

VISUALISING LARGE SEMANTIC DATASETS

A generic, scalable and aesthetic approach

SUVODEEP MAZUMDAR

Submitted in partial fulfilment of the requirements
for the degree of Doctor of Philosophy

Department of Computer Science
The University of Sheffield

November, 2013

Supervisor: Prof. Fabio Ciravegna
Co-Supervisor: Prof. Daniela Petrelli

ABSTRACT

This thesis aims at addressing a major issue in Semantic Web and organisational Knowledge Management: consuming large scale semantic data in a generic, scalable and pleasing manner. It proposes two solutions by de-constructing the issue into two sub problems: how can large semantic result sets be presented to users; and how can large semantic datasets be explored and queried. The first proposed solution is a dashboard-based multi-visualisation approach to present simultaneous views over different facets of the data. Challenges imposed by existing technology infrastructure resulted in the development of a set of design guidelines. These guidelines and lessons learnt from the development of the approach is the first contribution of this thesis.

The next stage of research initiated with the formulation of design principles from aesthetic design, Visual Analytics and Semantic Web principles derived from the literature. These principles provide guidelines to developers for building generic visualisation solutions for large scale semantic data and constitute the next contribution of the thesis. The second proposed solution is an interactive node-link visualisation approach that presents semantic concepts and their relations enriched with statistics of the underlying data. This solution was developed with an explicit attention to the proposed design principles.

The two solutions exploit basic rules and templates to translate low level user interactions into high level intents, and subsequently into formal queries in a generic manner. These translation rules and templates that enable generic exploration of large scale semantic data constitute the third contribution of the thesis. An iterative User-Centered Design methodology, with the active participation of nearly a hundred users including knowledge workers, managers, engineers, researchers and students over the duration of the research was employed to develop both solutions. The fourth contribution of this thesis is an argument for the continued active participation and involvement of all user communities to ensure the development of a highly effective, intuitive and appreciated solution.

PUBLICATIONS

The work presented in this thesis is present in the following publications:

- Mazumdar, S., Petrelli, D., Elbedweihy, K., Lanfranchi, V., & Ciravegna, F. Affective Graphs: The Visual Appeal of Linked Data. *Semantic Web — Interoperability, Usability, Applicability* (In Press).
- Mazumdar, S., Petrelli, D., & Ciravegna, F. Exploring user and system requirements of linked data visualization through a visual dashboard approach. *Semantic Web — Interoperability, Usability, Applicability* (In Press)
- Cano, A. E., Mazumdar, S., & Ciravegna, F. Social influence analysis in microblogging platforms - A topic-sensitive based approach. *Semantic Web — Interoperability, Usability, Applicability* (In Press)
- Elbedweihy, K., Mazumdar, S., Wrigley, S.N. & Ciravegna, F. (2014). NL-Graphs: A Hybrid Approach Toward Interactively Querying Semantic Data. *Research Track, 11th Extended Semantic Web Conference (ESWC 2014), 25-29 May, Anissaras, Crete, Greece.*
- Gentile, A. L., Lanfranchi, V., Mazumdar, S., Ciravegna, F. (2011). Extracting semantic user networks from informal communication exchanges. In *The Semantic Web — ISWC 2011* (pp. 209-224). Springer Berlin Heidelberg.
- Lanfranchi, V., Carvalho, R. F., Gentile, A. L., Mazumdar, S., Ciravegna, F. (2011). A semantic knowledge management framework for informal communication exchanges. In *Demo Track, 10th International Semantic Web Conference (ISWC 2011)*.
- Mazumdar, S., Varga, A., Lanfranchi, V., Petrelli, D., & Ciravegna, F. (2011). A knowledge dashboard for manufacturing industries. In *The Semantic Web: ESWC 2011 Workshops* (pp. 112-124). Springer Berlin Heidelberg.
- Rowe, M., & Mazumdar, S. (2010). Integrating and interpreting social data from heterogeneous sources. In *Linking of User Profiles and Applications in the Social Semantic Web, Extended Semantic Web Conference 2010*.
- Mazumdar, S. 2010. Visualization of Large Datasets using Semantic Web Technologies. *Poster Proceedings of the ESWC2010 PhD Symposium*.
- Petrelli, D., Mazumdar, S., Dadzie, A. S., & Ciravegna, F. (2009). Multi visualization and dynamic query for effective exploration of semantic data. In *The Semantic Web — ISWC 2009* (pp. 505-520). Springer Berlin Heidelberg.

ACKNOWLEDGMENTS

I would like to express my deep and sincere gratitude to my supervisor, Prof. Fabio Ciravegna for his extraordinary support and guidance throughout my PhD. His suggestions and creative ideas for solving complex problems has always been thought-provoking and highly stimulating. His ability to turn complex problems into creative research opportunities is highly inspirational and has taught me how to question and critically think about research problems. I am also extremely grateful for all his comments, suggestions and advice on my thesis and, despite his hectic schedule, he has always taken out time for me. He has been a motivation and an inspiration to me and I will always cherish his advice and guidance on research as well as life in general.

I would also like to thank my co-supervisor, Prof. Daniela Petrelli for her immense support and guidance during my PhD. Her highly creative approach toward solving problems and instantly sketching complex designs during discussions has helped me think creatively for innovative ideas. I am extremely grateful to her for teaching me how to read and write papers and providing advice on research as well as life. She has been an outstanding support to me, especially during my early years in research when she spent hours discussing ideas, teaching me relevant concepts, and helping me plan my research. Talking to her is always a privilege and I thank her for the hours she has spent discussing research, sharing new ideas and discussing possible future research.

Fabio and Daniela have been a great support for me and I have learnt far more from them than I can list — I thank them immensely for their time and effort.

Few people have an enduring impression on one's life. I am thankful and feel greatly privileged for their lasting influence on my life – Mr. T. Sinha and Dr. S. N. Sanyal. I would like to thank my teachers and guides, Mr. R. Bandyopadhyay, Mr. A. Chatterjee and Prof. P. Misra. I will always cherish their wisdom, knowledge and advice.

As a young researcher, I have received an incredible amount of help and guidance from a lot of researchers. I am very grateful to Jonathan Butters, Ravish Bhagdev, Matthew Rowe, Rodrigo Carvalho, Anna Lisa Gentile, Vitaveska Lanfranchi and Sam Chapman for the invaluable help, suggestions and advice they provided during my PhD research.

I am very grateful to all the members of the Organisations, Information and Knowledge (OAK) group for helping me throughout my research, despite their busy schedules and strict project deadlines. I would like to thank Amparo Elizabeth Cano Bassave, Grégoire Burel, Khadija Elbedweihy and Andrea Varga for all the fun times we shared — they made my life as a PhD student highly enjoyable and delightful.

I would also like to thank all the participants of my experiments, who had provided extremely valuable feedback, suggestion and advice. I thank them for their time and effort, despite their busy schedule. The members of the OAK group and K-Now deserve a special mention for their continued effort and participation in my user studies.

I am extremely grateful to Rolls Royce plc., for providing me with the opportunity to collaborate in an industrial setting. My visits and meetings at Rolls Royce, Derby

have been enlightening and stimulating. In particular, I would like to thank Andrew Harrison, Colin Cadas, Kevin Platt and Divya Tanjore for their invaluable comments, advice and suggestions. I would also like to thank all the members of the Samulet project for their collaboration. The members of the Grassportal project have also been very helpful and have provided me with a lot of feedback that have been very helpful for my research.

Many thanks to Praveen Appikonda, Mabrouka Maamra, Yogesh Sharma, Sambit Das, Prajit Chatterjee, Panchadeep Nath, Qin Yu, Neha Patel, Esperance Pinto, Sridatri Ghosh and Aparajit Gupta for not only being invaluable friends over the past many years, but also being an invaluable support to me. I would like to thank my brother, Indradeep Mazumdar for providing invaluable guidance to me and being an immense support to me and my family. His thoughtful, insightful and valuable comments have always helped me think critically and in more depth, helping me take clearer and better decisions. Thanks are also due to my sister in law, Richa Nahata and my newpew, Risheet Mazumdar.

I would also like to thank my wife, Piyasha Mukherjee, who has been a continuous source of strength for me. She is an incredible support and has been there beside me through good times and bad. I am extremely grateful to her for her continuous encouragement and motivation.

I would finally like to thank my parents for their love, support and belief in me. I thank them for the unconditional support they have given me, their patience with me and the sacrifices they have made for my happiness.

CONTENTS

1	INTRODUCTION	1
1.1	Motivations	3
1.2	Research Questions	6
1.3	Claims	6
1.4	Contributions and Thesis Structure	8
I	STATE OF THE ART	11
2	BACKGROUND	13
2.1	Information Behaviour	15
2.2	Information Seeking	16
2.3	Visual Analytics	19
2.3.1	Interaction with Visualisations — Dynamic Query	22
2.4	The Semantic Web	22
2.4.1	Ontologies	27
2.4.2	Linked Data	27
2.5	Knowledge Management	28
2.5.1	Business Intelligence	29
2.5.2	Business Intelligence Systems	30
2.6	User-Centered Design	33
2.6.1	Engaging Users in Design	34
2.6.2	Stages of User-Centered Design	37
2.7	Summary	40
3	VISUAL ANALYTICS IN THE SEMANTIC WEB	43
3.1	Consuming Semantic Data	44
3.1.1	Querying Semantic Data	45
3.1.2	Presenting Semantic Data	50
3.2	Knowledge Management and the Semantic Web	60
3.3	Ontologies in Knowledge Management	61
3.4	Semantic Knowledge Management Systems	64
3.5	Aesthetics and Art in Information visualisation	68
3.6	Principles of Aesthetic Design	74
3.7	Aesthetics and Semantic Web	75
3.8	Summary	77
II	METHODOLOGY	79
4	VISUALISING LARGE SCALE SEMANTIC DATA: AN APPROACH	81
4.1	Analysis of Requirements for Visual Analytic Solutions in Knowledge Management	81
4.1.1	Stakeholder Requirements	81
4.1.2	A User Centered Approach to Understanding Domains and Users	82
4.1.3	Understanding Domains	83
4.1.4	Studying the Domains	87
4.2	Requirements	89

4.2.1	Functional Requirements (FR)	89
4.2.2	Non Functional Requirements (NFR)	91
4.3	Solution — Semantic Web	92
4.4	The Solution Architecture	95
4.4.1	User Interface	97
4.4.2	Interaction Interpretation	97
4.4.3	Result Interpretation	98
4.5	Outline of Research	98
4.5.1	Visualising semantic data: engaging with users	99
4.5.2	Exploring and querying semantic data: engaging with users	103
4.6	Summary	106
III DEVELOPMENT OF TECHNOLOGY 109		
5	POINTS OF VIEW — VISUALISING SEMANTIC DATA	111
5.1	Introduction	111
5.2	Design Rationale	112
5.2.1	Design Layout	112
5.2.2	Visual Representation of Semantic Results	113
5.2.3	Visualisation Widgets	114
5.3	Architecture and Implementation	116
5.3.1	Configuration Step	117
5.3.2	Translating interactions to SPARQL	118
5.3.3	Tools Used	121
5.4	Points of View — A scenario of use	122
5.5	Conformance to Requirements	124
5.6	Evaluation	124
5.6.1	Focus Group Sessions with guided tasks	124
5.6.2	Focus Group Sessions with open tasks	129
5.6.3	Understanding public endpoints and System Requirements	134
5.6.4	Limitations	135
5.6.5	Evaluation within an Organisation	136
5.7	Generic Mechanisms of Interaction	140
5.7.1	Global Interactions	140
5.7.2	Local Interactions	142
5.8	Summary	143
6	AFFECTIVE GRAPHS — EXPLORING AND QUERYING SEMANTIC DATA	147
6.1	Introduction	147
6.2	Principles of Aesthetics for Semantic Data	148
6.3	Evaluating aesthetics of Semantic Web applications	149
6.3.1	Experiment design	150
6.3.2	Results	159
6.3.3	Limitations	160
6.4	Exploring and Querying Semantic Data — approach	160
6.5	Affective Graphs — A scenario of use	161
6.6	Design Rationale	164
6.6.1	Consistency in Visual Representations	165

6.6.2	Representing Semantic Concepts	165
6.6.3	Representing Semantic Relations	166
6.6.4	Layout	168
6.6.5	Designing Interactions	169
6.6.6	Designing an Integrated solution	169
6.7	Architecture	170
6.7.1	Query Mechanism	174
6.7.2	User Interactions and Contextual Information Presentation	176
6.7.3	Query Formulation	177
6.7.4	Result Presentation	179
6.7.5	Tools Used	180
6.8	Conformance to Requirements	180
6.9	Evaluation	180
6.9.1	Formative Evaluation with Domain Experts	182
6.9.2	User Evaluation with Casual and Expert users	187
6.9.3	Effect of prolonged use	193
6.9.4	Hybrid Approach Evaluation	202
6.9.5	Evaluating Aesthetic Properties of Affective Graphs	215
6.10	Generic Mechanisms of Interaction	217
6.11	Supporting User Needs in Semantic Web	220
6.12	Summary	222
IV	CONCLUSIONS AND FUTURE WORK	225
7	DISCUSSIONS	227
7.1	Visualising Semantic Result Sets: A Reflection	227
7.1.1	Lessons Learned and Guidelines	228
7.1.2	Novelty in the solution	231
7.2	Exploring and Querying Semantic data: A Reflection	233
7.2.1	Novelty in the solution	233
7.3	Impact on the State of the art	236
7.4	Summary	240
8	CONCLUSIONS	243
8.1	Revisiting the Research Questions and Claims	244
8.2	Impact	247
8.3	Continuing Research and Future Work	248
8.3.1	A Combined Approach — a proposal	248
8.3.2	Visualising Semantic Data	249
8.3.3	Visualising Social Media	251
8.4	Final Remarks	252
V	APPENDIX	255
A	AESTHETIC METRICS	257
B	EVALUATION QUESTIONNAIRES - POINTS OF VIEW FOCUS GROUP SESSION WITH OPEN TASKS	259
C	EVALUATION QUESTIONNAIRES - POINTS OF VIEW FOCUS GROUP SESSION WITH GUIDED TASKS	267

D	AFFECTIVE GRAPHS EVALUATION WITH DOMAIN EXPERTS	273
E	AFFECTIVE GRAPHS EVALUATION QUESTIONNAIRES: CASUAL AND EXPERT USERS	283
F	AFFECTIVE GRAPHS EVALUATION QUESTIONNAIRES: LEARNABILITY	285
G	AFFECTIVE GRAPHS EVALUATION TASKS: LEARNABILITY	295
	G.0.1 Simple Tasks (ST)	295
	G.0.2 Multiple Attributes Task (MAT)	295
	G.0.3 Multiple concepts Task (MCT)	295
	G.0.4 Complex Task (CT)	295
	G.0.5 Exploratory Task (ET)	295
	BIBLIOGRAPHY	297

LIST OF FIGURES

- Figure 1 Choo's representation of data, information and knowledge. Image from [CDT00] 14
- Figure 2 Representations of Knowledge hierarchy 15
- Figure 3 Wilson's and Spink and Cole's model describes the interplay among Information Seeking and Information Behaviour 17
- Figure 4 The role of lookup, learning and investigating in explaining exploratory search 18
- Figure 5 Focus areas of Visual Analytics, as illustrated in [TC05] 21
- Figure 6 A more detailed view of the various research areas contributing to Visual Analytics, illustrated in [KMSZ06] 21
- Figure 7 Dynamic HomeFinder — an example of dynamic query being used in a real-estate scenario. Image from [WS92] 23
- Figure 8 Results obtained by comparing the time taken by users interacting with dynamic query mechanisms, natural language query interfaces and paper based searching. Image from [WS92] 24
- Figure 9 Examples of Business Intelligence visualisation dashboards. Top left: IBM Cognos (Image from <http://jragdale.wordpress.com/tag/ibm-cognos>), Top right: SAP BI (Image from http://www.biondemand.com/sites/default/files/global/images/screenshots/biod_dashboard.jpg), Middle left: Microsoft BI (Image from <http://www.dashboardinsight.com/dashboards/screenshots/bi-analytics-dashboard.aspx>), Middle right: Oracle BI (Image from [ora]), Bottom left: Jaspersoft (Image from <https://www.jaspersoft.com/tour>) and Bottom right: Pentaho (Image from <http://www.pentaho.com>) 32
- Figure 10 Rubin's [Rub94] depiction of users as the centre of the development. Image from <http://www.neelworld.info/documents/ucd.pdf> 35
- Figure 11 The survey by [VMSC02] shows which methods were being used more frequently in practice as well as the ranking of five methods which in their opinion have the greatest impact by over 100 UCD practitioners. 37
- Figure 12 (Left) Conventional User-Centered Design (UCD), four-stage design model (Right): Extended user centred, five stage design model, introducing a new concept, understand. Image from [HRRS07] 38
- Figure 13 mSpace interface exploring Classical Music dataset. Image From <http://research.mspace.fm/mspace/what> 47

- Figure 14 View-Based approaches: GRQL (Top-Left) allows users to navigate ontologies as hierarchical lists and select concepts and properties of interest. Image from [ACK04]; Semantic Crystal (Top-Right) shows an entire ontology using interactive graphs, enabling users to select concepts and properties to set for their query. Image from [KBF07]; iSPARQL (Bottom-Left) allows users to build visual queries by adding new concepts and creating links between the query concepts. Image from <http://dbpedia.org/isparql/>; SEWASIE (Bottom-Right) enables users to select from pre-defined domain-specific templates as a starting point to drive querying. Image from [CDM⁺04]. 48
- Figure 15 An extension of the formality Continuum, illustrated by [KB10]. Faceted Browsers and form-based systems have been introduced in the model, before a completely formalised approach. Image adapted from [KBF07] 49
- Figure 16 Existing Semantic Systems positioned on Interactivity and Visual encoding scales — note how the system screenshots show progressively increasing visual elements used on the interfaces. 51
- Figure 17 Enriched Text systems such as PowerAqua present text and clickable URIs in structured tables and organised layouts based on content. Image from [LMU06] 52
- Figure 18 MuseumFinland interface, where a user has performed an initial query to look for all artifacts that are tools. Image from [HMS⁺05] 54
- Figure 19 Sig.ma interface showing the mashup returned after querying for Tim Berners-Lee. Image from <http://sig.ma/search?q=Tim%20Berners%20Lee> 55
- Figure 20 Tabulator provides users with two modes of viewing RDF data — an exploratory Outliner mode (Top) and analysis mode for viewing instances (Bottom Left and Bottom Right) in geographical maps and calendar views respectively. Images from [BLCC⁺06] 56
- Figure 21 DBpedia Mobile provides a map enriched with geocoded data instances from linked data (top left); clicking on locations provide more information about the resource (bottom left); users can add filters to find relevant information (top right); users can create new content by interacting with a simple form to add a new location, photo or review (bottom right). Images from [BB08] 57
- Figure 22 Examples of Graph Visualisation Tools. Top Left — Welkin (Image from <http://simile.mit.edu/welkin/>). Top Right — RDF Gravity (Image from <http://semweb.salzburgresearch.at/apps/rdf-gravity/>). Middle Left — Cytoscape (Image of RDFscape plugin, from [Splo8]). Middle Right — IsaViz (Image from <http://www.w3.org/2001/11/IsaViz/>), Bottom Left — Semantic Wonder Cloud (Image from [RATS10]), Bottom Right — Relfinder (Image from <http://www.visualdataweb.org/relfinder.php>) 59
- Figure 23 Davies' proposed architecture of Knowledge Management using Semantic Web technologies. Image recreated from [DFH03] 65

- Figure 24 QuizRDF. Image from [DW04] 67
- Figure 25 Results from K-Search, a hybrid search system showing document annotations and graphs of result sets as well as text snippets from the documents that match a keyword and semantic query. Image from [BCC⁺08] 67
- Figure 26 Spectacle Cluster Map 68
- Figure 27 (Left) A few existing applications mapped on the information aesthetics model proposed by [LM07]. (Right) Lau and Moere identifies how information visualisation and visualisation art map on their model 71
- Figure 28 Examples of aesthetic information visualisations 72
- Figure 29 Examples of beautiful information visualisations — providing information using artistic images 73
- Figure 30 An example scenario was explored where a biologist needs to understand the distribution of grasses belonging to a particular tribe (*panicae*). Step 1 – Step 3 Images from <http://www.kew.org/data/grasses-db.html>. Step 4 Image from <http://www.gbif.org/> 86
- Figure 31 Solution Architecture 96
- Figure 32 Development of solutions, using an iterative user centered design approach. The figure illustrates how the User Centered Design methodology (image at the center from [HRRS07]) was central to the development of the final solutions, by using several iterations of the design, development and evaluation stages. Two main solutions (*views*. and Affective Graphs) were developed. While being an extension to Affective Graphs, Affective Graphs (hybrid) is illustrated differently for better readability. The figure is not to be scaled, but aims at providing a relative timeline of when different phases of the user centered design were conducted 99
- Figure 33 A factory floor map showing the machine capabilities and performances. Image provided by Rolls Royce as a part of the Samulet Project 100
- Figure 34 Sketches for perceived system used during UCD phase — notice the widget-based dashboard (Right) used to illustrate the dashboard 101
- Figure 35 Visualisation of Semantic Result sets final interface design 102
- Figure 36 Sketches during the design phase for the development of Affective Graphs 104
- Figure 37 Querying Semantic Data final interface design 106
- Figure 38 The Web-based interface for grass data with generic filters (top) and four different views on the retrieved data set, namely: tag cloud, result list, pie chart and geo-plot. (Data is courtesy of the GrassPortal (<http://www.grassportal.org/>) Project and Kew Gardens). As can be observed, the major elements of the interface are: filtering interface (A), filter selections (C), search button(D) and visualisation space with widgets (B) 113

- Figure 39 Social data after the flood in Cumbria, UK in 2007 visualised using *.views*. (Data harvested from Flickr and Twitter). The Cascading Style Sheets (CSS) style in this instance has been modified from Figure 38, though the basic interface remains the same. 115
- Figure 40 Solution Architecture of the Interface 116
- Figure 41 Architecture of the visualisation interface 119
- Figure 42 Popup dialog providing details on individual instances — here, the details on the University of Sheffield. 121
- Figure 43 Global filtering for user-defined queries on the DBpedia dataset. On entering the values for filters, the number of results available is displayed in a button, clicking on which starts visualising the results. 123
- Figure 44 Automatic suggestions guide users for providing the right query. Typing a few characters (here, “pa”) starts suggesting possible filter values containing these characters, along with the total number of times they have occurred within the dataset. 124
- Figure 45 User-satisfaction questionnaire responses for students (Left) and domain and computer science experts (Right) show the overall improvement of the modified system. 131
- Figure 46 Plots showing the extremely high variation of query execution phase (98ms to 6.85s) in the backend processing. The size of the results (1 – 2200 results returned, 2 – 1100 results returned, 3 – 600 results returned and 4 – 100 results returned) are shown as individual bars on the y-axis, and the differently shaded x axis bars show the time taken to perform individual functions to retrieve the respective results. The plots show the relative times in different parts of the system 136
- Figure 47 A traditional database being used as the backend — as expected, most of the times the query processing stage takes the longest times, however this is a consistent behaviour. 137
- Figure 48 *.views*. deployed for Cost Reduction use case 138
- Figure 49 Evaluation Results 139
- Figure 50 Layouts used to compute interface aesthetic metrics. The layouts were obtained from the interfaces directly or from the relevant papers, publication materials, websites or user manuals. The layouts shown are of ten well-known Semantic Web interfaces (screenshots of the system are shown on the top, while images at the bottom show the respective markups): 1. NLP Reduce, 2. Ginseng, 3. Semantic Crystal, 4. Querix, 5. DBpediaFacet, 6. Sig.ma, 7. KSearch, 8. Tabulator, 9. PowerAqua, 10. mSpace and 11. Affective Graphs (later discussed in Section 6.9.5) 152

- Figure 51 Calculating Interface Metrics. Left: Illustration of how different measures were calculated. Image recreated, inspired from [NSA00]; Right: The Java application, Aesthetics Calculator shows a user left clicking-and-dragging the mouse to mark up a visual area. Upon releasing the mouse button, the new aesthetic measures are calculated and shown at the bottom 153
- Figure 52 Comparative evaluation of seven aesthetic metrics with ten Semantic Web tools. The eighth metric, Order and Complexity is computed as a sum of the others 159
- Figure 53 A screenshot of *AffectiveGraphs*, where the node currently on focus is 'Lake': Section A contains the interactive node-link representation of the data, Section B contains contextual information relevant to the concept currently being explored (here, Lake), Section C contains search elements and controls the visual rendering of the node-link graph, Section D shows the Simple Protocol and RDF Query Language (SPARQL) query being generated for search, Section E contains advanced features to modify the query. 162
- Figure 54 Comparison of randomly generated pie-chart colours for RGB (left) and HSB (right). Though the colours on the right are completely random, constant values of brightness and saturation generate a more pleasing chart. Images created in <http://sketch.processing.org/> 166
- Figure 55 Different visual representations of edges bear different ontological significance — hierarchical edges are represented as triangles showing the parent and child node, data properties are represented as satellite objects and object properties are represented as curves 167
- Figure 56 (i) Comparison of Straight Edges with curved Edges. Image from [FT04], (ii) Visual response to hovering over an object property 168
- Figure 57 Architecture 170
- Figure 58 Architecture 172
- Figure 59 Different visual representations of nodes and edges bear different ontological significance 174
- Figure 60 Visually building queries 174
- Figure 61 adding constraints 175
- Figure 62 Illustration to describe how query formulation takes place. Left — the user starts by adding a property to query; Middle — another property is added to the query; Right — a constraint is added to the query 177
- Figure 63 Results of the Formative Evaluations with domain and Knowledge Management experts 184
- Figure 64 Evaluation results for SUS scores, query input time and number of attempts for *Affective Graphs*: the users have been grouped into two types — experts and casual 190

- Figure 65 Analysis of performance of users with five types of tasks: ST: Simple, MAT: Multiple Attribute, MCT: Multiple Concept, CT: Complex, ET: Exploratory. Figure on the left shows a box-plot of the number of attempts required to solve a task satisfactorily, grouped into sessions and task types. Figure on the right shows a box-plot of the input time grouped into sessions and task types. Images from [MPE⁺14] 198
- Figure 66 Results of the User Evaluations. Left — Aesthetics questionnaire responses, Right Top — SUS scores, Right Bottom — Extended questionnaire responses. Image from [MPE⁺14] 200
- Figure 67 Incorporating a natural language search box to provide means for users to enter questions and keyword based queries (top). In this example, the user has just entered a query, and Affective Graphs renders the identified concepts and attempted to build a query from the retrieved terms. The queries are analysed to identify concepts which are then subsequently loaded in the visualisation, as well as displayed on the right Query panel 204
- Figure 68 Analysis of hybrid search approach, grouped by experts and casual users. Figure on the left shows the number of attempts for each query, where a few queries required more than one attempts. Figure on the right shows the input time taken to complete formulating a query. 208
- Figure 69 Two queries provided some challenges for users — Left, Q5 required the two properties isPartOf and populationTotal (marked in blue) to be added to query with a single constraint on populationTotal. Right, Q2 required artist and releaseDate (marked in blue) to be added to query with multiple constraints added to releaseDate 209
- Figure 70 Qualitative data analysis — Figure on the left shows a boxplot of SUS scores grouped into expert and casual users; Middle figure shows a boxplot of overall ease of approach, grouped into expert and casual users; Right figure shows the aesthetics scores provided by users 210
- Figure 71 Comparison of Affective Graphs with the highest scoring tool and the maximum scores in each aesthetic metric 216
- Figure 72 Improving readability (and speed) by limiting results for aggregate queries. Both figures show identical information (e.g., most resources are in London). However, the first widget is almost illegible. 230
- Figure 73 Improving readability for textual results by providing a page view of large result sets. Note the scroll bar for both figures indicating the number of textual results contained in the widget. 231
- Figure 74 Positioning the approaches on an extension of the formality Continuum, illustrated by [KB10]. Image adapted from [KBF07] 237

Figure 75	Relative positions of <i>.views.</i> and Affective Graphs on the Interactivity vs Visual Abstraction graph, as introduced in Section 3.1 238
Figure 76	Single-path navigation: top left shows an approach where users progressively select subsets of the data from Left to Right (Image from [sWRS06]); top right shows an approach where users navigate from Left to Right, but relevant information related to the right columns is presented on the left columns (Image from [WAs08]); Multi-path navigation support of Affective Graphs enables exploration in multiple directions and provide means for users to discover connections between the paths 239
Figure 77	A transition in the Information Need from generic to highly specific results in the inability of a dataset in providing specific information 250

LIST OF TABLES

Table 1	Table prepared by [AMkPo4] to indicate ways to engage the user communities in the software design cycle 36
Table 2	Aligning final interface design with functional and non-functional requirements 125
Table 3	User-satisfaction responses for domain experts on a 5-point Likert scale. 132
Table 4	Global User Interactions translated to SPARQL queries 142
Table 5	Local User Interactions translated to SPARQL queries 143
Table 6	General Principles for Aesthetic Linked Data Visualisation 149
Table 7	Principles for Aesthetic Node-Link Representations 150
Table 8	Aligning final interface design with functional and non-functional requirements 181
Table 9	A manual inspection of screen recordings show different exploratory behavior and uncover several issues 185
Table 10	User Interactions translated to SPARQL queries in Affective Graphs 219
Table 11	Design guidelines proposed by Dadzie and Rowe [DR11] visual information presentation catering to two types of users — lay and tech 221

ACRONYMS

API Application Programming Interface

BI	Business Intelligence
CAD	Computer Aided Designing
CAM	Computer Aided Manufacturing
CI	Competitive Intelligence
CSS	Cascading Style Sheets
CTA	Cognitive Task Analysis
DSS	Decision Support System
FOAF	Friend Of A Friend
GBIF	Global Biodiversity Information Facility
GIS	Geographical Information Systems
GML	Graph Markup Language
HCD	Human-Centered Design
HCI	Human-Computer Interaction
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
IDE	Integrated Development Environment
IE	Information Extraction
IR	Information Retrieval
IT	Information Technology
ISO	International Organization for Standardization
JSON	JavaScript Object Notation
KM	Knowledge Management
LD	Linked Data
LOD	Linked Open Data
NL	Natural Language
NLP	Natural Language Processing
OWL	Web Ontology Language
PNG	Portable Network Graphics
RDF	Resource Description Framework

RDFa	Resource Description Framework in attributes
RQL	RDF Query Language
SPARQL	Simple Protocol and RDF Query Language
SQL	Structured Query Language
SVG	Scalable Vector Graphics
SW	Semantic Web
UCD	User-Centered Design
UI	User Interface
UML	Unified Modelling Language
URI	Universal Resource Identifier
URL	Universal Resource Locator
VA	Visual Analytics
W ₃ C	World Wide Web Consortium
WWW	World Wide Web
XGML	Extensible Graph Markup Modelling Language
XML	Extensible Markup Language
YAGO	Yet Another Great Ontology

INTRODUCTION

A substantial amount (upto 55%) of an aerospace design engineer's time is occupied while seeking, processing, communicating and disseminating information [Rob10]. 40% of corporate users can not find essential information on their intranet [Felo4]. Feldman estimates that an organisation with 1000 knowledge workers loses at least \$6 million annually in time spent searching for knowledge, while the cost of reworking information owing to not being retrieved can be upto \$12 million. Only 21% of knowledge workers can find the information they look for in more than 85% of the cases [Felo4]. The foundations of traditional knowledge practices within organisations where isolated groups of individuals focussing on their own problems and datasets is now challenged with a move toward a highly collaborative environment exploiting expertise from various data sources, levels and domains. Traditional management styles and operational frameworks often lack the flexibility and agility to fully exploit the opportunities arising out of this information deluge [Rudog]. Modern organisations need to *"evolve into a knowledge-generating, knowledge-integrating and knowledge-protecting organisation"* [Teeoo] in order to effectively generate, manage and curate organisational knowledge.

Highly structured, self-descriptive pieces of information, interlinked in large networks to formalise constantly growing organisational knowledge have the potential of identifying highly valuable knowledge, a significant portion of which would remain hidden and buried otherwise. Therefore, I believe that the opportunities arising out of employing semantic techniques in knowledge practices within large organisations are considerable. Core enabling technologies, Application Programming Interface (API) and tools encompassing ontologies, ontology languages, annotation services and repositories developed by the Semantic Web community provide an excellent infrastructure for distributed information and Knowledge Management (KM) based on metadata, semantics and reasoning [CSG07, DLS07]. This generates excellent potential for building highly knowledgeable systems with various specialised reasoning services [FVHD⁺02] and a shared community knowledge infrastructure that can form an indispensable part of an organisation.

While great opportunities arise out of imbibing semantic techniques in knowledge practices in large organisations, significant challenges need to be addressed. Large organisations generate vast amounts of data, as a part of their daily activities. Rigorous documentation, performance reports, presentations, informal communication, digitisation of legacy data etc. are examples of the huge mass of data generated. The work presented in this thesis is an attempt to alleviate one of the significant challenges that arise as a result: effectively and efficiently consuming large semantic datasets. While other challenges such as annotation of large volumes of data, validation and organisation of semantic data, building and managing ontologies etc. are significant challenges that arise out of generation of semantic data, this thesis solely focusses on consumption.

Facilitating the consumption of semantic data is highly challenging, as analysts, knowledge workers and user communities need quick access to data. Most user interfaces

developed within the Semantic Web (SW) community are still research prototypes, and though promising, struggle to address issues such as scalability and genericity. With continuously growing semantic data, manual searches and basic displays make the task of quickly assessing large datasets difficult. This raises the need for a visualisation-based approach, as visualisations can help summarise and interpret the contents of large volumes of data. The need for visualisations as a solution for semantic organisational knowledge introduces further challenges. How can visualisations be used to communicate and interact with massive data sets, how can such solutions be intuitive, interactive and appealing, how can users interact with such solutions to express their information need and so on are some of the questions that are raised when such solutions are sought.

Though the attention is on providing a visualisation solution that is functional and attempts to address user information needs, however, a major issue in Knowledge Management and Information Retrieval systems is the lack of attention to aesthetic appeal. Despite aesthetics being considered an intangible characteristic of an interface, the results of employing aesthetic design can be highly significant in influencing ‘playfulness’ [PSP⁺13, MK01, LTo4] and usability [Hua, LWT05, CSP10, PSP⁺13] among users. Aesthetic design has been adopted in a few areas such as product design, civil engineering [CM07, DASH06] and web design [Tra04, KG10, TKI00]. The Information Visualisation community, in addition to artists, illustrators and web designers has also been producing beautiful visualisations and infographics¹ carefully designed to evoke an aesthetic response among readers². Much of this effort however, has been in illustrating facts and figures within articles or web sites³ instead of an interactive visualisation framework aimed at exploring large datasets. The translation of such principles into the fields of SW, KM and Information Retrieval (IR) has therefore been highly limited. Few works note the attention to and the need for aesthetics [HHU⁺11, GGP⁺10, DDSC07, RC04], but there is an urgent need for a more methodological approach toward interface design with a specific attention to aesthetics, from the initial conceptualisation and design stages

Motivated from a Knowledge Management (KM) perspective within large organisations, this thesis proposes design principles and recommendations that can help solution developers take a methodological approach toward addressing issues that can arise out of consuming semantic data in the enterprise. The design principles propose how interface designers can develop aesthetically pleasing solutions for consuming large volumes of semantic data. This thesis also presents two approaches for Visual Analytic solutions for exploring and querying large distributed knowledge sources within organisations. Evaluations conducted during various stages of development and their results are further presented. Findings from the evaluations show that while large semantic datasets can be visualised and effectively explored, constraints imposed by existing frameworks can limit a fully interactive experience. The findings also show that aesthet-

¹ Information Graphics are visual representations intended to present complex information quickly and clearly, <http://en.wikipedia.org/wiki/Infographic>

² <http://infosthetics.com> archives some excellent examples of aesthetically designed visualisations, which can inspire new ideas for visualising complex data

³ Several news agencies employ on infographics to illustrate facts to support their news stories or to support interactive story telling – e.g. <http://www.theguardian.com/data>, <http://www.nytimes.com/newsgraphics/2013/12/30/year-in-interactive-storytelling/#dataviz>

ically pleasing solutions can be developed for querying large repositories of information by employing Visual Analytic and aesthetic design principles.

A major contribution of this thesis is in establishing the need for user engagement at all stages of the development of solutions. Starting from initial requirement gathering and understanding domain knowledge to evaluating final solutions, the role of users and stakeholders is paramount. Involving users in all stages ensure the requirements and design solutions are well communicated, and the users and their information needs are well understood by developers.

The next section presents the motivation for the research in more details, and sets the stage for the thesis.

1.1 MOTIVATIONS

With data continuously generated as a result of daily activities within large organisations and new data sources (sensors, datasets etc.) introduced as sources of semantic data, a significant growth of semantic knowledge within organisations is observed. Several organisations have already started integrating Linked Data (LD) within their enterprise and many success stories have emerged that are seen as great examples of the Semantic Web effort in the enterprise. An excellent example is the BBC's adoption of the Linked Data initiative in releasing information as Linked Open Data. The BBC has been one of the earliest adopters of Linked Data among the large organisations, and several endpoints such as the programmes⁴, music⁵ and wildlife finder⁶ provide excellent examples of how proprietary data can be released for fostering development of applications [HB11]. Universal Music makes their data available as Linked Data to be easily queried on a custom interface⁷. The life sciences domain has also been an early adopter of Linked data for Knowledge Management tasks via several research projects such as Bio2RDF⁸, Linked Life Data⁹, CardioSHARE¹⁰, GenoQuery¹¹ etc [AKM09]. The same is observed with Governments releasing their data as a part of their commitment to transparency: a growing amount of information is continuously being made available to public as linked open data¹². If, on the one hand, this amass of data is exciting to have and exploitable by knowledge workers, on the other hand there is a need to quickly understand content spanning across various data repositories. The Semantic Web and Knowledge Management communities are facing new challenges in terms of consuming Semantic Data made available, e.g. dynamic discovery of sources, provenance and quality assessment, effective integration to name a few. However this is only one side of the coin as data are intended to be, in the end, for human consumption, not just for machine crunching.

4 <http://bbc.co.uk/programmes>

5 <http://bbc.co.uk/music>

6 <http://bbc.co.uk/wildlifefinder>

7 <http://umusic.co.uk/artists>

8 <http://bio2rdf.org/>

9 <http://linkedlifedata.com/>

10 <http://biordf.net/cardioSHARE/>

11 <https://www.lri.fr/~lemoine/GenoQuery/>

12 As of October 2013, <http://ckan.net> registered 21,549 datasets of different types such as Census Records, Railway Maps, Greenhouse Gas data, Airport data, Gene information and so on.

A multitude of visualisation methods have been proposed, ranging from standalone visualisation systems to web based mashups and browsers. However, most of such methods suffer from two major drawbacks: genericity and scalability.

Genericity refers to how generic an approach is with respect to domains and datasets. Significant effort is spent on visualising large collections of domain-specific data, resulting in exciting visualisation paradigms and techniques that cater to only one domain or dataset. This, however would generate a tremendous impact on other datasets, domains and communities if developed with genericity in consideration. For example, the Talis Research Funding Explorer¹³ provides an excellent example of how Visual Analytic principles can be incorporated with Semantic Web data to support analytical tasks. However, it would be a significant benefit to other use cases (such as an organisation exploring financial data, open governmental dataset on public administration, or global crime statistics) if there was an easy way of “plugging-in” another dataset from a different domain. Admittedly, much of the research output from a domain (e.g. genetic data analysis) is highly domain dependent and cannot be easily transferred into another domain (e.g. music publishing). This is where the need for generic technology frameworks arise, and I believe more research needs to be invested in such technologies, keeping in mind that some domain-specific aspects may also be needed to be implemented as fit-for-purpose solutions. Since the very essence of semantics itself is self-description, research involving semantic technologies need to be domain-independent. This has not been observed with visualisation systems existing in the Semantic Web and there is a need for similar generic approaches.

Another way genericity applies is in terms of a user’s experience, expertise and role. Historically, the act of gathering insight from visualisations have mostly been seen as an expert task, requiring the trained eyes of a professional, capable in making sense of images. The same appears to be observed in the Semantic Web community — visualisation systems are mostly designed for domain experts and rarely involve the consultation and participation of typical or lay users (the definition of lay and expert users in this context is the same as described by [DR11]). However, I believe that the use and uptake of visualisations can have a far-reaching effect on a wider audience. This has been shown to a great extent by the many visualisation paradigms that exist on the web as freely available tools. A long standing testament to the empowerment of lay users using visualisations is the popularity of mapping solutions such as Google Maps¹⁴, Bing Maps¹⁵ and Google Earth¹⁶. These are initiatives that are drawn from highly expert domains of cartography and geovisual analysis. However, such technologies have been welcomed and adopted by casual users with such ease that they have stood the test of time and expertise into constantly evolving solutions that are now an everyday necessity for millions of users.

Basic charting features provided by Microsoft Office¹⁷, Open Office¹⁸ and Google Charts¹⁹ have also gained massive popularity with lay users. Most users are comfort-

13 <http://bis.clients.talis.com/>

14 <http://www.google.co.uk/maps/preview>

15 <http://www.bing.com/maps/>

16 <http://earth.google.co.uk>

17 <http://office.microsoft.com/en-gb/>

18 <http://www.openoffice.org/>

19 <https://developers.google.com/chart/interactive/docs/spreadsheets>

able with building charts and graphs for their daily tasks and reporting activities, even though they possess very little or no database knowledge. As a step toward collaborative Visual Analytics, web-based solutions such as IBM's Many Eyes²⁰ encourage participation from the community to create new visualisations with their own datasets as well as reuse of existing ones. Unfortunately, the same environment does not exist in the Semantic Web community, where the focus for most visual solutions have been expert or trained users for domain-specific purposes: hence, a generic approach is highly needed, while keeping domain needs in consideration.

The second drawback, *scalability* applies in two ways — a scalable approach to dealing with large data; and a scalable approach to dealing with large schemas. While both essentially involve scalable solutions to visualising content, they need two different solutions. Unsurprisingly, the ability to efficiently query large and complex datasets does not solve the problem of scalability (as achievable with clever indexes and powerful processors), as scalability also deals with the ability to visualise large complex datasets. Moreover, the ability to visualise large datasets does not warrant that the user will be able to process, comprehend and understand the underlying data. Most of the techniques investigated so far in consuming Semantic data belong to one or more of the following categories: graph visualisations, mashups and browser-based solutions. Graph visualisations suffer from two fundamental problems — most of the solutions visualise the entire schema in a graph, thereby increasing the cognitive burden on users, and isolating them from the data (which is left invisible to the user unless he/she queries for it). Other solutions visualise individual data instances — If an instance contains a large graph, then the solution is unscalable as well as difficult to use. On the other hand, visualising instances does not provide sufficient insight to help understand an entire dataset. Mashups are an excellent way of interacting with semantic data, but also suffer from problems: most of them are built on temporal or topological navigational paradigms — this means that the data needs to contain the relevant information (e.g. time and geo-coded information) for the mashups to be useful. Browser based solutions can be of two types — template based and faceted. Such solutions involve less visualisations, and instead employ variants of interactive tables and lists. Faceted browsing works on a topical navigation paradigm, where the topics have already been pre-determined by the developer or data provider. Furthermore, such interfaces can allow users to follow only one path of exploration, thereby making it difficult for users to compare and contrast data. Template based browsing enables users to visualise only an instance of the data at a time. For example, Sig.ma²¹ provides all the information about a particular resource in one page. However, such methods only help when the user has a very specific information need related to a data instance within the dataset.

