, for example, breakdown. Similarly, while web searches are typically for web pages, it is not clear whether searching at the data level should return any object [3], specific types of objects [4], object relationships [3, 5], portions of RDF [6], entire ontologies [7, 8], and so on. Further, as the work on semantically Linked Data has separated the data from presentation, we are able to represent the data however we like, whether decided by interface designers or end-users [3]. The flipside, however, is that someone, either the interface designer or the end user, has to decide how to represent the data. In short, the freedom enabled semantically organized datasets, has in turn broadened our options and increased the number of decisions that designers, or end users, have to make. Recent work has shown, however, that increasing numbers of options can make us feel less confident in our decisions, and less happy with our

results [9], rather than making us feel empowered. What effect, then, does this have on confidence during search interface design, given that we now have more freedom to design? The new method discussed in this paper can support designers in creating carefully reasoned search interface designs.

The case study at the focus of this year's workshop[1] surrounds a semantically annotated archive of heterogeneous archived files. Such an archive contains many file-types and an unlimited number of information types, such as reports, emails, notes, scanned items, policy documents, procedural documents, memos, analyses, and multimedia (images, videos, audio, etc). Archivists working with such a dataset may be looking for many different types of results, may know what they are looking for, may be looking at relationships between results, and may be learning about a certain event from the content of multiple documents. Consequently, variation in the dataset, the freedom of metadata, and the freedom of representation, make it a grand challenge to design an effective interface for accessing and working with the data.

One challenge for archivists working in such a scenario is in searching, finding, exploring, and learning about the dataset. So how do we go about designing a search interface that works for users in this scenario? Recently, we presented a framework, called Sii[2], for analyzing search interface designs for how they support different styles of search and different searcher profiles [10, 11]. The analysis can be applied to established working systems and low-level prototypes alike. Consequently, we can analyze search interfaces at design time, and learn from prior art, to ask 'Is this new design going to support the right kinds of search for our users?' While it is not unusual to work from a user-centred design approach, this analysis method provides a representation of support simultaneously for different searching profiles, and can demonstrate, therefore, which of several design options will provide the most appropriate support for the intended users. User-cantered design, in this case, helps inform the profiles that users fit into.

In the remainder of this paper, after presenting some related work, an analysis of the Tabulator browser [3] is presented and discussed. The Tabulator provides an interface that allows users to browse linked relationships in a heterogeneous dataset, collect results according to certain relationships, and present them in several alternative visualizations. Following this analysis, the case study scenario of archivists is revisited to consider future design ideas.

2. The Tabulator

In line with the method required for using the Sii framework, detailed further in the next section, the discussion of the Tabulator interface below is presented as a series of component parts that each contributes to searching for information. There are 8 main search features of the Tabulator interface, and two less obvious features, which have been highlighted in Figures 1 and 2, and inputted into the Sii website for analysis. The foremost feature of the interface is the tree-based explorer (#1 in Figure 1). Using this

[1] http://swui.webscience.org/SWUI2009/archival-casestudy/

[2] http://mspace.fm/sii

explorer, the user can expand any one of the root nodes initially listed to see all of the attribute types associated with it, and one or more of their values (long lists are cut off and replaced with a 'more' button). The user can continue to navigate in this way as long as the values reached by expansion have further attributes to expand. As well as exploring in this way to find specific items of information, the user can also define a pattern and request, using the 'Find All' button, to see all such values. To assert such a pattern, the user can select the attributes and/or values in the explorer, so that they are highlighted in green. Alt-select allows the user to select multiple attributes or values for more complicated examples, as shown in Figure 1.