While all existing solutions are highly useful and provide creative solutions to their own research problems in their own right, I believe a more generic and scalable approach is needed to ensure more uptake and acceptance in the user community. Semantic data promises immense potential in user interface, visualisation and interaction design and exploiting semantics and building novel, high impact and highly innovative solutions is the motivation of this thesis.

²⁰ <http://www-958.ibm.com/software/analytics/manyeyes/>

²¹ <http://sig.ma>

1.2 RESEARCH QUESTIONS

This thesis explores innovative solutions to visualisation and interaction with large semantic data. Two key players in this context need to be considered — data and users. While semantic data indicate the technical complexities and the intricacies of semantic formalisms, the technologies are in fact, built for users who need help understanding and comprehending data. Clearly, these two motivations need a multi-disciplinary approach where technical solutions to effectively query and visualise data are merged with the needs of users, their cognitive perspectives, perception and aesthetic responses.

This thesis attempts to answer the main research question:

How can large scale, multi-dimensional, heterogeneous semantic data be effectively explored in a generic, usable and pleasing manner?

The above question is broken down into three constituent elements: effective and usable exploration, generic mechanism, and pleasurable experience.

R1 *How can visualisation interfaces provide effective means of exploring large scale semantic data ?*

R1.1 *What are the implications of using large scale semantic data on visualisation systems ?*

R1.2 *What are the implications on users ?*

R1.3 *What are the considerations that are needed in order to design such interfaces?*

R2 *How can visualisation interfaces help explore semantic data in a generic manner ?*

R2.1 *How can the process of visualisation of semantic data be formalised ?*

R2.2 *How can visualisations query semantic data in a generic manner ?*

R2.3 *How can user interactions drive data exploration in a generic manner ?*

R3 *How can visualisation interfaces be designed to be visually pleasing?*

R3.1 *Can we develop aesthetic principles for designing semantic web solutions?*

R3.2 *How can such principles be translated into practice?*

1.3 CLAIMS

The problem of consuming semantic data was analysed and broken into two sub parts: interaction and visualisation of semantic data and querying semantic data. The two solutions were designed using an iterative UCD approach, and as an outcome of evaluations, extensive literature survey and interview sessions, several design principles and recommendations were identified. The tangible results of the thesis are translated into design principles that designers and solution developers can refer to as a guide while developing aesthetically pleasing solutions for visualising large scale semantic data.

In addition to the development of design principles, there are three claims that the thesis makes while answering the research questions. The first claim arises out of the lessons learned from evaluations and user studies while developing the solution for visualising result sets:

C1: A multi-visualisation approach that provides simultaneous views of different facets of large datasets can provide effective means to explore large scale data. However, existing frameworks pose technical challenges that may restrict a fully interactive experience for users

The rationale behind multiple views is to provide simultaneous visualisations of different facets of the data so that users can formulate a very quick understanding and overview of the data from multiple perspectives. While different visualisations require multiple queries to explore data in different facets, system evaluations (Section 5.6.3) with the solution in a realistic scenario discovered that several key (and desirable) features had to be disabled owing to the unpredictability of supporting technology. Existing framework can pose technical challenges that are beyond the control of interface developers, hence, potentially affecting usability and user experience. Several recommendations have therefore been provided as a result, that can help solution developers and designers cope with the unreliability of back end technologies (Section 7.1.1). The evaluation with domain and technology experts validate the hypothesis of a multi-visualisation approach toward effectively exploring semantic datasets as presented in Sections 5.6.1 and 5.6.2. Establishing the validity of the approach in large organisations has been limited to an internal evaluation at Rolls Royce plc, reported in Section 5.6.5. The expenses with respect to expertise time and security restrictions as well as incorporating the approach within standard enterprise search frameworks have been a limitation that hindered a large scale extensive evaluation to validate the approach within large organisations. However, the positive results of the two small scale evaluations (with two use cases in an industrial setting), the acceptance of the solution as a TRL 6 solution and plans of incorporating the multi-visual approach in enterprise search systems at Rolls Royce plc indicate the potential value of the solution (this is discussed in more details in Section 8.2, discussing the impact of the solutions).

The second claim (C2) addresses how a generic approach toward exploring semantic data can be employed.

C2: Generic means for exploring semantic data can be developed by aligning user interactions with various combinations of generic and specific queries. This requires formalising the process of translating low level user interactions to semantic queries by interpreting user actions to high level intent.

A generic mechanism for exploring large unknown datasets can be achieved by separating user intent and querying processes (Sections 5.3 and 6.7). Users observe visualisations and interact with them in order to satisfy their information needs. A logical separation of all the processes from each other ensures a generic manner of translating high level user intents (e.g. a user wants to compare two visualisations) to low level user actions (e.g. dragging a graph next to another to compare trends). By making use of basic, standard and generic rules, the processes are connected to each other — this ensures that while genericity is maintained, domain or user specific rules can be easily incorporated. Interpretation rules that translate user interactions to high level user intents as well as translation rules to convert the user intents into highly formalised queries are presented as tables in the Sections 5.7 and 6.10. These rules form the backbone of a generic interaction framework that supports the two developed solutions.

The third claim arises out of an initial literature survey, and follow up development of the second solution to explore and query large semantic data:

C3: It is possible to develop guidelines of aesthetic design for exploration of large scale semantic data by combining aesthetic design, Visual Analytics and Semantic Web principles. These guidelines can then be used to develop user interfaces to help explore large scale data in a generic, usable and pleasing manner.

The claim C3 is initially presented as a list of guidelines for developing aesthetic interfaces for visualising and exploring semantic data. These guidelines were devised after an extensive survey of the literature in aesthetic design, visual analytics and semantic web. Section 6.2 presents the guidelines in two forms: generic aesthetic principles for Semantic Web interface development, and node-link aesthetic principles for interfaces employing node-link visualisation to present data. Initial evaluation of the existing user interfaces in the Semantic Web motivates the need for an explicit focus on aesthetic design (Section 6.3). The guidelines were then consulted, while developing a solution for exploring and querying semantic data aimed at providing users with a generic, effective, usable and pleasing interface. The claim is justified by four evaluations: the first as a small formative evaluation with domain experts who provide initial first feedback to aid the next step of design (Section 6.9.1); the second in a comparative setting against other approaches (Section 6.9.2); the third in an extended evaluation that examines how users behave with longer exposure to system (Section 6.9.3). Following addition of a natural language component to the solution, the final evaluation compared how experts and casual users use the system for large datasets (Section 6.9.4). Positive responses from users, as well as the high usability scores show the value of the aesthetic approach, and serves to validate the genericity, usability and aesthetic value of the solution. The first evaluation (Section 6.9.1) was conducted with aerospace domain experts in Rolls Royce plc, while conducted in a knowledge management setting was in a small scale, and aimed at capturing qualitative feedback from the users from a knowledge worker/-manager's point of view. Hence, the evaluation served to indicate the potential of the solution within large organisational knowledge management frameworks, but a larger evaluation within such settings is needed to conclusively validate the applicability of such approaches, specifically within large organisations.

1.4 CONTRIBUTIONS AND THESIS STRUCTURE

This thesis presents a body of work that investigates how large semantic datasets can be explored in a generic, visual and pleasing manner. The first contribution is presented as sets of guidelines — the first set (Section 7.1.1) aims at mitigating the issues surrounding the technological limitations in the Semantic Web (as identified in Section 5.6.3); the second set provides a list of guidelines (Section 6.2) that can be used to design aesthetically pleasing solutions for the Semantic Web. The first set of principles have been developed as a result of reflecting upon the experience gained from developing the first solution, while the second set of principles were developed by conducting an extensive literature survey.

The development of technology is the next set of contributions, as they provide the technical mechanisms to help visualise semantic data and help alleviate the issues with

existing technology. These technologies are, in effect tangible outcomes of research, and following their implementation, the developer has a greater insight into potential issues, lessons learned and guidelines to help mitigate potential shortcomings. Two approaches are proposed in the thesis, by deconstructing the problem of consuming semantic data into visualisation and querying. The first approach, *.views. (points of views)* is a multi-visualisation approach that presents different facets of semantic data to facilitate exploration of data from different perspectives (Chapter 5). The second approach, *Affective Graphs* employs aesthetic design and Visual Analytics principles to explore semantic data in a graph-based visual manner, providing easy solutions to querying semantic data (Chapter 6). The novel contributions and lessons learned that had emerged from the design, development and evaluations of these solutions are presented in the Chapter 7.

Using various Visual Analytic techniques, several design decisions have been taken, which can be adopted in other systems and seed new design ideas. Another contribution, is highlighting the need for an active user participation, which is highly needed within the Semantic Web and Knowledge Management communities. A close cooperation with user communities in various aspects of the development of the systems using an iterative user-centered design methodology has been extremely helpful in bridging the gap between design, implementation and user needs. This methodology has not been exploited to its fullest potential in the community so far, and there is a greater need to do so. The ways in which active user participation has been a part of the development of the solutions is discussed further in the Chapter 4

Broadly, the thesis consists of five distinct parts: Introduction (Chapter 1), State of the art (Chapters 2, 3), Methodology (Chapters 4), Development of technology (Chapters 5, 6) and Conclusions and Future work (Chapters 7,8). The following paragraphs summarise individual chapters and the work included in the thesis:

Chapter 2 discusses the state of the art in Information seeking behaviour and Information Visualisation. This chapter presents an introduction to Visual Analytics and also sets the stage for the research, discussing the foundations of the Semantic Web, linked data, knowledge management and business intelligence.

Chapter 3 introduces discusses how semantic data has been consumed and what are the types of interfaces that have been developed so far. This chapter then relates semantics with principles of aesthetics by highlighting various principles that are particularly relevant for Semantic Web solutions.

Chapter 4 presents the goals of the thesis and lists the various requirements expected from the solution. These requirements are generic, and the proposed solutions are further elaborated in this chapter. The chapter then concludes by aligning the requirements to the various features of the solution.

Chapter 5 discusses the development of the first solution, *.views..* The chapter discusses the design rationale behind the framework and then details the architecture and implementation. The chapter then discusses the two types of evaluations and addresses Research Question R1, providing evidence to Claim C1.

Chapter 6 addresses Research Question R2, and starts by proposing design guidelines for building aesthetic interfaces for the Semantic Web identified from the literature. The chapter then introduces the second system, *Affective Graphs* and discusses its development and the various design decisions taken. The architecture and the technical imple-

mentation details are also discussed in the chapter. The chapter further discusses four evaluations that were conducted and addresses R2, and discussions provide evidence to C2.

Chapter 7 discusses the findings from the development of the two solutions, and draws from the evaluations as well as a higher level of reflection. The chapter also summarises the novel contributions from the two solutions as well as presents the lessons learned while developing, designing and implementing the solutions.

Chapter 8 is the last part of the thesis and discusses the impact of the work reported in the thesis, highlighting some of the success stories. This chapter also discusses how the research questions have been answered and supports the claims with evidences. The chapter then discusses how I propose to continue the research and where I would like to focus on, in the near future.

Part I

STATE OF THE ART

I tell you and you forget. I show you and you remember. I involve you and you understand

— Confucius (500 BC)

BACKGROUND

Looking for information is an integral part of life, more so in the digital age of highly connected devices, high performance hardware and massive stores of data. Collecting information to answer one's questions can be a highly complex process involving various sub tasks like active searching, subconscious decisions, passive learning and so on. Breaking down search processes into smaller elements based on theoretical foundations from multiple disciplines of study like psychology, biology, information science and so on can help analyse the processes involved in Information Seeking behaviour. These studies can be invaluable in understanding how efficient and effective search and Information Retrieval systems can be designed to help users find answers to their questions quickly and easily. Information is consumed by users as a result of several processes — active processes such as browsing and querying as well as inactive processes like responding to environmental stimuli, learning out of life experience and so on. A survey on Information Seeking behaviour needs to start from understanding the term information itself.

Several definitions have evolved over the years attempting to formulate, describe and explain 'Information' from an information sciences perspective. Although expressed with subtle differences, these definitions essentially describe information as a stimulus that can change one's knowledge about a particular topic. [Cas02] proposes that information can be defined as any difference one perceives in their environment or within themselves. [Kri83] defines information from the perspective of uncertainty, as "*any stimulus that reduces uncertainty*". [Mar97], on the other hand provides a more generic definition of information as "*anything that can change a person's knowledge*". Wilson, looking at the role of 'data' in describing information explains the difficulty in associating a single definition also stems from the ambiguity that exists among 'data', 'information' and 'knowledge' [Wil81]. Ackoff defined information by contrasting with data, knowledge and wisdom where data constitute raw observations and measurements; information represents purposeful messages, built out relationships and interconnections within data; knowledge is built by applying information and data; wisdom is evaluated understanding, created by reflecting upon knowledge [Ack89].

Several researchers have distinguished these terms over the years- although they have often been used interchangeably with each other, they are subtly different from each other [SA00]. Essentially, data represents real-world discrete facts [CDT00, dav97], in terms of simple observations [DP97], text and symbols that do not answer particular problems without being interpreted [QD99, vdS96, SA00]. Information brings meaning, relevance and purpose to data [vdS96, CDT00, DP97, SA00], intended to change the perception of the recipient [dav97]. It can be a flow of meaningful messages and facts [Nono8] that are organised to depict a condition or situation [Wii94].

Knowledge, on the other hand constitutes truths, justified beliefs, commitments, judgments, expectations, methodologies and so on [Wii94, Nono8, CDT00]. Knowledge comprises of valuable information from the human mind [DP97] that helps answer questions

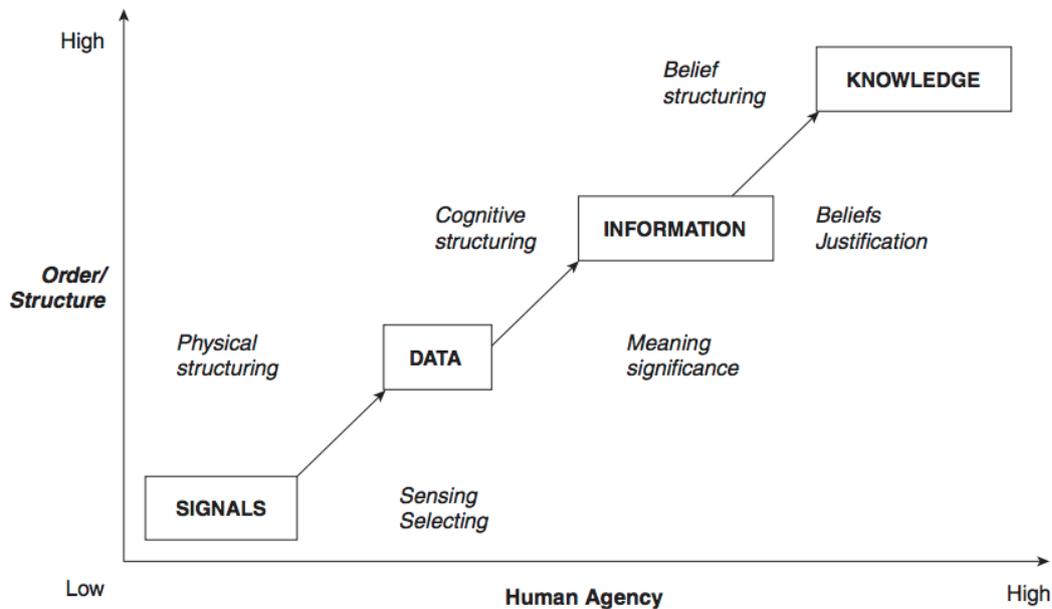
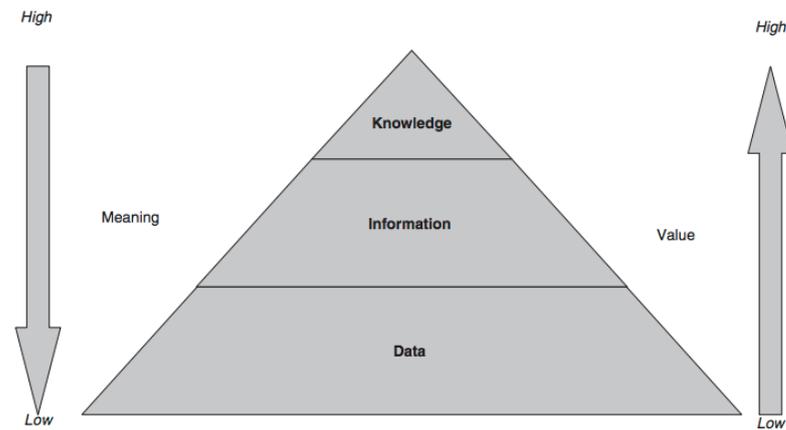


Figure 1: Choo's representation of data, information and knowledge. Image from [CDT00]

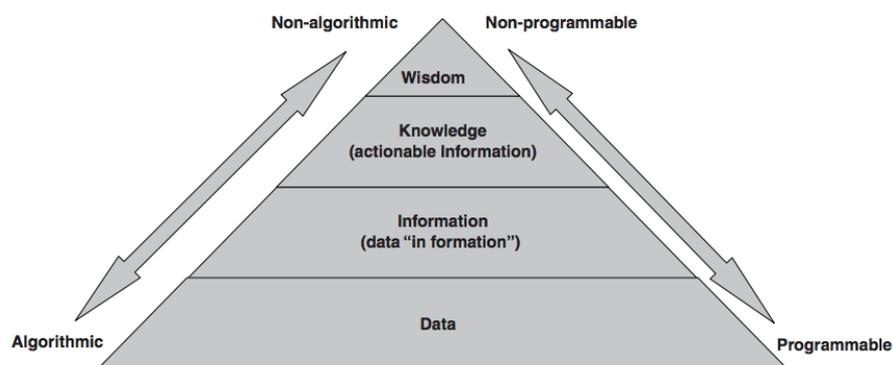
like why and how [QD99]. [BP97] define knowledge as “the invisible force that propels the most successful companies to stock market values which far exceed the visible assets of their financial balance sheet”, asserting a relation between knowledge to the financial success of a company. [SA00] describes knowledge as “the whole body of data and information that people bring to practical use in action, in order to carry out tasks and create new information”. The authors provide a nice example to distinguish the three — a morse code, consisting of a sequence of ‘on’ and ‘off’ signals is seen as raw data; the meaning attached to the signal e.g. an SOS message is seen as the information; information attached with purpose and competence, potentially causing an action is seen as knowledge e.g. rescue operation, upon receiving an SOS message. [AL99] provides a description of knowledge as “Knowledge is a justified personal belief that increases an individual's capacity to take effective action.”. [Tee00] notes “Knowledge is not primarily about facts and what we refer to as ‘content’. Rather, it is more about ‘context’. Knowledgeable people and organisations can frame problems and select, integrate and augment information to create understandings and answers”.

[TV01] further distinguishes knowledge into two types: individual and organisational. They describe knowledge as “the individual capability to draw distinctions, within a domain of action, based on an appreciation of context or theory, or both”. Proposing the role of individuals in contributing to the organisational knowledge, the authors describe it as “the capability members of an organisation have developed to draw distinctions in the process of carrying out their work, in particular concrete contexts, by enacting sets of generalisations whose application depends on historically evolved collective understandings”.

Several researchers have introduced ‘wisdom’ as another concept in the hierarchical representation of data, information and knowledge, also known as the DIKW hierarchy [Row07]. As with the others, there have been multiple definitions of wisdom over the years — [JV07] note wisdom to be an accumulated knowledge that allows one to understand how to apply concepts from one domain to a new situation ; [AG03] proposes



(a) The knowledge hierarchy, showing Data, information and Knowledge [CW04]



(b) The wisdom hierarchy, also referred to as DIKW hierarchy [AG03]

Figure 2: Representations of Knowledge hierarchy. Images from [Row07]

wisdom as “the highest level of abstraction, with vision foresight and the ability to see beyond the horizon”. [Jas11] describes wisdom as “the ability to act critically or practically in any given situation. It is based on ethical judgement related to an individual’s belief system”. Figure 29b describes how the DIKW hierarchy has been described.

2.1 INFORMATION BEHAVIOUR

Information Behaviour involves the study of the entire process of requiring, acquiring and using information. [Cas02] defines it as the “totality of human behavior in relation to sources and channels of information, including both active and passive Information Seeking, and information use”. This encompasses any information that has been gathered inadvertently such as glimpsing or encountering new information as well as information that is avoided or that does not evoke a response from the subject [Wiloo, Cas02]. While research into human Information Behaviour has historically incorporated studies related to Information Seeking, foraging, retrieving, organising and use [SC06], the focus of this discussion will be toward understanding the motivations and the processes involved with Information Seeking activities. Information seeking tasks typically arise out of an

Information Need, which motivates an individual to investigate if there is a necessity to look up different sources to find some information.

The progress in establishing a theoretical definition of the term ‘*Information Need*’ has been generally slow — mostly due to a combination of several factors like understanding context, difficulty in segregating ‘*wants*’ from ‘*expressed demand*’ and ‘*satisfied demand*’ and so on [Wiloo]. The slow progress in asserting a single definition to the many possible perspectives that Information Need has been explored [Kri83]. Broadly, Information Needs have been investigated from two perspectives — the nature of the need and the level of perception. While the nature of need could be a ‘*demand*’, ‘*want*’ or ‘*need*’, level of perception can be categorised as immediate (thereby requiring urgent action) or deferred (requiring an action sometime in the future).

[Caso2] defines Information Need as a “*recognition that your knowledge is inadequate to satisfy a goal that you have*”. Making sense of user needs is a complex task, one that involves understanding the knowledge of the user, system capabilities, context, human cognition and so on. It is important to note that a user’s Information Need should not be estimated based on the system being used, rather the main goal of the users. [SBC97] define Information Need as “*the perceived need for information that leads to someone using an Information Retrieval system in the first place*”. [Mar97] describes Information Need as the process in which humans purposefully engage in order to change their state of knowledge. [Atk73] defined Information need as “*a function of extrinsic uncertainty produced by a perceived discrepancy between the individual’s current level of certainty about important environmental objects and a criterion state he seeks to achieve*”.

2.2 INFORMATION SEEKING

Action motivated to address Information Needs, *Information Seeking* has been the subject of considerable research over the past decades — aimed at understanding search patterns, strategies and techniques. Information seeking, or the “*purposive seeking for information as a consequence of a need to satisfy some goal*” [Wiloo] can involve interaction with different sources of information such as the internet, magazines, articles, newspaper and so on. [SCo6] define Information Seeking as “*a subset of Information Behaviour that includes the purposive seeking of information in relation to a goal*”. Information Seeking and IR, though refer to finding information, are terms that are essentially different. More “*human oriented and open ended*”, Information Seeking relates to the process of acquiring information, whereas Information Retrieval implies finding information from a database management perspective, where individuals know a piece of information and thereafter organised it for future look-up by themselves or others [Mar97]. Although Information Seeking broadly relates to any attempt (both offline and online) at gaining information to satisfy one’s Information Need, most of the following discussions would be directed to Information Seeking in electronic environments.

[SCo6] describes how information searching, seeking and behaviour relate to each other as shown in Figure 3(a). Information behaviour is a wider field of study that includes environmental factors such as communication patterns as well as human behaviour. The authors further describe Information Seeking in two settings — work (occupation or school) related and non work related. Information searching is a narrower concept that is associated with humans using an Information Retrieval system. Informa-

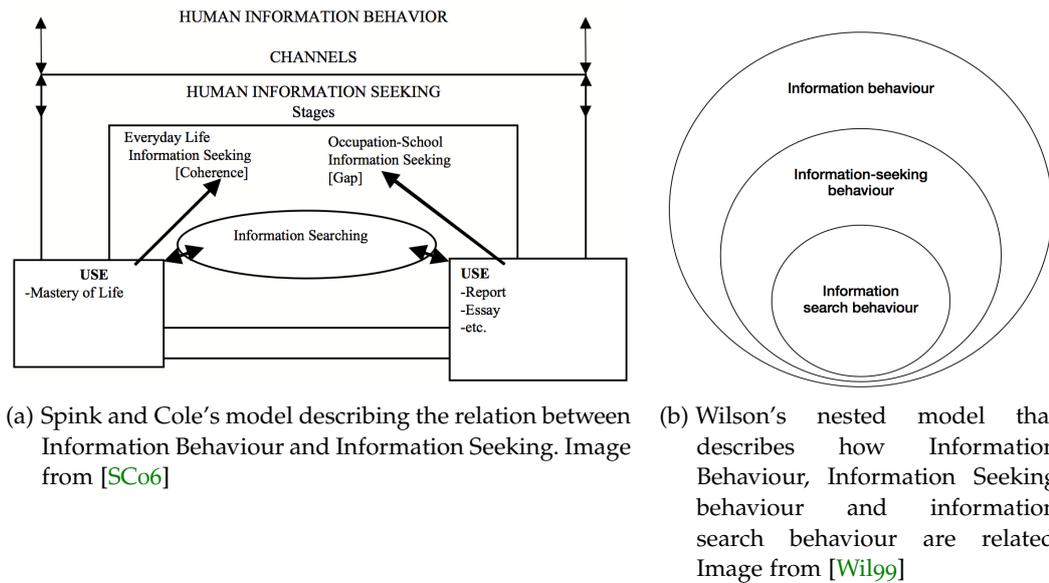


Figure 3: Wilson's and Spink and Cole's model describes the interplay among Information Seeking and Information Behaviour

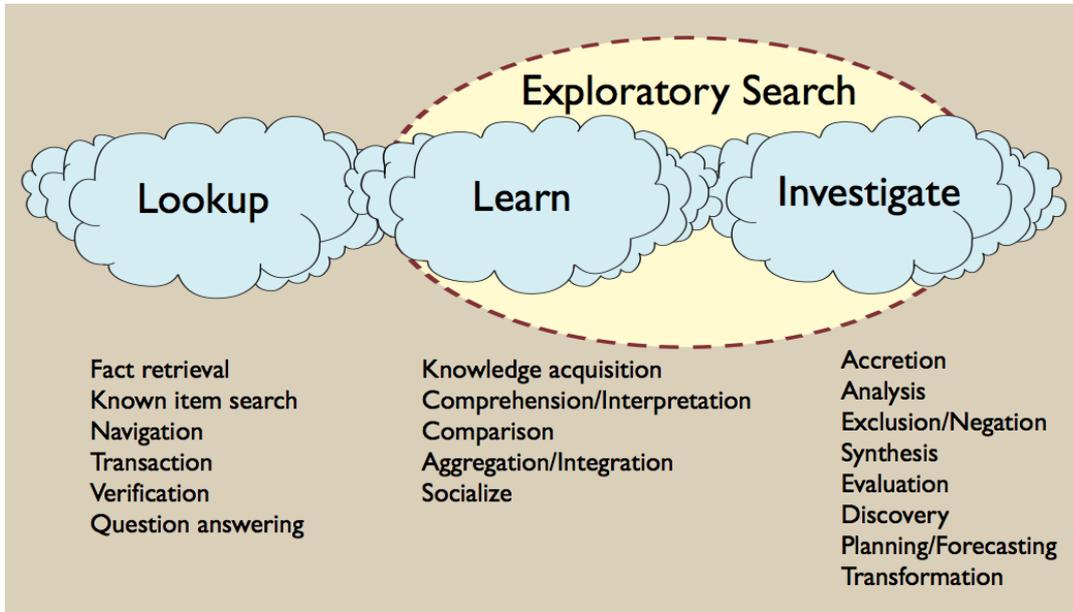
tion use is considered within the more specific context of information searching, as well as the broader context of Information Behaviour.

[Wil99] presents another model to describe how Information Behaviour and Information Seeking behaviour are related as shown in Figure 3(b). Wilson presents a nested model showing the different fields. Information behaviour can be seen as a general field of investigation, while Information Seeking behaviour can be seen as a sub-set of the field that encapsulates the different methodologies used by individuals to identify, discover and access information resources. Information search behaviour, on the other hand deals with the "interactions between information user (with or without an intermediary) and a computer-based information systems".

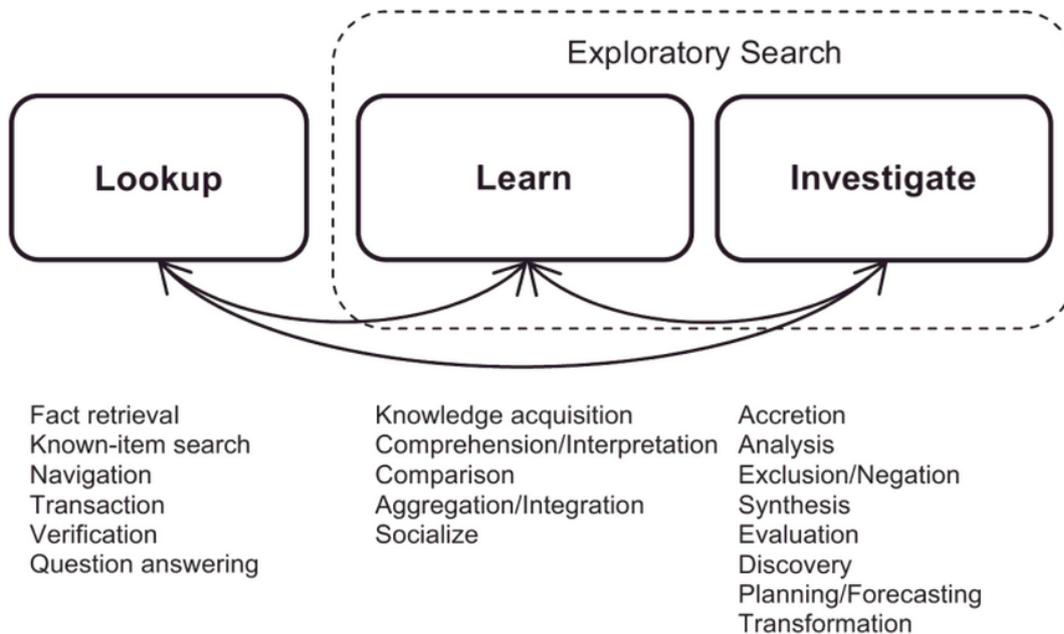
There has been significant research on understanding how humans look for information and identify the processes that we consciously or subconsciously undertake while attempting to answer our Information Needs. Over the past few decades, researchers have attempted to explain (and predict) the sequence of actions that people undertake while seeking information. This has given rise to many models (as flowcharts, diagrams, sequence of events etc.) which have evolved to well established theories for Information Seeking behaviour. While there are a few models proposed from a wider perspective that investigates the entire process of Information Behaviour, there are models that attempt to describe and explore the Information Seeking process.

[Cho99] notes that people make use of information and build knowledge, by utilising the representations, meanings and context within information and data. From an information management perspective, information gathered is typically stored, indexed and managed to improve future retrieval of the collected information that is encountered [Mar97]. However, from an end user's perspective, how a piece of information is delivered to him/her is equally important. [Bato3] identified the four Information Seeking modes — searching, browsing, monitoring and being aware. While Monitoring and being aware are two modes where information is received by a user while being

in a passive state, searching and browsing are the focus of discussion in this chapter, as it involves a user actively seeking for information. Users actively attempt to “answer questions or develop understanding around a particular question or topic area” in directed searching. Browsing, on the other hand involves users having “no special Information Need or interest”, but expose themselves to new interesting pieces of information. A shift from the “analytic approaches of query-document matching toward direct guidance at all stages of the Information Seeking process”, exploratory search is a different paradigm from the browsing and searching behaviour that Bates suggested [WDM⁺07].



(a) Exploratory search, as illustrated by [Maro6]



(b) The interplay of lookup, learning and investigating. Image from [WR09]

Figure 4: The role of lookup, learning and investigating in explaining exploratory search

[Maro6] describes search activities are typically classified into three types: *lookup* (carefully selected queries providing precise results), *learn* (development of new knowledge) and *investigate* (critical assessment of information like analysis, synthesis and evaluation) as shown in Figure 4 (a). Exploratory search, as suggested by [Maro6] is comprised of the learning and exploratory tasks. [WR09], on the other hand proposes that there is an interplay between the three tasks, that contribute toward an exploratory search activity — “lookup searches are embedded in learning or investigation, learning is an important part of investigation”. Thus, a better representation of exploratory search tasks would be as shown in Figure 4 (b). Lookup searches involve well directed queries to retrieving single facts and popular search engines like Google¹ and Bing² follow this approach. Exploratory searches, on the other hand, requires a greater user involvement, more system functionality and a greater interaction between users and systems.

Exploratory search activities can typically require multiple queries to formulate an understanding of an area of interest. It is also possible that exploratory search sessions can extend to days or weeks or even longer durations, depending on the task at hand. Most often, the motivation for such activities is an open-ended, persistent and multi-faceted Information Need. Such types of search activities may also require the engagement and collaboration of other individuals who are either information seekers or interested in the outcome of the tasks.

An Information Seeking behaviour, *browsing* is an informal and opportunistic activity that depends heavily on the information environment [Mar97]. Information seeking tasks that are not well defined or spread across multiple disciplines can be more effectively resolved by browsing. Gathering an overview of a certain topic or keeping informed about developments in a particular field can also resolved by browsing. There are several other reasons why information seekers resort to browsing as a method for finding information like clarifying an information problem, monitoring a process, discovering or learning, developing a formal strategy and so on. Another interesting reason is that browsing requires a smaller cognitive load as compared to analytical search strategies, which requires a lot of cognitive processing to recollect significant terms that are conceptually relevant to the topic of interest. Our perceptual abilities to recognise information dictates how useful our browsing experience would be.

There have been several ways of presenting information to users while they look for information — while performing a variety of activities like active searching using queries in an Information Retrieval system, casually looking for information using browsers, performing searches using visualisation tools and so on.

2.3 VISUAL ANALYTICS

Visual Analytics (Visual Analytics (VA)), or the science of analytical reasoning supported by highly interactive visualisation interfaces [TC05] offers help in providing means to users to understand their data. Interactive visual interfaces that visualise large volumes of highly complex domain-specific information can be immensely valuable for decision-makers as they reduce the time for analysing raw data. With information visualisation tools being widely available for users to gain an overview of the data, there has been sig-

¹ <http://www.google.com>

² <http://www.bing.com/>

nificant research into VA. Such technologies can be used to synthesise information and derive insight from massive dynamic, ambiguous and often conflicting datasets. Visualising datasets can not only help in detecting the expected trends and patterns, but also discover the unexpected. Over the past decades, there has been a paradigm shift from *confirmatory data analysis* to *exploratory data analysis* [Tuk77]. Confirmatory data analysis methods involve the user to formulate a hypothesis about the data which is then proved/disproved by gathering evidence from the data. Exploratory data analysis methods require the users to explore and analyse the datasets, without any pre-conceived hypotheses. The aim in such methods is to discover implicit information from the data. Though confirmatory data analysis by statistical methods can often provide means to answer known questions *the known unknowns*, visualisation tools can provide means to answer the questions unknown to us *the unknown unknowns* by exploratory data analysis. Where information visualisation aims only at visualising the underlying data, VA provides means for the user to interact with the visualisations, thereby encouraging discovery of new seemingly ‘hidden’ information. The power of VA lies in revealing ‘hidden’ trends or patterns in underlying data.

A multidisciplinary field, VA has several focus areas. Two main models have been proposed — The model proposed by Thomas and Cook is divided in four parts — *Analytical Reasoning techniques* to provide deep insights to users for supporting assessment, planning and decision making (Figure 5); *Visual Representations and Interaction Techniques* to exploit the cognitive processes involved in seeing, exploring and understanding large amounts of information; *Data Representations and Transformations* to convert data into visualisations to support analysis; techniques to support *Production, Presentation and Dissemination* of analysis efforts [TC05, TC06]. In an analysis activity, the last area (i.e production, presentation and dissemination) is often the most over-looked and time-consuming of all the areas [TC06]. However, this area is an integral part of VA activities, as it is the only process that is visible to larger audiences. Hence, it is extremely important to produce suitable material that ‘traps’ analysis activities, presents them in a meaningful way and disseminates to the right audiences in the appropriate manner.

An alternative model (Figure 6) provides a more detailed view of the scope of VA, by combining the ‘advantages of machines with strengths of humans’ [KMSZ06]. This model explains how VA effectively integrates several research areas like *Information visualisation*, *Human Factors* and *Data Analysis*. Methodologies from Information Analytics, Geospatial Analytics and Scientific Analytics are integrated to provide Information Visualisation systems. Human factors (Interaction, Cognition, Perception, Collaboration, Presentation and Dissemination) are essential to provide a coherent link between humans and computers. Methodologies developed in Knowledge Discovery, Data Management, Knowledge Representation and Statistical Analysis also contribute to an effective VA system, to support data analysis.

In spite of the potential benefits of VA in understanding and discovering information, most visualisation methods fall short in visualising high volumes of data. [KMSZ06] discuss their views on the problems associated with visualising large-scale datasets. Loading large datasets and computing individual visual items to provide interactive visualisations requires high computational capability, often unavailable in many organisations. The visualisation interfaces are often restricted by the medium itself — i.e. the computer screens: the number of pixels available for applications is limited, hence, pro-



Figure 5: Focus areas of Visual Analytics, as illustrated in [TC05]

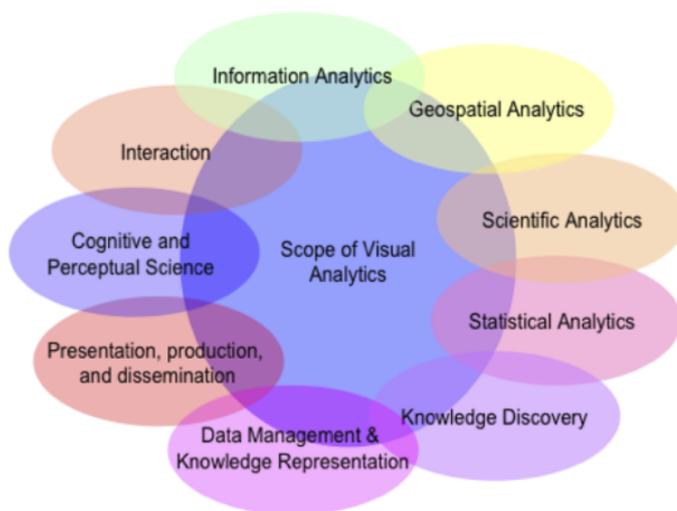


Figure 6: A more detailed view of the various research areas contributing to Visual Analytics, illustrated in [KMSZ06]

viding inappropriate visualisations which require extremely high resolutions for the visual interfaces available. While certain data reduction techniques like filtering, aggregation, compression, principle component analysis provide help in scaling down large data into a smaller section to increase visibility and comprehensibility, they result in loss of details. Even if visualisations are obtained for complex large-scale data, it might not necessarily be effective for human comprehension, reasoning and cognition. Poorly designed visualisations can be misunderstood, misrepresented and misleading and care should be taken to ensure that the visualisations are intuitive, aesthetic and presentable to users.

2.3.1 Interaction with Visualisations — Dynamic Query

Direct manipulation of databases by using interactive graphical widgets provides ways for users to formulate complex queries to explore their data sets [AWS92]. Interacting with databases mostly involve the users to type queries in high level query languages like Structured Query Language (SQL). This requires expert knowledge on the query languages, specially if the users require to formulate complex queries. In addition, it also requires knowledge of the underlying databases. However, for novice users to significantly benefit from such techniques require them to devote a lot of time and effort for learning. Dynamic queries provide the users with interactive and simple ways to formulate complex queries. [GR94] suggests three goals of direct manipulation- select relevant sub sets of data; select the appropriate attributes of the data; and aggregating data elements. Dynamic query allows the users to manipulate interactive elements in interfaces to control different attributes of the data being visualised. The effect of such an approach is immediately conveyed to the users as they can see the changes on the visualisations. Such an approach can benefit the users immensely in the following ways:

- Easy to learn and remember (esp. for novice users)
- Flexible, easily reversible actions
- User is always in control
- Provides instant visual feedback for actions
- Limits the types of errors
- Reduces need for error messages

[WS92] presents an interface, Dynamic HomeFinder which applies dynamic query mechanisms in a real-estate scenario. The interface presents the users with several query widgets and a map of an area (Figure 7). Users can manipulate these query widgets to identify houses that fulfill their requirements. All the queries are then ANDed and passed to the database to select the relevant results. The interface on the right presents all the ways the users can interact with the database in order to select the all houses that match the user's criteria. The benefit of such mechanisms can be clearly seen from their results (Figure 8). The results show that there is a significant reduction in time taken for users to find the required houses. Dynamic query mechanism was tested with paper based and natural language query based ways of finding data.

2.4 THE SEMANTIC WEB

In 2001, Tim Berners-Lee, the inventor of HyperText Markup Language (HTML), the basis of World Wide Web proposed the Semantic Web (SW), an extension of the existing web [BLHL01]. The main drawback of existing web technologies is that the web is currently built for humans to read. Machines are unable to perform intelligent processing on these pages, apart from inferring meanings from the layouts of pages or connections between them and they fall short of understanding the meaning to a web page. This limits the capability of existing tools to perform intelligent functions like agents automatically

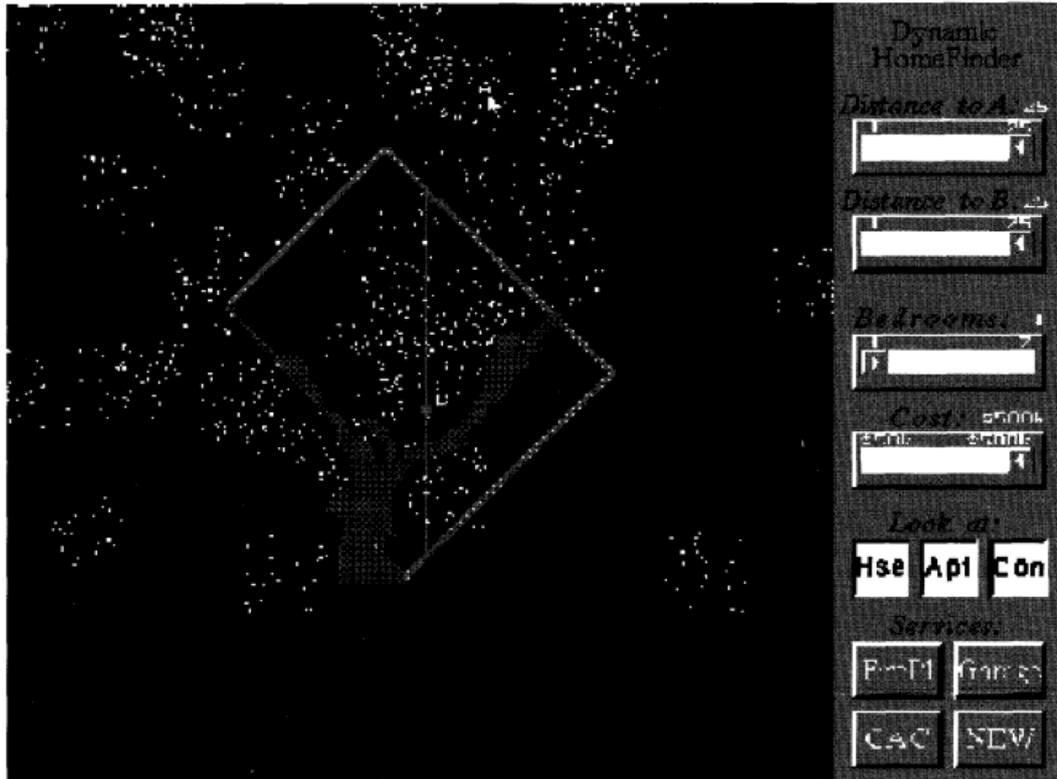


Figure 7: Dynamic HomeFinder — an example of dynamic query being used in a real-estate scenario. Image from [WS92]

querying services. The Semantic Web empowers machines to “understand” information presented in the web, by making use of the structured information associated with each web page. *SW* aims at providing machines with the ability to consume vast amounts of information automatically.