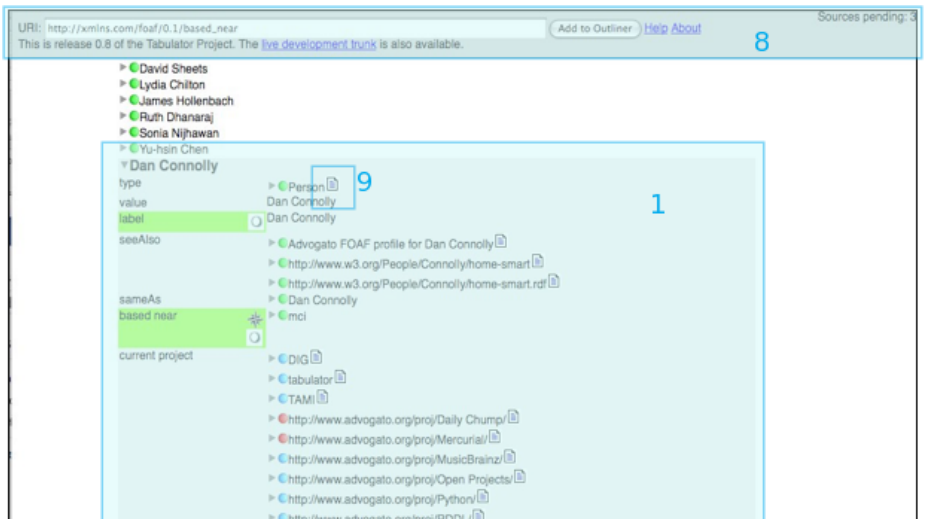

Figure 1: The top part of the Tabulator, which contains a tree-based browser, can expand endlessly downwards depending on the number of expanded nodes.

For example, a user might expand a 'developer team' node to see all of its attributes, such as its office location and its developers, and expand the details of one team member, and highlight: the name, date of birth, current living location and picture. Pressing 'Find All' will find these details for all the members of the team and pass them to the analysis features, described below. If, however, there is a team manager with these same details, he will also be found, as the user did not highlight 'developer' as a constraint. The user may add this constraint and select 'Find All' to pass the new findings to the analysis modules, as a new result set. Further, the user may decide that they want to see the whole team, regardless if they are missing either their date of birth, or home town, and may mark them as optional with the radio button seen within the green highlight.

There are 5 analysis modules available (#2-#6 in Figure 2), that make up 5 separate features: the table view, the map view, the calendar view, the timeline view, and the SPARQL code view, which allows the user to directly edit a query in the SPARQL language used to retrieve from the Semantic Web. The 'Find All' button passes sets of results to these views to be displayed. In the team example above, the table view would show four columns, with the team members' names, dates of birth, locations

and pictures. As the query contains a location field, these can be displayed on the map view. Multiple result sets can be shown on the map view at once if required. Similarly, as the team member query above has a date field, the user can show their dates of birth in either the calendar or the timeline view, where result sets can be combined if required. The SPARQL viewer provides a query by example interface, allowing the user to edit the queries that produced existing result sets, and use them to create new queries, and thus new results sets.

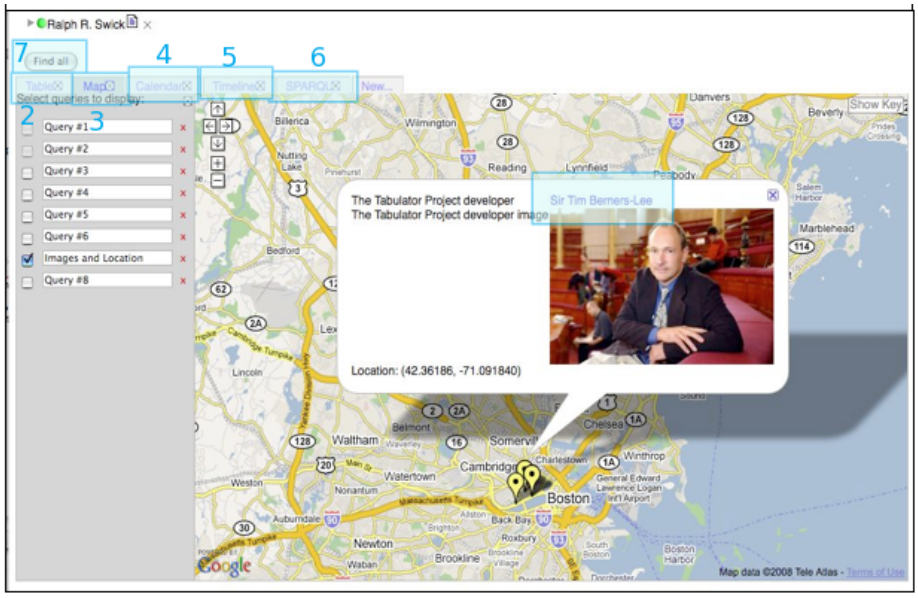

Figure 2: The bottom part of the Tabulator Browser contains five analysis modules, and its placement is controlled by the height of the tree browser above (Figure 1). The bottom part of the tree browser is visible just at the top.

The first unobvious feature of the interface is, in fact, the 'Find All' button (#7 in Figure 2), which serves to create results sets from the patterns defined in the explorer, and passes them to the analysis modules. This has been identified as a separate function as it is not required to explore or to analyze, but is required to move from exploring to analyzing.

Another noticeable feature of the interface is the URI bar that is permanently visible at the top of the screen (#8 in Figure 1). Primarily, the URI bar is used to display the complete URI of the last item selected within the Explorer. This allows the user to both check the provenance of an item selected, and copy and save it if necessary. The URI Bar may also be used to add certain parts of the Semantic Web to the browser, as a new root node on the interface. This can be achieved by pasting a URI into the URI Bar and pressing 'Add to Outliner', where Outliner is the name used for the explorer.

The penultimate feature to identify in the Tabulator is the RDF Popup button (#9 in Figure 1). This allows the user to view the original source data, in the RDF format of

something found in the explorer. The final feature of the Tabulator to identify is that any item found in the analysis modules may be loaded as a new starting node in the explorer, by double clicking on it (#10). So in the team member example, the user may wish to start exploring again from one particular member, or one particular location or date.

3. Analysis

The Sii method was applied using the online website[3] by the author of the framework, who is also the lead author of this paper. The method is designed for solo or small group use, similar to Heuristic Evaluations [12], where the analyses can be strengthened by either experience or the corroboration between multiple evaluators. The process involves 1) identifying the features that contribute to a search interface, and 2) assessing how many moves, or actions, it takes to use each of them to achieve 32 known searching tactics [13, 14]. Zero is used when a feature does not support a tactic. The analysis produces three graphs, shown below, that show 1) the total support for each feature, 2) the total support for each tactic, and then 3) an average support provided for each of 16 different searching profiles [15], where Sii maps certain tactics to the needs of each searcher profile. While it is not possible to describe all of these tactics and searcher profiles in detail here, the Sii method is fully described, along with the tactic and each user type, in previous publications [10, 11]. In each graph, though, taller bars or higher peaks represent greater support. A table of the 16 searcher profiles is shown in Table 1 next to the profile graph (Figure 5).

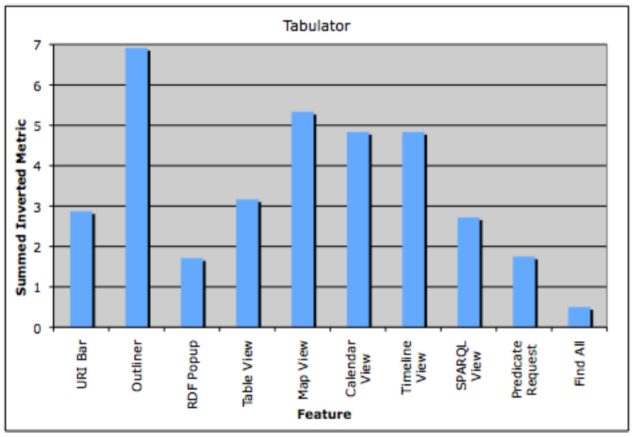

Figure 3: shows the total support for search provided by each feature of the Tabulator browser, where taller bars represent greater support.

Figure 3 alone both confirms some expectations and reveals some interesting insights. First, it is not surprising, perhaps, that the Explorer provides the broadest

[3] http://mspace.fm/sii/project.php?pid=00000015

amount of support for search, compared to all the other features within the Tabulator. Second, it is probably not surprising that the different visualizations at the bottom of the interface make up the subsequently tall bars within the graph, as these provide the means to analyze the results further.