In order to consume data, machines should be able to reason upon structured information using inference rules. However, for this process to be generalised, the machines must use standardised technologies. [BLHL01] proposed the use of Extensible Markup Language (*XML*) (eXtensible Markup Language) and Resource Description Framework (*RDF*) (Resource Description Framework) for this purpose. *XML* provides the ability for users to define their own terms for annotating information they publish on their web pages. This can be hidden from view of humans so that only machines can extract information based on these user-defined tags. Scripts can then scrape the web pages to identify user-defined tags to obtain a highly structured set of information. However, machines can make use of the structured information only if they are aware what the tags actually mean. Different users might use various tags in their own ways to describe similar objects (example, a ‘car’ could also be referred to as a ‘vehicle’ or ‘automobile’) or use the same tag to describe different objects (example, ‘school’ could refer to an educational institution or a collection of fish; ‘bank’ could refer to a financial institution or the edge of a river).

This ambiguity can cause a hindrance to software programs reasoning on such data. Context is often useful in such situations, but not always the ideal solution. Universal

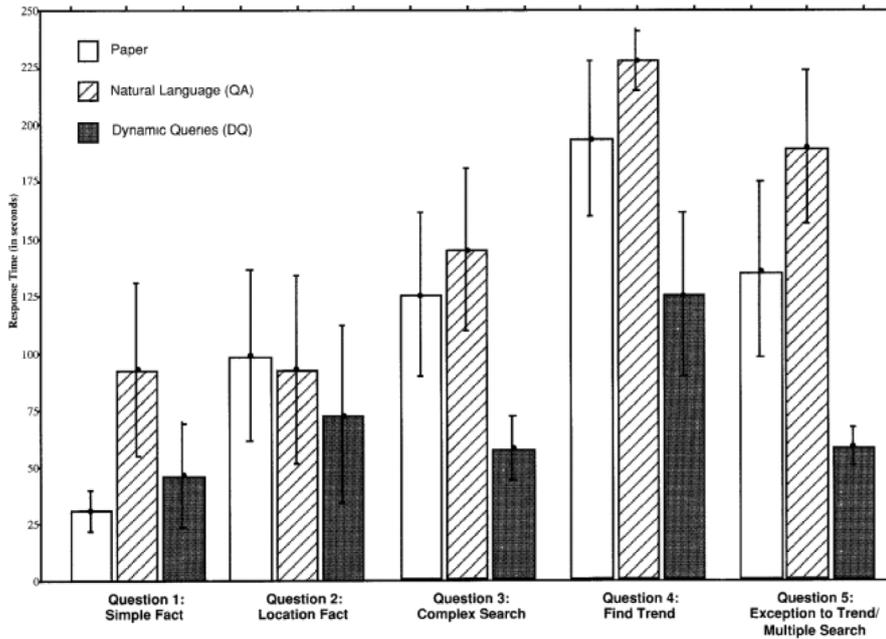


Figure 8: Results obtained by comparing the time taken by users interacting with dynamic query mechanisms, natural language query interfaces and paper based searching. Image from [WS92]

Resource Identifiers (Universal Resource Identifier ([URI](#))) are unique identifiers that can be assigned to every unique concept. Users can define their own [URI](#) for a concept that they want to introduce. [URIs](#) can be re-used and shared to avoid duplication of effort. [RDF](#) attempts to encode meanings between these [URI](#) concepts by associating them into triples. Following from natural languages, [RDF](#) triples contains Subjects, Predicates (also known as property or relation) and Objects. Subjects may be a blank node or a [URI](#) reference; objects may be a blank node, literal or [URI](#) reference, while predicates are [URI](#) references. [RDF](#) triples are conventionally written in the order of subject, predicate, object. Sets of [RDF](#) triples constitute to form an [RDF](#) graph, where the nodes are the subjects and objects of triples and the links between the nodes are the properties or predicates [Cono4].

```
<rdf:Description rdf:nodeID="node14ptj21ncx6627">
  <gml:pos>-3.5666667,54.65</gml:pos>
</rdf:Description>
```

The above [RDF](#) statements describe a node with [GPS](#) coordinates, where the node `14ptj21ncx6627` (subject) has the literal `-3.5666667,54.65` (object) as position (property). These descriptions can be used later on to describe other things, such as:

```
<rdf:Description rdf:about="http://twitter.com/chris-smith26/6330626216">
  <sioc:hasCreator rdf:resource="http://twitter.com/chris-smith26"/>
  <created xmlns="http://purl.org/dc/terms/">2009-12-04 07:24:26.0</created>
  <rdf:type rdf:resource="http://rdfs.org/sioc/ns#Post"/>
  <sioc:hasContent>RT @wigtononline: Cumbria school kids song for
  #CumbriaFloods http://tinyurl.com/ye39qtg Charity helps victims
  Please retweet
</sioc:hasContent>
```

```
<subject xmlns="http://purl.org/dc/terms/">CumbriaFloods</subject>
<itr:has_Localization rdf:nodeID="node14ptj21ncx6627"/>
</rdf:Description>
```

The above statements provide an example where a social post (tweet) on Twitter has been described by the node (ID = node14ptj21ncx6627), identifying the geographic location where the tweet was posted from.

Triples can be serialised as [RDF](#) files or dumps, which can then be loaded on to triple stores like Sesame³, Virtuoso⁴ or Jena SDB⁵ for storing and querying. Triple stores or RDF stores are graph based frameworks that support storing and querying semantic data, much like a Database Management System (DBMS) in the traditional database world [[OSGms08](#), [YWZ⁺08](#)].

In order to provide a more compact, natural and readable alternative to [XML](#) syntax for defining [RDF](#), [[BLCo8](#)] specifies the Notation 3 language. The language aims to improve the ways [RDF](#) is expressed and allow smoother integration of rules using variables, formulae and quantifications (such as @forSome, @forAll) into [RDF](#). The readability of [RDF](#) triples is improved by using features like [URI](#) abbreviation; repetition of objects for the same subject with commas; repetition of predicates for the same subjects with semicolons; blank node syntax; using formulae within “{” and “}”; grammar rules.

[[AB08](#)] proposes a way of integrating web sites and [RDF](#) data by including [RDF](#) statements within a page’s content. Making use of the extensible nature of XHTML, publishers can embed Resource Description Framework in attributes ([RDFa](#)) statements in web pages, which can be parsed by programs that extract and index [RDF](#) statements. Ivan Herman’s site, (<http://www.w3.org/People/Ivan/>) provides some examples of embedding [RDFa](#) statements within a web page —

```
<span rel="foaf:holdsAccount">
<span about="http://www.facebook.com/ivan.herman" typeof="foaf:OnlineAccount">
<a rel="foaf:accountServiceHomepage" href="http://www.facebook.com"> facebook</a>
(acc. name <span property="foaf:accountName">ivan.herman</span>),
</span>
</span>
```

The [HTML](#) code shows an example where the author provides information about his Social Networking sites using [RDFa](#) statements. The ‘about’ statement on the second line is used to indicate that the facebook link is a type of online account, which is defined in the Friend Of A Friend ([FOAF](#)) ontology. Similarly, the [FOAF](#) ontology is referred to in the fourth line while mentioning the account name. Programs (such as Sindice) scraping the website will look for these statements and build the [RDF](#) triples accordingly, and can load them into a triple store to be available for querying.

While the Resource Description Framework ([RDF](#)) is a graph-based data model for representing semantic information, [SPARQL](#)⁶ is the proposed query language for [RDF](#). The RDF Data Access Working Group released a first public working draft of [SPARQL](#) in 2004, which presently is now a World Wide Web Consortium ([W3C](#)) Candidate Recommendation [[PAG09](#), [QL08](#)]. [SPARQL](#) is a graph-matching query language, where patterns

3 OpenRDF Sesame, <http://www.openrdf.org/>, Last accessed 25/06/12

4 OpenLink software Virtuoso, <http://virtuoso.openlinksw.com/>, Last accessed 25/06/12

5 Jena SPARQL Database, <http://openjena.org/SDB/>, Last accessed 25/06/12

6 <http://www.w3.org/TR/rdf-sparql-query/>

in the data are matched with ones specified by users, based on specific restrictions. Outputs of [SPARQL](#) queries are of different types, based on specifications of the queries, such as yes/no, value of variables matching patterns, construction of new triples from values and descriptions about resources [[PAG09](#)]. [SPARQL](#) queries are constituted of triples, where each subject, predicate and object can be either a variable (indicated by a preceding '?') or value. The following set of examples illustrate some [SPARQL](#) queries, based on the Semantic Web Dog Food⁷ data.

```
SELECT DISTINCT ?object WHERE {
?subject a ?object
}LIMIT 100
```

The example is a very basic query to list 100 distinct concepts that exist within the dataset. The value of the predicate, 'a' (which is treated as a keyword, as an alternative for the <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>)⁸ restricts the returned triples to only those triples that describe the type of subjects.

```
SELECT DISTINCT ?subject WHERE {
?subject a <http://swrc.ontoware.org/ontology#InProceedings>
}LIMIT 100
```

The above example shows a basic query that lists 100 resources that are the type of `InProceedings`. Here, the only variable, subject is restricted by the value of the object (<http://swrc.ontoware.org/ontology#InProceedings>) and predicate (a).

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX purl: <http://purl.org/dc/elements/1.1/>
PREFIX swrc: <http://swrc.ontoware.org/ontology#>
SELECT DISTINCT ?label,?creator,?year WHERE {
?subject a <http://swrc.ontoware.org/ontology#InProceedings>.
?subject rdfs:label ?label.
?subject purl:creator ?creator.
?subject ?prop ?year.
}
```

The above example provides a slightly more complicated query, looking for the label, authors (creator) and publication year of all 'in proceedings' articles. Further pattern-matching based on values of variables are provided by filter mechanisms.

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX purl: <http://purl.org/dc/elements/1.1/>
PREFIX swrc: <http://swrc.ontoware.org/ontology#>
SELECT DISTINCT ?label,?creator WHERE {
?subject a <http://swrc.ontoware.org/ontology#InProceedings>.
?subject rdfs:label ?label.
?subject purl:creator ?creator.
?subject ?prop ?year.
FILTER (?year=2009 AND regex(?label,"visual","i")).
}
```

The example shows a more complex query, listing the label and authors of all the publications in the year 2009⁹ and the label contains the term 'visual' (case insensitive).

⁷ <http://data.semanticweb.org/>

⁸ <http://www.w3.org/TR/rdf-sparql-query/#abbrevRdfType>

⁹ Note that this query would not work in the [RDF](#) dump provided by Semantic Web Dog Food, since the year is represented as a string variable. In such a case, the filter triple should be set as `FILTER (?year="2009" AND regex(?label,"visual","i"))`

2.4.1 Ontologies

Since users do not have control on other user's concepts and URIs, the same concepts may be defined by different URIs. This could cause confusion among software agents when analysing information shared among different users. Large sets of RDF data using the URIs could vary among different publishers. RDF schemas¹⁰ could help in providing a level of abstraction from the data to indicate the purpose of classes and properties by providing further descriptions. However, they are incapable of providing means to reason over the data. Ontologies, which are formal specifications that conceptualise a domain can provide advanced ways of reasoning over RDF data. Such specifications provide a structured way of identifying the concepts in a domain and their relations with each other. Inference rules set up by their authors can also be used by software agents for reasoning. Different ontologies can also be 'mapped' together using equivalence relations. Data sources, based on different ontologies can be connected together using these equivalence relations. The Web Ontology Language (Web Ontology Language (OWL), [MH04]) is used to define ontologies. OWL provides three sub-languages, depending on the requirements of the implementers — OWL *Lite*, OWL *DL* and OWL *Full*, arranged according to levels of expressivity. The OWL vocabulary allows users to describe various relations among classes and properties like disjoints, cardinality, unions, intersections and so on. The following example shows a portion of an ontology:

```
<owl:Class rdf:ID="Call">
  <rdfs:subClassOf rdf:resource="http://xmlns.com/wordnet/1.6/Announcement"/>
  <rdfs:label>Call</rdfs:label>
</owl:Class>

<owl:Class rdf:ID="AcademicEvent">
  <rdfs:subClassOf rdf:resource="#OrganisedEvent"/>
  <rdfs:label>Academic Event</rdfs:label>
</owl:Class>

<owl:ObjectProperty rdf:ID="forEvent">
  <rdfs:domain rdf:resource="#Call"/>
  <rdfs:range rdf:resource="#AcademicEvent"/>
  <rdfs:label>for event</rdfs:label>
</owl:ObjectProperty>
```

The above statements are extracts from an ontology¹¹ discussing a conference (European Semantic Web Conference, ESWC 2006¹²). The statements show two classes, 'Call' and 'AcademicEvent'. An object property linking these two classes is shown, where the statements would be interpreted as Calls (which are a type of Announcement, described by the resource '<http://xmlns.com/wordnet/1.6/Announcement>') are being announced for Academic Events (which are a type of Organised Events)

2.4.2 Linked Data

LD, or the "set of best practices for publishing and connecting structured data on the web" [BHBL09] has been exponentially rising over the past few years. Since traditionally

¹⁰ <http://www.w3.org/TR/rdf-schema/>

¹¹ ESWC Conference Ontology, <http://www.eswc2006.org/technologies/ontology-content/2006-09-21.rdf>, Last Accessed 25/06/12

¹² European Semantic Web Conference, <http://www.eswc2006.org/>, Last Accessed 25/06/12

data providers have been publishing data on the web as CSV (Comma Separated Values is a file format that can store tabular text data, where each column value in a row is separated from the next by a comma) or XML¹³ (Extensible Markup Language is a set of rules for saving data where data values are marked up within conceptual tags) files, or HTML tables, a lot of the structure and semantics of the data cannot be maintained. Linked Data, or the Web of data refers to data on the Web, where it is machine-processable and is linked to other data sets, which may be further linked to several other data sets. Unlike traditional Web of hypertext documents, which uses HTML for describing links between hypertext documents, the Web of Data connects arbitrary things described in RDF by describing their relationships. [BL06] provides the following set of principles or guidelines for data providers to publish and connect their data, while adhering to the architecture and standards of the Web

- Identify things by URIs to enable people/objets to look them up
- Provide relevant and useful information using the standard presentations (RDF/XML) when anyone looks them up
- Include links to other URIs to maintain data interconnections and help discovery of more things.

These principles make the Web of Data a “tightly interwoven” layer with the traditional Web. The Linking Open Data project¹⁴ is a community effort to publish various data sets under open licenses as RDF on the Web, connected with RDF links by following the underlying principles of publishing LD. The project, started in 2007 provides a great example for demonstrating the value of SW and LD. Users and programs can browse, visualise, query or reason over this cloud of large, interlinked data. The data size and coverage has been rapidly growing over the past few years, starting in 2007 with roughly 500 million RDF triples and 120,000 RDF links between data sources. As of September 2011, the size of the cloud was over 31 billion RDF triples and 504 million RDF links¹⁵. Developers can gain access to such data by querying publicly accessible LD endpoints. However, the data is unknown to developers or end users, as the data is not visible unless the endpoints are queried upon.

The following section investigates knowledge practices employed within organisations, knowledge workers and user communities that are existent within organisations. Understanding the various knowledge practices in companies and their existing Knowledge Management problems can be helpful in identifying solutions that can span across the hierarchical structure of organisations. The section also presents a few Business Intelligence solutions that are available as off-the-shelf products.

2.5 KNOWLEDGE MANAGEMENT

Businesses are presently facing ever increasing challenges in an era of highly dynamic and uncertain economic conditions. One of the most important assets of organisations,

¹³ XML, http://www.w3schools.com/xml/xml_what_is.asp, Last Accessed 25/06/12

¹⁴ Linking Open Data Project, <http://esw.w3.org/topic/SweoIG/TaskForces/CommunityProjects/LinkingOpenData>, Last accessed 25/06/12

¹⁵ <http://thedatahub.org/> contains thousands of datasets that can be downloaded by users and developers

knowledge is a key factor in determining the financial success of a company. While the combination and application of tangible resources are responsible for deploying services and manufacturing products, an organisation's know-how is ultimately what drives such activities [ALo1]. The success of an organisation depends highly on how well the company exploits their knowledge and memory [DFHo3]. The discussion in this chapter and the thesis is mostly on the KM efforts in knowledge-intensive firms as opposed to labour or capital intensive firms. Such firms are typically characterised by significant problem solving and non-standardised production; creativity on the part of the organisational environment and the practitioner; high reliance on individuals and less on capital; most employees require higher education level and professionalisation; materials and assets are not the central factor, most important factors are in the minds of employees, networks, customer relationships, manuals and so on; high dependence on the loyalty of key personnel and vulnerability when they leave the organisation [NAEWo6].

With shifting markets, technological advancements, increase in global competition and rapidly increasing innovative products, organisations need to continually re-define themselves to stay ahead in the competition. Companies employing advanced knowledge practices to create, use and disseminate information within themselves to build new technologies and products can gain a significant competitive advantage in a highly uncertain era [Nono8]. Organisations should also leverage existing knowledge and create new knowledge by developing an "absorptive capacity" which can help use existing knowledge to recognise new information, evaluate, assimilate and use it to develop new knowledge and capabilities [CL90, GMSo1].

[SAoo] lists a few benefits of knowledge engineering in general, as a field of study: identify opportunities and bottlenecks in the development, distribution and application of knowledge within organisations; gather a greater understanding of the structures and processes applied by knowledge workers; build better, easier and maintainable knowledge systems. Knowledge systems, employing methodologies developed in knowledge engineering research aim to acquire, capture, categorise, classify, archive, manage and present organisational knowledge in a way that is easily accessible to users. Such systems can help organisations in many ways — faster decision-making, increased productivity and increased quality of decision-making, enhanced problem solving to name a few. [BGG96] discusses their findings on the expected benefits of knowledge systems and if such benefits were realised, as a result of an empirical study on survey data collected from questionnaires to individuals in the industry and organisations.

2.5.1 Business Intelligence

Business Intelligence (BI) is a concept of management philosophy that involves the collection of processes and software to support understanding of large datasets, retrieval and analysis of information and making decisions within organisations [EB11]. Several descriptions of Business Intelligence exist: [PLK05, GK86, GG86] describes BI as a concept and a tool that helps organisations manage and refine information and make effective decisions. [PLK05] points that BI includes two aspects: the information and knowledge that describes the organisations, processes, customers, competitors, economic and financial status; and the processes that produces the intelligence. [Rud09] describes BI as the capabilities that are required to turn data into intelligence. [Sho13] surveys the

literature and discusses that BI has been defined from two main standpoints — a technology view and a process view. From the technical view, BI involves a set of technologies that facilitates the following processes in a sequential manner: gathering and storage of data; transformation of data to information; transformation of information to knowledge; and utilisation of the knowledge to take decisions. The process view defines BI as a continuous supporting process of decision making where internal and external data is gathered, analysed and aggregated to provide insights.

In their survey paper, [CCC⁺09] notes that the various definitions of BI can be grouped into three categories: managerial/process; technological; product. The managerial/process-based definition describes BI as the process of gathering data from internal or external sources and its subsequent analysis to aid decision-making processes. The technological definition describes BI as the tools and technologies that allow recording, recovery and analysis of information. The product-based definition describes BI as the result/product of extensive analysis of detailed business data and analysis practices using BI tools.

Since the 1970s, early systems at assisting decision-making processes within organisations were introduced as Decision Support System (DSS), which took several forms — executive information, online analytical processing (OLAP) and predictive analytics [WW07]. Gartner analyst Howard Dresner in the 1990s coined the term BI as “a set of concepts and methods to improve business decision making by using fact-based support systems” [Pow03, WW07]. Power provides a comprehensive history of how Decision Support Systems developed and evolved. In fact, Dresner noted that the very definition of BI has been distorted over the years with different vendors marketing their BI solutions in their own ways [Koto6]. [Sho13] notes that a disagreement in formalising a definition of BI among researchers with the term BI has also been due to the confusion between BI and Competitive Intelligence (CI) — where CI encapsulates the entire competitive environment including the internal and external information. While considerable difference exist in a formal definition, the essence of BI remains utilising data to aid decision-making processes within the organisation, by translating data to information and knowledge.

Gartner, in their survey of 1400 chief information officers noted that BI projects were the number one priority, and are recognised as instrumental in driving business effectiveness and innovation [WW07]. [CKKSo2], in 2002 noted that the two major technology investments by organisations that have shown good return on investment are KM and BI, and predicted the two technologies would blend, where techniques from one is translated into the other. While BI is a set of tools and technologies to gather, access and analyse large amounts of data, KM represents a set of practices of the creation, development and application of subjective human knowledge. [WWo8] proposes the application of data mining to bridge the gap between the two fields.

2.5.2 Business Intelligence Systems

Several players exist in the Business Intelligence field, and organisations presently invest heavily in developing solutions for the enterprise. Software giants such as Microsoft, IBM, SAP and Oracle have all stepped into the BI field, establishing themselves as major vendors. One of the most popular solutions was developed by Cognos, which was later acquired by IBM in 2008. IBM Cognos (Figure 9, top left) offers several functionalities to help decision-making processes: query and reporting, performance dashboards, perfor-

mance metric monitoring, analysis of trends and patterns, collaborative features, mobile, real-time performance monitoring and advanced visualisations¹⁶. The Cognos suite is also integrated with other IBM products such as ManyEyes, SPSS Predictive Analytics, Lotus Connect and so on. By integrating IBM Cognos with IBM's popular ManyEyes program, users have a large range of visualisations to choose from. The type of data can be matched to the visualisation, which can in turn be composed into reports, or dashboards to monitor processes. The framework is flexible to encode different types of data with visual properties such as position, shape, size, color etc. [Ili13] provides a tutorial for how visualisations can be built based on the data and the best practices of doing so using the IBM Many Eyes system¹⁷. The software is used by thousands of companies around the world¹⁸, and is ranked in the leaders of Gartner's Magic Quadrant for Business Intelligence¹⁹²⁰. Gartner's Magic Quadrant report [RSSHo9] notes infrastructure, metadata management, workflow and collaboration, reporting, querying MS Office integration, advanced visualisation and scorecarding as the strengths of the software, while highlighting poor performance as a drawback. Another major vendor, SAP entered into business intelligence field with the acquisition of Business Objects. The SAP Business Intelligence Solutions (Figure 9, top right) provide a large range of solutions with similar features as IBM Cognos, such as reporting, dashboard, mobile and integration with different data sources²¹. SAP Lumira provides data visualisation facilities and can be used to visualise and aggregate from different datasets. Reporting and ad hoc query capabilities were noted to be as the top strength for the software, while low levels of customer support was noted to be a drawback [RSSHo9].

Another major player, Microsoft entered the BI market fairly late with a competitive price and a well-packaged BI product²²(Figure 9, middle left). The software, packaged with other Microsoft products such as Excel, SQL Server, SharePoint Server along with the large user base of Microsoft application developers makes Microsoft's BI a highly prospective solution [RSSHo9]. Microsoft's BI product covers three categories: Personal, Team and Corporate BI²³. Four main technologies drive the solutions: Excel (personal), Excel Services and Office SharePoint Server (team) and PerformancePoint Server (corporate). Personal BI enables users to create quick spreadsheet based reports, perform quick analysis on data from different sources with a minimal Information Technology (IT) requirement. Team BI enables users to share their analysis in dashboards, securely accessible to collaborators. It also allows integration with Microsoft Office products to facilitate collaborative activities. Organisation BI, on the other hand provides a wide range of organisational facilitates such as planning, budgeting, forecasting tasks. Users can build single dashboards and consolidated reports containing KPIs, analytic reports and views on the data sources. Long development cycles, poorer metadata management, differ-

16 from <http://www-01.ibm.com/software/analytics/cognos/solutions.html>

17 https://www-304.ibm.com/connections/blogs/predictiveanalytics/entry/extensible_visualization_brings_new_visualization_options_and_advantage_to_cognos_active_report?lang=en-us

18 <http://www-01.ibm.com/software/uk/data/cognos/>

19 <http://en.wikipedia.org/wiki/Cognos>

20 <http://searchcio.techtarget.com/definition/Cognos>

21 <http://www.sap.com/pc/analytics/business-intelligence/software/overview/index.html>

22 <http://www.microsoft.com/en-us/bi/Products.aspx>

23 [http://technet.microsoft.com/en-us/library/dd365010\(v=office.12\).aspx](http://technet.microsoft.com/en-us/library/dd365010(v=office.12).aspx)

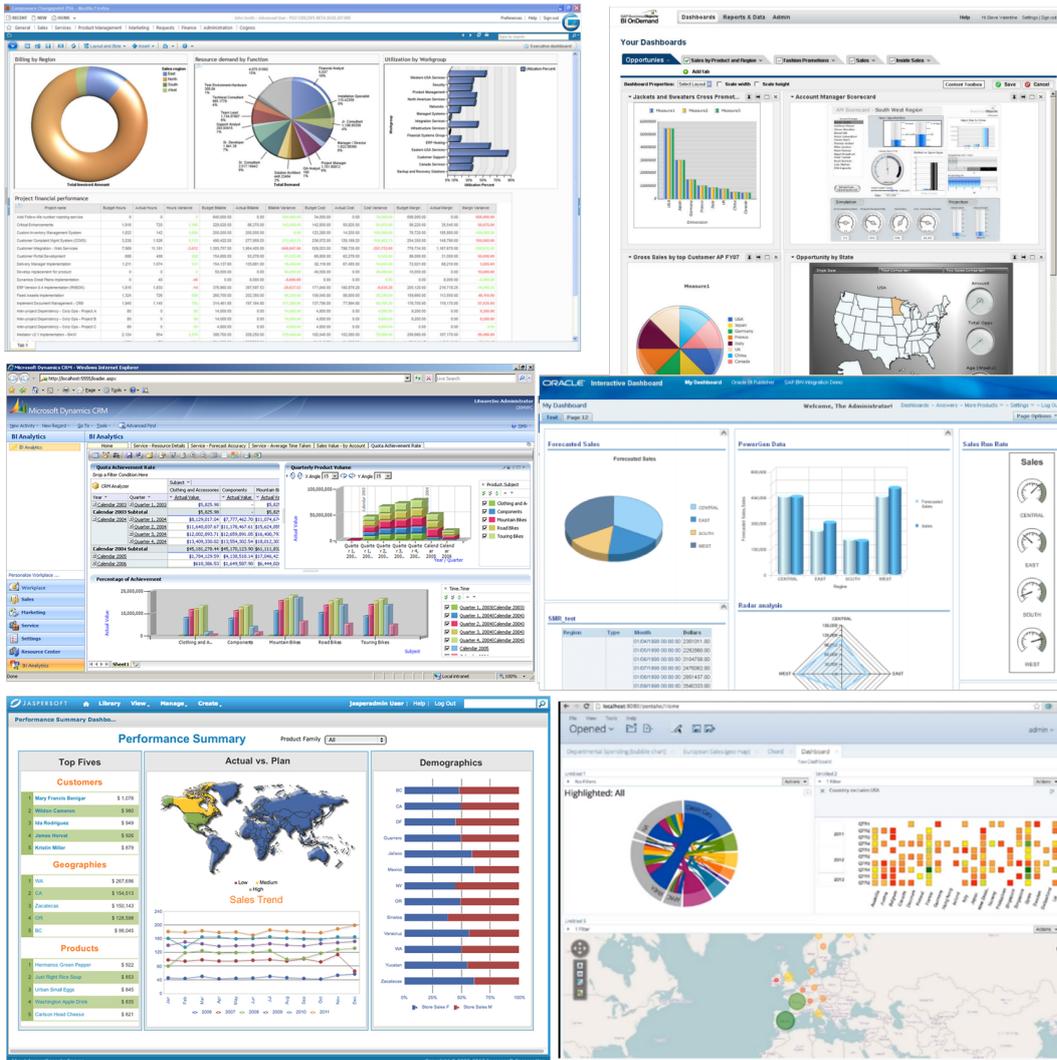


Figure 9: Examples of Business Intelligence visualisation dashboards. Top left: IBM Cognos (Image from <http://jagsdale.wordpress.com/tag/ibm-cognos>), Top right: SAP BI (Image from http://www.biondemand.com/sites/default/files/global/images/screenshots/biod_dashboard.jpg), Middle left: Microsoft BI (Image from <http://www.dashboardinsight.com/dashboards/screenshots/bi-analytics-dashboard.aspx>), Middle right: Oracle BI (Image from [ora]), Bottom left: Jaspersoft (Image from <https://www.jaspersoft.com/tour>) and Bottom right: Pentaho (Image from <http://www.pentaho.com>)

ent metadata model across products and lack of direct sales channel are a few of the challenges they face [RSSHog].

With a large base of customers using Oracle applications, database and middleware, Oracle is another well-established solution provider. Acquiring Siebel Systems²⁴ and Hyperion Solutions²⁵, Oracle moved into the BI market with an offering of a complete, open and architecturally unified BI system (Figure 9, middle right). The software pro-

²⁴ <http://www.oracle.com/uk/products/applications/siebel/overview/index.html>
²⁵ <http://www.oracle.com/us/corporate/acquisitions/hyperion/index.html>

vides similar functionalities as expected — ad hoc query and analysis, OLAP, web-based interactive dashboards, collaborative workspaces and scorecards. Oracle exploits federated query capabilities to source data from multiple sources and common enterprise information model maintains consistency over different applications. Gartner, in their survey noted that Oracle faces several challenges: reporting tools lacked capabilities, customer support and a general slower introduction of innovative technologies such as in-memory processing, visualisation and search.

While several major vendors such as the ones discussed have a large share of the BI market, several open-source solutions have also gained momentum and are being considered by many organisations. Jaspersoft (Figure 9, bottom left) exploits its extended unified metadata capability for abstracting business-level information from lower level data models and provides a wide range of charts and ad hoc query and analysis capabilities. It provides easy interaction for non technical users to develop reports without IT knowledge. Jaspersoft provides several features such as report designer, cluster-aware caching, data visualisation and exploration, data virtualisation and in-memory engine²⁶. Gartner notes that in spite of its favourable feedback from users, it lacks in supporting a large number of users. Pentaho²⁷(Figure 9, bottom right), providing a suite of extensive BI capabilities including query reporting, data mining and a BI platform, is another significant player in the open-source BI market. The components of Pentaho can be used either as an out-of-the-box solution or integrated with other solutions. The software is user-friendly and the presentation and dashboard capabilities are flexible [TvdB10]. Pentaho provides several features such as automatic table designer, cost-benefit analysis of aggregation at different levels, automatic generation and population of aggregated tables. Both Pentaho and Jaspersoft follow a subscription based model on their GNU Public License, and have been rated higher than other vendors in the customer support category [RSSHo9].

As can be observed, most of the BI solutions provide several means for users to aggregate information from different data sources (organisational databases, excel spreadsheets, individual datastores), perform analyses, what-if scenarios, ad hoc query processing etc. and subsequently build dashboards and reports containing scorecards and different types of visualisations. Most of the solutions also provide facilities for collaboration and sharing within teams as well as throughout the organisation. The prime function of BI solutions, therefore is in performance reporting and monitoring by assimilating information from a variety of sources, and performing some analysis on the information collected.

2.6 USER-CENTERED DESIGN

Originating from the seminal work of Norman and Draper [ND86], UCD requires users to be involved in the development of a product. It is a multi-disciplinary approach grounded on user studies and active involvement of users for a precise understanding of users and task requirements. [RSP02] discusses three primary reasons for involving real users as opposed to the traditional approach of interviewing proxy-users and managers: developers gain a better understanding of user's needs and goals thereby

²⁶ <http://www.jaspersoft.com/meet-the-new-jaspersoft-5>

²⁷ <http://www.pentaho.com>

resulting in a more usable to product; to ensure users have a realistic expectation from the product; and users feel a sense of ownership towards the product, resulting in being more receptive.

2.6.1 *Engaging Users in Design*

Gould and Lewis [GL85] discuss three principles of system design that should be followed for a useful and easy way to use a computer system: early focus on users and tasks (designers must understand user and tasks by studying their cognitive, behavioural, anthropomorphic and attitudinal characteristics); Empirical Measurement (users should use simulations and prototypes, to be recorded and evaluated for analysis) and Iterative Design (there should be several cycles of design, test and measure and re-design). The authors stress the need to involve direct users of the systems and understand them as opposed to identifying, describing or stereotyping them as well as human discussions with intermediaries. Designers can also be trained by users to use existing systems and relate them with the tasks so they can have a greater understanding of systems users are comfortable with and their tasks. [RSP02] suggests five principles to illustrate the early focus on users and tasks:

- User’s tasks and goals are the driving force behind the development — the principle recommends that instead of trying to fit a technology in an area, it is important to understand how to align available technologies to better support user tasks and goals.
- Users’ behaviour and context of use are studied and the system is designed to support them — this suggests understanding user behaviour, priorities, preferences and implicit intentions while performing their tasks.
- Users’ characteristics are captured and designed for — it is important to understand and account for the physical as well as cognitive characteristics of users that can also affect how some users interact with the systems. e.g. some users may be colour-blind, or have personal preferences while performing tasks etc.
- Users are consulted throughout development from earliest phases to the latest and their input is seriously taken into account — users can be involved in various ways and they need to be respected by the developers
- All design decisions are taken within the context of the users, their work, and their environment — designers need to be aware of users while making their design decisions even though users may not directly design the product.

The illustration provided by [Rub94] in Figure 10 sets the stage for the role of users in designing products: users can be seen as the “center of focus” as the development proceeds. The user is depicted at the centre of a double circle, where the inner ring consists of context, objectives, environment and goals. The outer ring consists of task detail, task content, task organisation and task flow.

The Table 1, prepared by [AMkP04] summarises the ways users can be typically involved during the system development.

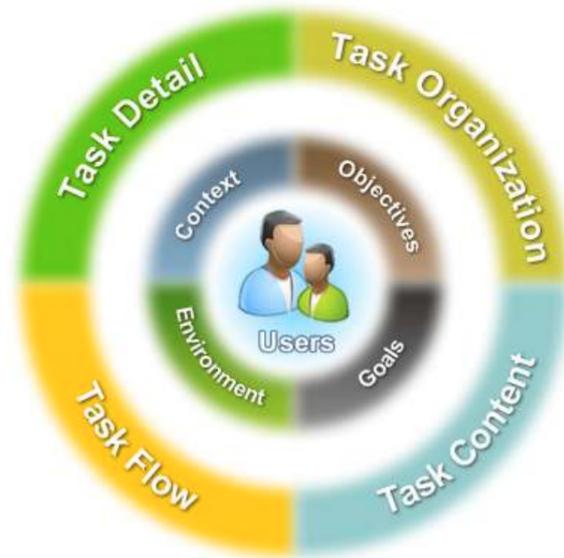


Figure 10: Rubin's [Rub94] depiction of users as the centre of the development. Image from <http://www.neelsworld.info/documents/ucd.pdf>

Since the time Norman initially coined the term “User-centered design”, there has been several definitions within the community indicating varying degree of participation of users as well as understanding of tasks [GGB⁺05]. Norman discusses UCD as a philosophy based on the needs and interests of the user, with an emphasis on making products usable and understandable [Nor04]. He proposes three conceptual models: the user model, the designer model and the system image. The key to a good design is consistency between the three models: the designer model is the conceptualisation of the system that the designer has in mind, the user model is used by the user to explain the operation of the system and the system image reflects the design of the system. The designer needs to ensure that the designer model is consistently translated to build an appropriate system image, which the user interprets to understand the system. With the progress in developing ISO 9241 *Ergonomic requirements for office work with visual display terminals*, a need for a separate high-level document developed which would explain human factor activities performed during the design of interactive systems. This need drove the efforts into a working committee to develop guidance for project managers. The results were published as ISO 13407:1999 *Human-centred design processes for interactive systems* (Human-Centered Design (HCD)). While UCD and HCD may suggest different implications, they have often been used in the literature (as well as in this thesis) as synonyms referring to the same principle of having the involvement of users in the development of systems. The following principles were listed in the report for human-centred design:

- the active involvement of users and a clear understanding of user and task requirements
- an appropriate allocation of function between users and technology
- the iteration of design solutions

Technique	Purpose	Stage of Design Cycle
Background Interviews and questionnaires	Collecting data related to the users and expectations of users; evaluation of design alternatives, prototypes and final artifact	At the beginning of the design project
Sequence of work interviews and questionnaires	Collecting data related to the sequence of work to be performed with the artifact	Early in the design cycle
Focus groups	Include a wide range of stakeholders to discuss issues and requirements	Early in the design cycle
On-site observations	Collecting information concerning the environment in which the artifact will be used	Early in the design cycle
Role-playing, walk-throughs, simulations	Evaluation of alternative designs and gaining additional information about user needs and expectations; prototype evaluation	Early and mid-point in the design cycle
Usability testing	Collecting quantities data related to measurable usability criteria	Final stage of the design cycle
Interviews and questionnaires	Collecting qualitative data related to user satisfaction with the artifact	Final stage of the design cycle

Table 1: Table prepared by [AMkPo4] to indicate ways to engage the user communities in the software design cycle

- the iteration of design solutions

The recommendations of the working committee included activities such as planning of the Human-Centered process, specification of user and organisational requirements, production and evaluation of solutions on the basis of the requirements. ISO 13407 defines HCD as *Human-centred design is an approach to interactive system development that focuses specifically on making systems usable. It is a multi-disciplinary activity*". In their survey of UCD practices among UCD practitioners, [VMSCo2] defined UCD as "UCD is herein considered, in a broad sense, the practice of the following principles, the active involvement of users for a clear understanding of user and task requirements, iterative design and evaluation, and a multi-disciplinary approach. UCD methods are modular or identifiable processes involved in UCD practice. You should NOT think of UCD as merely usability testing or software engineering". [AMkPo4] describes UCD as a "design processes in which end-users influence how a design takes shape". [Hen07] describes UCD as "User-Centered Design (UCD) is a user interface design process that focuses on usability goals, user characteristics, environment, tasks, and workflow in the design of an interface. UCD follows a series of well-defined methods and techniques for analysis, design, and evaluation of mainstream hardware, software, and web interfaces. The UCD process is an iterative process, where design and evaluation steps are built in from the first stage of projects, through implementation". Though slightly varying definitions of UCD exist, the role of the user in developing the final product is the primary theme of the design — the techniques and principles how they are involved are several, and which ones are most appropriate are often left to the developer and design community to decide.

[VMSCo2] surveyed over 100 UCD practitioners in order to understand how much UCD is being employed in designing systems as well as which methods are being employed by researchers. Figure 11 illustrates how UCD methods are presently used by practitioners:

	Ranking					Average Ranking	Frequency
	1	2	3	4	5		
Field studies (include contextual inquiry)	12	6	5	2	1	2.00	28
User requirements analysis	3	3	0	0	1	2.00	7
Iterative design	17	21	9	5	2	2.15	65
Usability evaluation	12	8	10	7	1	2.39	43
Task analysis	6	8	6	7	1	2.61	34
Focus groups	5	2	2	1	4	2.79	16
Formal heuristic evaluation	3	2	5	2	2	2.86	15
User interviews	2	0	3	4	0	3.00	11
Prototype without user testing	1	3	5	4	1	3.07	15
Surveys	0	2	2	1	1	3.17	9
Informal expert review	4	6	3	10	6	3.28	31
Card sorting	0	1	1	0	1	3.33	5
Participatory design	1	0	1	2	1	3.40	7
No code/too sketchy to be categorized							64

Figure 11: The survey by [VMSCo2] shows which methods were being used more frequently in practice as well as the ranking of five methods which in their opinion have the greatest impact by over 100 UCD practitioners.

As can be noted from the figure, and discussed by Vredenburg, the most used method was Iterative design, with Usability Evaluation and Task Analysis following up. The third most used method was Informal Expert review. The results from the survey show that the widely used methods are mostly the less expensive ones. The authors conclude that though the role of UCD methods in improving the usefulness and usability of a product is highly recognised, in practice, the adoption of methods in the organisations has been uneven. There is also a disconnect in the understanding of UCD's benefits in the longer term as opposed to the costs involved in the short term. Though the benefits of UCD are clear to all parties, there has also been some resistance to UCD, owing to the short-term expenses [Nie93] and such techniques have not been exploited to the fullest by organisations [CKI88]. In addition to resource constraints, another recent survey [RRH00] indicated the lack of formal usability training and UCD expertise is also an obstacle for the relatively less uptake of UCD in organisations.

2.6.2 Stages of User-Centered Design

Conventionally, a software design cycle uses four stages, involving the same kinds of activities — *Study, Design, Build, Evaluate*. During the research or design process, this cycle generally iterates several times, and the cycle can be entered into at any point [HRRS07]. Research projects can often start with evaluating existing technologies to understand how they can be improved. In such cases, the design model might be entered from the evaluate stage. (Figure 12, Left) shows the conventional UCD approach and their phases.

In addition to the stages in the conventional, [HRRS07] proposes the introduction of another stage in the cycle, Understand (Figure 12, Right). The authors propose the

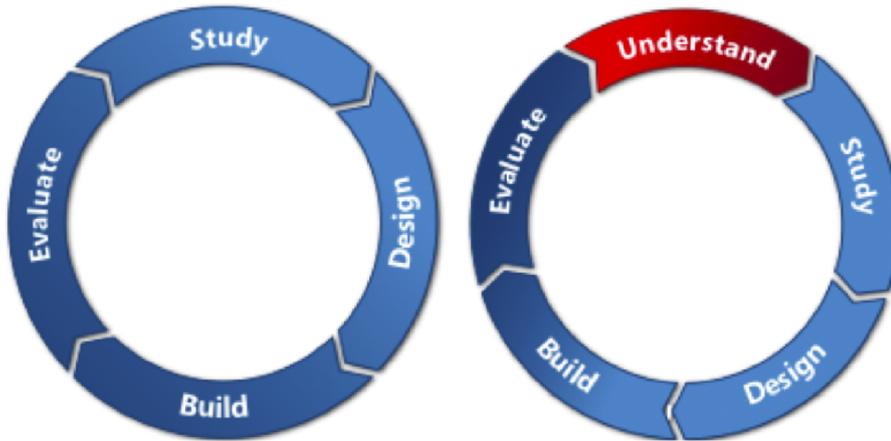


Figure 12: (Left) Conventional UCD, four-stage design model (Right): Extended user centred, five stage design model, introducing a new concept, understand. Image from [HRRSo7]

understand phase as an initial phase that focusses on human values and aims to identify the values that are to be researched upon and designed for. This stage is a multi-disciplinary one, driven by understanding from other fields such as philosophy, psychology, art, literacy theory, cultural studies, anthropology and so on. The stage needs discussion with all parties: end-users, developer community, stakeholders and technology designers in order to understand the values users expect from the system. This stage also requires an understanding of the domain and the kind of users that are expected to be using the system. At the end of this stage, decisions such as the target user types, focus of the system are taken as well as fundamental research questions are generated, which are then fed into the next (study) stage.

The *Study* stage deals with looking at an individual's interactions and tasks or work practices that they undertake to perform their daily job. This stage needs a complex multi-disciplinary analysis, focussed on the forces that drive individuals to engage with technologies and how such technologies are embedded in the world. This takes into account the user's interactions with computers, software programs etc by studying their work habits, practices and environments. The study stage is not just limited to understanding the user's computer use, but also the interactions users have in everyday world with objects and their environment. Ethnographic studies or observations are commonly conducted at this stage, and controlled user studies and focussed studies can also follow up after basic studies are conducted. The outcome of this stage is a thorough understanding and an insight of user's social and environmental factors and how they interact with their environment. With this understanding, various possible technical solutions can be sketched, which serve as inputs to the next (design) stage.