One perhaps surprising result is that, while the table view may provide the most often used representation for analysis, the map, calendar, and timeline views provide more support for search. This prompts the question, which has probably not been asked as of yet: what about their design is different to the table view? Consulting the inputted data online in more detail reveals that compared to the table view, the other views are interactive. With the map, for example, the user is able to zoom in on specific groups of results, thus reducing the number of results found. There is currently no means within the table view to manipulate the results and so the subsequent question is, therefore, how could the table view be altered to permit further investigation.

Another perhaps surprising result is the support for search provided by the URI bar that is persistent at the top of the screen. Investigating the inputted data online reveals that, as this persistently shows the URI of the last item clicked on, that it can be used for a number of monitoring tactics. As it can also be used as an input to control the main explorer, the URI Bar can also be used for tactics such as expanding, narrowing, and restarting ones search.

Finally, although it appears only to serve as a means to fill the analysis views below, the 'Find All' button, in of itself, supports the tactic of recording ones search. If it merely populated the views, rather than creating query objects that can be compared or combined, then it would not support any particular tactic at all.

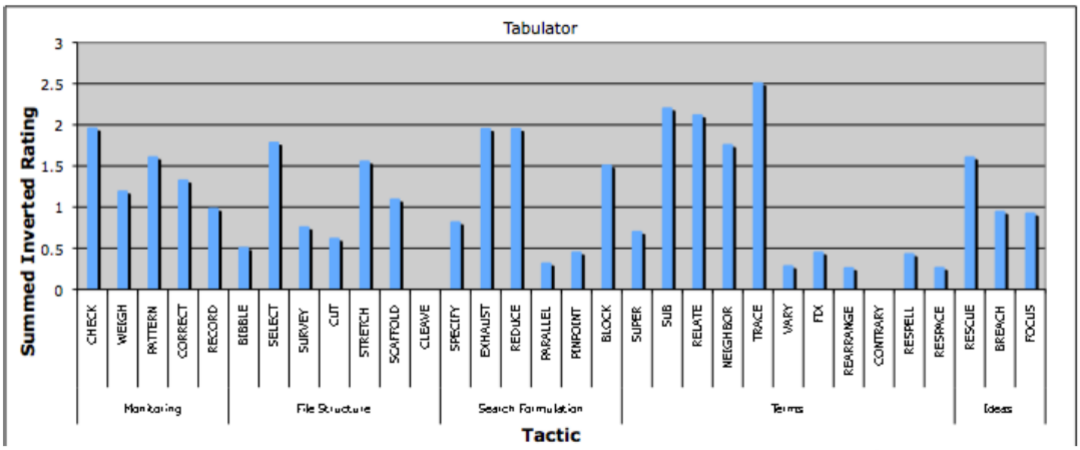

Figure 4: shows the total support provided for each of 32 known search tactics provided by the design of the tabulator browser, where taller bars represent stronger support.

From Figure 4 we can see that there are two tactics that are entirely unsupported; although results from other analyses[4] show that these are often the hardest to support.

[4] Other analyses are available on the Sii website (http://mspace.fm/sii

CONTRARY, for example, is to find the opposite of something, which is inherently different from showing everything but something (BLOCK). While TRACE, consulting results to find new search constraints, is often well supported, the tabulator supports this better than actually defining or altering ones search constraints. Consulting the input table reveals that this is due to the many ways of visualizing results, but that the only way to specify ones searches is through the single explorer interface.

One key tactic is to SPECIFY one's constraints, and we can see that it has much more support, compared to some other tactics relating to refining search constraints. This supports the some of the criticisms of the Tabulator, stating that it can be hard for a user to specify what they would like to find with the Tabulator interface.

It is also clear in the graph, that the first half of the term tactics receive much more support than those in the latter half. This shows that it is easier to expand and narrow upon ones search than it is to specify variations within them. That is, a user is restricted to either specifying a specific value of a particular attribute, or that they would like any value of a particular attribute. It is difficult using the browse-then-analyze model of the Tabulator to explore variations in either phase, as the results of a user's actions are so distantly removed from the actions themselves.