The *Design* stage is a more creative stage where the developers and designers identify the design goals. This is a stage that involves art and design more than programming and implementing. This stage explores the various artistic possibilities like sketches and considers the user's culture and operation environment (e.g. schools, stations, churches etc). This stage also requires an understanding of the existing technologies to be used in addition to the new system. This stage can also consider potential new technology such as hardware, software or infrastructure to generate better designs and needs an

in-depth thought of the costs of offering new designs to enable a better way of working and different experiences.

The *Build* stage involves building different possible versions of the perceived system, following the results from the design stage. The build stage can involve low-fidelity prototypes, paper sketches or even stable high-fidelity systems, ready to be used for long term user studies. Partially implemented systems can also be developed at this stage, where some functions of the systems will be performed by a human who is hidden to the user (this technique is known as the “*Wizard of Oz*” technique).

Evaluate stage evaluates the outcome of the build stage. The type of evaluation, however, should be decided in the previous stages. The goal of this stage is to identify how well the system performs as opposed to how well the system was perceived to perform in the previous stage. Though visualisations are intended to help users perform their daily activities, it is rarely the focus of attention. How well users can perform their current activities, with the aid of a particular software is an important question needed to be answered to understand the benefits of the system. Visualisation systems are designed to improve efficiency, understanding or performance of users in their daily analytical tasks. It is necessary to quantify these factors in order to understand the effect of the system on the end-users. However, some factors like aesthetics, usability or ease of learning are difficult to quantify. Hence, a qualitative evaluation is also encouraged to understand how well the system performs from a subjective point of view. Various questions should be addressed while evaluating a software system — e.g. Accessibility, Acceptability, Efficiency, Effectiveness, User experience, Usability and so on. However, care needs to be taken to understand the resources and constraints involved during the process — for example, cost, time, system representations, participants, test environment and equipments, data capture tools, analysis tools, security and so on. Situated evaluation focuses on how a system supports people’s work in context. Evaluation can be conducted in several ways — with (a basic system) or without (mockups of the intended system) a working system; with or without (expert review) real users; with (in laboratory) or without (as a field study) the context of use. There are various approaches to evaluation — Heuristic evaluation [NM90], cognitive walk-through [WRLP94], think aloud study [ES93], contextual inquiry [BH99], CASSM (Concept-based Analysis of Surface and Structural Misfits) [BGFMo8] etc.

Heuristic evaluation is a method for usability analysis where users look at an interface and come up with positive and negative points about it. In spite of exhaustive lists of usability guidelines, evaluators often find them intimidating. [NM90] presents their nine simplified usability heuristics as follows:

- Simple and natural dialogue,
- Speak the user’s language,
- Minimise user memory load,
- Be consistent,
- Provide feedback,
- Provide clearly marked exits,
- Provide shortcuts,

- Good error messages and
- Prevent errors

Cognitive Walkthrough [WRLP94] is a review process, aimed at evaluating usability of systems. Here, the proposed design is presented to a group, which evaluates the interface, using a pre-defined sequence of actions to complete a task. [ES93] proposed a model for generating verbal reports during evaluations, to better understand the interpretive processes of the experimenter. *Thinking aloud*, as they suggest, is the ideal way to gain information about the subject's information processes. *Contextual inquiry* [BH99] is aimed at understanding the users and their day-to-day work. The designers, in this process interview the users in their workplace to understand their work. They observe the users as they work and then ask questions in every step to understand their strategies and motivations. [BGFMo8] addresses the misfit between users and system requirements, which are often not highlighted using heuristic evaluation or walkthroughs. CASSM focuses on the user's conceptual model and how it fits in the system developed. The users are presented with the system, and they judge it with respect to the presence, absence or difficulty of each concept. They can also consider relationships between concepts and actions that can be performed on concepts. Finally, the user judgments on the concepts are then assessed in accordance to several pre-defined structural misfit criterion.

Designers and developers need to iterate over these steps and develop the final product with the involvement of the users. The higher the number of iterations, the closer the system will be toward the best design.

The following section introduces Semantic Web as an information infrastructure that provides a highly structured and self-descriptive framework to facilitate a generic mechanism of managing information. The next section also surveys the interfaces that have been developed so far within the Semantic Web. Semantic Web, in the context of the thesis is viewed as the key enabler for providing Knowledge Management solutions in the enterprise.

2.7 SUMMARY

Several research areas like Information Seeking behaviour, searching techniques, visualisation, human computer interaction and so on have been actively pursued by researchers for decades. Research into areas such as modelling aesthetics and human emotions, cognitive psychology and so on can be traced back to over a century. A firm theoretical understanding is required from a large number of research areas to develop interactive, intuitive, usable and aesthetically pleasing interfaces. It is also important to realise that a significant amount of effort and thought should be dedicated to designing interfaces before implementing them. An aesthetically pleasing interface can evoke a positive response from a user, thereby making users more open to new creative solutions. Principles of aesthetics can guide designers into building products that can be more friendly to users and eventually improving their acceptability.

This chapter discussed the literature from different perspectives— starting from the act of Information Seeking and behaviour, then to the information infrastructure of the Semantic Web, and finally to the application domain of knowledge management and

business intelligence. The chapter also discusses how developers and solution designers can engage with user communities in different stages of UCD process. This chapter served to form the foundation for the basic concepts discussed in the thesis. The following chapter introduces visual analytics in the Semantic Web and surveys the interfaces that have been developed so far. The chapter also discusses aesthetics in the context of the Semantic Web, and investigates different aesthetic measures for assessing the aesthetic quality of some existing Semantic Web tools. Semantic Web, in the context of the thesis is viewed as the key enabler for providing Knowledge Management solutions in the enterprise, and hence, the reviews, discussions and research surrounds Semantic Web and Linked Data.

[SW](#) and [LD](#) provide a machine-readable and understandable way of formalising information across different platforms. However, since data is eventually meant for human consumption, there is a need to present such information in an intuitive and meaningful manner. This task is further complicated with the ever-increasing volume of data continually generated. Extracting actionable information from large volumes of data is a highly complex task for analysts and decision makers. ‘Visual Analytics’ aims to reduce this complexity by visually representing information to enable users directly interact with the information, gain insight and draw conclusions, thereby aiding decision-making processes [[KMS⁺08](#)]. The opportunities that arise from combining Linked Data and Visual Analytics help promote a mutually beneficial research direction: Linked Data can benefit greatly by Visual Analytics — enabling discovery of hidden trends and patterns; Visual Analytics can benefit by the development and evaluation of scalable web-based Visual Analysis techniques for large distributed networks [[GSK⁺10](#)].

Several researchers have attempted to support complex querying and/or visualising query results. [[HLTE11](#)] classifies such attempts in two categories — simple and complex approaches. Complex approaches (such as SPARQLViz¹ and iSPARQL²) include advanced user interfaces and query constructs, designed for experienced users and experts. Simple approaches such as mSpace [[sWRS06](#)],/facet [[HvOH06](#)] or Parallax [[HK09](#)], on the other hand are designed for casual users, but are limited in answering more complex queries. Such interfaces employ basic visualisations such as lists, tables, maps, matrices or scatter plots to represent information, thereby limiting the analytical dimensions being represented. Data Visualisation in the Semantic Web need a more careful consideration. This is due to more content being added to human readable content to make it machine-processable such as RDFa³ and microdata⁴. However, the additional information being added is highly structured and well connected — this creates more opportunities to visually represent structured data in a standardised manner.

Green proposed a few guidelines to motivate Visual Analytics research for discovery and knowledge building, based on their human cognition model [[GRF09](#)] —

- Provide multiple views (foster discovery of patterns using different views, as humans have different ways of processing information)
- Direct interaction (interaction to be provided without interfering with the user’s train of thought);
- Central role of interaction (interaction enables user and machine to share knowledge);

¹ SPARQLViz, <http://sparqlviz.sourceforge.net/> is a plugin for IsaViz, that enables users to build queries from a SPARQL query interface

² OpenLink’s iSPARQL interface graphically renders a user’s SPARQL query, showing how query concepts and relations are linked

³ <http://www.w3.org/TR/xhtml-rdfa-primer/>

⁴ <http://www.w3.org/TR/2011/WD-microdata-20110525/>

- Insulation of reasoning flow (visualisation should not hamper the rhythm of reasoning);
- Intimate (seamless) interaction;

Most relevant to *SW* research is the recent work by Dadzie and Rowe [DR11], which presents design guidelines for Linked Data by exploring the literature. Starting from high level design guidelines [Shn96], the authors propose the need for Linked Data tools to support multi-dimensional, hierarchical and network data. Additionally, the tools should also provide support for identifying/highlighting relationships within the data and the ability to export data to users/applications for re-use. The authors explored design guidelines for Linked Data interfaces from the point of view of lay-users (regular web users without knowledge of ontologies or *RDF*) and tech savvy users. This work is discussed further and the developed technologies are aligned with the principles identified by Dadzie and Rowe in Chapter 8.

Many *VA* tools outside the *SW* community such as [WHP⁺04, AAW07, WJF⁺06, KBH06, KNS⁺06, MMME11] have been built over the years. However, such tools continue to provide inspiration for Visual Analytics research in the Semantic Web community. Traditional plotting techniques such as Scatter Plots [CM88], Pie Charts and Parallel Coordinate Plots [Ins09] as well as newer techniques such as Spiral Graphs [CK98] and Fisheye lenses [SB92] can be incorporated with different forms of visualisations. A few Visual Analytics systems have also been built specifically for the Semantic Web, such as Stefaner's Elastic Tag Maps⁵, Elastic Lists [SM07] and the work done by [HLTE11, WvHdV⁺10, YAPM08, MO] which have been developed for different application areas such as social network analysis, movement data analysis, bibliographic reference analysis, event detection and so on.

3.1 CONSUMING SEMANTIC DATA

The hypertext Web for sharing documents has been very popular and spread quickly among users. A reason attributed to this is that authors of *HTML* documents could publish their documents and immediately see the results. Documents could be linked to several others by hyperlinks and such links could be easily navigated. However, the *SW* and *LD*, in spite of providing ways of uniquely identifying resources with *URIs* that can link to other objects with explicit relations has not been supported well with techniques for human-consumption, apart from just looking up *URIs* and relations [BLCC⁺06]. There have been several recent efforts to help exploring, browsing and visualising Semantic Web data and *LD*. A vast multitude of semantic search engines such as Swoogle⁶, hakia⁷, SWSE (Semantic Web Search Engine)⁸, SemSearch [LUM06] have been developed that accept user inputs as natural language and provide textual results. Users can interact with semantic tools in a variety of ways that range from formal to highly visual and interactive. As a result, semantic tools present results in several ways: textual lists, tables and visually enhanced result sets.

5 http://well-formed-data.net/experiments/tag_maps_v5/

6 <http://swoogle.umbc.edu/>, Last accessed 25/06/2012

7 <http://hakia.com/>, Last accessed 25/06/2012

8 <http://swse.deri.org/>, Last accessed 25/06/2012

Triplestores such as OpenLink’s virtuoso and openRDF’s Sesame provide users with a web interface for querying semantic data. Such interfaces accept formalised SPARQL queries as user input and return result sets in a tabular format. Question answering systems such as Power aqua⁹ [LMU06], NLP Reduce [KBf07] and Search systems such as Swoogle, SWSE, SemSearch [LUMo6], Sindice¹⁰, sig.ma¹¹ mostly require natural language text input and provide results as a textual list of retrieved triples/documents. Apart from tabular and textual representation of information, semantic systems can also visually enhance result sets by making use of various forms of visual encoding (e.g. presenting a geographical map of places, graph of data points etc.). Several types of interfaces and visualisations have been developed since the early stages of the Semantic Web — from LD browsers, mashups and dashboards to domain-specific visualisations. Related work is discussed in two directions — Interactions involved in querying Semantic Data and Techniques used for presenting Semantic Data.

While most tools have a distinct visual separation between querying and results presentation, some follow a combinatory approach that integrates querying techniques within the same interface as viewing results. A good example of such an interface is the Research Funding Explorer¹², that employs a multiple coordinated visualisations of different facets of data — a geographical map plotting the relevant geographical areas of interest, a timeline that indicates a temporal distribution of the data and a bar chart that shows a different feature of the data. Here, interacting with one visual element triggers queries that further refine the dataset, thereby resulting in a new subset of the data being visualised. Most mashup tools follow a similar multiple-coordinated view approach — where interactions with widgets or visual elements trigger several queries.

3.1.1 Querying Semantic Data

Highly structured and formalised data are stored in triplestores and are made available to programs via queryable endpoints. This section discusses various methods of querying such endpoints, by making use of interaction mechanisms to build complex formalised queries. The systems discussed in this section are distinguished into different types strictly based on the interactions required from the user to formulate queries. Although all systems would need a translation from any type of interaction mechanism to strict formal queries, the interest is in how end users interact with systems and how much formalised inputs users have to provide.

3.1.1.1 Formal Querying

The most structured approach toward querying in the Semantic Web is formal queries such as SPARQL or RDF Query Language (RQL) [KB10] where the intended users are generally highly proficient in query languages. The process is similar to traditional database systems, where expert users enter formal queries in a text input element. Owing to their highly structured, defined and formalised syntax, a significant amount of training as well as understanding of Semantic Web principles is needed by users in or-

⁹ <http://technologies.kmi.open.ac.uk/poweraqua/>

¹⁰ Sindice, <http://sindice.com/>, Last accessed 25/06/12

¹¹ <http://Sig.ma>, <http://sig.ma/>, Last accessed 25/06/12

¹² Research Funding Explorer: <http://data.gov.uk/apps/research-funding-explorer>

der to find information. Such queries are often time-consuming as it requires users to consult specification documents to effectively build queries. In spite of several features being included in **SPARQL** that can help reduce the amount of effort involved in formal query authoring (e.g. prefixes, keywords(a)), the process is a complex one, often requiring users to seek external help (manuals, specifications, tutorials etc.). Such systems generally do not preprocess the data and do not provide any kind of Natural Language interface [WRE⁺10].

3.1.1.2 *Text-Based Approach*

Whilst only a minority of users prefer constructing formal queries to explore their data, most users lack the necessary technical expertise. Such users need a quick and efficient solution for lookups, answering highly specific queries or just getting a feel of the data at hand. This has caused a need for informal query languages and user interfaces that can provide assistance to users while expressing their Information Need. Several approaches for exploring and querying Linked Data have been developed over the past few years. Most of the applications developed to date have adopted a text-based approach, where a user's interaction with the system is largely text driven requiring the user to enter textual queries (as free Natural Language (Natural Language (NL)) question answering approach, as in **NLP Reduce** [KBF07], **PowerAqua** [LMU06], **Freya** [DAC11], **Querix** and **Ginseng** [KBF07]; form-based approach as in **K-Search** [BCC⁺08]; or keyword-based approach as in **Sem Search** [LUM06], **Sindice** [TDO08], **SWSE** [HHU⁺11] and **Falcons** [CGQ08]).

Free NL such as NLP Reduce and PowerAqua provide high expressiveness by allowing users to formulate questions or keywords using terms of their choice [EWC⁺12a]. However, Natural Language Processing (NLP) techniques are well-known for several issues such as generic technology (reliance on domain-specific ontologies), ambiguous (grammar, syntax, context contribute toward disambiguating terms and entities) [SBB⁺02, SAN07], multiword expression [Jac97, SBB⁺02], multilinguality [BCCGP04] etc. Controlled NL such as Ginseng provide support during query formulation by using suggestions of valid query terms found in the underlying schema. However, this introduces interaction challenges that increase frustration among users owing to the restrictive nature of the querying approach [EWC⁺12a].

3.1.1.3 *Faceted Approach*

Faceted approaches, on the other hand such as **mSpace** [sWRS06], **museumFinland** [HMS⁺05] and **/facet** [HvOH06] require users to progressively build queries to reach their final result sets — users first select a higher level category (Figure 13). Following this selection, a list of lower hierarchical categories pertaining to the previous selection are displayed. This process continues till the user reaches the data instances of interest. Depending on the implementation, categories of the facets can either be pre-determined, or configured by the user. The progressive queries are typically 'hidden' and separated in the logical layer of the system, so to ensure users are unaware of the technical complexities of faceted queries.



Figure 13: mSpace interface exploring Classical Music dataset. Image From <http://research.mspace.fm/mspace/what>

3.1.1.4 View-Based Approach

Enabling a more visual approach toward building semantic queries, **Semantic Crystal** [KBf07] is a view-based tool that visualises an entire ontology, with interactive elements that can accept user inputs. Several other view-based search systems such as **GRQL** [ACK04] and **SEWASIE** [CDM⁺04] have also been developed to aid querying semantic data, shown in Figure 14. In contrast to Semantic Crystal (where the entire ontology is visualised during initialisation), GRQL enables users to navigate ontologies by clicking on classes to show related properties. Clicking on properties expand the view to select the range of the clicked property. SEWASIE presents users with a set of pre-defined domain specific patterns (or templates) to choose from. These patterns provide users with a starting point which can be extended and customised to build formal queries. OpenLink's **iSPARQL**¹³ is another graph-based tool that requires the user to actively build **SPARQL** queries. Unlike the other view-based tools, iSPARQL does not provide users with a view of the underlying ontology, expecting the user to have a good understanding of **SPARQL** querying and the domain knowledge.

View-Based approaches provide the most support to users by visualising the search space in order to support understanding the underlying data and possible queries [EWC⁺12a]. However, view-based systems, specially graph-based do not scale up to larger datasets and schemas. While users often take a long time to formulate queries using view-based systems, they can provide means to build precise queries [KBf07].

Figure 15 shows an extended version the formality continuum proposed by [KB10] where the different systems are placed on a scale that describes the relative degree

¹³ <http://oat.openlinksw.com/isparql/index.html>

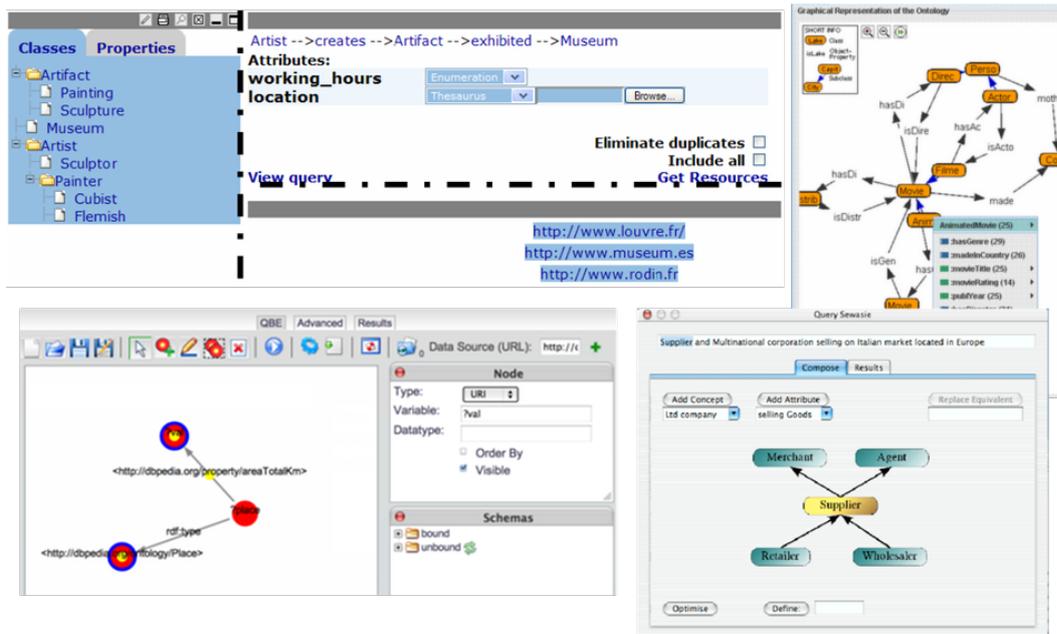


Figure 14: View-Based approaches: GRQL (Top-Left) allows users to navigate ontologies as hierarchical lists and select concepts and properties of interest. Image from [ACK04]; Semantic Crystal (Top-Right) shows an entire ontology using interactive graphs, enabling users to select concepts and properties to set for their query. Image from [KBF07]; iSPARQL (Bottom-Left) allows users to build visual queries by adding new concepts and creating links between the query concepts. Image from <http://dbpedia.org/isparql/>; SEWASIE (Bottom-Right) enables users to select from predefined domain-specific templates as a starting point to drive querying. Image from [CDM⁺04].

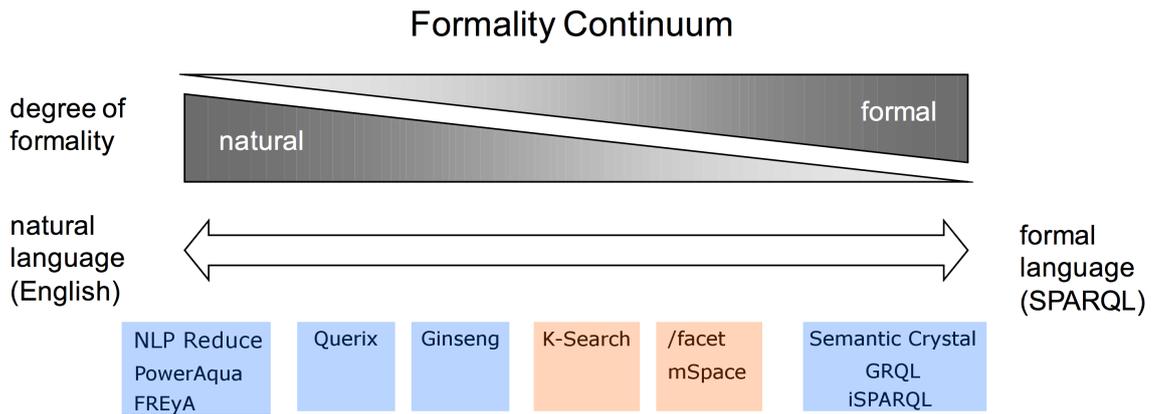


Figure 15: An extension of the formality Continuum, illustrated by [KB10]. Faceted Browsers and form-based systems have been introduced in the model, before a completely formalised approach. Image adapted from [KBf07]

of freedom, naturalness, structuredness and formality of the query mechanisms being employed. The model has been extended with two more mechanisms of querying — formed based and faceted. These approaches are frequently used in Semantic Web systems, and it is important to understand the freedom or formalisations such systems offer.

The model attempts to formalise how full Natural Language based systems and formal logic based systems are related, on a scale of naturalness and formality. Systems such as NLP-Reduce and Querix accept full or slightly controlled Natural Language questions. Systems such as Ginseng accept queries built with further controlled vocabulary and syntax, but the resultant query is also expected to be in Natural Language. The next three types of systems deviate from the path of Natural Language queries, and take a more formal and restrictive approach. Slightly more formal in approach and less in Natural Language expressiveness, form-based systems provide users with multiple fields in forms. The users enter their Information Need in the form and the resultant query is translated from form data to SPARQL query. Faceted browsing interfaces have been considered to be slightly more formal than form-based as the users are restricted in the way the queries are formulated — the order of the interactions progressively lead the user to a final query. The final approach is a visual one, where the query and data is represented as graphs (or other visual metaphor) and interaction with visual objects on-screen results in a formalised query. Whilst specific implementation details can dictate how these types of systems can be positioned on the continuum, this discussion takes only the aforementioned systems into consideration as generic examples of the approaches.

3.1.2 Presenting Semantic Data

The availability of open source browser-based charting tools and graphics libraries such as Processing.js¹⁴, d3.js¹⁵, Highcharts, Raphaël¹⁶, Sgvizler¹⁷ etc. and browser scripting toolkits such as jQuery¹⁸, \$dom¹⁹ and so on have fostered an environment where developers, consumers and web enthusiasts can easily access Linked Open Data endpoints²⁰ such as dbpedia²¹, data.gov.uk²² or bbc²³ to build usable browser-based applications. [SRN11] presents an extensive comparative discussion to help developers choose the toolkits and libraries based on their requirements and motivations. The following discussion presents some of the well known tools and systems that have been developed to present semantic data to users. The tools range from text-based tabular displays and mashups to more complex visualisation systems. The systems designed so far are plotted on a graph showing the amount of interactivity and the level of visual encoding applied on the data, as shown in Figure 16. The positions of the systems are not absolute and not meant to be scaled, but act merely as indication of their relative interactivity and visual encoding. The systems are classified on the basis of the interactivity they support and the extent of visual encoding on data instances into the following categories:

- Text-Only Systems
- Enriched Text
- Enriched Text + Rendered Images
- Enriched Text + Rendered Images + Basic Visual Encoding
- Advanced Visualisations

The X-axis indicates the amount of visual encoding that an underlying data element has undergone — minimal in data tables and maximum in interactive graphical visualisations. The Y-axis indicates the amount of interactivity that the tool supports — basic keyboard inputs to mouse gestures like click-and-drag, zoom as the maximum. While there are several systems developed

3.1.2.1 Text-Only Systems

A semantic system that mostly presents information in textual format, either as list of instances or table of triples has been categorised in the group of Minimal Visual Encoding systems. Types of such systems are triplestore query forms, Semantic Web search engines and LD browsers. A few systems have been discussed below — the list, though not

14 <http://processingjs.org/>

15 <http://d3js.org/>

16 <http://raphaeljs.com/>

17 <http://code.google.com/p/sgvizler/>

18 <http://jquery.com/>

19 <https://github.com/julienw/dollardom/>

20 <http://www.w3.org/wiki/DataSetRDFDumps> provides a catalogue of all the datasets that are available to download

21 <http://dbpedia.org/sparql>

22 UK Government's open data available for query at <http://data.gov.uk/sparql>

23 BBC programmes and music dataset query-able at <http://bbc.openlinksw.com/sparql>

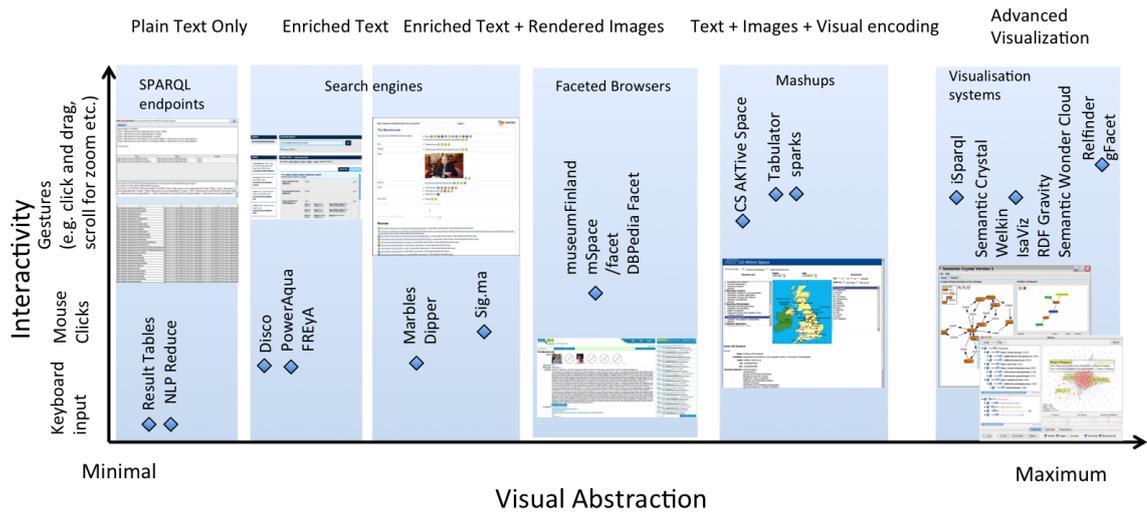


Figure 16: Existing Semantic Systems positioned on Interactivity and Visual encoding scales — note how the system screenshots show progressively increasing visual elements used on the interfaces.

exhaustive, is however representative of the data presentation mechanisms employed by most such systems. Text-based interfaces form one of the most common types of systems that have been developed by the Semantic Web community. From an information visualisation perspective, these types of interfaces are the simplest, requiring minimal visual encoding. Most of such tools are search systems and browsers that present result sets in tables and lists, while the query mechanism could be one query language (i.e. free-formed Natural Language, formalised questions, keyword-based, form-based) or a combination of multiple query languages (i.e. hybrid search systems). Though some LD browsers may apply some visual abstractions on the data such as group result sets in facets or colour-coding or highlighting results (based on data type or content) and so on, the visual representation of data instances are mostly textual. Tools such as NLP Reduce and result tables from triplestores are examples of such systems. Interactions in such systems are typically limited, allowing mostly keyboard inputs to type queries in either highly formal or Natural Language queries. The results from a query are mostly presented in plain text as lists or tables.

3.1.2.2 Enriched Text

Basic search engines, question answering systems and Natural Language interfaces such as **PowerAqua** [LMU06], **Disco**²⁴, **Marbles**²⁵, and **Dipper**²⁶ provide lists and tables that present result sets. The text is mostly presented as URIs, providing means to access follow-on information. The result sets are often visually separated into different categories of information in an interface. Such systems can also encode text into fonts, colour categories and present icons providing further information. Interactions in such

²⁴ Disco Hyperdata Browser: <http://www4.wiwiss.fu-berlin.de/bizer/ng4j/disco>

²⁵ <http://www5.wiwiss.fu-berlin.de/marbles>

²⁶ Dipperâ Talis Platform Browser: <http://api.talis.com/stores/iand-dev1/items/dipper.html>

The screenshot displays the PowerAqua Question Answering interface. At the top, there's a header 'PowerAqua QUESTION ANSWERING'. Below it, the 'ASK ANOTHER QUESTION' section contains a search bar with the query 'Give me English actors that act in Titanic' and an 'Ask' button. A checkbox for 'Make use of WATSON' is also present. The 'SOURCES' section lists several data sources with their respective fact counts. The 'LINGUISTIC TRIPLES' section shows the query triple: <actors / English . act . Titanic> <English . ? . actors>. The main results area is titled 'We found 5 answers in total from 1 ontologies' and lists three actors: Bernard Hill, Brian Aherne, and Frances Fisher. Each actor entry includes a score of 1 and a list of relevant facts and merged answers, such as 'Titanic_%281997_film%29' and 'England_(England_synonym)'.

Figure 17: Enriched Text systems such as PowerAqua present text and clickable URIs in structured tables and organised layouts based on content. Image from [LMUo6]

systems are mostly text-based keyboard inputs and mouse clicks enable users to follow hypertext links.

3.1.2.3 Enriched Text + Rendered Images

Faceted browsers, enable a greater interactivity by presenting users with a progressively refined list of categories of data, based on previous selections (mostly by clicking categories in multiple filters). Examples of such interfaces are */facet* [HvOHo6], **DBpedia Faceted Search**²⁷ Longwell and **mSpace** [sWRSo6]. However, in most cases, such interfaces do not provide much support to visualise the structure of the data as well and connections between concepts and data instances. mSpace [sWRSo6] is a browser for exploring its RDF database related to classical music. The main principle behind the browser is that the information is categorised into high dimensional spaces. These spaces are then presented one subset at a time. The subsets of information are provided in several hierarchical columns of instances belonging to their corresponding hierarchy. The columns are arranged from left to right, the leftmost column being the highest in the hierarchy. mSpace provides the users means to sort the data as they wish using several facets, which are populated with values appropriate to the choices of the previous facets. Figure 13 shows an example scenario where the user explores the information related to Classical Music.

The user selects 'Modern' from the highest category of era, which provides a further set of composers during that era. The user then selects Edward Elgar from the

²⁷ <http://wiki.dbpedia.org/FacetedSearch>

list of the composers, which populates the piece column with a list of the composer's compositions. Users can add or remove categories or sort them as they wish, thus providing them with control over the browsing, as they wish. Such systems generally have a greater classification of result elements, and present result sets in a more readable manner.

MuseumFinland [HMS⁺05] is a web-based portal for museum collections that allow users to browse all the inventories using multiple pre-defined facets. The system uses several different ontologies for different content like artifacts (tangible objects), materials (substances used to make artifacts), situations (social events, situations, processes), actors (persons, organisations), locations (geographic locations), time (periods, centuries, era), collections (museum collections). The total number of classes and individuals in the ontologies are about 10,000. The user is initially presented with nine facets (derived from the previously mentioned ontologies), which are artifacts, materials, creator, location of creation, time of creation, user, location of usage, situation of usage and collection. Figure 18 shows the results when the user has queried for all artifacts which are tools. The column on the right shows the individual items returned as results, grouped according to one of the nine categories. Users can also further refine their queries by clicking on the column on the left to select another category. Inspired from MuseumFinland, [HvOHo6] presents a domain-independent faceted browsing interface */facet*. Users can select their preferred facets, thereby visualising instances of the data in a hierarchical list in the context of the user. */facet* also provide users with means to filter over the data set in three ways — within all instances, within an individual facet or within all active facets. The tool also accepts a configuration file, stored in RDF format to define which facets should be displayed.

Sig.ma (Figure 19) is a tool to enable exploration of the Web of Data. The system consists of two subsystems, a user interface and a backend, which performs textual and hierarchical analysis on the abstracts of the resources and their graph structures. The backend further aggregates this data using information from external resources and social tagging systems. Websites embedded with RDFa, RDF and microformats are scraped and indexed by Sindice, which is then queried by users via the tool. The results are provided in a web page which can then provide forward links to provide more information about entities in the page. Owing to the heterogeneity of the data sources, Sig.ma may return un-related data, which can be deleted by users. These user actions are recorded by the tool to prioritise results the next time. A previous system, developed in the X-Media project [PMDCo9] is worth mentioning here, where users are provided with multiple widgets customised for individual visualisations at the same time so that they can understand the different aspects of the underlying data.

CS-AKtive [GAC⁺04] is a browser that provides means to view researchers according to their research areas as defined by ACM classification system as well as geographical locations. The CS AKTive Space interface is based on the mSpace model for browsing. **Longwell** [BHH⁺06] is a generic faceted RDF browser, that can provide faceted browsing for arbitrary data, but however requires the knowledge of a domain-expert to configure the different facets. The tool provides users with different levels of customisation depending on the level of coding that is involved.

The screenshot shows the MuseoSuomi website interface. At the top, there is a search bar with the text "Sanahaku:" and a search button. Below the search bar, there are several filters and categories. The main content area displays a list of artifacts, each with a small image and a description. The artifacts are grouped by category, such as "Esimuotyyppi" and "Materiaali".

Sanahaku: tarkenna hakua

Hakuehdot
Kategoria: Esimetyyppi > työvälineet (ryhmittele kahteet) (0/0/0)

Kohteet ryhmiteltyinä kategorian työvälineet mukaisesti
(näytä ilman ryhmitelyä)
tekstiäkäyttövälineet, kohteet 1-4/219 (ryhmittele kahteet)

Esimuotyyppi [käsiksi > työvälineet \(kaaka luokitella\)](#)
[tekstiäkäyttövälineet](#) (219),
[kansanlaakannon työvälineet](#) (1),
[kukittelemattomat työvälineet](#) (36),
[maataloustyövälineet](#) (7), [metallityövälineet](#) (1),
[sähkö- ja huonontamavälineet](#) (4),
[käsiohjavälineet](#) (9), [metallityövälineet](#) (4),
[työkäsit](#) (22)

Materiaali [\(kaaka luokitella\)](#) (ryhmittele kahteet)
[materiaalit](#) (241)

Valmistaja [\(kaaka luokitella\)](#) (ryhmittele kahteet)
[herätilot](#) (9), [tuotemerkit](#) (2),
[yhtiöt](#) (38)

Valmistuspaikka [\(kaaka luokitella\)](#) (ryhmittele kahteet)
[Afriikka](#) (2), [Etelä-Amerikka](#) (1),
[Eurooppa](#) (84)

Valmistusaika [\(kaaka luokitella\)](#) (ryhmittele kahteet)
[aikakaudet](#) (90), [vuosisadat](#) (89)

Käyttäjät [\(kaaka luokitella\)](#) (ryhmittele kahteet)
[herätilot](#) (54), [lastokset](#) (1),
[yhtiöt](#) (3)

Käyttöpaikka [\(kaaka luokitella\)](#) (ryhmittele kahteet)
[Eurooppa](#) (71)

Käyttötilanne [\(kaaka luokitella\)](#) (ryhmittele kahteet)
[harrastus- ja kansalaistoiminta](#) (4),
[kohteelle tehtävät toimenpiteet](#) (17),
[maatalous ja karjankasvat](#) (2),
[ruoan- ja ruoanvalmistus](#) (3),
[toimijoiden yleiset prosessit](#) (2),
[eläimet](#) (9), [valmistustekniikat](#) (179)

Kokoelma [\(kaaka luokitella\)](#) (ryhmittele kahteet)
[Erikoisen kaupunginmuseon kokoelmat](#) (54),
[Kansallismuseon kokoelmat](#) (193),
[Lahden kaupunginmuseon kokoelmat](#) (50)

kohteet
 kehräpöy, kuosali (NBA SU4527 50)
 kehruslautaa, kehräpöy, kuosali (NBA SU5069 26)
 rukinlappi (ECM 100 1)
 sseilöde, vääntämälappio, vääntämäpyörä (NBA SU2449 7) (edellisenä) / (seuraava)

kansanlaakannon työvälineet, kohteet 1-1/1 (ryhmittele kahteet)

suonirauta suonenraskentärauta (ECM 2711 1) (edellisenä) / (seuraava)

kuokittelemattomat työvälineet, kohteet 1-4/36 (ryhmittele kahteet)

kohteet
 nappikoukku-napituskoukku (ECM 3594 264)
 kietkamäläin, komsiöhäina (NBA SU4922 32)
 palohosat-palohosat (ECM 614 1)
 huonilasta (NBA SU4135 166)

Figure 18: MuseumFinland interface, where a user has performed an initial query to look for all artifacts that are tools. Image from [HMS⁺05]

Interactions in such systems are generally limited, allowing keyboard-based text inputs in either Natural Language or keyword based. Mouse clicks are also enabled, which results in either a new query being triggered or following links.

3.1.2.4 Enriched Text + Rendered Images + Visual Encoding

Mashup services²⁸ and multiple view based systems such as **Sparks**²⁹, **tabulator** [BLCC⁺06] and so on present users with multiple views over different features of the data. However, most of these views are either pre-configured (e.g. visualisation systems built for

²⁸ A mashup is a small, specialised application that combine data retrieved via APIs from several sources and processed to be presented in a different way than the original data provider(s) [HB11]

²⁹ Sparks Prism, <http://sparksrdf.github.com/>

The screenshot shows the Sig.ma interface for a mashup about Tim Berners-Lee. The main content area displays a profile for Tim Berners-Lee with two photos, a comment, and various RDF properties like 'given name', 'family name', 'comment', 'is creator of', 'is alternate of', 'alternate', 'birth year', 'birth place', 'birthday', 'category', and 'is child of'. The right sidebar shows a list of 12 sources, each with a title, number of facts, and a date. The sources include entries from DBpedia, Informa, Webz, Smeerch.it, NYLON, and Untitled document. The interface also features a search bar at the top, navigation buttons (Help, About, Forum), and a version number (1.1.28).

Figure 19: Sig.ma interface showing the mashup returned after querying for Tim Berners-Lee. Image from <http://sig.ma/search?q=Tim%20Berners%20Lee>

particular datasets) or designed for specific facets of the data (e.g. spatial data visualised in maps, temporal data visualised in timelines or calendar views).

Tabulator [BLCC⁺o6] has a generic approach to visualising data sets. It provides two modes of looking at data sets — exploratory and analysis. While exploratory mode enables users to have an overview of the data without any pre-conceived ideas (using a tree-structured outliner window to browse RDF data) about the data set, analysis mode enables users to define queries from the exploratory mode to visualise the results. The outliner view initially begins with several RDF nodes with clickable links that expands or collapses any further information. Nodes are coloured to provide the retrieval status of the node. Upon highlighting elements in the outliner view, users can form queries which can be submitted to the backend. The results are then further visualised in tables, calendar or maps in different tabs. Figure 20 shows the outliner mode (Top) used for exploration of RDF data, and the map (Bottom Left) and calendar (Bottom Right) views produced as a result of queries formed from exploration mode.

Providing means to access LD from mobile phones, **DBpedia Mobile** [BB08] is a good example of a mashup service that enriches standard maps with geo-coded LD. The service can extract a user's location and display a map with points of interest located near him/her. Different types of information are displayed using relevant icons, mapped according to Yet Another Great Ontology (YAGO) categories. Clicking on icons loads a summary view which provides more information on the particular resource, retrieved by querying DBpedia and Flickr. Additionally, the summary view also presents user-generated content like reviews and nearby DBpedia mobile users. The summary view can also link the user to a photo view (further information retrieved from DBpedia and Flickr) and a full view (shows all the available information related to the particu-

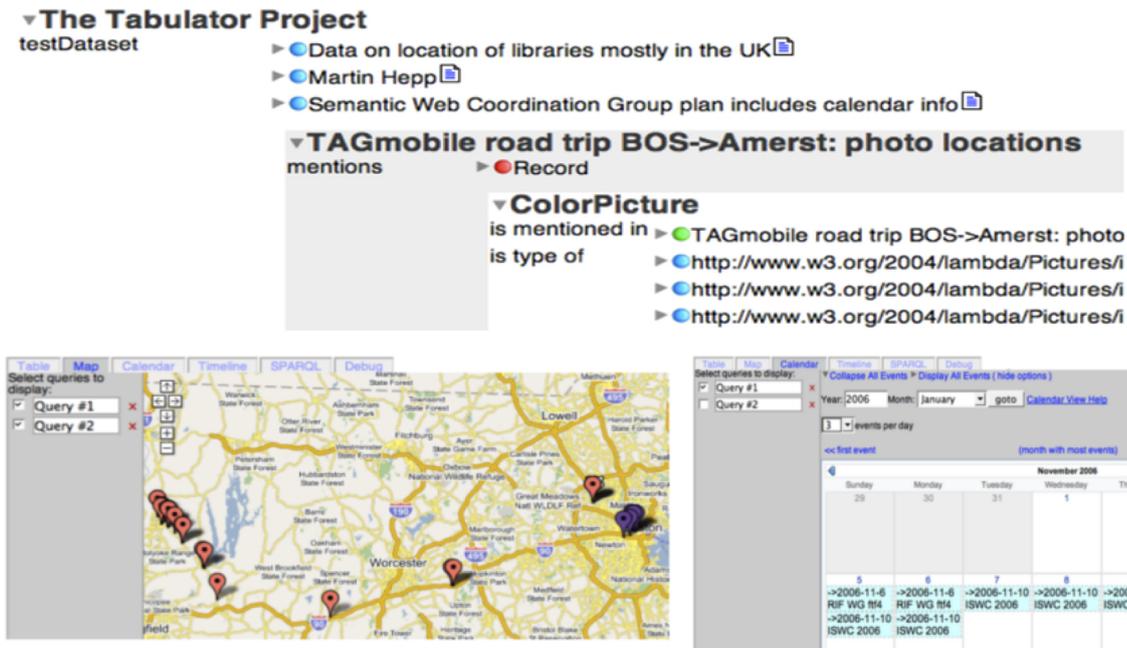


Figure 20: Tabulator provides users with two modes of viewing RDF data — an exploratory Outliner mode (Top) and analysis mode for viewing instances (Bottom Left and Bottom Right) in geographical maps and calendar views respectively. Images from [BLCC⁺06]

lar resource, retrieved from LD sources like GeoNames³⁰, flickr wrapper³¹, revyu³² and DBpedia).