Figure 5 is designed to convey how different types of searchers are supported. The 16 searcher profiles are made up of four dimensions of two options, as displayed in Table 1. Like the pattern created by the pairs of options in the table, Figure 5, shown in Figure 5, also has patterns. These four dimensions lead to four interrogation angles, discussed in turn.

Method of Search. The first and the second half of the graph, for example, are almost identical, indicating that the Tabulator is just as supportive for people who are scanning or searching, where the latter is characterized by searching for a known item. The second half of the graph is slightly higher, however, representing slightly better support for those who are searching.

Goal of Search. There is also a clear pattern across the different quarters of the graph, where the odd quarters are noticeably higher than the even quarters. Unlike many browsers, this means that users who are intending to learn more generally about a topic are better supported than those who are specifically aiming to retrieve a certain piece of information.

Mode of Search. The most prominent difference seen is between the odd and even eighths of the graph. This drop indicates that it is significantly harder to use for people who can specify exactly what they need, than it is for people who are likely to recognize the information they need when they see it. This emphasizes one of the results shown in Figure 4 and matches the opinion held by some that it is actually hard to use the Tabulator to find specific information, and that users are almost entirely dependant on what is presented to them as they explore. Ultimately, the user is required to begin at varying starting points, and to seek the information they can only navigate through links and associations. Most existing web browsers provide keyword search paradigms to search for and jump directly to the information they need, and allow navigation from there.

Resource Being Sought. The final pattern seen is between the odd and even sixteenths of the graph, which are slightly higher for the latter part of each pair. This indicates that it is slightly easier to find metadata than it is to find particular

information objects. This is perhaps not surprising for a browser of the data-web, which promotes exploration of inter-object associations.

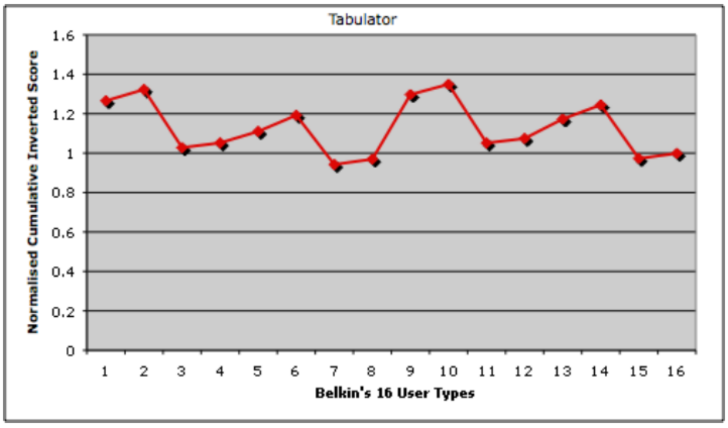

Figure 5: showing the average support provided for each of 16 searcher profiles by the design of the Tabulator browser, where peaks represent stronger support.

Table 1: 16 Searcher Profiles, defined by Belkin et al. [15]

ISS	Method	Goal	Mode	Resource
1	Scan	Learn	Recognize	Information
2	Scan	Learn	Recognize	Meta-Information
3	Scan	Learn	Specify	Information
4	Scan	Learn	Specify	Meta-Information
5	Scan	Select	Recognize	Information
6	Scan	Select	Recognize	Meta-Information
7	Scan	Select	Specify	Information
8	Scan	Select	Specify	Meta-Information
9	Search	Learn	Recognize	Information
10	Search	Learn	Recognize	Meta-Information
11	Search	Learn	Specify	Information
12	Search	Learn	Specify	Meta-Information
13	Search	Select	Recognize	Information
14	Search	Select	Recognize	Meta-Information
15	Search	Select	Specify	Information
16	Search	Select	Specify	Meta-Information

4. Discussion

It is in the third graph that we can most easily consider how a design is fit for the designated users. In starting with prior-art we can see that the kind of support provided by the Tabulator is useful for users who are recognizing information and trying to learn. The support is equally balanced, almost, for those who are looking for potential items (Scanning) or for people who are looking for items that they know exist (Searching). But the support is notably less for people who can easily Specify

what they need. This lack of support is controlled mainly by the tree-based browser that requires users to sequentially expand relationships from a certain starting point, and navigate towards their desired result. The final element to note from the graph is the slightly increased support for metadata, since the main method of browsing deals almost entirely with metadata.