DBpedia Mobile also provides filtering options, which enables users to select the information they are interested in — users can set the type of information (YAGO categories such as Libraries, Museums, Banks, Restaurants etc.), label (text filters containing specific characters), population (value for the population of a place) and so on. Interactions with such filters generate SPARQL queries (users can also write their own SPARQL query if they choose to) which are forwarded to the DBpedia Mobile's RDF store. In addition to consuming LD, users can also create their own content by uploading reviews, photos or their own locations.

Sparks³³ provides a good example of coordinated multiple visualisations in a dashboard-like interface, aimed at exploring LD. The Sparks interface mainly consists of a geographical map, displaying geo-located data elements similar to DBpedia Mobile. Interactive filter elements like sliders and tag clouds allow users to click and select the relevant subsets of the data.

User input in such systems are mostly textual, via keyboards. Result displays allow clicking on links resulting in generation of a query or following up links. Such systems can also abstract data into visual objects and visualised in a panel/canvas. These visual-

30 <http://www.geonames.org/>

31 <http://www4.wiwiw.fu-berlin.de/flickrwrapper/>

32 <http://revyu.com/>

33 Sparks Prism, <http://sparksrdf.github.com/>

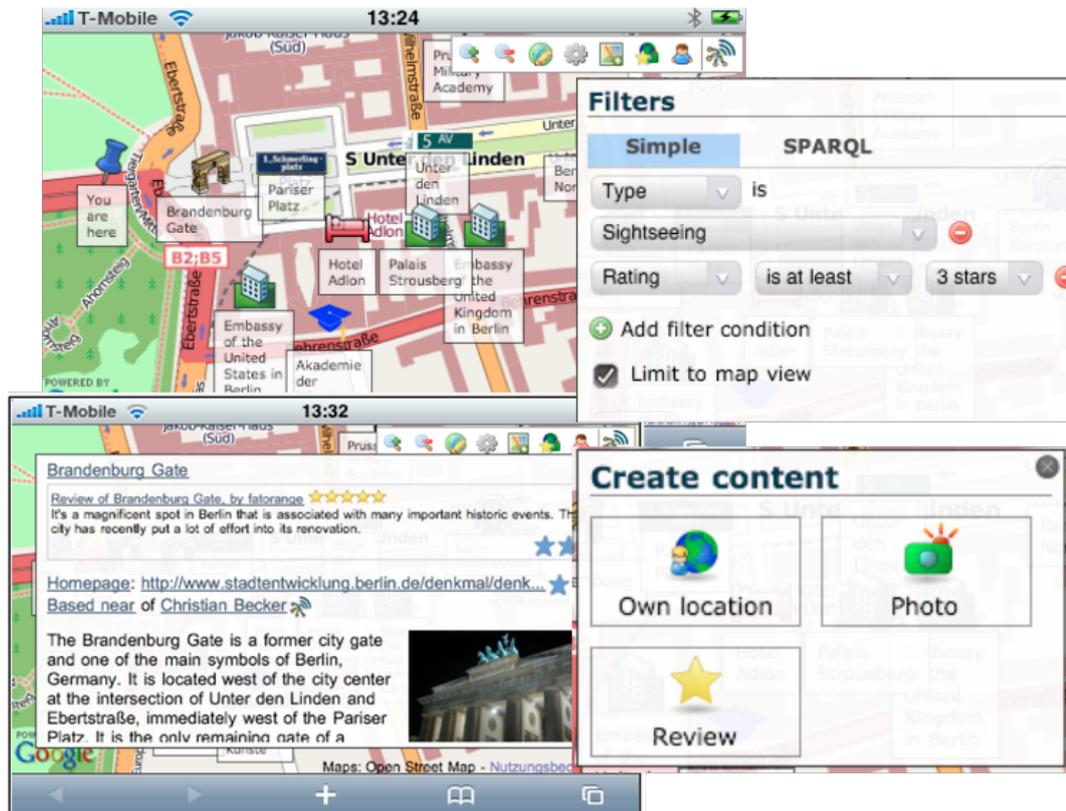


Figure 21: DBpedia Mobile provides a map enriched with geocoded data instances from linked data (top left); clicking on locations provide more information about the resource (bottom left); users can add filters to find relevant information (top right); users can create new content by interacting with a simple form to add a new location, photo or review (bottom right). Images from [BB08]

isations can support additional interactions such as zoom, scroll or click-and-drag (e.g. geographical maps, timelines etc.)

3.1.2.5 Advanced Visualisation

Cytoscape³⁴, an interesting interface originally developed for the bio-informatics platform is a well-known tool that can be used for visualising semantic data. The tool is capable of visualising large (more than 100,000 nodes) graphs in Graph Markup Language (GML) and Extensible Graph Markup Modelling Language (XGMML) formats and filtering over them. With some improvements, Cytoscape could handle large graphs in RDF and OWL formats. RelFinder³⁵ is a system that helps reveal relationships between concepts and presents this information as a graph. The interface requires users to enter the names of two resources of interest. Once the user selects the respective resources from automatic suggestions, the system creates two nodes and positions them in a graphical layout. The system then queries for all the relation connecting the two concepts and creates paths connecting the nodes, based on the discovered relations.

³⁴ Cytoscape, <http://www.cytoscape.org/>

³⁵ <http://www.visualdataweb.org/relfinder.php>

Welkin³⁶, as shown in Figure 22 (Top Left) is a java-based graph visualisation tool that loads **RDF** models and provides a graphical representation of the data along with lists of predicates and resources. The tool provides a user to get a mental model of an unknown data set, to get a quick idea about the connections and links within a data set. It can also help **RDF** authors in loading current versions of the **RDF** graph to check for inconsistencies or errors that might go unnoticed using other browsing techniques. Using the JUNG Graph API³⁷, **RDF Gravity**³⁸ shown in Figure 22 (Top Right) provides a filterable interactive graph visualisation of single or multiple **RDF** files. The tool also provides support for text-based searches as well as multiple selections, clicks, drags and zoom. **IsaViz**³⁹ shown in Figure 22 (Middle Right) is a visual editor for browsing and authoring **RDF** models. The tool also supports exporting to **RDF/XML**, Notation 3, N-triple formats as well as image formats like Portable Network Graphics (**PNG**) and Scalable Vector Graphics (**SVG**).

Originally developed as a bioinformatics platform, **Cytoscape**⁴⁰ shown in Figure 22 (Middle Left) integrates gene expression profiles with molecular interaction networks. The tool is capable of visualising large (more than 100,000 nodes) graphs in **GML** and **XGMML** formats and filtering over them. However, with some improvements, Cytoscape could handle large graphs in **RDF** and **OWL** formats. Several other visualisation tools have been developed which are aimed at large scale **RDF** graph visualisations⁴¹. However, most of these visualisations are either at the cost of interactivity or generality.

Semantic Wonder Cloud [**RATS10**], (Figure 22, Bottom Left) provides means for users to explore the DBpedia dataset by providing a Google Wonder Wheel -like visualisation. Users select a node to start their exploration, by selecting from a list of automatic suggestions of DBpedia resources after a text input. The system then generates a graph, keeping the selected resource at the center and a set of 10 most similar resources, with each resource being plotted as a circular node. The size of the node varies according to the relevance of the resource to the central node. The user's exploratory activity then constitutes of clicking on nodes of interest, and rebuilding the related nodes accordingly. A web browser-like interface maintains the exploration history of the user, thereby enabling him/her to move backward or forward to the previously explored resources.

RelFinder (Figure 22, Bottom Right) is a system that helps reveal relationships between concepts and presents this information as a graph. The interface requires users to enter the names of two resources of interest. Once the user selects the respective resources from automatic suggestions, the system creates two nodes and positions them in a graphical layout. The system then queries for all the relation connecting the two concepts and creates paths connecting the nodes, based on the discovered relations. The relations are added on the graph, with the shortest paths being rendered first. The nodes are interactive, allowing users to click, drag and drop them. Filters can also be applied that can allow the users to select relations based on the number of objects in between the nodes as well as specific ontological classes that the objects belong to.

36 Welkin, <http://simile.mit.edu/welkin/>, Last Accessed 25/06/12

37 Java Universal Network/Graph, <http://swik.net/jung>

38 RDF Gravity, <http://semweb.salzburgresearch.at/apps/rdf-gravity/>, Last Accessed 25/06/12

39 IsaViz, <http://www.w3.org/2001/11/IsaViz/>, Last Accessed 25/06/12

40 Cytoscape, <http://www.cytoscape.org/>, Last visited 25/06/12

41 Large-scale RDF Graph Visualisation Tools, <http://www.mkbergman.com/414/large-scale-rdf-graph-visualization-tools>, Last Accessed 25/06/12

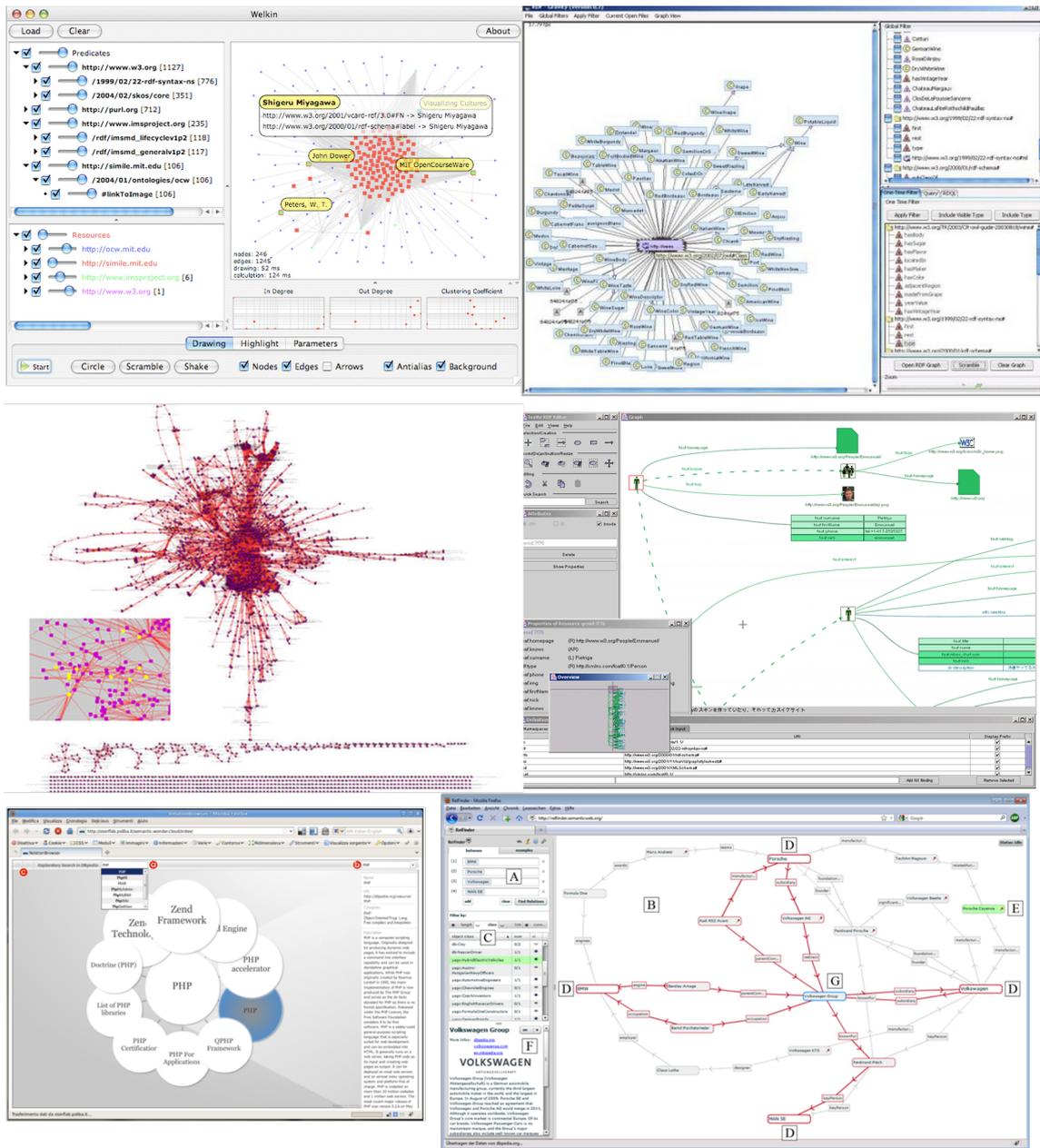


Figure 22: Examples of Graph Visualisation Tools. Top Left — Welkin (Image from <http://simile.mit.edu/welkin/>). Top Right — RDF Gravity (Image from <http://semweb.salzburgresearch.at/apps/rdf-gravity/>). Middle Left — Cytoscape (Image of RDFScope plugin, from [Splo8]). Middle Right — IsaViz (Image from <http://www.w3.org/2001/11/IsaViz/>), Bottom Left — Semantic Wonder Cloud (Image from [RATS10]), Bottom Right — Relfinder (Image from <http://www.visualdataweb.org/relfinder.php>)

Such systems typically support multiple interactions — text entry using keyboard, short cut key combinations, mouse and keyboard combination shortcuts and so on. Mouse gestures such as click-and-drag, scroll, hover, right-clicks (for context-menus)

etc. are often used to provide interactive means of building queries or exploring result sets.

3.2 KNOWLEDGE MANAGEMENT AND THE SEMANTIC WEB

Identifying sources of knowledge, ranging from external vendors, partners, suppliers and research institutes to internal employee knowledge, strategic and managerial processes etc. is only the first step in establishing a knowledge process within an organisation. Multiple sources of information signify that interoperability between different organisations, working groups, individuals and even document types require special attention — terminology used within one domain can be highly varied from another domain. Over the past few decades, more efficient knowledge practices have been incorporated in organisations, thereby improving the way companies organise, manage and use their knowledge. In spite of these advancements, current Knowledge Management systems face several challenges. The challenges lie in areas such as Information Seeking, extraction, maintenance and automatic document generation [DFH03, DFKO02]. The authors describe the issues as follows:

- **Searching** for information — keyword based searches retrieve information that may be irrelevant, and used in different contexts in the retrieved documents. Applying Semantic Web principles can help providing more streamlined results, more relevant to the Information Need of the user.
- Information **extraction** — manual browsing and reading is required to extract information from sources, due to the lack of automatic agents that can identify relevant information from textual documents and relate it to information extracted from other sources
- **Maintenance** of large unstructured/weakly structured text sources is a challenging and time-consuming task — requiring mechanised semantic representations that help detect anomalies.
- Automatic document **generation** that can be dynamically reconfigured based on user profiles and other relevant aspects — this requires intelligent agents to analyse the semantics of semi-structured information.

Advancements in Semantic Web has triggered a new impulse in Knowledge Management research that aims to facilitate mechanisms for machines and humans to have access, reason upon and exploit a ‘unified information medium’. The expressivity and functionality achieved with ontologies and Semantic Web technologies exceeds previous efforts in computer based Knowledge Management systems [SP10]. Developed for Artificial Intelligence, ontologies are the backbone and empower Semantic Web technologies to perform intelligent inferences, deductions and build connections. An ontology provides “*a shared and common understanding of a domain that can be communicated between people and application systems*” [DFH03]. Standards and protocols governing Semantic Web principles ensure that everything within the organisation can be accessed and exploited in a generic way. At the same time, ontologies provide employees with means to extend and formalise their own datasets and knowledge processes.

Similar to the internet, large organisations possess large number of documents that are weakly structured and contain multimedia information. Such documents mostly reside in large document repositories and the company intranet. Finding and maintaining information from such repositories is an extremely challenging process, owing to weakly structured and formulated representation. Furthermore, significantly increasing amounts of documents pose a Knowledge Management problem, since managing, maintaining and organising such large document sets is an extremely complex and time-consuming task if dealt by humans. This calls for automated intelligent agents and systems that can help categorise, classify, extract and manage information in a seamless manner. Semantic Web provides means to annotate documents, web pages and multimedia content with machine readable and processable meta-information, aligned with domain specific ontologies. This enables an “*incredibly large network of human knowledge*” that can be complemented with “*machine processability*” [DFH03] thereby creating new opportunities and mechanisms to explore a highly structured, extensible and rich network of interconnected knowledge repositories, connecting various information silos, domains, business areas, working groups and so on.

3.3 ONTOLOGIES IN KNOWLEDGE MANAGEMENT

[Obe13] lists several benefits that can arise out of incorporating ontologies in Knowledge Management: innovative business solutions (as ontologies facilitate hard to achieve business scenarios), increased productivity of information workers (ontologies facilitate visualisation capabilities and interaction possibilities), improved enterprise information management (ontologies provide formal descriptions of enterprise information) and increased productivity of software engineering (automation of web service discovery and composition; cost and time reduction by application of ontologies in software engineering; and quality improvements). [Dau11] discusses several benefits in utilising semantic technologies in the enterprise: data integration and framework (open data protocols avoid data silos and make data accessible over the web, using URIs and metadata to describe data); agile schema development; semantic search capabilities; and collaborative/social computing. Prior to their application in Semantic Web, ontologies have been used and built in artificial intelligence research. Research in the artificial intelligence community has been predominantly in two areas: Form-oriented and Content-oriented. While form oriented research involved logic and knowledge representation of domain knowledge, content oriented research involved content of knowledge [MI96]. The authors noted the urgent need of technologies to support content-oriented research and proposed ontology engineering as a field of research, that aims to “*provide a basis of building models of all things in which computer science is interested*”. They proposed a list of eight levels how ontologies should be used (shallowest to deepest):

- Level 1: As a common vocabulary to communicate among distributed agents.
- Level 2: As a conceptual schema of relational databases
- Level 3: As a backbone information for user of a user of particular knowledge base
- Level 4: For answering competence questions

- Level 5: As a means for standardising terminology, meanings of concepts, components of target objects and components of tasks.
- Level 6: As a means for structural as well as semantic transformation of data bases, considering the differences of the meaning of conceptual schema
- Level 7: Reusing knowledge of a knowledge base
- Level 8: Reorganising a knowledge base

While complex domain ontologies with extensive formal semantics can assist reasoning systems draw inferences and conclusions, light-weight ontologies are more often used for information extraction tasks. Several annotation services have been developed over the years to assist in semantically marking up documents and web pages.

Manual creation of ontologies is a challenging process and consists of a set of eight steps [Nm01, Av04]:

- determine scope (understand the purpose and application of the ontology, who will use and maintain it, which domain will the ontology represent)
- consider reuse (widely available ontologies from third parties can provide a good starting point to create an ontology)
- enumerate terms (identify the main terms and their properties that are going to be discussed)
- define taxonomy (using a top-down, bottom-up or a combinational approach, build a hierarchy of the identified concepts to describe the domain using concepts such as *rdfs:subClassOf* and *owl:subClassOf*)
- define properties (identify object and data properties associated with classes that have been identified — this step is often performed in conjunction with the previous)
- define facets (make use of the expressivity provided by OWL — enriching properties with cardinality, required values and relational characteristics)
- define instances (instantiate ontology classes from legacy data or information extracted from text)
- check for anomalies (detect inconsistencies)

The tutorial by [Nm01] covers the entire process of ontology creation from scratch in a greater detail. It is important to note that most of these steps are fields of research in their own right requiring extensive deliberation among domain experts.

The Semantic Web provides the means to make use of existing knowledge to describe a domain. For example, multiple domains need to describe the notion of time. A generic description of time, its units of measurement and various representations can be created with one ontology, and it could be simply reused in various definitions of different domains [Nm01]. [BMT05] describes ontology reuse as “*the process in which available (ontological) knowledge is used as input to generate new ontologies*”. The content of the knowledge sources and the domain overlap dictates how ontologies will be reused

— either by integration or by ontology merging. Reusing ontologies help in reducing engineering costs as it prevents building new ontologies that are already existing. However, the authors note that it could be possible that incorporating an existing ontology could involve certain costs such as finding, getting familiar with, and adapt ontologies to fit the purpose of a new ontology.

Drawing analogy from methods in software engineering, [Pin99] notes that large domain ontologies should be comprised of smaller manageable separate ontologies used as building blocks. Pinto describes the two perspectives of reusing ontologies: building an ontology by reusing other ontologies which are parts of the final ontology (ontology integration); and building an ontology by “*merging different ontologies on the same subject into a single one that ‘unifies’ all of them*”. [Avo4] highlights the variety of ontologies and possible sources that are available to be used- codified bodies of existing knowledge (ontologies carefully constructed by experts over the years), integrated vocabularies (merge several independently developed ontologies), upper-level ontologies (generally applicable ontologies, as opposed to highly domain level ones), topic hierarchies (sets of terms, loosely arranged in a hierarchical fashion), Linguistic resources (resources such as WordNet provides a good starting point for building ontologies), ontology libraries (repositories such as Swoogle⁴² index ontologies to be retrieved using keyword searches).

[Obe13] discusses the challenges of incorporating ontologies in the enterprise: cost-benefit ratio (incorporating ontologies can involve significant costs such as hiring ontology experts, training, technical integration costs, modelling, maintenance, technology buy-in or redevelopment costs); training (training existing employees to adapt to new ontology practices and logic-based systems); calculating benefits (measuring how much of a benefit will occur once ontology practices have been adopted); technical integration (deciding build or buy the solution, maturity of the available tools, scalability); multiple ontologies with different ontology languages (editors, reasoners, stores need to be integrated); modelling (domain experts may not possess technical expertise to model ontologies, cost involved with modelling, use case specific ontology modelling).

[Avo4] highlights two major challenges faced by Semantic Web technologies for a wider scale adoption and implementation: support the re-engineering task of semantic enrichment and providing means for maintaining and adopting semantic metadata. While research efforts over the past few years have provided with matured ontology engineering tools, manual ontology acquisition is still a challenging process. By making use of machine learning techniques, such issues and challenges could be solved — machine learning can support tasks like extraction of ontologies from existing web data, extraction of relational data and metadata from existing web data, merging and mapping ontologies, maintaining ontologies by analysing instance data, improving applications by observing users. Machine learning techniques can support the following three types of techniques — Natural Language ontologies (large ontologies, requiring few updates, representing background knowledge of the system, help expanding user queries), domain ontologies (contains knowledge about a restricted domain, requires the involvement of a knowledge engineer to be developed) and ontology instances (can be generated automatically and more frequently, ontology remains unchanged).

42 <http://swoogle.umbc.edu/>

3.4 SEMANTIC KNOWLEDGE MANAGEMENT SYSTEMS

[DFH03] proposes a generic architecture for Knowledge Management based on the Semantic Web. The architecture combines together several sub-systems to form an integrated system using a different style of engineering [Av04] for the On-To-Knowledge project⁴³. The on-to-knowledge system is discussed here, as it provides a good example of a complete system with an integrated semantic Knowledge Management system — most of the Semantic Web research in Knowledge Management has been focussed on isolated sub systems.

The process of Knowledge Management starts with the data repositories, consisting of large sets of semi-structured and unstructured documents. Information from these documents are then extracted and the documents are annotated based on ontologies that reside on ontology repository. The resultant RDF graphs are then stored in the annotated data repository. Ontology middleware and reasoning provides administrative support and infrastructure to ease integration with applications. Ontology editors such as protégé⁴⁴ and neon toolkit⁴⁵ can be used by knowledge engineers and domain experts to refine or update domain definitions. Users and tools can then access the information and use it by browsing, visualising, sharing with colleagues or perform automatic summarisation and so on.

The architecture is a way of describing how several Semantic Web tools can be incorporated into “a single lightweight architecture using Semantic Web standards to achieve interoperability between independently engineered tools” [Av04]. The following describe how the authors achieve various functionalities in the sub-systems of the Knowledge Management system (though other researchers may have followed different techniques e.g. for information extraction as briefly discussed, the basic approach remains the same — extract useful information, annotate documents, store annotations and use):

- The **knowledge acquisition** phase is responsible for extracting information from semi structured (e.g. HTML tables, spreadsheets) and unstructured (Natural Language text) information sources. While the authors use a combination of statistical methods and shallow Natural Language technology to extract key concepts from unstructured documents, wrappers, induction and pattern recognition techniques are used to extract the content from semi-structured documents. In the information extraction phase, Kim [PKM⁺03] uses named entity recognisers to identify unique concepts from documents (i.e. text, HTML, XML, email etc) using light weight ontologies. SemTag performs automated semantic tagging of large corpora with terms from a standard ontology — the documents are first retrieved, tokenised and processed to find instances in the spotting pass. The next step, learning pass attempts to determine the distribution of terms. The final step, tagging pass disambiguates all the terms and finally adds an annotation into the database [DEG⁺03]. S-CREAM (Semi-automatic CREATION of Metadata) requires a user to manually annotate a corpus using a domain ontology. The system then attempts to learn the annotations, using which it creates new annotations for new documents [CHSo4]. Semantic information can also be directly entered from the point

43 <http://www.ontoknowledge.org/>

44 <http://protege.stanford.edu/>

45 http://neon-toolkit.org/wiki/Main_

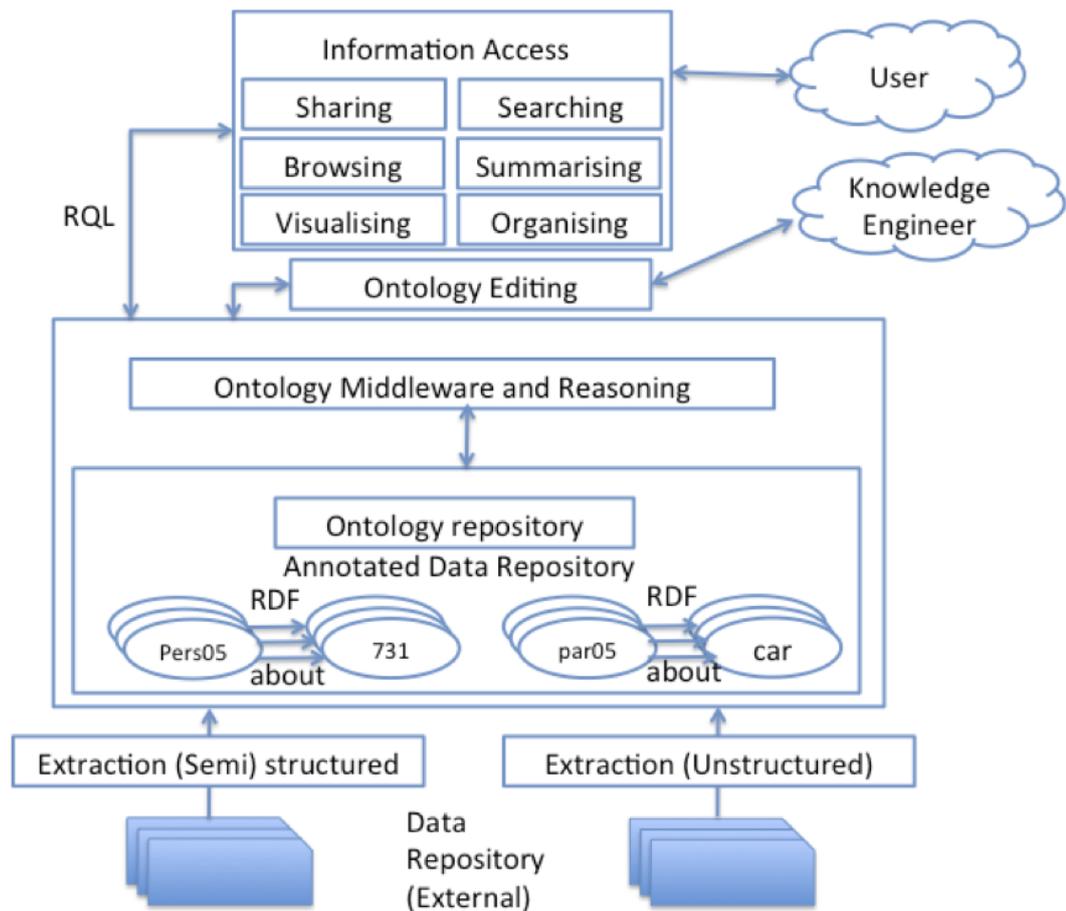


Figure 23: Davies' proposed architecture of Knowledge Management using Semantic Web technologies. Image recreated from [DFH03]

of knowledge creation, using forms that capture the context and semantics of the text [BCC⁺08].

- **Knowledge storage** — The resultant set of concepts from the knowledge acquisition step is then represented as **RDF** and **RDF** schema. The descriptions are then stored and made available to be retrieved at a later stage, using structured query languages such as **SPARQL**. The ontologies and its instances can be typically stored in standard **RDF** schema repositories such as **virtuoso**, **sesame** or **4store**.
- **Knowledge Maintenance** plays an important role in any organisation — since ontologies are the backbone of any Semantic Web application, ontologies need to be maintained and change needs to be controlled. This calls for providing functionality that enables change management, access control, transaction management and so on. Sophisticated editing environments allow knowledge engineers to retrieve ontologies from the ontology repository and edit them, finally placing them back to the repository.

- The **use** of knowledge can be done by several ways — sharing relevant information, enabling search across documents using ontologies, browsing and visualisation being a few.

While it is important to get a broader perspective on various semantic-web enabled Knowledge Management processes in an organisation, how such information is used in an organisation is of more relevance in the context of this chapter and the thesis in general. Specifically, the question that is interesting to ask is ‘what are the information visualisation techniques and interfaces that have been used to represent organisational knowledge so far?’. Though essentially research prototypes, several systems and applications have been developed specifically to assist in organisational Knowledge Management tasks — the following are just a few, in addition to several others discussed in Chapter 4. Research efforts have mostly focussed toward search systems

[DW04] discusses that the lack of semantic metadata in World Wide Web (WWW) and intranet documents creates a need for a keyword based approach and a capability for searching across metadata in order to ensure that there is a greater coverage. If any document does not contain any metadata, it would still be searchable using traditional free text search. The authors present QuizRDF, a system that allows searching across documents using keywords as well as RDF annotations. In addition, it allows users to browse and query the ontology. A second motivation of the authors for such an approach toward querying is to allow a mix of browsing and searching activity, which promises to provide a more powerful tool for Information Seeking activity. A further motivation for a combinational approach lies in the fact that RDF descriptions would not be enough to replace the content of the entire document — providing the ability to search the content not captured in RDF ensures that the users have the ability to search any section of any document.

QuizRDF provides users with a drop-down list containing all the resource types stored in the index and a text box to enter Natural Language text, as shown in Figure 24. The figure shows the results obtained when a user has selected ‘Employee’ and entered first and last names as values for the data properties. The object properties are listed alongside the drop-down menu — ‘HasSkills’ and ‘WorksInProject’. These ontological properties and classes are clickable, allowing users to browse the ontology.

[BCC⁺08] lists similar reasons where documents may have incomplete annotations or manual annotations may be spurious to motivate their hybrid search system. The authors provide a further motivation where the ontology used to annotate a particular document may not be representative of the context of the document — a different ontology annotating a document meant for a different purpose would signify that the annotations do not reflect the true nature and content of the document. The authors present a hybrid search system, K-Search, by combining keyword search and semantic search. The system provides a form-based entry interface to the users, where keywords including boolean operations can be entered into a form field (keyword search) and metadata constraints can be entered by clicking on the ontology tree and entering restrictions on the form thereby generated.

The results are presented to the users in a sortable tabular display, showing a textual snippet of the keyword, as well as the metadata restrictions. Clicking on a document in the table shows a preview of the document, along with highlighting the annotations

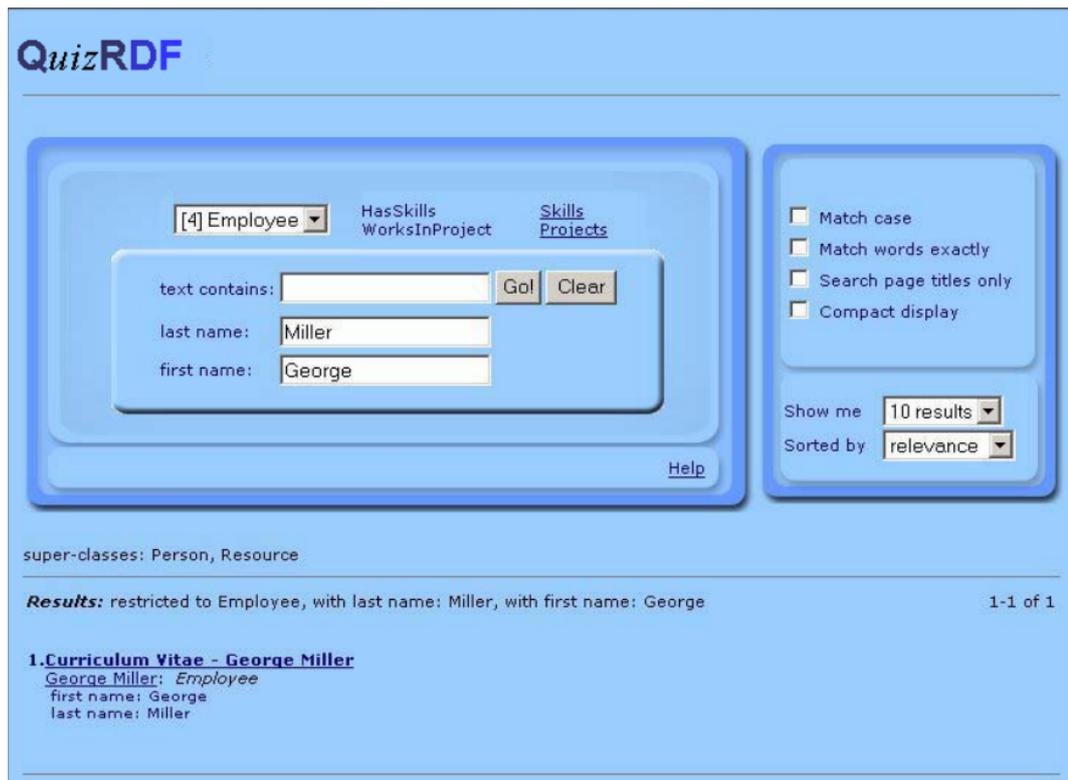


Figure 24: QuizRDF. Image from [DW04]

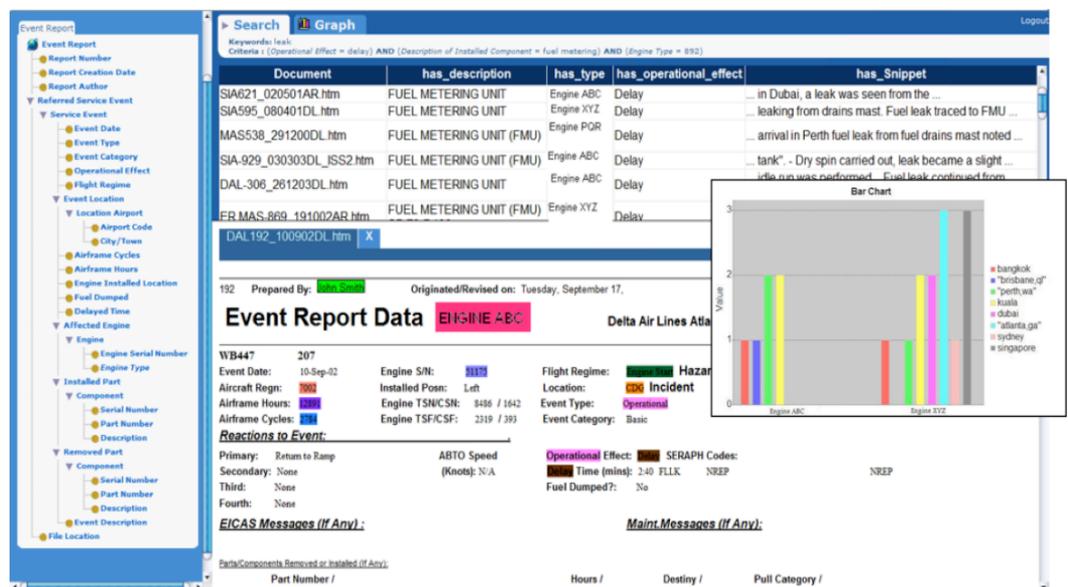


Figure 25: Results from K-Search, a hybrid search system showing document annotations and graphs of result sets as well as text snippets from the documents that match a keyword and semantic query. Image from [BCC+08]

within the document. In addition to the document preview, the system also shows a graph produced from the results, as shown in Figure 25.

[FW02] presents an interesting application of information visualisation in information management — to be used by two types of users: information consumers to ‘receive relevant structured content in their personalised portal’ and information managers to filter and structure the information and maintaining user profiles. The Spectacle Cluster Map provides a visualisation of the instances of several classes, using the classes as its main organisational principle, as shown in Figure 26.

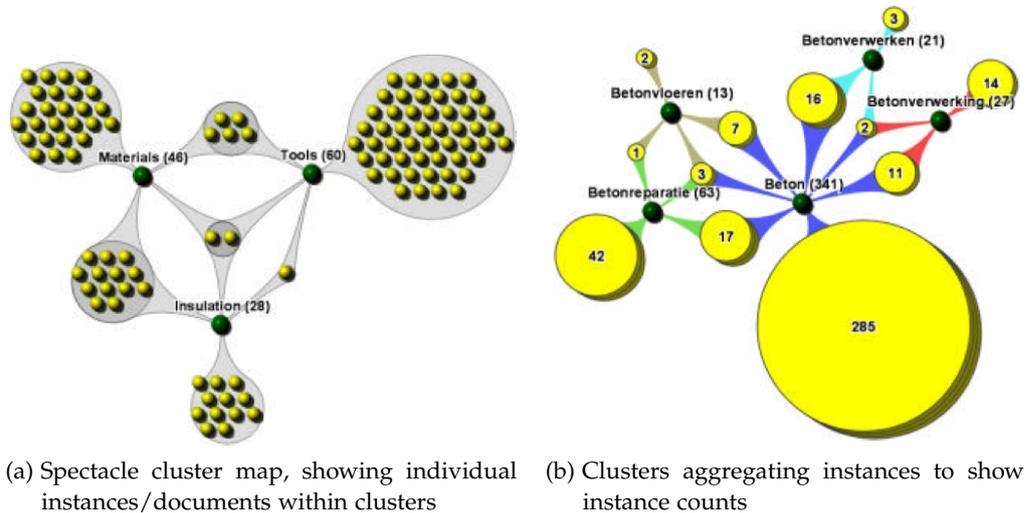


Figure 26: Spectacle Cluster Map. Images From [FW02]

While smaller spheres indicate instances (here, documents), larger spheres represent classes. The instances are clustered together on the basis of their class memberships. A balloon shaped edge connecting a cluster to a class indicates the instances are members of that class — clusters can have multiple memberships too. Such visualisations are helpful to provide a lot of information to users — specifically related to semantic closeness of instances. The authors discuss a few applications where information managers can make use of such applications — acquiring insight, monitoring information repositories and visualising automatic classification quality. In order to deal with scalability, the system aggregates instance clusters and provides the total number of documents/instances belonging to a particular cluster, as shown in Figure 26, right.

3.5 AESTHETICS AND ART IN INFORMATION VISUALISATION

Deriving an understanding of the term *aesthetics* from its antonym (anaesthetic), which means to “dull or deaden, causing sleepiness and numbness”, [CM06] explains aesthetics as something which awakens the senses and “enlivens or invigorates both body and mind”. [Heko6] explains aesthetics as “the pleasure attained from sensory perception”. [SH07] provides some clarification on several misunderstandings concerning the meaning of aesthetics and notes that aesthetics is not strictly applied to the visual domain, but can also include other sensory perception like sounds, touch, smell and so on. Aesthetics apply not only to art, but also to natural phenomena such as people, weather conditions or landscapes.

Though the focus of information visualisation lies in the functionalities and effectiveness of visualisations in communicating information, the potential positive influence of aesthetics on tasks-based activities are often neglected [LM07]. *Information Aesthetics*, combining creative design with information visualisation is a growing area of research. Developing interactive systems which can evoke positive responses from end users (like being comfortable using the system, having an enjoyable user experience and easy to use) is an important aspect of interaction design [RSP02]. Understanding the emotions that are elicited by users while a system is being used is an area of research which is of great interest, specially in the field of human computer interaction and information visualisation. Several researchers, over the past years have attempted to model and explain the human perception of aesthetics : four such models have been described.

A well thought design for an interface can go a long way in improving the experience for users. [Noro4] discusses that beautifully designed products make users feel positive and good, thereby putting them in a state of mind that makes them more receptive and open. This positive change encourages them to be creative and arrive at solutions that can assist them in dealing with their problems. They propose a model of emotion that attempts to explain how humans react to stressful and pleasurable situations, by distinguishing the human brain into three levels: (1) *A visceral level* which responds rapidly to stimuli, decides what is safe or dangerous, pleasurable or unpleasant, good or bad, and is responsible for triggering emotions like sadness, anger, joy and so on; (2) *A behavioural level* which dictates most of the human activities like talking, driving, writing and typing; (3) *A reflective level* which brings about conscious thought. The primary notion is that the positive or negative state that we are in dictates how we think. If an individual is frustrated, scared or angry, he is concerned about how to overcome his fear or solve his problems, thus resulting in being tensed. On the other hand, if an individual is happy and comfortable, his body is relaxed and assumes a less focussed stance, thereby enabling creativity.

Jordan proposes a different model on the pleasurable experiences of interacting with objects [Jor02]. The model attempts to explain the different kinds of pleasure that one has, which designers need to be aware of while designing a product. The kinds of pleasure that are presented are: *physio-pleasure*, *socio-pleasure*, *psycho-pleasure* and *ideo-pleasure*. *Physio-pleasure* relate to pleasure from the sensory organs like sight, touch, taste and smell. *Socio-pleasure* refers to pleasure from being in the company of other people like friends, family, loved ones and so on. *Psycho-pleasure* relates to the emotional and cognitive feelings people experience when interacting with products like easy to use and satisfying websites. *Ideo-pleasure* refers to the aesthetics of a product as well as the personal and cultural attributes of the product. Designers should understand their product and the kind of pleasure it is required to deliver — it is not necessary to deliver products that tries to provide all the kinds of pleasures.

[LBOA04] attempts to explain the psychological theories behind aesthetic experiences that we experience in everyday life by analysing the features of modern art. Their model consists of five stages: *perception*, *explicit classification*, *implicit classification*, *cognitive mastering* and *evaluation*. A work of art is the input to the model, while the result of the five stages are two outputs: aesthetic judgement and aesthetic emotion. Upon exposure to the input (work of art), an individual attempts to quickly perceptually analyse it — this can be on the basis of several features like colour, contrast, symmetry, grouping and so

on. The next stage, implicit memory integration does not require a conscious involvement from the individual. This stage deals with three features that contribute toward a final aesthetic judgement — familiarity (the more exposed a person is to a work of art, the higher is the aesthetic preference), prototypicality (how representative an object is to a class of objects. e.g. art work of an artist or from an art school) and peak-shift phenomenon (how strongly some objects amplify the characteristics of familiar objects). Explicit classification relies on the knowledge and experience of the individual (an experienced individual would notice the content of the art in more details like the style of painting). The last three stages, explicit classification, cognitive mastering and evaluation are presented as a feedback — loop, where the cognitive mastering stage can either help revealing a satisfactory understanding or increase or decrease the level of ambiguity. Experts refer to their knowledge of style and visual features to evaluate their understanding of the art after the cognitive mastering stage, whereas casual individuals evaluate the same by referring to the content and external sources. [Faio4] discusses several other models that have been proposed to explain our perception of aesthetics in art.

[MW04, WWM08] proposed an alternative model that attempts to describe a holistic experience by four threads — *sensual*, *emotional*, *compositional* and *spatio-temporal*. The sensual thread refers to our sensory and bodily experiences when interacting with a device (e.g. look and feel of mobile phones, thrill, fear and so on); Emotional thread relates to the feelings of oneself or others (e.g. sorrow, joy, happiness, anger are all feelings that are either directed to an object, oneself or another person); Compositional thread refers to the narrative of an experience (e.g. the who, what, how); Spatio-temporal thread relates to how space and time effect our experiences (e.g. the feeling of time slowing down or speeding up. These threads are expected to help designers understand the relationship between technology and experience. The framework can aid in developing a greater insight into designing a product as a whole, based on the experiences of its users, rather than a combination of smaller characteristics.

Toward the goal of understanding aesthetics from an information visualisation perspective, Lau and Moere propose a model of information aesthetics as defined by two factors — *data focus* and *mapping techniques*.

The model aims to analyse information aesthetics in an information visualisation perspective and artistic perspective. While mapping technique describes the process to represent abstract datasets as visual representations, data focus describes the ability of a visualisation to communicate information to users Figure 27 (Right). Figure 27 (Left) shows a few existing tools mapped on their model of information aesthetics.

An example of information aesthetics, *Circos* (Figure 28 (a), [KSB⁺09]) visualises variations in genomic structures (but portable to other data domains like auto industry, presidential debates etc.) as circular ideograms. The layout helps in visualising relationships among data objects. The system is an attempt at bringing aesthetically pleasing and easy to use interfaces to scientific research. Apart from circular ideograms, *Circos* also supports other visualisations such as line charts, scatter plots, heatmaps and histograms. An information user can provide his data in text files and select the items to be plotted using a configuration file, which generates the images. *Circos* also allows users to upload a tabular data from a spreadsheet (or use random data) to quickly visualise

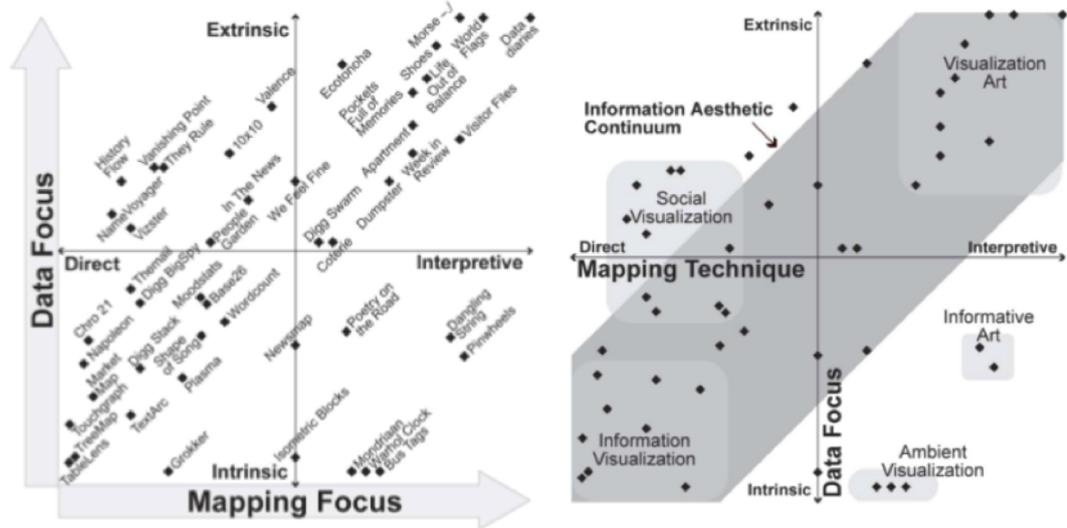


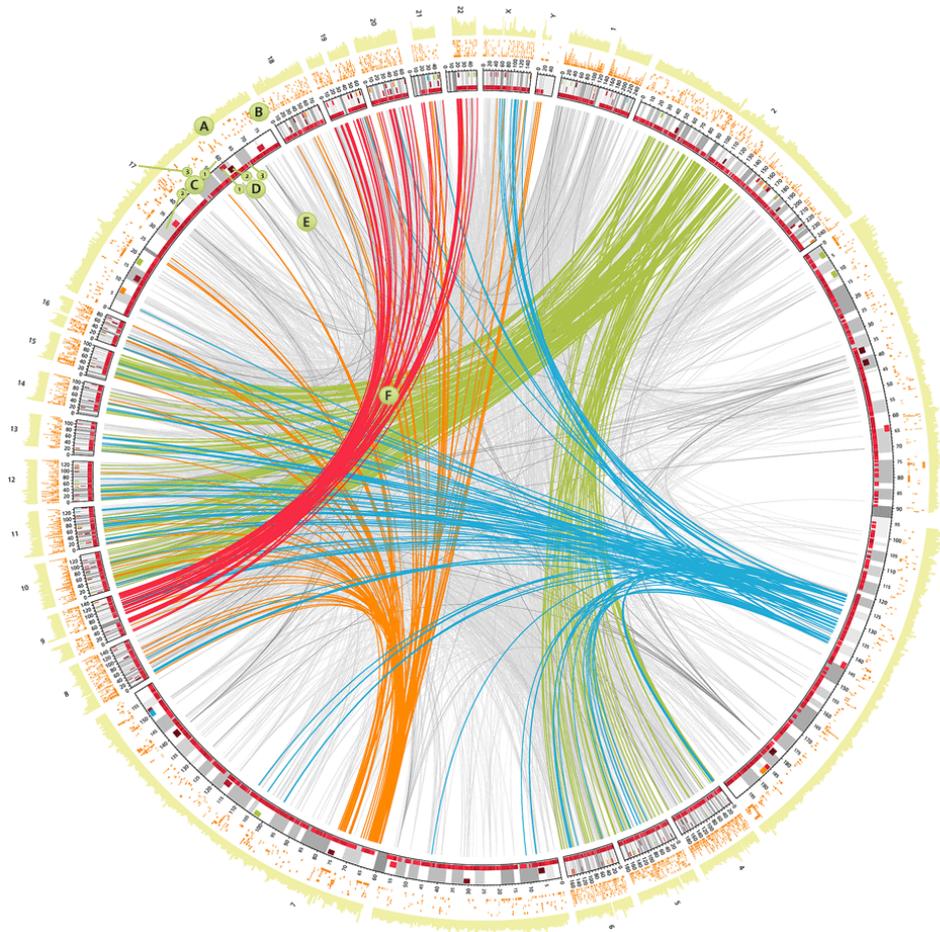
Figure 27: (Left) A few existing applications mapped on the information aesthetics model proposed by [LMo7]. (Right) Lau and Moere identifies how information visualisation and visualisation art map on their model

their data. The availability of large numbers of tutorials and user guides makes it very easy for users to use the software and apply it on their data.

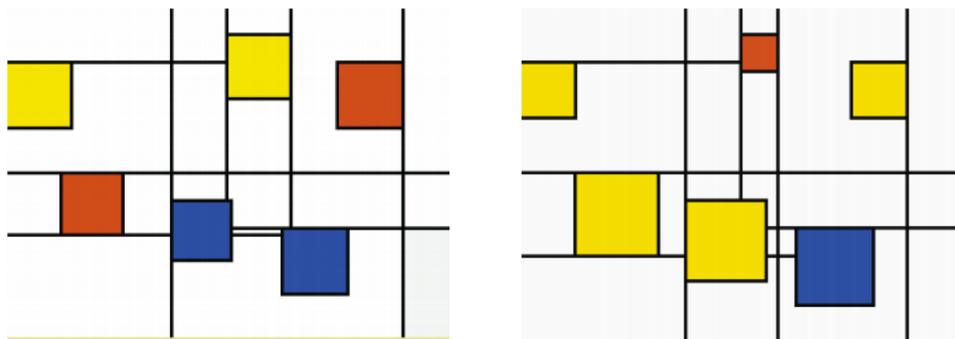
visualisations that are not rendered on computer screens, or *Ambient Information visualisation* also play an important role in providing information to users. The idea of Ambient information visualisation is to provide information to the user by presenting visualisations in the boundary of the user's location. A subset of Ambient Information visualisation, *Informative Art* requires the visualisation to be presented to the user in it's everyday environment. This enables to communicate aesthetically pleasing visualisations during our daily life. Dutch artist Piet Mondrian provides an effective template which can be used to produce information art. Figure 28 (b) shows an example where Information Art can be used to convey information to users [SLHo3]. The colours in Figure 28 (b) indicate the conditions of the weather, yellow indicates sunny or clear weather, blue indicates downfall (rain or snow) and red indicates cloudy weather. The position of the blocks roughly indicate their position around the world and the size of the blocks are mapped to the temperature of the city (larger blocks indicate greater temperatures). Similar visualisations have been developed that map employee's email exchanges in an organisation and bus traffic.

Often visualising large volumes of data as real-world objects, informative art provides a creative way of visualising data like in the High Altitude ⁴⁶(Figure 29 (a)) example that provides an interesting visualisation of the development of the leading stock market indices (like Dow Jones, Hang Seng, Nikkei and Dax) over the last 20-30 years as mountains. Combining the world of natural forces that shape a mountain's structure with the artificial economic situations, Najjar creates beautiful images of mountains with impressive cliff and rock formations

⁴⁶ High Altitude, http://infosthetics.com/archives/2010/03/high_altitude_the_stock_market_trends_as_realistic_mountain_ranges.html, Last Accessed 30/01/2011



(a) Circos visualises variations in genomic structures as circular ideograms. Image from Circos website (http://circos.ca/intro/genomic_data/) depicting the location of genes involved in diseases



(b) Example of Informative Art — Visualising weather in six cities during two different times of the year. (Left) Temperature distribution among six international cities in winter. (Right) Temperature distribution among six international cities in summer. Image from [SLH03]

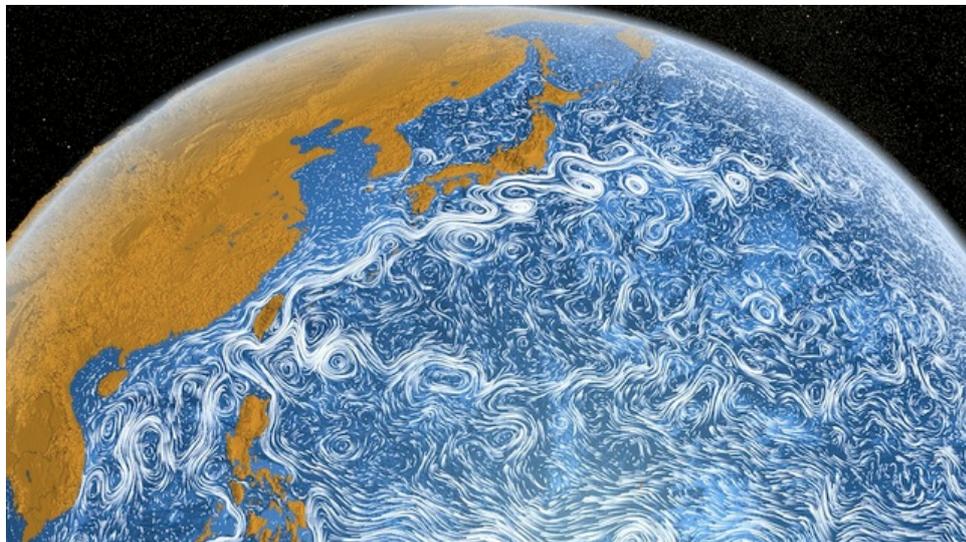
Figure 28: Examples of aesthetic information visualisations

NASA’s perpetual ocean⁴⁷ beautifully visualises surface ocean currents around the world using scientific data (Figure 29 (b)) like high resolution model of the global ocean

⁴⁷ http://infosthetics.com/archives/2012/03/nasa_perpetual_ocean_the_ocean_surface_currents_around_the_world.html



(a) High Altitude visualises stock market indices as mountain peaks, modified from mountain pictures. Image from http://infosthetics.com/archives/2010/03/high-altitude_the_stock_market_trends_as_realistic_mountain_ranges.html



(b) Ocean currents visualised on a map of the world, developed by NASA's Scientific visualisation Studio. Image from http://infosthetics.com/archives/2012/03/nasa-perpetual_ocean_the_ocean_surface_currents_around_the_world.html

Figure 29: Examples of beautiful information visualisations — providing information using artistic images

and sea-ice movements, sea surface levels, surface wind stress readings, sea surface temperature that help capture currents. The data constituting tens of thousands of ocean currents was collected over a period of two years between 2005 and 2007, and was rendered as a video. Such visualisations can help users identify major and minor current distributions all across the globe, and can potentially help in a multitude of applications like disaster recovery planning, weather prediction, optimisation of navigation and so on. While there is an increasing amount of effort recently being invested in developing aesthetically pleasing interfaces, there has been a significant amount of interest in developing a standardised set of principles that can be applied to interfaces in order to ensure the interface is aesthetically pleasing to users.

3.6 PRINCIPLES OF AESTHETIC DESIGN

Usability studies, focus groups, user interviews and studies can be a great way of understanding the usability and pleasantness of an interactive system. While such studies can only happen once a system has been implemented, it is left too late in the system's development cycle to pay an attention to the aesthetics. This motivates the need to generalise certain principles that can help ascertain the initial reaction of a user when he is initiated to the tool (this initial reaction is highly related to the Stage 1 in the model proposed by [Waro4] and Norman's visceral level [Noro4]). This process typically takes milliseconds and is mostly subconsciously realised by the user. The familiarity of a user with the system or product decides how much time and effort will be devoted by the individual on studying the product. Aesthetic principles as "*heuristics that consumers access to make aesthetic judgements*" [KG10] are developed within us through thousands of years of aesthetic experience. These principles are imbibed within us, and if products are built in accordance to these principles, they have a high chance of being accepted as aesthetically pleasing [Noro4].

Identifying principles and best practices for developing aesthetic design is a research field on its own, continuously evolving over the past decades, focussing on several application areas such as art, music, dance and more recently, information visualisation and interactive interfaces. Such design principles are generally not binding on the designer, and often non conformance can help in developing a better product ([ALB11] provides an example where Harry Beck, in 1933 tweaked the London underground map by straightening subway lines and spacing the stops evenly to better represent the sequence of stops). The following aspects are generally accepted as the aesthetic principles for layouts: balance, emphasis, movement, pattern, proportion, harmony and variety [Kimo6]. However, this list is not always necessary to be applied — there are variations of the list.

[MG05] presents the following list of principles applied in web design cases: balance (symmetrical and asymmetrical), rhythm (regular, flowing, progressive), proportion (proportion of dimension), dominance (dominant, sub-dominant, subordinate) and unity (the relationship between the visual components and elements and the complete visual scene). Apart from the aforementioned aspects, [MG05] discusses other concepts that are related which can affect an individual's aesthetic perception of a system: contrast (representing the notion of dynamic conflict between the different visual elements), positive and negative space (positive space refers to the objects in the environment, while negative space refers to the environment itself), rule of thirds (divide the frame of reference into a third and placing the objects on the lines in between), visual center (natural placement of visual focus), colour and typography (certain colour templates are more effective at presenting information)

Highlighting over a century of research in cognitive sciences, human perception, art, psychology, industrial design, [SH07] identified certain principles that can help product designers create better products that have a higher likelihood of being successful. The authors classified the principles in three categories: psychophysical, organisational and meaningful properties. Psychophysical properties are measurable and quantifiable characteristics of objects like colour, size and intensity. These properties have been extensively studied in product design and various findings such as the preferential or-

der of colours (blue, green or red, and yellow), the impact of brightness and contrast in product design and so on have emerged from multiple user studies. Organisational properties can be properties that dictate the structure of an object such as the perception of lines, edges, contours, or basic geometric shapes — such features need the active perception of individuals and can be measured and quantified in principle. The authors discuss that order, balanced compositions (e.g. eye tracking studies have revealed that changing the balance of an original artwork leads to confusion among viewers), symmetry (e.g. the preference of symmetrical faces over asymmetrical ones) and proportions (e.g. the preference of proportion depends on the object being designed, as well as the familiarity of a proportion) are inherent in designs that are accepted as good examples. The authors also note the need for complex structures and variety otherwise too many regular and balanced patterns would make the perception of our world as boring and uninteresting. Meaningful properties, on the other hand are subjective and are based on our previous experience and knowledge. Familiarity with objects and repeated exposure can make perceptual and cognitive processing more fluent, thereby increasing the perception of pleasantness. Repeatability however has its drawbacks as that can induce boredom and thereby lose its appeal. As humans, we categorise objects into distinguishable groups (prototypical behaviour) and the easier it is for us to classify an object into a class of objects, the more aesthetically attractive it is perceived. [SH07] also notes that the perception of an aesthetically pleasing design also depends on several other features like our individual tastes, culture, knowledge, experience, sensitivity and so on.

3.7 AESTHETICS AND SEMANTIC WEB

Schwarz et al. and Lang [SWM⁺04, Lan09] noted the correlation between the time taken to process an object and human aesthetic response — the perception of beauty can be explained as a function of the fluency in the processing of the object. Two phases of the human cognitive system that come into play are the *preattentive phase* (low level processes before processing the sensory information) and *interpretive phase* (representations that are learned). The perception of aesthetics is therefore based on the “*combination of cognitive and sensory modes of experiences*” [FDPL05]. The pre-attentive processing stage exists before conscious processing, and occurs at Norman’s visceral level [Nor04, MPWG12]. This raises the question — how can information be represented to be quickly processed by our preattentive processes? Very interesting to this context is the work conducted by Healey in [HBE92, HBE95, HBE96], where the authors investigate visualisation of multivariate data using preattentive processing in a rapid manner (less than 250ms).

The experiments conducted by Healey drew several interesting conclusions such as:

- Hue can be used as a mechanism to rapidly and accurately determine a target (example, an anomaly);
- Form(shape) can be used to determine targets if hue is not varied; varying hue affects the ability to determine a form-defined target;
- Varying form does not affect the ability to determine a hue-defined target; location is not a deterministic factor in identifying a target.

Several other cognitive aspects have also been proposed elsewhere, such as a minimalistic approach [Tuf86], symmetrical layouts, Golden Ratio [FDPL05, Eico3] and so on.

Tufte's work, in [Tuf86] is also significant for identifying aesthetic principles for information visualisation. He lists several guidelines for building attractive displays of statistical information:

- have a properly chosen format and design
- use words, numbers, and drawing together
- reflect a balance, a proportion, a sense of relevant scale
- display and accessible complexity of detail
- often have a narrative quality, a story to tell about the data
- are drawn in a professional manner, with the technical details of production done with care
- avoid content-free decoration

Bennet [BRSG07] discussed Gestalt principles applied to graph drawing from two perspectives — perceptual grouping (the ability to extract low-level primitive visual features from images and formulate a higher-level structure, e.g. grouping simple and stable figures that are similar in shape, located nearby etc.) and perceptual segregation (the ability to separate features from images and grouping them into mutually exclusive areas in order to construct a useful representation of the image, e.g. symmetry, orientation, contours). Several principles were also noted by [MG05, Kim06] such as *balance* (symmetrical and asymmetrical), *rhythm* (regular, flowing, progressive), *proportion* (proportion of dimension), *dominance* (dominant, sub-dominant, subordinate), *unity* (the relationship between the visual components and elements and the complete visual scene), *emphasis*, *movement*, *pattern*, *harmony* and *variety*.

Beck [BBD09] investigated aesthetic dimensions for dynamic graphs and proposed principles for general aesthetics, dynamic aesthetics and scalability applicable to three types of graph representation techniques — node-link, matrix and list. Among the general aesthetic principles, the authors list principles such as reduce *visual clutter*, reduce *spatial aliases*, *spatial matching of multiple representatives* and *maximise compactness*. Beck also notes while dealing with dynamic data, it is important to preserve user's mental map⁴⁸ in order to facilitate graph comparisons and ensure the user requires minimum cognitive load⁴⁹ to compare present graphs with previous one to perceive changes. Beck also points out that temporal aliases⁵⁰ should be minimised so that a continuity between consecutive frames is established.

Owing to the graph-based nature of Semantic Web, in addition to being multidimensional and multivariate, principles of graph visualisation are applicable in certain contexts. The nature of Semantic Web data, in addition to being multidimensional and multivariate, is graphical. Several principles of graph visualisation aesthetics are therefore applicable in this context [BRSG07, Eico03]. Eichelberger list several aesthetic principles

⁴⁸ Mental Map is the abstract structural information a user gathers by looking at graph layouts

⁴⁹ Cognitive Load is the amount of information needed by the working memory of a user in order to process a visualisation

⁵⁰ Visual elements that are mistaken for one another due to their temporal placement

to be followed while drawing class diagrams in Unified Modelling Language (UML)⁵¹, which we believe are highly pertinent and can be considered while designing similar LD applications. The following principles are the most appropriate for data visualisation in the Semantic Web, specifically for graph based visualisations:

- Separate hierarchical and non-hierarchical relations, hierarchy should be clearly visible
- Centrally position parents or children- this is particularly useful for hierarchical representations such as Semantic Web data. Child nodes to be located at median position of its parents.
- Nodes should be clustered according to semantic reasons — semantically similar nodes should be positioned close
- Avoid, if possible crossings and overlappings on edges
- Vicinity of comment nodes — comments connected to other models should be located as close as possible to the connected nodes. In a LD setting, this can be a way of connecting multiple ontologies/datasets/graphs.
- Adornments should be clearly assigned to their model elements. In a LD setting, graphs adornments/additional specifications (e.g. labels, icons etc.) should be standardised based on the same group/ functionality.
- Respect graph drawing constraints — aspect ratio, compact drawing, symmetry, minimisation of edge bends

3.8 SUMMARY

Over a decade of research in Semantic Web applications and user interfaces has produced several interesting search and visualisation systems, that presents information to users by employing a variety of techniques from basic textual lists to highly interactive visual elements, graphs and charts. The growing amount of linked open data has encouraged developers and researchers to build rich visualisations and mashups that can exploit such datasets. This has promoted a change from developing solutions meant for proprietary datasets and structures to generic technologies that is applicable to any dataset that conforms to a standardised format.

This literature survey presented related work in two ways: how users interact with systems while expressing their Information Need; and how systems present semantic data while displaying query results. Kaufmann [KBf07, KB10], in her thesis explained that the ideal querying system should need mechanisms for users to enter queries which are formal as well as natural — i.e. there should be a balance of naturalness and formality in the way users enter their query. Upon reflection, this can be seen as an interesting approach, since very few systems allow users to specify their Information Need in a controlled vocabulary. While satisfying the constant demand for a well-performing keyword-based google-like search engine is highly challenging, the opportunities provided by Semantic Web and highly structured datasets are immense. Structured data can

⁵¹ UML specification <http://www.omg.org/spec/>

make highly structured queries possible and thus, a balance in formal and structured query mechanisms is important. In the evaluation later conducted by Kaufmann (using NLP Reduce, Querix, Ginseng and Semantic Crystal), it was observed that the most liked interface and query language was Querix (controlled Natural Language). However, a more formal approach like Ginseng was highly disliked by users and the system was reported to be highly restrictive. Interestingly enough, a highly formal approach like Semantic Crystal was judged to be the second best liked system after Querix.

While there is a need for a balance in formality and naturalness of query mechanisms, I believe each query mechanism is useful for its own purpose. Given the vast opportunity arising out of structured data, a highly formal approach is essential to build highly precise queries, which may be extremely difficult for a full Natural Language system to employ. An ideal solution, in my opinion would be a combination of several approaches, applied in conjunction with each other based on tasks, situations and requirements.

Related work in presentation of semantic data has been discussed in two dimensions: visual representation and interactivity. Only a few systems among a plethora of existing systems have been considered — that attempts to encapsulate each representative section. Whilst minimal visual representations are simple, clear and easy to use, they lack interactivity. Plain text lists and tables do not make use of the opportunities provided by semantics. Supporting navigation across and within datasets, search engines and mashups enable users to follow links, view data using multiple paradigms and gather a greater understanding of result sets. Visualisation systems present semantic data in a more visual manner and support greater interaction. It is important to note that while Kaufmann's formality continuum expresses such systems as a highly formal one, the visual representation employed by such systems are based on the nature of semantic data. Hence, while the interaction during query formulation is a highly formal one, the visual representation of such data is a highly natural one.

This chapter also explored related work on aesthetics and Visual Analytics from a more specific context of the *SW*. An explicit focus on aesthetics in the Semantic Web has been lacking and little research has been done in the area. One might argue that the same principles of interface design should be applicable in Semantic Web interface design without any extra effort. However, in order to fully exploit the highly interlinked and well connected nature of semantic data, new kinds of visualisation and interaction techniques are needed. Formalising an approach to design aesthetic interfaces can ease the process of introducing new visualisations to research prototypes and production systems in a way that is pleasant and appealing to end users. An analysis of the literature identified several key principles and guidelines that can be applicable in Semantic Web. These principles are presented in two groups — general aesthetic principles and principles for node-link representations.

The following chapter presents the methodology that was employed to develop the solutions discussed in the thesis. Two solutions are presented – the first for visualising semantic result sets in a multi-visualisation approach, the second for exploring and querying semantic data using a interactive and visual mechanism. The next chapter also discusses how users have been engaged while developing these solutions, using an iterative user centered design process.

Part II

METHODOLOGY

As outlined in the research questions presented in Chapter 1, this thesis investigates how Visual Analytic approaches can be employed to effectively explore large scale distributed semantic datasets within the context of Knowledge Management. The thesis claims that such approaches can be developed by implementing a generic solution that exploits familiar interaction mechanisms, designed with specific attention to aesthetics and developed with the close cooperation of user communities as iterative processes. This chapter discusses how requirements were drawn for addressing the research questions, and how approaches were developed in order to address the requirements. Adopting a [UCD](#) methodology, the approaches were continuously refined after every iteration of evaluation following several subsequent re-design stages. An analysis of the requirements for Visual Analytic solutions for Knowledge Management is discussed as an initial introduction to the chapter.

4.1 ANALYSIS OF REQUIREMENTS FOR VISUAL ANALYTIC SOLUTIONS IN KNOWLEDGE MANAGEMENT

4.1.1 *Stakeholder Requirements*

The work discussed in this thesis is aimed at three categories of key stakeholders:

1. Businesses and Organisations in a Knowledge Management perspective;
2. Data providers and owners; and
3. User communities (any typical user, domain experts etc.), developers and enthusiasts;

With the growth of an organisation, there is a need to introduce highly specialised workforce. In order to streamline various processes, organisations are further grouped into departments, horizontal and vertical levels. This segregation can happen on the basis of one or a combination of several criteria like business area, domain, function, clientele, etc. For example, a functional segregation of a company would be to divide the workforce into accounts, engineering, manufacturing, design, sales units and so on. While the size of these units can vary, they are further subdivided on the basis of authority- most of these units consist of several tiers of supervisors, each answerable to his/her superior. At the top, executives and directors oversee the performance of the several layers of management, who are in turn responsible for the majority of the employees. This, in essence results in an organisational structure resembling a pyramid, where the senior decision makers are low in numbers, at the top and the large workforce of employees, specialised with their own skill sets at the bottom. A typical organisation consists of a hierarchy of employees, each level being target users for the developed solutions.

As they play different roles in the organisation, the employees in these levels have their own set of Information Needs. In a typical manufacturing organisation, product manufacturers, maintenance staff, product designers, testers etc. have a very specific skillset. Hence, their contribution would directly relate to the products that are being manufactured — e.g. design a better/cheaper product, manufacture products based on specifications, perform maintenance operations and so on. Such tasks need employees to record daily activity and decisions in well structured formats, thereby documenting every stage of their work. One of the most important Information Need, in case of adversity is understanding if any similar conditions have occurred in the past, and what were the decisions undertaken at that time to rectify the issue. This requires accessing historical records of all maintenance/design/manufacturing activities to learn from previous experience. The Information Need in such scenarios can be answered by highly directed queries to single/multiple data sources to access the related records.

Managers, on the other hand do not work directly with the products, though they may be involved with the management of relevant processes. A manager's primary goal is to understand how to improve and streamline processes and evaluate performances of different teams and units within the organisation. The goal of senior managers and key decision makers is to understand how to streamline processes within the organisation, increase profits, accumulate financial information to take budgetary decisions, understanding where company efforts should be focussed and so on. Managers and senior managers need to explore organisational data to discover information, often without having a fixed Information Need — exploring the company archives to find interesting facts and figures and discovering performance patterns being a few examples. The main Information Need for such employees is aggregating different types of operational and financial data to draw conclusions from a high level perspective.

Considering the three tiers of an organisation, I believe that the ultimate Information Needs of the employees consist of two types :

1. Finding highly specific information by drilling-down to individual data points (querying); and
2. Assessing and exploring information from a high level to gather an overview (exploring);

4.1.2 *A User Centered Approach to Understanding Domains and Users*

The types of users that have been involved throughout the process were managers, software developers, researchers and knowledge workers from the aerospace engineering and life sciences domains. Several students were also involved, who had volunteered to evaluate the prototypes. Users who were involved also possessed a wide range of expertise with respect to Semantic Web and search systems — from no knowledge of semantic technologies to highly proficient in search systems and Semantic Web experts. In addition to students, several participants from research projects were recruited. Three projects provided support during the research phase and provided valuable datasets,

domain-specific expertise and invaluable feedback sessions — SAMULET¹, SILOET² (aerospace engineering) and Grassportal³ (plant life sciences). Regular project meetings and discussions with project members also provided valuable inputs which were fed-back during the next cycle of development. An independent evaluation conducted by the SEALS project⁴ comparing one of the systems with other established systems provided very helpful comments to seed the next stage of development. Following the user centered design process, it was possible to continuously involve users thereby creating progressively improving systems. This also provided ways to better understand users, their expectations and requirements.

The involvement of users from different domains has ensured that the research was true to the UCD methodology at all times. Since the technology being sought is a generic one, portable to any domain, the involvement of users has been from different domains. Different stages of the user centered design have spanned across the three year research period, constituting parts of the development cycle of the two systems implemented. This section discusses the understanding and study phase, where initial sets of interviews were conducted with users from two different domains, and an analysis of the available datasets was conducted.

4.1.3 *Understanding Domains*

The Understanding phase in UCD focusses on human values and draws reflective thoughts and conceptual analysis from different strands of research. This stage involves interviewing stakeholders and users to understand what are the possible outcomes of using the technology to be designed. There were three sets of distinct “Understand” stages, each pertaining to a particular solution. The first Understand stage during the development of a system, ViziSocial [RM10] was during the first few weeks of research. At this stage, it was important to understand from a higher perspective how Visual Analytic solutions can aid in providing insight into comprehending data. This stage coincided with the literature survey, where an inspection of the available literature was also being conducted in order to understand the motivations and role of visualisations from the perspective of Knowledge Management as well as situational awareness.

4.1.3.1 *Understanding the Aerospace Domain*

The second Understand phase was conducted over a significantly longer period, extending to the end of the the initial five months of research. During this phase, several stakeholders at a manufacturing organisation (Rolls Royce) were interviewed and on site factory-visits of manufacturing facilities were also conducted. Motivated by an organisational Knowledge Management perspective in the manufacturing unit of the aerospace industry, stakeholders from several departments and application areas were interviewed. The interview sessions were structured in two distinct stages: understand-

¹ http://www.rolls-royce.com/about/technology/research_programmes/manufacturing_technologies/

² http://www.rolls-royce.com/about/technology/research_programmes/gas_turbine_programmes/siloet.jsp

³ <http://www.grassportal.org/>

⁴ <http://www.seals-project.eu/>

ing data, domain and information needs; and understanding the visualisation needs, existing techniques and open-ended ideas.

INFORMATION NEEDS This stage aimed at gathering an understanding of processes, business units, individuals and expertise involved in manufacturing processes. Six stakeholders were invited for focus groups and interview sessions where a set of 24 prepared questions were asked in order to stimulate discussions in addition to a few open-ended themes for discussions. The questions were prepared by experts in manufacturing and were highly specific to the aerospace domain as well as the SAMULET project and a few examples are as follows:

- *What standard features are in a commodity?*
- *What is current knowledge base on commodity / standard feature? E.g. how is it linked to commodity, manufacturing process routing, manufacturing location, current cost, cost reduction history.*
- *What impact will changing metal type / feature design have on the overall cost?*

The stakeholders were interviewed individually as a part of a group meeting. The interviews were structured, with a lead interviewer asking the stakeholders a seed question from the list of questions. Once an answer was provided, the rest of the attendees reflected on the response, stimulating further discussions. Each interview session was conducted for an hour with a minimum of three participants, and a maximum of six excluding the interviewee. The interviews involved senior representatives from Computer Aided Manufacturing team, Cost Reduction team, Patent team, Concessions team, and a Process Development manager. While all stakeholders were involved in manufacturing within the aerospace industry, their area of interest and specialisation focussed on different business processes. The Computer Aided Manufacturing (CAM) team was interested in better knowledge practices that surrounds a standard feature⁵ of a component. The team's interest was also on manufacturing processes, machines and factories — understanding the performance of the units in a global as well as local scale. The Cost Reduction team's interest was in organising and gaining easier access to ideas and initiatives aimed at reducing manufacturing or operational costs of the unit. The Patent team's main interest was managing and understanding patent requests across different business units, as well as understanding competitor patent applications. This is a challenging task, as competitors use different and often evolving technical nomenclature to depict physical objects and characteristics. Concessions users are concerned with assessing whether to use engine parts that present features that have been manufactured with a slight deviation from the original drawing. These deviations can include anything, from defective blades to bolts that have been manufactured with the wrong dimensions. Their main job is to make decisions on whether a concession should be granted for the use of a part that presents manufacture deviations to build an engine. The interest of a process development manager was to gain an understanding of dynamic data, particularly relevant to machines and factory layouts.

⁵ A standard feature is a manufacturing term that denotes a basic feature of a component such as 'round hole', 'cylinder' etc.

The interviews with manufacturing units provided significant insight into manufacturing processes and Information Needs of the key players. However, such targeted questionnaires were helpful only in providing more information about the highly specific domain of manufacturing in aerospace domain. More helpful, in the context of this thesis was what are the visualisation needs of such user communities.

VISUALISATION NEEDS Three other visualisation-specific questions ensued after the set of questions were discussed:

1. *What are the visualisations that you presently use ?*
2. *What are the visualisations that are needed for your daily activity?*
3. *What are the visualisations that would be nice to have ?*

The first of the three questions was aimed at understanding how users presently make use of their data. This also gave an idea of the level of familiarity they have with visualisations and interactions. Most of the interviewees mentioned the existing visualisations to be basic — bar graphs and pie charts mostly in order to report performance data. Indeed, most of the user groups in manufacturing and design communities are heavy users of Computer Aided Manufacturing and Computer Aided Design systems, where entire designs are visualised in three-dimensional views. However, since the context of this thesis is in a generic visualisation approach, the interest was in understanding what are the standard visualisations the communities were familiar with.

The next question attempted to understand what were the visualisation expectations from users. The users were shown several example visualisations that had been developed previously such as in [PMDC09]. Further discussions ensued from the demonstration of previously designed solutions and users recommended the need for geographical maps and engine maps. The third question encouraged users to think of blue-sky solutions to their Knowledge Management needs. This discussion, however was highly limited as it was noted that users appeared to be constrained by the various possible visualisations that were already demonstrated to them as well as the practical applicability of new solutions within an organisational setting.

Most of the interviewees appeared to use basic charts and graphs as presentations while reporting about efficiency and performance of their teams. Complex visualisations were mostly used in highly specific tasks such as Computer Aided Design or scientific data presentation. Factory visits at the company manufacturing facilities identified similar visualisations such as timetables, team work sequence diagrams⁶, bar charts, and pie charts. The same was replicated across the different manufacturing domains and stakeholders. A common suggestion from every stakeholder was to incorporate easily navigable factory maps and engine maps within an interface to visualise performance data.

⁶ Manufacturing workforce is organised into smaller teams, and their working shifts are indicated in timelines and gantt chart-inspired diagrams

4.1.3.2 Understanding Life Sciences

The association with another project, GrassPortal⁷ was also very helpful in understanding the domain of life sciences, more specifically global biodiversity of grasses. As a part of the research project, several scientists from the field were involved in synthesising taxonomic, phylogenetic⁸, biographic and environmental data related to the distribution and characterisation of grass species across the world. Since existing work is multi-stranded and highly diversified among various institutions and research organisations, the effort of the GrassPortal project was in integrating heterogeneous datasets into a comprehensive source of information for biologists and ecologists.

The first step in understanding the domain was to interview project members in order to understand the life sciences domain, and in particular, biological data. Different plant characteristics and distribution were also examined. Life science practitioners were also interviewed in order to understand how they presently accessed data in their daily tasks and how the information is made use of. An example scenario describes how the community presently manually aggregates information from several sources in order to fulfil an Information Need.

Step 1. Search

Step 2. Select Individual Grasses

Step 3. Forward query to another dataset

Step 4: Select grass to obtain distribution

Figure 30 illustrates a four-step process for obtaining grass distribution data. Step 1 shows a search for 'panicae' on the GrassBase website. Step 2 shows selecting individual grasses from the search results. Step 3 shows forwarding the query to another dataset, the Herbarium Catalogue. Step 4 shows selecting a grass to obtain its distribution on the Global Biodiversity Information Facility (GBIF) map.

Figure 30: An example scenario was explored where a biologist needs to understand the distribution of grasses belonging to a particular tribe (panicae). Step 1 – Step 3 Images from <http://www.kew.org/data/grasses-db.html>. Step 4 Image from <http://www.gbif.org/>

In the example (Figure 30), a biologist looks for the distribution of grasses that are annual and belong to the tribe 'panicae'. The only way he/she can find the information

⁷ <http://www.grassportal.org/>

⁸ Phylogenetics is a study of evolutionary relationships among groups of organisms (as from <http://en.wikipedia.org/wiki/Phylogenetics>)

is by querying the database at GrassBase⁹. The individual grass instances can then be selected and queried at another database, GBIF¹⁰ to be visualised on a map. Since there are no present means to aggregate information, the biologist needs to do so manually by creating a temporary database and then visualising the results on a map.

The third stage of understanding (Sept – Nov, 2010; Jun – Jul, 2011; Jul 2012) and Phase 2 (Jun – Jul 2011) was as a result of revisiting the domains and datasets following an evaluation cycle.

4.1.4 *Studying the Domains*

The two domains are disconnected from the other, and can help in understanding standard visualisations and interface requirements from a generic perspective. The most significant study phase occurred during the initial few months of research (Dec 2009 – Feb 2010): at this phase, in addition to factory visits at manufacturing facilities and user interviews, several aerospace datasets were made available. Furthermore, with the help of the Grassportal project, there were several types of data that were made available to be explored for understanding possible ways of visualising:

4.1.4.1 *The Aerospace Domain*

A jet engine has a life span of 40-50 years, during which regular maintenance activities are conducted on the engine. This includes engine overhauls, vibration tests, performance tests, maintenance reports and so on. Each of these processes involve careful and detailed documentation which results in a multitude of different types of documents being generated, each of which are in large volumes. In addition to maintenance records, engines generate a gigabyte of data every hour they are in operation, which are also examined and recorded. All of this information is warehoused by the service domain in a large aerospace company. Aerospace designers also produce several types of documents for example, Computer Aided Designing (CAD) files, new design ideas and so on. Manufacturing departments also require to produce different types of documents — manufacturing guides, machining ideas, cost reduction documents, machines performance data across different manufacturing facilities and so on. Harvesting such information across different domains and document types is immensely important to an organisation, since it provides a way for all users to access information. For example, service domain engineers might face an unexpected situation involving a particular engine. It is vital for engineers to take immediate steps to understand the situation and take urgent counter-measures. Having quick and easy access to all the related situations involving the same or any other engine, manufacturing processes involved in manufacturing the engine, design decisions taken and so on would be vital. Providing intuitive and interactive means to visualise such data would be extremely beneficial in such cases.

Some examples of the datasets that have been examined are as follows:

SERVICE INVESTIGATION REPORT A report generated by service engineers as a result of an investigation into an event. The report details the involved parts numbers,

⁹ GrassBase, <http://www.kew.org/data/grasses-db.html>

¹⁰ Global BioDiversity Information Facility

reasons for the event and findings after analysis. Such documents are valuable to design engineers to identify the common parts and reasons for common events.

“RED-TOP” REPORT Sensitive reports which require urgent attention, which act as flags to the lead engineers and managers for different parts.

TECHNICAL VARIANCE Reports generated by service engineers when small variations of the dimensions of components are requested. Technical variance requests are essentially requests to identify if a component is suitable for flight.

COST REDUCTION IDEAS Documents and presentations generated by manufacturing engineers which discuss ideas (for e.g. new manufacturing techniques, new machining processes etc.) which can be used to reduce costs of manufacturing. Each Cost Reduction initiative is classified into several other categories like status, opportunities or hazards, year of completion etc. Users are concerned with quickly and efficiently accessing historical decisions and estimations of such initiatives as well as understanding the status of the latest initiatives being considered. Cost reduction initiatives can originate from several different supply chain units such as compressors, combustors, fans, assembly and so on and initiatives can be widely varied from modifying designs to changing suppliers or materials. Cost reduction users currently have little means to query and visualise such constantly increasing data to enable them make quick summaries, apart from manipulating excel filters and tables. This is a highly time and resource consuming process, requiring a lot of effort from the end users to find information from an excel sheet. However, often users have to go through several hours of understanding the data they have retrieved in order to make use of it.

CONCESSIONS Concessions users are concerned with making decisions regarding whether to use engine parts that present features that have been manufactured with a slight deviation from the original drawing. These deviations can include anything, from defective blades to bolts that have been manufactured with the wrong dimensions. Their main job is to make decisions on whether a concession should be granted for the use of a part that presents manufacture defects to build an engine.

The data in general represented reports, documents, intranet links, shop visit reports and so on. The key identified concepts that can provide standard visualisations were identified as time (timelines, calendar views), geographical (geographical maps, area maps), categories (pie charts, bar charts, tables), performance indicators (line plots), engine components (topological views).

4.1.4.2 *Animal and Plant Life Sciences Domain*

An initial study involved studying the domain of animal and plant life sciences, more specifically Grass species distribution. One of the most important problems associated with grass descriptions and plant sciences in general is nomenclature. Over the period of few centuries grasses have been discovered in different areas of the world and have been named by researchers. However, there is a need to standardise grass names. This creates a situation where a grass species has been named in various ways by different scientists in different areas of the world. These synonymous grasses need to be identified

according to their characteristics. Grasses can be characterised by identifying different physical features — for example, colour of the stigmas or the length of their blades. A dataset of 5000 grass species have been studied where each grass is identified by 1090 unique characteristics. The distribution of such grasses across the world has also been studied. The datasets studied are as follows:

GRASS CHARACTERISTICS A set of 5000 unique grasses described by 1090 unique characteristics, available at GrassBase¹¹

GBIF DATASET Global Biodiversity Information Facility¹² data describing the coordinates of instances of grass species discovered across the globe.

The key concepts that can provide standard visualisations were identified as individual characters (data tables), geographical locations (geographical maps, area maps), categories of characters (pie charts, bar charts, tables), morphological characteristics (topological view of grass), time of discovery (timeline).

4.2 REQUIREMENTS

Several requirements for a solution to address the research questions were drawn. The requirements were identified and grouped into two main categories: Functional and Non Functional.

4.2.1 *Functional Requirements (FR)*

An interactive visualisation system provides an interface between a user and the data, by communicating the status of the data to the user using visual means. The user, on the other hand manipulates and interacts with visual objects and provides input to the system to communicate their intentions. Functional requirements translate the research goal to the primary expectations from the solution. Development of the solution needs to focus on carefully addressing each requirement and the solutions to the individual requirements constitute the fundamental building blocks of the solution. The following lists the requirements that the solution must address:

1. Solution must visualise data — The importance of visualisation, in this context is in quick and effective communication of the contents of large datasets. The central focus of the thesis is how to employ such techniques to explore large volumes of semantic data in a simple manner.
2. Solution must enable users to express their Information Need — Users must be able to communicate with the solution by expressing their Information Needs. Several approaches can be followed in answering the requirement such as using keyboard or mouse gestures, voice commands or sensors. It is important to ensure that the solution can capture user actions and translate them to interpret their intention.