In returning to the workshop's case study, however, the focus is from starting with a design challenge and potential users, and trying to create a new search interface that supports the access to and use of a heterogeneous archive. It is important, therefore, to start with the tasks that will be important to archivists in this scenario. Here we discuss two common example task profiles for a large heterogeneous archive: quickly retrieving a known document from within the archive and making sense of an event, for example, across multiple documents in the archive. For the first of these, it is particularly important to support the 15th and 16th searcher profiles, where the user is able to specify a known item to select. In particular, if the aim is simply to grab a known report, for example, then the 16th is the most important.

For the latter sense-making scenario, as stated in the workshop's case study, recall is particularly important, so that the archivists can make use of all available information. Consequently, the emphasis of an interface should be for Learning (Goal), rather than trying to Select specific pieces of information. Further, the emphasis should also lie on Scanning as a method, rather than Searching for a known item. The Mode, however, should vary, in that users should find it easy to Specify their initial needs, and recognize other important relationships. Finally, in such a metadata rich environment, where the semantic annotations rife, support for metadata maybe just as important, if not more important, than the documents themselves. Consequently, the main group of searcher profiles to be supported range from profiles 1 to 4.

With this analysis in mind, we can review the support that the Tabulator would provide in these identified searcher profiles. Clearly, half of the sense-making searcher profiles require the ability to specify easily what, amongst all the data, is being sought. Similarly, the 16th profile also depends on being able to specify quickly what is being sought. For these users, the Tabulator does not provide strong support. The interaction provided by the Tabulator, however, does provide strong support for exploring from a given point. Profiles 1 and 2, for example, are some of the most supported by the tabulator, and could be very useful for exploring, manipulating, and analyzing sets of returned results. To support archivists, however, different functionality is required to reach the starting points before beginning such an analysis.

In part of working out how to support these users, it is useful to know which tactics are valuable to people who can specify their needs (available in a technical report [16]). These tactics include: CUT, SCAFFOLD, CLEAVE, SPECIFY, EXHAUST, and the latter Term Tactics in Figure 4. In particular, these tactics involve being able to have take actions that have a large effect on reducing the number of results that are being returned. From other analyses of alternative search systems[5], faceted metadata can provide quite effective support for specifying multiple constraints. mSpace [4] and /facet [5] are two faceted browsers that can make use of semantic relationships to produce facets of metadata. Further, if supported by numeric indicators of how many

[5] Other analyses are available online at http://mspace.fm/sii

results are associated with items in facets [2], facets can permit users to CUT down the results dramatically, reorder results, etc. It is hard, however, to use term tactics that involve playing with, and analyzing the effect of, varying search terminology. For this, keyword search can be effective, if implemented effectively to support refinements such as spelling corrections and query expansion techniques. In particular, semantics may be very helpful for producing effective expansion recommendations.

5. Conclusions

In this paper, we have made two contributions. 1) We have demonstrated how a recent analytical framework, designed to analyze information seeking interfaces, can be used to begin addressing the interactive requirements for rich and varied heterogeneous datasets that are afforded by semantic annotations. 2) To provide an example, the Tabulator browser, developed by the team who envisioned the Semantic Web, has been analyzed for how it would support the needs of archivists in the case study scenario for the workshop. It is clear that the Tabulator would partially support the sense-making of large heterogeneous archives, but would struggle to support archivists in specifying areas and regions of the dataset to analyze. Both the reference to other prior art, and the discussion of particular tactics that support specifying during search, can inform how we should try to increase support for archivists.