¹¹ GrassBase, <http://www.kew.org/data/grasses-db.html>

¹² <http://www.gbif.org/>

3. Solution must facilitate querying — Since the aim of the research is to aid a user's access to the data of interest, the solution must provide ways to query data stores via visual and interactive mechanisms.
4. Solution must provide means to explore unknown data — While users may be well acquainted with their own datasets and domains, accessing datasets outside the usual ones should be a requirement in itself. More so, in large organisations sharing information across different application areas, verticals, business groups and so on. Most of the existing solutions rely on users, solution developers or data owners possessing prior knowledge regarding the dataset they are currently exploring. Hence, the solution must be easily and quickly portable and effective with very basic knowledge of a new dataset.
5. Solution must be scalable — With the growth of organisational data and rapidly increasing structured data, the need to effectively query and explore large scale data is growing. The solution must be able to cope with large datasets, in the order of billions of pieces of information. Existing visualisation systems are not highly scalable, while the scalable systems are mostly only text driven, offering minimal or no visual insight into the data.
6. Solution must provide visual summaries — Most of the existing visualisation systems present views of single instances or a small subset of the data. However, in order to scale up to larger datasets, the solution must categorise and organise the data into representative sub groups. Different techniques can be employed to achieve this, such as aggregating instances based on one or more features or pre-defined classification of data and so on.
7. Solution must facilitate simultaneous exploration of different facets of the data — The solution must exploit multi-dimensionality of data and provide multiple simultaneous views over different facets of the data. This will help users and analysts observe different dimensions of the underlying data at the same time, and will increase their overall understanding of the data from multiple perspectives.
8. Solution must provide support for advanced querying of data — Though target user groups can be casual users, the solution must take into consideration of the requirements of advanced and expert users. Such users are determined to possess expert knowledge in querying and can formulate highly complex queries in order to find information. The solution, while being simplistic in nature and approach, should also provide advanced settings for such users to customise its behaviour.
9. Solution must fit within organisational Knowledge Management frameworks — Different working groups, user communities and business units within one organisation may have their own knowledge practices as well as hierarchical structures. Hence, a flexible approach for the solution is a necessity in order to be applicable within an organisational Knowledge Management framework.

4.2.2 *Non Functional Requirements (NFR)*

While functional requirements dictate the fundamental features of the solution, non functional requirements determine the qualities and characteristics of the solution. The following lists the non functional requirements of the solution:

1. Solution must be domain and application independent — A generic approach to interact with Linked Data would be true to the essence of the Semantic Web as well as be extensible to other datasets, users and domains. Most of the existing solutions are highly domain or dataset dependent, only catering to one type of data. Generic solutions do exist, but lack in other areas (e.g. not scalable or a text template-based approach and so on)
2. Solution must be intuitive — An intuitive interface can help users comprehend, use and understand an unknown system, thereby increasing productivity and efficiency. New users must be able to use the system without much training.
3. Solution must be generic — Users with different range of expertise and domain knowledge must be able to use the system with minimal training. It is essential that the solution must not overwhelm users with too much information as well as technical jargon. Strict query formalisms provide means to structure data into standard formal representations. This, in addition to a self-descriptive nature of semantic data, can be exploited to provide intuitive and meaningful interfaces, understandable to casual users.
4. Solution must be aesthetically pleasing — While a lot of effort can be invested in building user interfaces and visualisations, making them aesthetically pleasing to users is a challenge in itself. Users are often willing to try searching for information multiple times if they find the interface pleasing and friendly.
5. Querying in a simple manner — The complexities of semantic querying must be hidden from plain view of the user. While the approach of building queries can be highly complex, the users must not be intimidated by and deterred from performing their tasks. Users with very basic or no knowledge should be able to use the solution easily, with little training.
6. Provide familiar representations — The solution must provide representations of semantic data as visual objects that are familiar to users. Following from the general requirement of an intuitive interface, a familiar visual object is intuitive and easy to use. This will ensure that users are not confused with unfamiliar objects and infer unintended meanings and associations (e.g. using familiar shapes such as circles to describe concepts as opposed to an irregular polygon)
7. Provide consistent representations — The solution must represent semantic data in a consistent manner. Objects defined to describe a particular type of data should be consistent throughout the interface, in addition to being familiar. This reduces complications and unnecessary confusion from a user.
8. Familiar and consistent interaction mechanisms — The solution must ensure users are familiar with the interaction mechanisms, based on their prior experience with

tools employing similar interaction paradigms. The interaction mechanisms must also be consistent throughout the solution (e.g. a particular interaction on a concept should invoke the same response as of another concept, unless deemed to be semantically different)

9. Access to provenance information — Trust and provenance are highly significant in an organisation, specifically when critical information is being shared across various user groups. The solution must be able to provide means for users to access provenance information when desired, in order to validate the authenticity and legitimacy of a piece of information when questioned.
10. Shortcuts — The solution must provide different shortcuts for repetitive or intensive tasks (e.g. key-combinations or gestures). Though not seen as extremely important, such features can provide users with more efficient techniques.

4.3 SOLUTION — SEMANTIC WEB

Requirements analysis for a generic, large scale and portable solution motivates several architectural decisions to employ specific information infrastructures. This section discusses the motivations for following a Semantic Web approach toward addressing the requirements. Two main requirements, NFR₁ and NFR₃ govern this decision — while NFR₁ calls for a solution that is domain and application independent, NFR₃ calls for a solution that is user and role independent. These requirements motivate the development of a flexible solution, disconnected from domain, users or application-specific design choices or Information Needs. Easy deployment in a large range of applications and technical settings also call for a self-descriptive and machine-processable information infrastructure that can formalise communication between services and software agents.

The inherent nature of semantic data, being highly interconnected, multi-dimensional and self-descriptive, provides an excellent framework as a foundation for the solution. Owing to a self-descriptive nature, semantic data can be easily aggregated from multiple sources in a standardised manner, thereby providing an ideal environment within an organisational framework (R₉). Highly graphical datasets can be easily imported into semantic stores that can be easily queried and reasoned upon. Developing rules, logic and reasoning also provide the ideal setting that can help the solutions to be deployed in various conditions with users. Semantic Web formalisms also provide a highly systematic way of querying graph-based data — this can be exploited by aligning user interactions and visualisations for generating dynamic queries. Hence the role of Semantic Web as a unifying platform, providing heterogeneous data from multiple sources, following strict standards and well defined schema provides an excellent research framework that can help address the requirements.

The Semantic Web framework provides additional features to identifying resources and entities with unique identifiers, therefore facilitating disambiguation tasks as well as uniquely representing distinct objects, entities, features and characteristics within a domain. The increasing availability of open data in a highly structured and well connected format also provided an excellent research and development platform for building solutions. Such datasets are available in several domains and different levels

of granularity. Using a Semantic Web information infrastructure with a wide range of datasets provide a realistic scenario of porting solutions from one domain or application area to others.

The solution to address the requirements and accomplish the broader research goal is proposed as a template-based faceted approach toward querying semantic datastores, aided by visual and highly interactive means. There are several alternatives to exploring/interacting with semantic data that have been proposed in the past, most of which lack one way or many in addressing all of the requirements of the solution. The following list summarises how other solutions lack in addressing the functional requirements:

- Natural Language Interfaces are means to answer specific questions and mostly present information in a text-based approach. Such an approach does not address requirements such as FR₁, FR₄, FR₆, FR₇, FR₈ due to the lack of exploratory visual mechanisms and advanced mechanisms for querying.
- Forms allow users to enter restricted values for specific fields to be finally built into a complete SPARQL query. Forms are mostly either pre-defined (therefore not easily portable) or user defined (mostly requiring technical knowledge of the dataset, domain and user requirements). Such approaches mostly use text and form-based views to communicate information to users, hence do not address requirements such as FR₁, FR₄, FR₆, FR₇.
- RDF Data Browsers provide template-based views of RDF data. While this provides a highly readable way to view data for individual instances, such browsers are mostly text based and typically lack visualisations and offer minimal insight to users that can help them understand the underlying data. Hence, these approaches do not address requirements such as FR₁, FR₂, FR₃, FR₅, FR₆, FR₇.
- Mashups present a view of a dataset using pre-defined visual techniques (such as maps, timelines etc). While intuitive, usable and efficient, mashups are restricted to pre-defined visualisations and datatypes. This makes mashups highly specific to datasets and domains and not easily portable to a completely different domain, hence such approaches do not satisfy requirements FR₄, FR₇, FR₈.
- Faceted Browsers provide users with means to progressively build queries based on multiple facets of data, to drill down to the instances of data of interest. Such systems are mostly pre-configured to present a set of facets and are thus difficult to be ported to different datasets or domains. Additionally, most of such are either text based or follow a tabular format of data presentation, thereby lacking visual presentation and advanced interactive mechanisms such as requirements FR₁, FR₆, FR₇, FR₈.
- RDF Graph Visualisation tools follow highly visual and interactive approaches toward visualising semantic data. However, most systems are aimed at visualising individual data instances or ontologies. While useful for specific purposes, such an approach lacks in providing insight for underlying datasets. Providing a graphical representation of a data instance is helpful, but can often be overburdening for a user. Such approaches lack in addressing requirements such as FR₅ and FR₆.

The proposed solution uses a combination of several approaches previously used (RDF graph visualisation, faceted browser, data browsers, forms and mashups), combining the strengths of different approaches to provide a high level of interactivity and visual insight. The proposed approach also explicitly studies aesthetic design principles and Visual Analytic techniques to be incorporated into a methodological approach toward developing semantic interfaces.

In the proposed solution, several steps of interpretation and translation provides a bridge between users and the data. User input is captured via interactive elements and translated into highly formal SPARQL query by making use of different layers of internal mappings between user actions and query templates. A similar process transforms structured results into data objects as a multi-step process by using different layers of internal mappings between data objects to visual representations. A thorough review of the literature covering different research areas such as VA, Information Visualisation, Interaction Design and Aesthetic Design identified best practices and principles governing the development of interactive visual interfaces. A major part of the solution is in the way it is implemented: the entire process is encapsulated within an iterative UCD approach, involving different user communities that provide continuous feedback and suggestions to help arrive to a more refined and improved solution.

4.3.0.1 *Studying Semantic Datasets in the Open Domain*

With the decision to adopt a Semantic Web infrastructure, an additional domain that was added at this point was the open domain, particularly with open datasets released by organisations for public consumption. The availability of massive volumes of semantic data as Linked Open Data (LOD) has provided an excellent research platform for the development of Semantic Web solutions. The Linked Data setting of providing information from different domains, datasets, specificity in a highly structured and interconnected way via query-able services serves as an ideal replication of a standard organisational platform. Additionally, an open platform can serve as an excellent test bed for users outside organisational hierarchies without security and privacy concerns. Standard Semantic Web principles ensures that Linked Data across all platforms are uniform, and porting a solution developed for Linked Open Data into an organisational framework is easy. Hence, despite being motivated from a Knowledge Management perspective and the solutions designed within the context of organisational Knowledge Management, this thesis exploits standard semantic datasets and Linked Open Data to validate and evaluate solutions. The additional need within organisations, however is a layer of security and authentication protocols which is out of scope for this thesis.

In addition to organisational and research datasets, an exploration into open semantic datasets was conducted. While evaluations within the industry and academic projects was conducted in order to understand how domain experts interacted with the systems, an explicit focus was made on open datasets. This was required so that truly independent feedback could be obtained, where external users could evaluate the systems, without the constraints and restrictions imposed within organisational frameworks such as privacy, data protection, authentication and security. Several well recognised datasets within the Semantic Web community were analysed and used as a part of evaluations

such as Mooney Natural Language Learning data¹³, Semantic Web Dog Food data¹⁴ and DBpedia¹⁵. The Mooney Natural Language Learning data consists of geographical data, providing details of cities, states, rivers, lakes etc. in the USA. The dataset contains 5689 triples, with 402 instances of cities, 51 instances of states, 50 instances of mountains. In total, descriptions of 742 distinct objects are provided in the dataset. Semantic Web Dog Food data provides details of publications within the Semantic Web research area, including authors, papers, institutions, research areas and so on. There are 230390 triples, describing 30673 distinct objects including 8840 persons, 3717 In-Proceedings papers, 886 talks, 434 events and so on. The largest dataset, DBpedia provides structured information extracted from Wikipedia, as a queryable service. The dataset (english) presently describes 4.0 million things (the entire dataset including 119 languages describes 24.9 million things), with 2.46 billion RDF triples, where 470 million triples belong to the English subset of the data¹⁶. In addition to the previous datasets, several other datasets have been examined and used to develop the solutions, such as MusicBrainz¹⁷, Semantic Bible¹⁸, Jamendo¹⁹, Movies Dataset²⁰.

Most of the open datasets contain information that most users would be highly conversant with, such as geography, movies or music. Hence, they provided an excellent opportunity to evaluate with most users as they are familiar with the terminologies, data as well as most of the formalisations. At the same time, some datasets like Semantic Web Dog Food provided academic publication data, which only a few users such as academics and researchers have familiarity with. The combination of generic and domain specific datasets provided an excellent opportunity to replicate organisational domain-specific and generic datasets, where user communities understand familiar datasets but unfamiliar with datasets from other domains.

4.4 THE SOLUTION ARCHITECTURE

As previously discussed, the goal of the research is to provide design principles for developing Semantic Web visual interfaces that can help users visualise, access, explore and query semantic data. Hence, the two key players are users and data, and how they can be brought together in a seamless and pleasant manner is the research focus. The solution is an extension of the primary notion that visualisations offer insight and can aid users in quickly gaining a good understanding of their data — hence a highly visual and interactive approach is employed. The two primary tasks of consuming semantic data are separated, and the design principles are drawn from the design, evaluation and implementation of the solution throughout the period of research:

1. Visualising semantic result sets; and
2. Querying semantic data sets.

¹³ <http://www.cs.utexas.edu/users/ml/nldata.html>

¹⁴ <http://data.semanticweb.org/>

¹⁵ <http://dbpedia.org/About>

¹⁶ <http://blog.dbpedia.org/category/dataset-releases/>

¹⁷ <http://linkedbrainz.c4dmpresents.org/content/rdf-dump>

¹⁸ <http://www.semanticbible.com/>

¹⁹ <http://dbtune.org/jamendo/>

²⁰ https://babbage.inf.unibz.it/trac/obdapublic/wiki/Example_MovieOntology

The Figure 31 describes how I address the requirements and achieve the goal of the thesis using several functional blocks throughout the process of a user expressing their Information Need to querying a datastore, and finally translating the results back into visual representations.

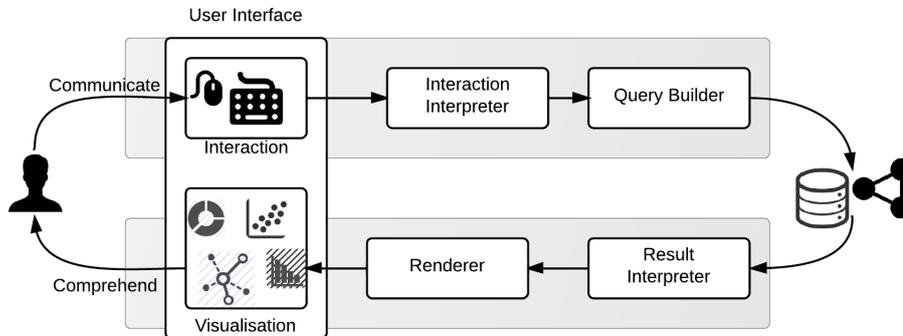


Figure 31: Solution Architecture

Exploring and querying semantic data is seen as a cyclic process, where a user is provided with an initial view of the data. The user, upon comprehending the content of the data interacts with the interface. The interactions are then interpreted (Interaction Interpreter) to understand what the intention of the user is. The user intent is then used to understand which kind of query it most relates to, by making use of pre-defined internal mappings. A mechanism that maps interactions to query templates is used to select the relevant query template. Interactions such as double click on a concept (in a force directed graph representation of RDF data) are mapped to queries such as “*show me all the instances of the concept*”. The queries built as a result of the Query Building stage are then validated and completed to well formed formal query syntax. The query is then passed to the data stores, to be executed. This completes the processes arising out of interactions with the user (Querying semantic data sets).

The results obtained from the semantic stores undergo the next set of processes (Visualising semantic result sets), resulting in visual rendition of result sets. The results are initially parsed to determine their structure and contents. The data objects are then converted into formal representations, based on information gathered during analysis of the results — e.g. a Java data class, or JavaScript Object Notation (JSON) object that represents a waypoint in a geographical map or a point in a scatter plot. The data objects are then translated into visual representations. The Rendering can be part of third party software (i.e. Google Maps, Highcharts) in addition to user-defined customisations on rendering (i.e. colour coding of Google Map waypoints).

Whilst all processes in the solution are essential, three components are the most important: User Interface, Interaction Interpretation and Result Interpretation. The other components such as Renderer, Query Builder, Parser etc. are standard processes for any typical solution. The novelty in this solution are however, in the three essential components. The components are further discussed as follows:

4.4.1 *User Interface*

The User Interface forms the user-facing side of the solution, and provides the medium of communication between users and data. Most important part of the solution is the User interface, and its role is expected to hide all the complexities of semantic data querying and in return present information in a seamless, pleasant and interactive manner. The requirements and design choices listed in the previous sections mostly relate to the user interface. In the context of the thesis, the most important elements of the user interface are visualisation elements and interactive objects. In the proposed solution, users mostly interact with the visualisations via interactive visual objects and basic input mechanisms. As mentioned previously, the solution distinguished into two separate tasks: visualisation and querying. Hence, there are two different approaches proposed for querying and visualisation.

My solution to visualising semantic datasets is to employ multiple simultaneous generic visualisations, where each visualisation represents a specific facet of the data (thereby addressing FR1, FR6, FR7). Visualisations and the organisation of the interface is highly flexible. Generic and familiar visualisations are employed, with the flexibility of adding customised domain-specific visualisation modules in the architecture. The ability to present information using generic visualisations help in exploring unknown data (FR4) in a generic manner. Interaction with the visualisations trigger queries and enable users to “drill-down” into data of interest (FR2, FR3).

The solution toward querying semantic data is highly interactive and guided by visualisations. Users are presented with basic visualisations (FR6) that communicate the content of the dataset being visualised. Each subsequent interaction generates new visualisations to reflect a further subset of the data, progressively helping the user explore the data. Making use of subsequent aggregate visualisations, users can get a significant understanding of large sets of data (FR5). Users can then explicitly query for data by interacting with the visualisations (FR3, FR8).

4.4.2 *Interaction Interpretation*

Approaches such as natural language search or form based employs text-based approaches to obtain user queries. These queries are typically entered into text fields within the interface, and are then read by the system when a search is triggered. Obtaining queries from mouse gestures, on the other hand is a more complex process as it involves understanding the context surrounding the action. Several scenarios need to be identified, that map mouse gestures (and key combinations) to user intent. Additionally, user intent can be at the logical level (to inspect details of a subset of data) or interface level (zoom). For example, a mouse gesture of scrolling the mouse wheel (user action) is interpreted as an intent to zoom in the interface (interface level), which identifies that the user intends to inspect the details of the area being zoomed in (logical level). These actions — intent maps are typically present in all interfaces, but are much more complex in highly interactive and visual systems.

Once the interaction of the user has been interpreted into a high level intention, it is converted into specific queries by the Query Builder. The conversion is based on selecting best query template from a set of candidate templates. More advanced techniques

such as machine learning (learning from interactions with other systems) can also be employed in order to translate high level intents into structured queries.

4.4.3 *Result Interpretation*

The solution interprets results and translates them in two phases — the first phase interprets the results into identifying the data objects. This process analyses the result headers to determine the result objects and data types, and then map them to real world entities. For example, latitude longitude readings indicate a geolocation, whereas time indicates temporal information. With the knowledge of what the data represents, the result objects are translated into respective data objects, understandable by the visualisation modules. Each of these objects is then rendered into a visual representation at a later stage.

4.4.3.1 *Considerations on Parallel Visualisations across domains*

Studying the three domains (open data, aerospace, and animal and plant life sciences) identified several concepts that can be generalised for any domain. Concepts such as time, geographic locations, categories, morphological or topological characteristics are concepts which can be identified among most domains. Visualising generic data can now be considered as a set of common concepts have been established. These concepts can now be visualised in a generic way. For example, a generic way to view geographical data would be using geographical or area maps; a generic way of visualising categories and aggregates would be using pie charts or bar charts.

Following studies of the domains, user communities, Knowledge Management processes and gathering an understanding of the shortcomings of existing approaches, several requirements of the solution were identified.

4.5 OUTLINE OF RESEARCH

Central to the work done in this thesis is the overarching theme of user engagement – the User Centered Design method is applied extensively, and is the first time it has been done so in such a scale within the Semantic Web community. Figure 32 presents an infographic of how the research was carried out, and the several iterations that took place during the development of the solutions. The key element of the figure is the illustration of a user centered design process, illustrating the different stages: Understand, Study, Design, Build and Evaluate. For illustrative purposes, the tasks Understand and Study (US) have been grouped into one element while Design and Build (BD) are indicated as one element. The figure presents how the different phases have proceeded, in a cyclic process with one phase feeding-into the next. The phases closer to the UCD illustration (US_{1,2}, BD₁, E₁) are earlier in the order of occurrence of the events. The further away the phases are positioned, the later they occurred. The right way to follow the illustration would be in a clock-wise spiral, radiating out gradually. As discussed previously, this research proposes two solutions: the first for visualising large semantic result sets and the second for exploring and querying large semantic data. The following section

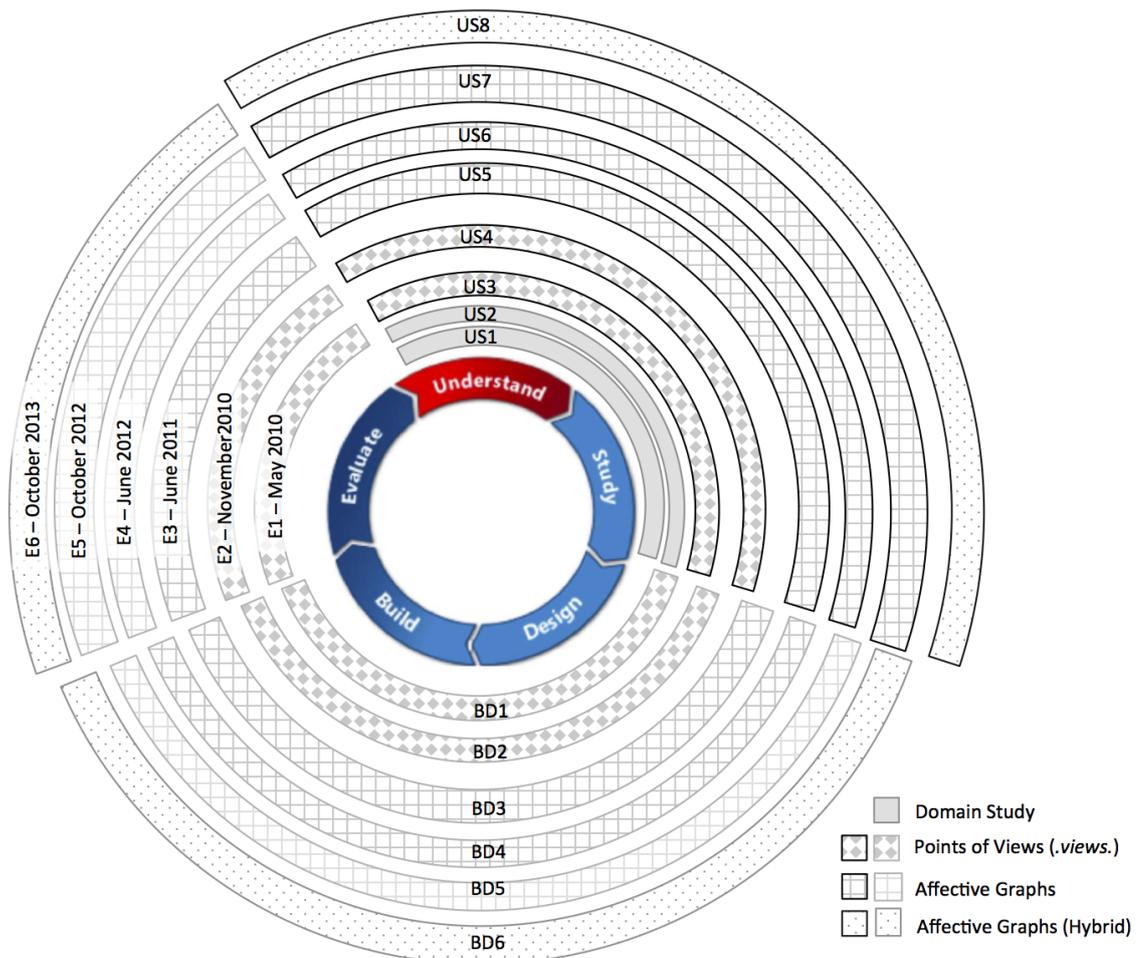


Figure 32: Development of solutions, using an iterative user centered design approach. The figure illustrates how the User Centered Design methodology (image at the center from [HRRS07]) was central to the development of the final solutions, by using several iterations of the design, development and evaluation stages. Two main solutions (.views. and Affective Graphs) were developed. While being an extension to Affective Graphs, Affective Graphs (hybrid) is illustrated differently for better readability. The figure is not to be scaled, but aims at providing a relative timeline of when different phases of the user centered design were conducted

discusses how users have been involved in developing the solution for visualising large semantic result sets.

4.5.1 Visualising semantic data: engaging with users

An initial pilot system was developed earlier to familiarise with the technical infrastructure of semantic data, as well as provide an initial estimate of how a potential solution could be developed [RM10]. With this understanding, the first phase of research (US1) focussed on studying the aerospace engineering domain, user needs and information sources. This required interviewing potential users, knowledge management experts and domain experts at Rolls-Royce plc. premises. Focus groups and group meetings

were organised to gather a greater understanding of the domain. Studying the domain was also conducted by analysis of several aerospace datasets shared by Rolls-Royce plc, as a part of the Samulet project. These processes were detailed in the Sections 4.1.3.1 and 4.1.4.1. This section also highlights how solution evolved as a result of engagement with users.

In addition to gathering an understanding of the domain and datasets, user interviews and focus groups also served to stimulate discussions on a potential solution, most desired and essential components and features of the solution. This was conducted by brainstorming several ideas, including how interactions could be designed, possible filtering and querying mechanisms and visualisations. These ideas were sketched into low fidelity mockups, which were then further discussed with the stakeholders at Rolls Royce. This provided some insight into how the useful features of the final system would be integrated into a coherent visualisation framework. Initial consultations with the stakeholders at Rolls Royce highlighted several issues and proposed modifications. However, as access to end-users were limited at this stage, the users provided more information on the type of data or architectural requirements rather than user-interface requirements. Some users, having prior experience with the visualisation system could relate to the proposed approach and provided some useful information on the widget-based visualisation. Some suggestions were also put forward, which discussed visualising trends of machine capabilities over time, as shown in Figure 33.

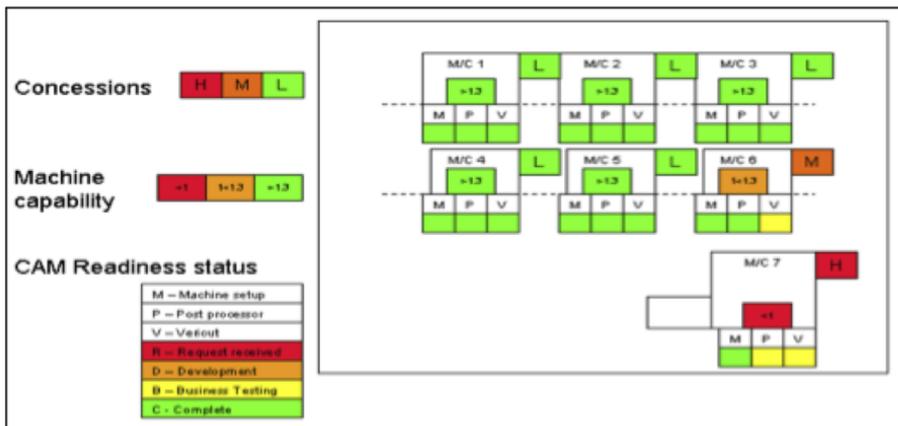


Figure 33: A factory floor map showing the machine capabilities and performances. Image provided by Rolls Royce as a part of the Samulet Project

The overall design of the solution was proposed as shown in the sketches presented in the Figure 34. Figure on the left shows an existing search system [BCC⁺08] with some modifications and the proposed visualisation system. The approach that was considered was that users would have two distinct environments: for querying data using forms, and for visualising data. The two systems would be developed in a way such that they can communicate with each other in a natural manner, so that the user can seamlessly navigate between the two.

The user would start from the ontology view in Section A, Figure 34 (left) within the existing search system. The ontology view would provide a summary of the main concepts within the ontology. Users, upon selecting concepts, would be able to build

a form of useful concepts. The form would be filled-in with the concept values, and then would be submitted to a semantic datastore. The results would then be shown in a tabular form, in the Section B. Clicking on individual documents of interest would show the document details in Section C. The button 'export' (D) provides a visual summary of the results received in the visualisation interface (Figure 34 (Right)).

Upon exporting the results, various facets of the result would be visualised in different visualisations in Section B. The different visualisations proposed were pie charts, bar charts, engine maps (for aerospace engineering), table display and geographical map. The Section A reflects which filters have been added (as imported from the previous query system). The filters are interactive, thereby simultaneously updating the views on Section B, employing dynamic querying mechanisms. The Section C would provide a set of visualisations that can be dropped on the Section B, triggering the creation of another visualisation widget. At any point, the users could return to the previous system and make new queries (at which point, any changes to the query in the visualisation interface would be transferred back to the search interface).

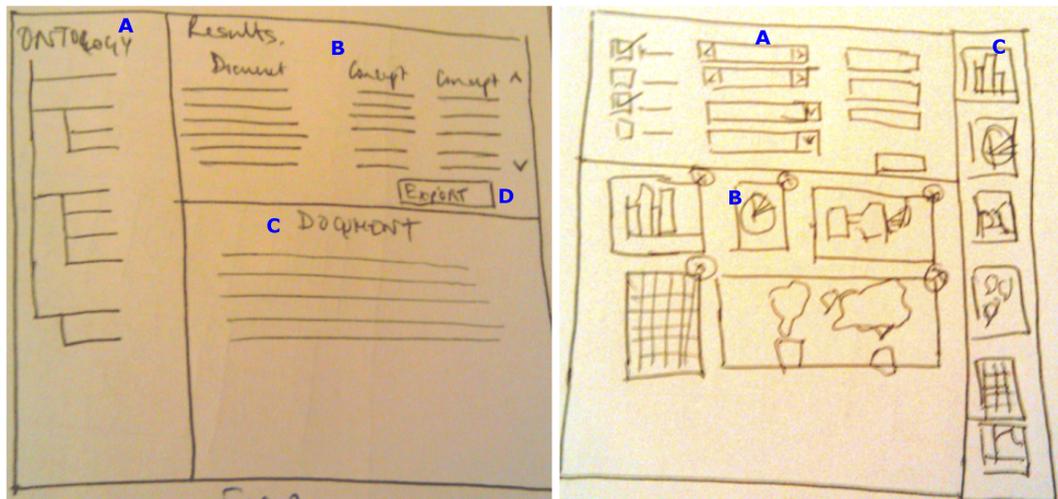


Figure 34: Sketches for perceived system used during UCD phase — notice the widget-based dashboard (Right) used to illustrate the dashboard

While highly appreciated in the initial design stage, it was quickly realised that such an approach would be cognitively demanding on the user's part, as constantly shifting focus between a highly textual and highly visual approach would be difficult for users. The next iteration which involved walk throughs of several scenarios with users identified that a better design would be to just focus on a single interface, and combine the form-based approach of the search system with a visual approach. Hence, the Section A of Figure 34 (right) would be modified to a form, where users can add ontology concepts. The queries would be triggered from Section A, and the results would be visualised in Section B. The Section C was also determined as unhelpful, since that can be replaced by a more space-saving feature such as a drop-down list, settings button or a pre-configuration step.

Following gathering an understanding of the aerospace domain, the research stemmed into two different direction: the first direction was targeted toward understanding an-

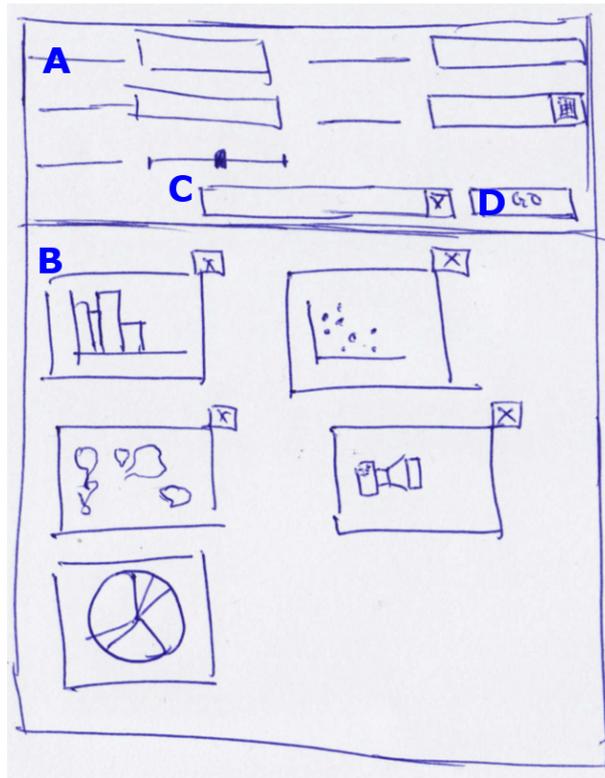


Figure 35: Visualisation of Semantic Result sets final interface design

other domain (plant life sciences), while the second was in designing a solution to address the needs of aerospace domain in a generic manner. Understanding of the plant life sciences domain (US2) also involved interviewing domain and technology experts, in addition to analysing datasets shared by the Grassportal project. These processes were discussed in more details in Sections 4.1.3.2 and 4.1.4.2. The design and development of the first iteration of the first solution coincided with the study of the plant life sciences domain and resulted in a solution for visualising semantic result sets, Points of Views, abbreviated as *.views*. (BD1).

The final designs for the two interfaces are shown in Figure 35- as will be observed later, most of the features in the final implementation have been true to these designs. Figure 35 shows the final proposed design for the visualisation interface. There are four primary sections to the design: a list of initial filters that is judged to be the most useful filters (A), a drop-down list of all available filters that can be selected by users (C), a 'search' button (D) and a visualisation container that provides all the visualisation widgets in a well-organised manner (B). The visualisation widgets can be rearranged or removed if the user wishes to. User's selection in the list of filters result in addition of new fields on the filter list (A), thereby enabling the user to set constraints on the filter fields. Upon setting a desired set of filters, the user can click on the 'search' button (here, shown as 'Go'). The click action triggers queries to the semantic data store and visualises the result set accordingly.

The first evaluation of *.views*. (E1), following implementation of the prototype provided helpful insights that were then combined with a greater understanding of the

domains. The first user evaluation was organised as several focus group sessions with students from the relevant domain is discussed in Section 5.6.1. This evaluation was conducted in May 2010. The students were initially presented with a set of guided tasks, and worked either individually or as a part of a group of 2-3 students. The participants provided qualitative information as questionnaire feedback, comments as well as a part of focus group sessions.

Following an analysis of the evaluation data (US3), screen recordings and user comments, the interface was redesigned (BD2). The changes on the interface was mainly driven by the points mentioned by the users during E1, as well as observing their interactions and activities from screen recordings. Several key features were included, which were suggested earlier and the new version of the system was developed over the next six months. The second version of *.views*. was evaluated as a part of the GrassPortal (E2) during November 2010. This evaluation is discussed in more details in Section 5.6.2. The evaluation involved computer scientists and biologists, grouped in pairs. The pair was asked to use the system in open tasks, that were relevant to their areas of research. The participants were then provided with questionnaires and invited for a short interview. Finally, a focus group discussed the system in more details, providing more information on how the system can be improved and possible new features and visualisations. Following the evaluations and analysis of the results (US4), a need for a greater emphasis in querying mechanisms drove the remainder of the research. The positive appeal of an interactive and visual mechanism of exploring datasets also steered the motivation for a visual approach toward interactively querying semantic datasets.

4.5.2 *Exploring and querying semantic data: engaging with users*

Development of the query mechanism was in four iterations, where a solution for interactively and visually querying and exploring large semantic datasets is proposed as Affective Graphs. This is indicated in the figure as three Affective Graphs cycles and one Affective Graphs (Hybrid) cycle. The initial design and development of Affective Graphs was a very basic prototype of the visual and interactive approach (BD3)– very few features were initially conceptualised, as it was important to understand how such an approach can be adopted. This was then evaluated within a knowledge management context, with domain experts from Rolls Royce plc (E3). Participants were provided with the early version of the system and after performing a few guided tasks, were asked to select any task of their choice to find information and explore the dataset. The evaluation is discussed in details in Section 6.9.1. Following the evaluation, an analysis of the data revealed several interesting key findings (US5), which validated the approach of a visual and interactive mechanism of querying data. More importantly, it identified several interaction and user interface issues. Discussions in interviews and a focus group involving all the participants also highlighted the areas of improvements and possible features that can be introduced. This fed into the next iteration of development (BD4), where a modified version of the interface was presented to expert and casual users, provided with a set of guided tasks (E4). The evaluation with the expert and casual users followed a comparative setting, comparing the approach with several other semantic systems employing different query mechanisms, and is presented in the Section 6.9.2.

and satellite relations depicting data properties (B). The diagram shows how users can create short queries and how visual objects can be translated into formal SPARQL syntax. The initial attempts were at understanding how users could visualise semantic data — in the true essence of a graph visualisation, the idea was to present concepts as circular nodes, with relations connecting them as curved links. The example shows how users could identify objects to be queried (Place, Person, Athlete, Soccer Player and Mountain has been highlighted). Initial discussions identified several issues with the approach — how would users know how much data is contained within objects or relations, how would users apply constraints, how would the query be displayed were examples of initial considerations that prompted a re-look at the design.

The figure on top right shows a storyboard session where following a re-design step, a user explores an aerospace dataset in a hypothetical scenario, following multiple trains of thoughts and exploring different hierarchical relations and contrasting the images at the same time. Several changes can be observed here — a concept is now represented as a pie-chart of the underlying content, grouped into subclasses. A context window, C is added which reflects the present status of the system — where the query under construction is displayed. A text display (“take-off”) shows how constraints can be displayed and added into the system. After a basic implementation of this design, several criticisms were raised- the interface presently lacked organisation, what was the final formal query being generated and how to engage advanced users were issues that required a design step.

The figure at the bottom left shows how the next iteration of re-design shaped the system — the top element shows the graphical space with concepts (A) being connected to each other with properties (B), context-menus. The idea of concepts displaying underlying statistics was appreciated by users in the previous design stages. The block on the right of the diagram (C) shows a contextual display element which is intended to display all the information contextually relevant to the concept in focus similar to the previous design. Section D was introduced, which shows formal query and configurations for advanced users to tweak the final query. Section E shows a tabular display that presents results returned by the data store. Basic text rendering displays the result set in a more readable manner (a bold text font would indicate a new instance is being displayed).

The sketch on the bottom- right shows how another type of querying mechanism can be included in the system — Section E shows a form based approach that can take user input and transform into view-based approach. This would enable users to enter broad queries, visualise them to fine-tune and finally construct a highly specific query. The panel on the left (F) was created as a result of a brainstorming session, where it was proposed to introduce other types of interactive mechanisms such as check boxes and sliders to fine-tune queries. This feature was not implemented eventually as it was decided that combining form based approach with a view based one would require a greater understanding and considerably more research than another iteration of a design cycle.

Figure 37 shows the final proposed design for the querying interface. There are five primary sections to the design: a query panel that provides search functionality (A), a visualisation panel (B) that presents concepts (C) and their properties (D), a context window (E), an advanced settings panel to be used by experts (F) and a results view (G).

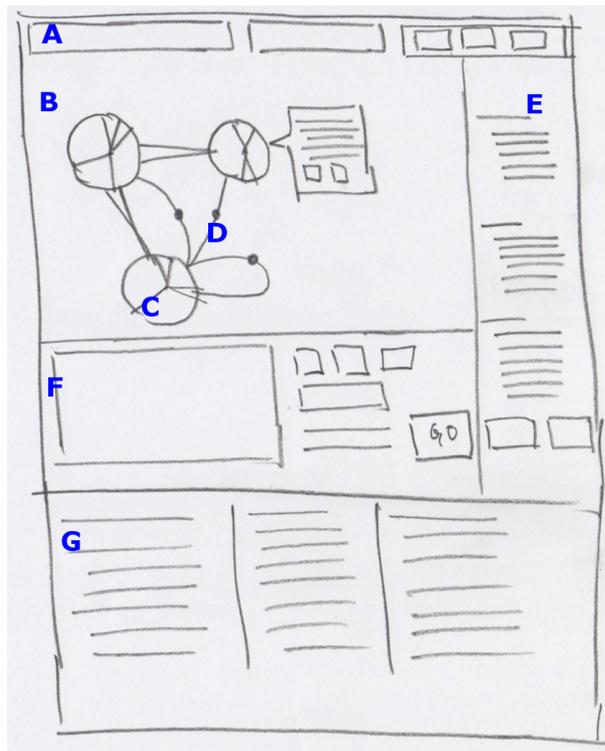


Figure 37: Querying Semantic Data final interface design

In addition, the visualisation panel provides other User Interface elements such as drop-down context menus and dialog boxes that are used to provide further information or gather user intent as shown in Figure 36.

4.6 SUMMARY

Throughout the period of the research, the continued involvement of user communities and stakeholders have been an extremely positive and encouraging experience. Suggestions in the form of minor interface changes to significant visualisation re-designs have been considered throughout, resulting in a stronger, and better designed system. A systematic study of the domain also highlighted present knowledge practices in the different domains and helped identify the shortfalls of such practices. This seeded the development of technology that was deemed highly useful and necessary for users. While the focus of the systems during these phases were highly domain-dependent, the outcome was independent of domains, user communities and applicability. Follow on evaluations with several other domains such as scientific publishing and geography established the validity of the approaches in porting to other datasets and domains.

The next two chapters discuss how these designs and approaches were implemented and developed into prototype solutions. Two solutions were developed, as a result of several iterations and the approaches were evaluated using users. The following chapter discusses the development of the first solution, *.views.* (points of view) and presents semantic result sets in a dashboard interface, providing multiple simultaneous visualisations of different facets of the results. The chapter discusses how *.views.* was developed

and the rationale behind the design. The chapter also discusses the technical implementation and the different user evaluation sessions of the system. A quantitative evaluation with a public endpoint highlighted the technical challenges that are posed to solution developers in a realistic setting.