In future work, like with other analyses available on the Sii website, we can strengthen our understanding of Semantic Web user interfaces by analysing other existing interfaces and comparing them. One of the key values of the Sii framework, however, is in being able to model and analyze the support provided by new search design ideas. The freedom of the Semantic Web means that there are many new searching interactions that could be generated. By adding these ideas to a Sii analysis, we can test to see the tactics and searcher profiles that they support. Such an analysis of designs is further supported when we can analyze designs from the particular searcher profiles we are trying to support. The Sii method adds rigor and structure to the early design of search interfaces, encouraging us to make more carefully reasoned decisions as we explore the many new opportunities that semantically Linked Data is affording.

References

1. Bizer, C., T. Heath, K. Idehen, and T. Berners-Lee, *Linked data on the web (LDOW2008)*, in *Proceeding of the 17th international conference on World Wide Web*. 2008, ACM: Beijing, China. p. 1265-1266.
2. Wilson, M.L. and m.c. schraefel. *mSpace: what do numbers and totals mean in a flexible semantic browser*. in *Proceedings of the 3rd International Semantic Web User Interaction Workshop (SWUI'06)*. 2006. Athens, GA, USA.

3. Berners-Lee, T., Y. Chen, L. Chilton, D. Connolly, R. Dhanaraj, et al. *Tabulator: Exploring and Analyzing linked data on the Semantic Web*. in *Proceedings of the 3rd Int. Semantic Web User Interaction Workshop, Athens, USA*. 2006.
4. schraefel, m.c., D.A. Smith, A. Owens, A. Russell, C. Harris, et al. *The evolving mSpace platform: leveraging the semantic web on the trail of the memex*. in *HYPERTEXT ?05: Proceedings of the sixteenth ACM conference on Hypertext and hypermedia*. 2005. New York, NY, USA: ACM Press.
5. Hildebreand, M., J.v. Ossenbruggen, and L. Hardman. */facet: a browser for heterogeneous semantic web repositories*. in *Proceedings of the 5th International Conference on the Semantic Web (ISWC'06)*. 2006. Athens, GA, USA.
6. Ding, L., T. Finin, A. Joshi, R. Pan, R. Cost, et al. *Swoogle: A search and metadata engine for the semantic web*. in *CIKM'04*. 2004: ACM New York, NY, USA.
7. Alani, H. and C. Brewster. *Ontology ranking based on the analysis of concept structures*. in *Proceedings of the 3rd international conference on Knowledge capture*. 2005: ACM New York, NY, USA.
8. Gao, M., C. Liu, and F. Chen. *An ontology search engine based on semantic analysis*. in *ICITA'05. Third International Conference on Information Technology and Applications, 2005*. 2005: IEEE.
9. Schwartz, B., *The Paradox of Choice: Why More Is Less*. 2005: Harper Perennial.
10. Wilson, M.L. and m.c. schraefel. *Sii: the lightweight analytical search interface inspector*. in *JCDL09 Workshop: Lightweight User-Friendly Evaluation Knowledge for Digital Librarians*. 2009.
11. Wilson, M.L., m.c. schraefel, and R.W. White, *Evaluating Advanced Search Interfaces using Established Information-Seeking Models*. Journal of the American Society for Information Science and Technology, 2009. **60**(7): p. 1407-1422.
12. Nielsen, J., *Heuristic evaluation*, in *Usability inspection methods*, J. Nielsen and R.L. Mack, Editors. 1994, John Wiley & Sons, Inc. p. 25-62.
13. Bates, M.J., *Information search tactics*. Journal of the American Society for Information Science, 1979. **30**(4): p. 205-214.
14. Bates, M.J., *Idea tactics*. Journal of the American Society for Information Science, 1979. **30**(5): p. 280-289.
15. Belkin, N.J., P.G. Marchetti, and C. Cool, *Braque: design of an interface to support user interaction in information retrieval*. Information Processing and Management, 1993. **29**(3): p. 325-344.
16. Wilson, M.L., *A Transfer Report on the Development of a Framework to Evaluate Search Interfaces for their Support of Different User Types and Search Tactics*. 2008, School of Electronics and Computer Science, University of Southampton.