Part III

DEVELOPMENT OF TECHNOLOGY

5.1 INTRODUCTION

Chapters 1 and 4 introduced the research questions, requirements and the subsequent formulation of the requirements for the solution. The solution was proposed as a two-fold approach, where the first would focus on visualising semantic result sets and the second on querying semantic data. This chapter presents the first approach (Section 4.4.1), which proposes semantic result sets to be visualised by employing simultaneous generic visualisations, each representing a specific facet of the result. In addition to generic visualisation modules, domain-specific ones can be plugged-in if the need arises. Interactive elements within each module can ensure users can ‘drill-down’ to the specific data instances of interest. Here, multiple simultaneous visualisation is realised in a dashboard metaphor, which presents information in organised visualisation widgets arranged in a grid, but can be re-arranged if a user wishes to. Two main factors are of interest while developing the approach:

1. Understanding how a dashboard approach could support the user in quickly exploring result sets and appreciating the multi-faceted nature of the underlying data; and
2. Understanding the technical implications and constraints in providing a generic visualisation service over Linked Data, either stored locally or remotely accessible via endpoints, and which technical constraints affect user interaction.

The key factors are therefore — *user needs* and *system requirements*. The user needs signify how users perceive a dashboard approach to exploring unknown data. This includes their experience interacting with a system that involves multiple coordinated visualisations using standard generic visualisations of different facets of the same dataset. The primary importance of the user needs is to validate and understand the hypothesis that generic visualisations can be effectively combined to build a domain-independent interface that facilitates exploration of unknown semantic data. System requirements, on the other hand signify how existing technology (publicly available Linked Data endpoints) can support such approaches. Matching user needs and system requirements can provide a way to realistically evaluate the technical constraints and possibilities with respect to what users expect and understand the compromises required from the perspective of a user interface.

This chapter discusses the implementation, design and evaluation of the dashboard-based framework, Points of View (*.views.*) and reports on the two factors. The next section discusses the UCD approach adopted in developing the solution, and presents sketches of the proposed solution. The results and details of the study are also disseminated in the following publications: [MVL⁺12, MPCar]

5.2 DESIGN RATIONALE

The rationale behind the design of the approach is discussed in three areas: the overall design layout, representation of results in a generic manner, and presentation of visualisations as widgets. A combination of the design decisions in the three areas have resulted in the final developed system. The design decisions were also influenced by the feedback from various users throughout the UCD process, as discussed in Chapter 4. The final implementation of the system, as shown in Figure 38 is true to the design presented in Figure 35 — the most significant elements of the implementation are marked-up to correspond with the respective elements of the design.

5.2.1 Design Layout

A tool that provides the user with a flexible way to look at the data from many perspectives needs to be customizable as the most effective type of visualisation highly depends on the data type and the task in hand [HBO10]. This design decision on effective customisation led to the adoption of a dashboard layout [Fewo6]. The Section 4.3 discusses how the other ways of querying semantic data fall short in addressing the requirements of the solution. While mashups and other multiple coordinated visualisation systems can provide different visualisations in the same interface, a dashboard-based approach is more structured and can provide spatial segregation of different types of views in a logical manner. Dashboards also provide simultaneous visual summaries of large sets of information in a limited amount of space (here, a single web page). Effective dashboards should be able to provide all the information in a meaningful, correct and intuitive way [Fewo6]. While widely used in BI Systems since the 80s, dashboard-like user interfaces are increasingly common in other domains only now. Popular websites like igoogle¹ and Yahoo!² use a design inspired by dashboard layout, by providing contextual widgets, each tuned to display a specific set of information. Website analytics systems such as Google Analytics³ also use a similar approach to visualising website traffic data. While the design and dashboard approach is inspired from business intelligence and personal information management systems, the development of such approaches on a Semantic Web setting ensures such designs are applicable across domains, user groups and application areas. Furthermore, the use of semantic technologies promise more reasoning and inferencing capabilities, which can be modelled in a more generic manner.

A similar approach is also employed by Content Management Systems and website builders such as Wordpress⁴, Drupal⁵, Liferay⁶ and so on. The design rationale embedded in Points of View (*views.*), is to create a dashboard for generic Linked Data by making visualisations available in customizable widgets as shown in Figure 38.

¹ igoogle interface, <http://www.google.com/ig>

² Yahoo, <http://uk.yahoo.com>

³ <http://www.google.com/analytics/>

⁴ <http://wordpress.com/>

⁵ <https://drupal.org/>

⁶ <http://www.liferay.com/>

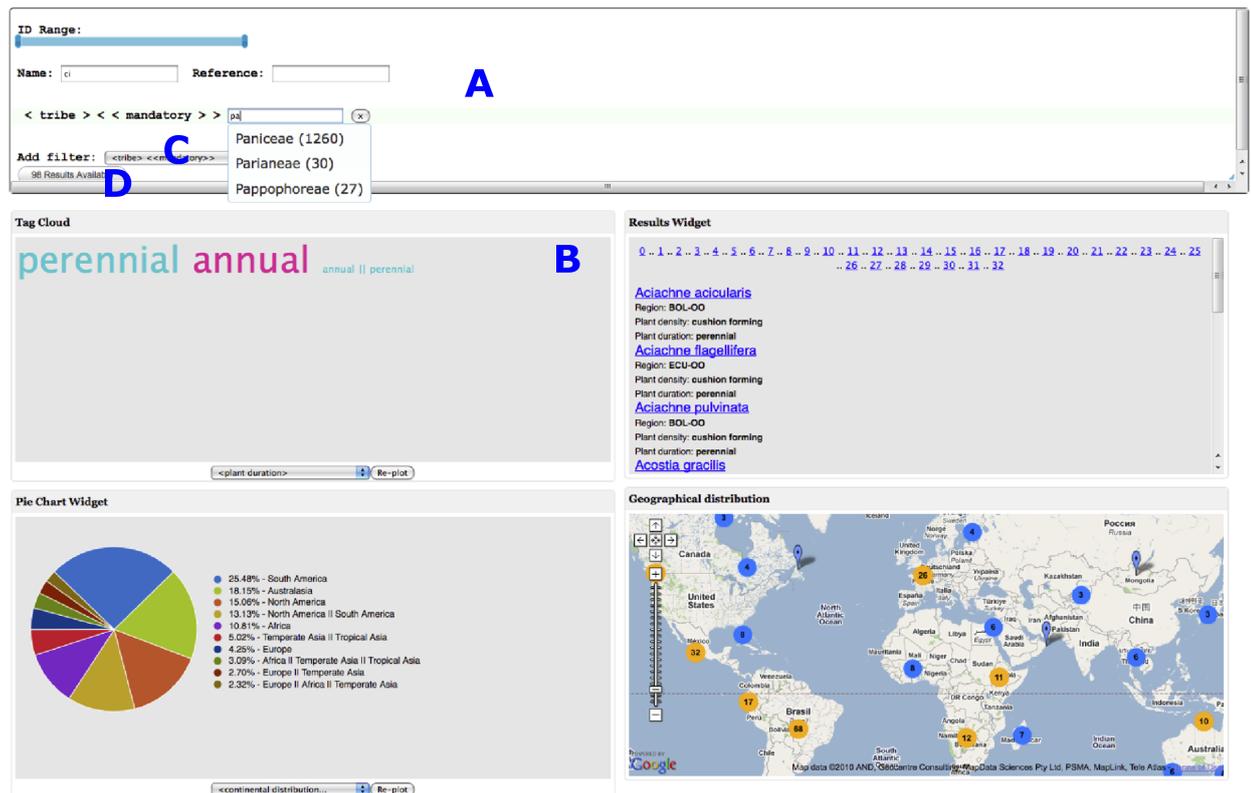


Figure 38: The Web-based interface for grass data with generic filters (top) and four different views on the retrieved data set, namely: tag cloud, result list, pie chart and geo-plot. (Data is courtesy of the GrassPortal (<http://www.grassportal.org/>) Project and Kew Gardens). As can be observed, the major elements of the interface are: filtering interface (A), filter selections (C), search button(D) and visualisation space with widgets (B)

5.2.2 Visual Representation of Semantic Results

Semantic result sets, returned as a result of querying endpoints are presented as a list of data instances that match the query, with the properties specified in the query and their values. As can be expected, visualisation of result sets are mostly applicable to visualising individual data instances, as opposed to concepts, classes and properties. Visualising graphical semantic data is therefore not the focus of the approach, unless an unforeseen (domain) specific need arises. Network visualisations can assist in such cases, but the expected use of such visualisation would be to present networks of physical entities within result sets such as people or products.

Though semantic results can be returned in various formats (XML, RDF, JSON, HTML, plain text etc.), the semantics behind each data instance is restricted to the specified query terms. Additional queries to the endpoints can provide more semantic information, requiring further processing. In this approach, multiple visualisations require the same result set to be presented in different techniques: such as category-based, facet-based and list-based. Category-based visualisations group the list of results into distinct categories and present them in tag clouds, pie or bar charts. Based on the topic of the

charts, further queries may need to be executed so the categories can be built accordingly. Specific visualisations (i.e. timelines, geographical maps) may require a similar process, where further queries are required to be executed in order to discover the distribution of the same result set based on a particular facet (such as temporal or geographical). List based visualisations are simplistic presentations of the result instances in a list or sortable table.

While several different possibilities of visualising semantic data exist, present solution only implements the most popular ones. Each query needs to be fine-tuned to retrieve further information related to the visual widget in question. A standard technique to present temporal data such as dates or years (inferred from the property definition and values) can be as timelines, calendar layouts and line plots. A prior search for the ranges (i.e. Range of a semantic property) of properties can identify possible properties that may be used for temporal data. Several ranges are well-used within the community as standard datatypes for such properties such as `xsd:dateTime`, `xsd:gYear`, `xsd:gMonth`, `xsd:gDay`, etc.⁷. Similarly, spatial data (inferred from property definition and values) can be presented in geographical maps using Geographical Information Systems (GIS) techniques. Example ranges for properties with spatial data may be `geo:Lat`, `geo:Long` etc.⁸.

Generic result sets can be further grouped into categories by triggering further queries to classify the results into different types. Several properties have been used as a standard within the Semantic Web community in multiple datasets such as `rdf:type`, `dc:terms:subject` etc. which can provide helpful ways of classifying the result sets. The categories of the data can be presented using category based visualisations such as tag-clouds or pie charts. Lists of results, on the other hand should be presented by exploiting `rdfs:label` (a standard property within the Semantic Web community) or another logical variant to present data instances as human-readable lists and possibly providing further information by following URIs when clicked.

5.2.3 Visualisation Widgets

As the sought solution is targeted to be consumed by different user communities within (and outside) organisations with their own sets of Information Needs, it is not predictable which visualisation users will find more useful given their task. For example, government data on schools performance could be better visualised as individual items on a map for parents trying to decide the best choice for their children, but would be more meaningful to public officials who want to compare school performance trends across the country if it was aggregated in tables. In a Knowledge Management scenario, where previous instances of engine service events are queried, a manager may wish to understand how the events are distributed geographically. However, a manufacturing unit may wish to view the result in an aggregated list, grouping the events into engine types. Therefore multiple views over the same data seem to be indispensable to support the understanding of the value of Linked Data and facilitate its use and consumption. Each visualisation is presented as a visual widget, that can be re-arranged within the

⁷ <http://www.w3.org/TR/xmlschema-2/#isoformats>

⁸ http://www.w3.org/2003/01/geo/wgs84_pos

dashboard layout, and can be minimised or closed. New visualisation widgets can be added with ease, and the facets of visualisations can be modified in a flexible manner.

Although not all visualisations are equal and a specific view can show or hinder interesting phenomena in the data [HBO10, Few06], it is important to explore the issue of visualising Linked Data as broadly as possible, leaving the introduction of visualisation constraints (i.e. which data type should be visualised, how and for which purpose) for a later stage. Identifying the best visualisation mechanism for a particular type of data, given a specific user community and task is itself a significant research challenge that falls outside the scope of this thesis.

In summary, the aim is to facilitate the visualisation of a generic Linked Data set in a way that is familiar and easy to understand and customise.

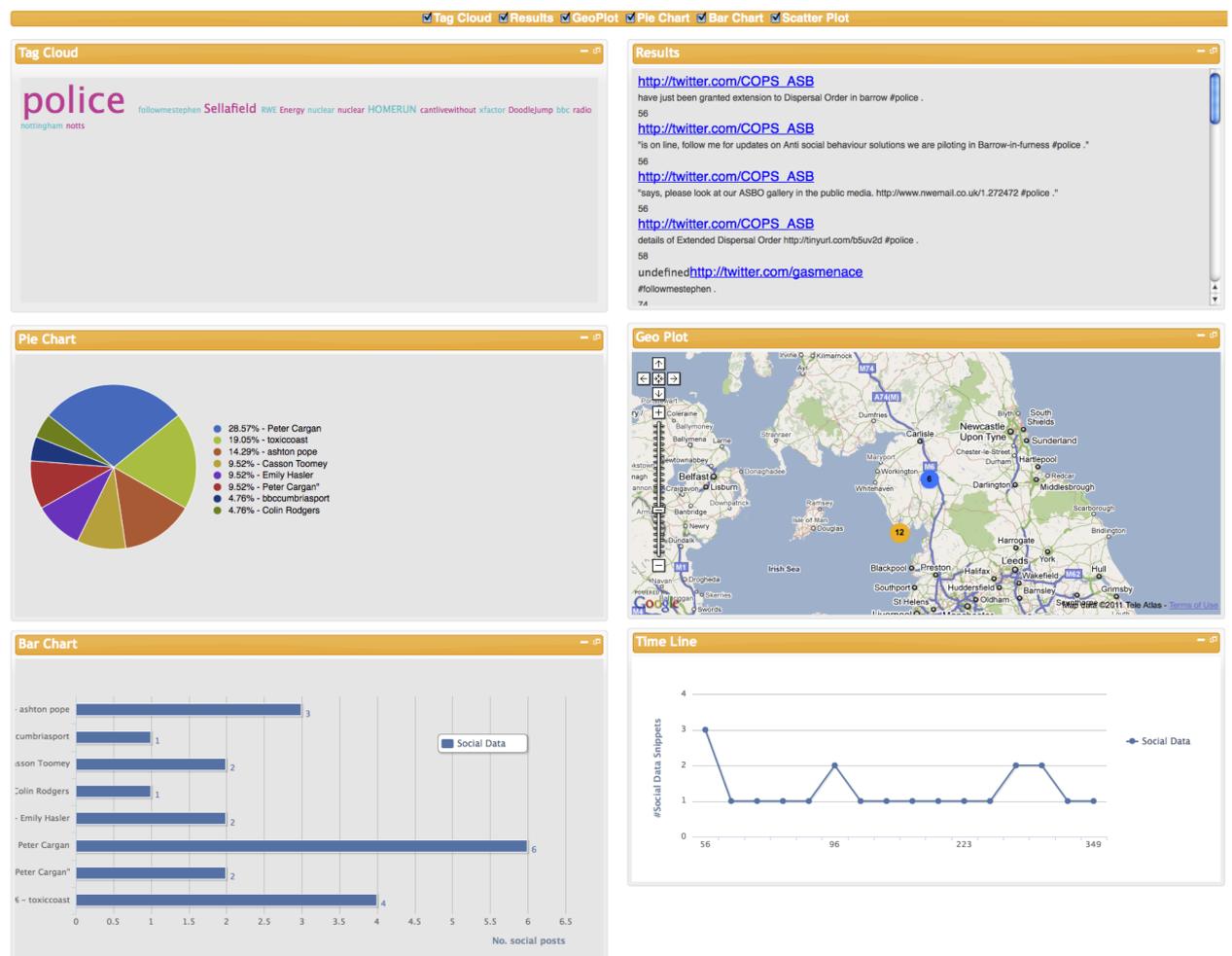


Figure 39: Social data after the flood in Cumbria, UK in 2007 visualised using *.views*. (Data harvested from Flickr and Twitter). The CSS style in this instance has been modified from Figure 38, though the basic interface remains the same.

The final design of the solution (presented earlier, in Figure 35) was generated after several design iterations with various user groups (as discussed in Chapter 4) and reflects the proposed dashboard. The multiple widgets target different facets of the same data, and hereby provide a comprehensive view of the underlying data to users.

5.3 ARCHITECTURE AND IMPLEMENTATION

The basic architecture of *.views* fits with the general solution architecture as shown in Figure 31, where the two approaches are presented. The two sections, querying semantic data (top) and visualising semantic data (bottom) used in conjunction provide the bridges between users and data. The logical architecture of *.views*. (Figure 40) is an extension of the visualisation section in the figure, expanding on the ‘Result Interpreter’.

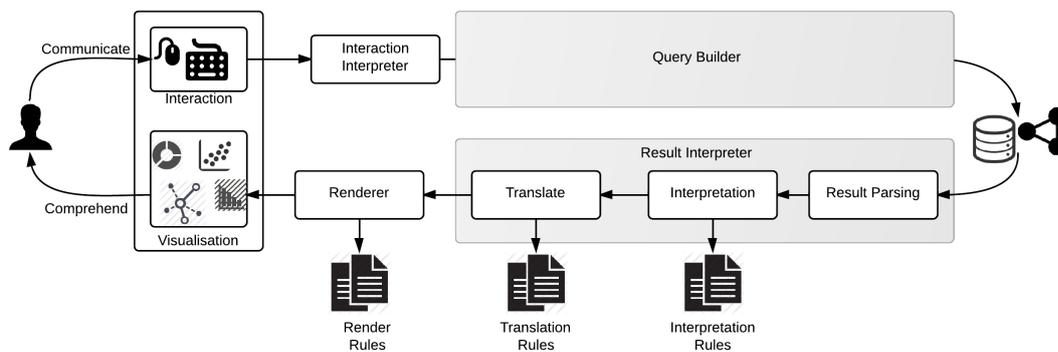


Figure 40: Solution Architecture of the Interface

The Query Builder consists of one significant block, but includes several stages of processing. As a starting point, the users interact with the systems and each physical interaction gets translated into user intents. These intents are encoded within the system and several rules guide how actions are interpreted as intents. The query most related to an individual user intent is then selected from a bank of query structures using very basic mapping rules. This is a simplistic process, since each widget has only a few possible interactions and query template. Hence, there are only a few types of query structures that need to be mapped with user intents. Being a rather simplistic process, the Query Builder has been presented as a single unit constituting a few very simple steps. The result from the Query Builder is a formal query, transferred to the Linked Data endpoint.

The more complex end of the solution, however is in the solution for the visualisation, more specifically ‘Result Interpreter’. This section of the solution consists of three major steps: Result Parsing, Interpretation and Translate. The Result returned from the Linked Data endpoint is parsed in two ways: result header is extracted to identify the result elements; contents are parsed to align the elements to data. Following the result parser, an interpretation stage maps result elements to physical entities (e.g. a latitude-longitude data element indicates spatial information, thereby referring to a physical geographical location). This is done using several interpretation rules that map result objects and data types into physical entities (such as time, location, component etc). The result contents are then translated into data objects, that can be used by the visualisation modules. The translation is based on rules, e.g. temporal information is translated into [JSON](#) objects that can be interpreted by the timeline visualisation widget and geographical locations are translated into marker objects that can be interpreted by the Google Maps [API](#). Each

of the objects are then rendered into visualisations by the respective visualisation widget (as per [API specifications](#)).

From a technical implementation standpoint, *.views*. is composed of two sub-systems: the front-end provides visualisations and user interactions, the back-end deals solely with querying the endpoints (Figure 41, backend on the left, front-end on the right).

5.3.1 Configuration Step

To start with a new data set, *.views*. has to go through a configuration step: a file contains the mapping between the widgets and the data feature, as well the corresponding endpoint to query. The following set of properties are defined in the configuration file, the content of which is represented in a tabular form:

visualisation Widget	Ontology property
Endpoint: "http://dbpedia.org/sparql"	
Instance Type: "http://dbpedia.org/ontology/Place"	
Geographical Map	"http://www.georss.org/georss/point"
Pie Chart	"http://dbpedia.org/property/city"
Bar Chart	"http://dbpedia.org/property/state"

.views. loads the configuration file during initialisation and builds [SPARQL](#) queries accordingly. The visualisation properties⁹ define how the respective widgets will be plotted, in a piechart, barchart, and on a map respectively — the plots would be built out of the distinct values of the properties (predicates) of the resource being queried for (instance type); 'Endpoint' defines which [RDF](#) store will be queried; and 'Instance Type' defines the type of instances that will be retrieved. Preparing a configuration file requires a certain understanding of a new dataset. Although Linked Data providers are likely to offer descriptions of their data models, a pre-configuration step can identify properties that are good candidates for certain visual widgets. For example, aggregate-based widgets like pie charts and bar charts are effective visualisations for faceted properties that have a short list of possible values occurring multiple times across the dataset whereas a tag cloud better suit a situation where the list of possible values are much larger. An example query to retrieve a list of properties that may fit in this criteria could be:

```
1. SELECT DISTINCT ?concept, count(distinct ?value)
2. AS ?count WHERE {
3.   ?s ?concept ?value.
4. } ORDER BY DESC (?count)
```

The result would be an ordered list of all properties along with the total number of unique values. Distribution of the distinct values across the data set can be retrieved by iterating through each property and querying for the distribution of its unique values. An example query (where the current property being investigated is 'city') would be:

```
1. SELECT ?val COUNT(DISTINCT ?obj) AS ?count WHERE {
2.   ?obj <http://dbpedia.org/ontology/city> ?val.
3. }ORDER BY DESC(?count)
```

⁹ These categories, essentially refer to properties (predicates) defined within the dataset. When visualised, the distinct values of the properties would be aggregated and plotted within the pie, bar and geographical plots

The resulting distribution can be analysed and a set of interesting properties identified. It is to be noted, however, that users can have different criteria for expressing interesting charts: a pie chart with too many sections, having equal area could be more interesting to some users than a pie chart with many sections with minimal area and a few sections with larger areas. Indeed, to determine which type of visualisation better fit which data for which task is still a matter of research, but results coming from the field of visual analytics [HBO10, Fewo06] are promising and allow us to forecast a time when this step of associating data features to visualisation widgets is done automatically or semi-automatically.

The pre-configuration step can either be a back-end process (the user enters a new dataset endpoint Universal Resource Locator (URL) and several PHP scripts automatically executes in the background, thereby selecting several possible properties) or a user-directed process (where the user can actively query the endpoint with a few pre-defined scripts on an interactive ‘setup’ environment to identify the respective properties). A fully automatic pre-configuration step can be time and resource intensive for its large number of calls to an endpoint and may result in unexpected time outs and performance issues, based on findings discussed in Section 5.6.3. In this implementation, a user-directed definition of the properties was used in the configuration file supported by queries similar to those above. However, once the system has loaded, the user has the flexibility to modify the faceting fields from each widget.

5.3.2 Translating interactions to SPARQL

To explain the *.views.* interface, lets consider the flow starting from the user interaction when querying the DBpedia SPARQL endpoint. The user is shown an HTML page with default widgets: on loading the page, a script sends a SPARQL query to the back end to retrieve all the properties in the dataset. An example SPARQL query would be as follows:

```
1. SELECT DISTINCT ?s
2. WHERE {
3.   ?s a <http://dbpedia.org/ontology/Place>.
4.   ?s <http://www.georss.org/georss/point> ?location.
5.   ?s ?property ?value.
6. } ORDER BY (?property)
```

The example query looks for all places in DBpedia that have a referenceable geolocation, but any other configurable constrain, e.g. a time frame, could be used too (by modifying the configuration file). Once the query is passed to the backend, a PHP script passes the query to the SPARQL endpoint using ARC¹⁰ classes. The response from the endpoint is then parsed by the backend and converted to JSON¹¹ format, which is then passed to the frontend. The frontend, upon receiving this response, parses the JSON object to populate its list of properties that will support the user in selecting the global filters.

This is captured by the drop-down select list (next to “Add filter:” in the Figure 44). The following shows an example query, where the user has selected ‘country’ and ‘type’ as filters and entered the value for country as ‘united_kingdom’. The user then starts typing ‘uni’ as the value for ‘type’.

¹⁰ ARC RDF system, <http://arc.semsol.org/>

¹¹ JavaScript Object Notation, <http://www.json.org/>

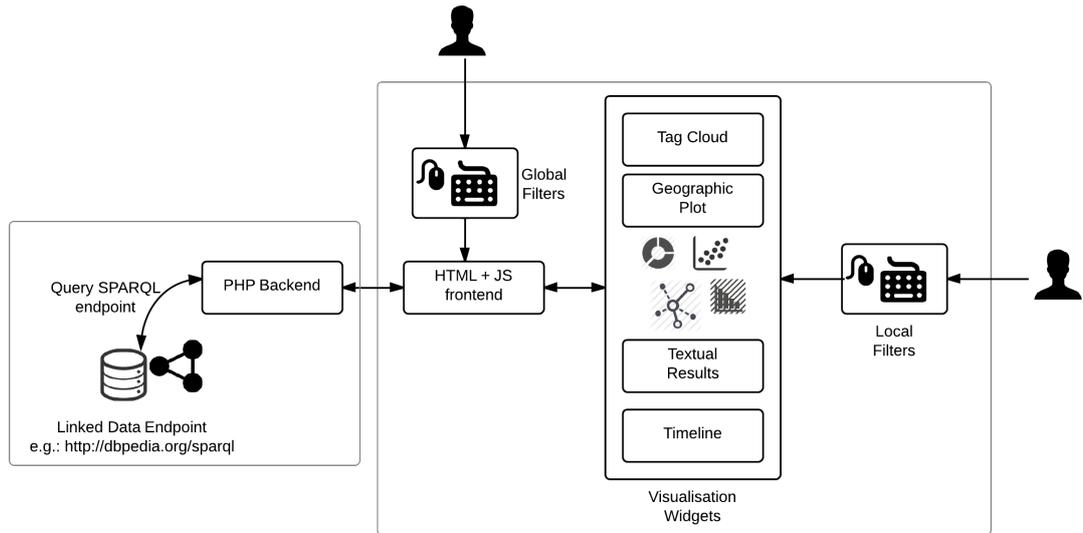


Figure 41: Architecture of the visualisation interface

```

1. SELECT DISTINCT ?type WHERE {
2.   ?s <http://dbpedia.org/property/country> ?country.
3.   ?s <http://dbpedia.org/ontology/type> ?type.
4.   FILTER (regex(?type,"uni","i") &&
5.           regex(?country,"united_kingdom","i")).
6.}

```

The query would return the types of instances that contain the character sequence ‘uni’ as its type and ‘united_kingdom’ as its country. The user can then select one of the suggestions and that would add the filter term as a global query. In the current implementation, automatic suggestions have been disabled due to back-end performance issues (as will be discussed in Section 5.6.3). Currently, the *SPARQL* queries do not contain any *FILTER* constraints. Instead, the user types the *URI* (or a matching literal) to have a valid global filter set up.

Once the user has followed the steps of selecting global filters and entering filter terms (as shown in Figure 43 and 44), *views* immediately displays the number of matches in the dataset, number returned after another query is sent to the endpoint. An example would be:

```

1. SELECT COUNT (DISTINCT ?s) AS ?count WHERE {
2.   ?s <http://dbpedia.org/property/country> <http://dbpedia.org/resource/United_Kingdom>.
3.   ?s <http://dbpedia.org/ontology/type> <http://dbpedia.org/resource/Public_university>
4. }

```

This query counts the unique instances of public universities that are located in United Kingdom¹². Clicking on the filtering interface (on ‘68 results available’ in Figure 43) triggers the simultaneous display of the widgets.

¹² All the references to the filter values are as URIs and not plain text to improve system performance, as discussed in Section 7.1.1

Unique queries (tuned by the individual widgets) are passed to the backend, which then responds with the results provided by the endpoint (which are further converted to [JSON](#)). In the previous example with public universities across United Kingdom, if the pie chart is focussed on visualising the results based on cities, the following query would be generated from the pie chart widget.

```
SELECT DISTINCT ?piecategory COUNT (?instance) AS ?count
WHERE {
  ?instance <http://dbpedia.org/property/country> <http://dbpedia.org/resource/United_Kingdom>.
  ?instance <http://dbpedia.org/ontology/type> <http://dbpedia.org/resource/Public_university>.
  ?instance <http://dbpedia.org/property/city> ?piecategory.
}ORDER BY DESC(?count)
```

The focus (or faceting field) of each widget is defined in the configuration file, but, for some widget, the user can alter it by selecting a new field from a drop-down list, e.g. Figure 38 shows that the mapping for both the tag cloud and pie chart can be changed using the drop down list in the bottom of the widget, whereas the map display is fixed. This flexibility ensures that the user has complete control over which facet of the data are explored at any time. The change of the faceting field (in this example, setting 'county' instead of previously defined 'city') from the drop-down list triggers a [SPARQL](#) query to the backend, essentially the same query, but with a different final triple pattern:

```
?instance <http://dbpedia.org/property/county> ?piecategory.
```

The back-end responds with a [JSON](#) object, which contains an ordered list of counties for the universities in United Kingdom. Each widget receives a similar JSON object, which is then parsed in its own way to provide the specialised visualisations¹³.

Once the individual widgets are loaded with their visualisations, the user can further interact with local filters and drill down to individual instances or group of homogeneous instances as in the case of maps and tag cloud. Local filters are generated either when selecting a different faceting field from a drop-down list (as discussed previously) or clicking on instances. The following example [SPARQL](#) query is generated when a section (in our example, the city London) of a pie chart is clicked:

```
SELECT ?instance ?property ?value
WHERE {
  ?instance <http://dbpedia.org/property/country> <http://dbpedia.org/resource/United_Kingdom>.
  ?instance <http://dbpedia.org/ontology/type> <http://dbpedia.org/resource/Public_university>.
  ?instance <http://dbpedia.org/property/city> <http://dbpedia.org/resource/London>.
  ?instance ?property ?value.
}group by ?instance
```

The back-end responds with a [JSON](#) object containing all the information regarding the selected instance(s). JavaScript modules then parse the object to create an [HTML](#) string that gets rendered on a popup dialog, as shown in Figure 42.

This allows a separation of the user from the raw data instances. The approach of providing aggregated views and combinations of data instances as visualisations enables

¹³ Several open-source JavaScript libraries have been used, to implement the different visualisation widgets, namely Highcharts charting library, <http://www.highcharts.com/> for timeline and bar-chart, Raphaël <http://dmitrybaranovskiy.github.io/raphael/> for the pie-chart and Engine Map and Google Maps <http://code.google.com/apis/maps/index.html> for the geo-visualisation.



Figure 42: Popup dialog providing details on individual instances — here, the details on the University of Sheffield.

users to have a high-level overview of the data. However, users can also drill-down to individual instances of data, which provides them direct access to the underlying data. The benefit of such a mechanism is that the users would not need to be semantic-web or database experts — their interactions would identify the subset of the data they are interested in.

5.3.3 Tools Used

views. was built using a client-server architecture. A web based interface was developed using [HTML](#) and Javascript. The visualisation tools being used are gRaphaël¹⁴, Highcharts¹⁵, Processingjs¹⁶, Google Maps¹⁷ and Raphaël¹⁸. [CSS](#)¹⁹ and jqueryUI²⁰ provide styling and a few interface elements, while jquery²¹ handles interaction with the server. The backend consists of PHP²² scripts, using ARC²³ to interact with Linked Data endpoints.

¹⁴ <http://g.raphaeljs.com/>

¹⁵ <http://www.highcharts.com/>

¹⁶ <http://processingjs.org/js>

¹⁷ <https://developers.google.com/maps/documentation/javascript/>

¹⁸ <http://raphaeljs.com/>

¹⁹ <http://www.w3schools.com/css/>

²⁰ <http://jqueryui.com/>

²¹ <http://jquery.com/>

²² <http://php.net>

²³ <https://github.com/semsol/arc2/wiki>

5.4 POINTS OF VIEW — A SCENARIO OF USE

Figure 38 and Figure 39 show the same interface applied on different datasets (though the CSS styles²⁴ applied in the examples are different). The different visualisations provide complementary information to the user as shown in Figure 39: the most discussed topic was the police (from the tag cloud top left), two areas were affected (from the geographical view, mid right), and when Twitter and Flickr registered higher activities (timeline, bottom right). The different visual widgets act on the same data set, each parsing it according to the type of visualisation they provide, e.g. geo-plotting extracted geo-information, timeline focussing on time values, etc.

It is important to note that some visualisations could be meaningless with certain data, e.g. if time is not provided a timeline would be empty. Therefore users can enable or disable widgets or re-arrange them (via drag-and-drop) depending on their needs, preferences and the data in hand. For example, numeric data would be better visualised as a table, a pie chart or a bar chart than as a list. The visualisation widgets developed so far include: a tag-cloud; a result list with links; a geographical plot; a timeline; a pie chart; a bar chart (all in Figure 39). Additionally, two aerospace domain-specific visualisations, Engine Map and Factory Map has been developed to cater to aerospace domain users [MVL⁺12]. Although this list is surely not exhaustive, the main interest was in providing a generic framework that could be expanded with other (generic as well as domain-specific) visual widgets than an exhaustive, but closed, tool. Indeed, *.views*. acts as a visualisation platform for Linked Data where new visualisation widgets can be plugged-in as and when they are developed, via API calls²⁵.

Essential for an effective use is to provide simple mechanisms to query the data set. As first experimented in [PMDCo9], *.views*. uses the concept of dynamic query [AWS92]: the interface provides graphical direct manipulation widgets, e.g. lists or slide-bars; while interacting, the user automatically queries the underlining database and the data in the filtered set is displayed. This approach supports Schneiderman’s well-known design paradigm “*overview first, zoom and filter, then details-on-demand*” [Shn96]: the full set is displayed first, the user uses the filters to select the subset of interest, then clicks on a view to dig into the details, e.g. at individual instances.

.views. provides two different types of filtering mechanisms: global and local. Global filters act on the whole data set and affect all of the visualisation widgets; local filters are attached to a single view (or widget), e.g. zooming in a geographical view to see details; clicking on a section of a pie chart to see the relevant subset of data. Global filters are automatically generated out of the data set, while local filters could be already embedded in some views (e.g. on maps) but some need implementation (e.g. pie chart selection). The local filters are essentially function callbacks from interactive elements that are aimed at capturing user input. Most standard charting tools provide such facilities, while the functionality is decided based on the type of visualisation as well as an intuitive expectation from the interaction (e.g. a user could expect a list of relevant results when clicking on a pie chart segment). Global filters are composed to retrieve the result set: items selected from a drop-down menu can be set to a specific value for data

²⁴ Cascading Style Sheets, <http://www.w3.org/Style/CSS/>

²⁵ Though existing framework requires function calls, a cleaner API access is planned in the near future

querying (Figure 43). Local filters support digging-into the retrieved set from different perspectives.



Figure 43: Global filtering for user-defined queries on the DBpedia dataset. On entering the values for filters, the number of results available is displayed in a button, clicking on which starts visualising the results.

Once the system has started up, the user queries the data by selecting the appropriate global filters that restrict the entire dataset to the subset of interest. The filters are selected from a drop-down list that is automatically generated during initialisation by querying the backend for all the query-able concepts. Upon selecting a filter from the list and clicking on the (+) button, the filter gets added in the filtering interface as shown in Figure 43 where the user had previously selected ‘country’ and ‘type’ as global filters. The interface displays the filter name (as retrieved from the dataset being explored) as a label, and provides a text box, which enables the user to type the respective filter values. As for the global filters, *views* automatically provides suggestions on the possible values: while the user is typing, SPARQL queries are sent to the backend to collect possible alternatives then displayed as suggestion list. Figure 44 shows an example taken from the grass data set: to a user typing “pa” the system suggests Paniceae, Parianeae and Pappophoreae as possible values for the filter named ‘< tribe > < < mandatory > >’, with the number of occurrences in the data in brackets²⁶. The suggestion list is automatically extracted from the data and therefore provides an insight into the underlying data fostering understanding for users unfamiliar with the set.

When the setting of global filters and their respective values is complete and the button ‘# Results Available’ is clicked, the backend is queried. The results are then simultaneously displayed in all of the visualisation widgets available on the interface. The user can then explore a single visualisation by making use of local filters — such as: clicking on portions of aggregate plots like the bars in a bar-chart, zooming into geographical maps etc.

In summary, the novelty of this approach is in providing a generic mechanism that automatically generates the user interface (both filters and visualisations) from a set of existing data and its structure without relying on any pre-determined domain-specific document templates. *views* tightly pairs the actual data set with graphical widgets giving the user the power to directly interact and explore the data. Contrary to other approaches that start from the domain description, *views* prevents querying empty data set thus saving users’ time and frustration.

²⁶ This feature is disabled while querying SPARQL endpoints, as discussed in Section 5.6.3

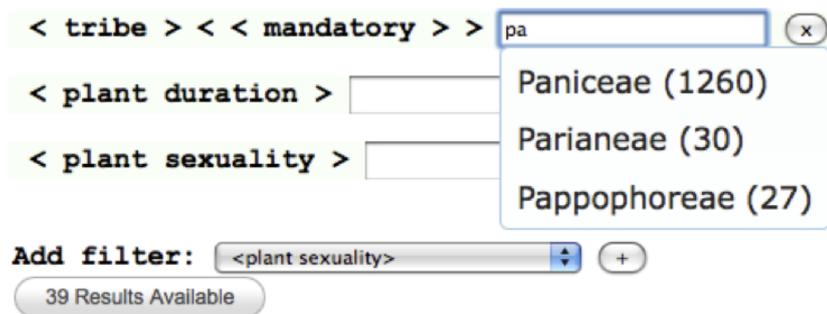


Figure 44: Automatic suggestions guide users for providing the right query. Typing a few characters (here, “pa”) starts suggesting possible filter values containing these characters, along with the total number of times they have occurred within the dataset.

5.5 CONFORMANCE TO REQUIREMENTS

The following table (Table 2) shows how the functional and non-functional requirements are addressed in the design for *.views.*. The table shows that apart from the FR8 (advanced querying mechanisms), most of the requirements are addressed. This is because *.views.* is developed as a part of a two-step solution, and the main contribution of its approach is in presenting semantic result sets.

5.6 EVALUATION

As discussed previously, it is essential for any visualisation of Linked Data to take into account user needs. Following a user-centred design approach [RSP02], a group of potential end-users has been involved in the formative evaluation of *.views.* A formative evaluation differs from a summative evaluation in several ways²⁷: it is done earlier in the design-development cycle, it aims at exploring the design space (e.g. alternative possibilities) and to have an overall sense of the user reaction to the system under design. As such it uses less formal techniques than a summative evaluation, but provide richer data to support understanding and, eventually, redesign.

Two sets of formative evaluations were carried out over one year, as described in the following sections.

5.6.1 Focus Group Sessions with guided tasks

The first version of *.views.* was presented to users for the first time during this evaluation, and hence, significant issues were identified in the process. The system was still under development, and at that stage, several key features were missing. However, in order to get a better understanding of the perception of users of the system and identify areas of improvements and key features required, it was necessary to conduct a user evaluation in a more informal setting so that qualitative feedback can be gathered as a part of discussions, .

²⁷ A summative evaluation occurs later in the development phase, when decisions have been already taken, and aims at ascertain the status of the system, e.g. by measuring its usability.

Requirement	Visualisation of Semantic Result Sets
FR1. Ability to visualise	Multiple widgets for visualising different facets of the data
FR2. Express Information Need	Users can select concepts and enter their queries in the respective fields, as well as use global and local filters to express Information Needs
FR3. Facilitate querying	A form-based approach that provides means to add/remove filters, which are translated into formal queries
FR4. Unknown data	Highly generic and basic visualisations can be used to visualise results and a generic mechanism integrated with Semantic Web principles to support any semantic data
FR5. Scalable	Multiple visualisations of aggregates can provide an overview of different facets of large result sets
FR6. Visual Summaries	Presenting aggregate data in visualisations
FR7. Simultaneous views	Visualising different facets of the same data in different visual widgets
FR8. Advanced querying mechanisms	
FR9. Knowledge Management	Applying basic visualisations to interpret different facets of any dataset ensures a flexible approach
NFR1. Domain independent	Visualisations and means for querying and translating user actions are highly generic (but can also support domain-specific needs)
NFR2. Intuitive	Visualisations, interactions and dashboard approach are inspired from standard designs and interaction mechanisms
NFR3. Generic	Familiar interactions and visualisations make caters to all types of users
NFR4. Aesthetically Pleasing	Not developed with explicit attention to aesthetics, but user feedback in different iterations positively impacted aesthetics
NFR5. Simplistic Query	Interaction-driven querying mechanism require users to use simple gestures on interactive form elements
NFR6. Familiar Representations	A combination of familiar generic visualisations are employed to simultaneously present data
NFR7. Consistent Representations	Use of familiar visual paradigms to represent standard data elements (e.g. a geolocation plotted as a waypoint)
NFR8. Interaction Mechanisms	Familiar interaction mechanisms expected from similar systems (e.g. clicking on charts, panning and zooming etc.)
NFR9. Provenance	Each instance can be drilled-down to their URIs, to be resolved to a further information page provided by data owners
NFR10. Shortcuts	Familiar shortcuts provided within visualisations (i.e. interaction events with visualisations to load data instances)

Table 2: Aligning final interface design with functional and non-functional requirements

5.6.1.1 *Method*

The first evaluation was a formative evaluation, focussing on identifying usability issues, future improvements and addition of key features. Employing a focus group technique with hands-on sessions, the evaluation aimed at providing evidence of use (via observations), participants' comments and suggestions that were to be used later to re-design the system.

5.6.1.2 *Procedure and Setup*

A dataset, consisting of grass phylogenetic data and their global distribution was loaded on *.views.* and K-Search. The dataset is described in Section 4.1.4.2. The evaluation was conducted as several focus group sessions over a period of a week during May 2010. Each focus group session involved 1 to 3 participants, amounting to the engagement of 8 participants. A focus group session lasted between 1.5 hours and 2 hours.

Following an initial briefing about the project and the goal of the evaluation, participants were provided a 15 minute demonstration of the data and the system. The participants were then given control of the system – in the case of multiple users, only one user was given control of the system, the remaining participants, sharing the screen. Groups were taken through the process of creating a query and exploring the result-set with a sample query. The evaluation then proceeded with a set of guided tasks, as shown in Appendix C. The tasks were then divided between each participant in the group, and the mouse control was swapped midway.

Once a group completed the guided tasks, each participant was provided with a questionnaire (Appendix C) with a few open ended questions to list the most positive, negative aspects of the system. A final group discussion followed once all the participants completed their questionnaire responses, that aimed to summarise their comments on the system.

5.6.1.3 *Participants*

8 students were recruited from the Animal and Plant Sciences department at the University of Sheffield. Students ranged from first year undergraduate students to masters graduated. All of the students were familiar with the domain, and one participant was highly conversant with the dataset. All of the participants were familiar with search techniques and regularly used the internet for looking for information. Most participants were not familiar with graphical visualisation systems, apart from basic Microsoft Office experience.

5.6.1.4 *Data Collection and Analysis*

Back-end search and interaction logs provided objective data. Doubts, comments and questions posed by participants during the evaluations were noted down with times. Additionally, audio and screen recordings of the entire evaluation were collected. The data collected was analysed from a higher level to identify possible interaction issues, future features as well as unique interaction patterns. User feedback in the form of questionnaire responses, answers to open ended questions (as shown in Appendix C), group discussion session and within-experiment discussions formed subjective data.

An analysis was conducted to understand how the system was judged by the users in various categories. The observation notes, feedback and comments from participants were then grouped into two types of comments – issue or request. Requests were the features users would have liked to see in the system following a redesign, whereas issues are the potential issues that users had encountered during conducting their tasks.

5.6.1.5 Results

The response was overall positive: the system was judged easy (72%), satisfying (72%), stimulating (78%), fast (90%) and reliable (72%). The comments from all the users were then collated and analysed, which led to several interesting ideas emerging: users appreciated the option of adding customised filters to select the data of their choice. Comments like *“You can use a large number of filters and so be as specific or vague as you want. All the information was displayed well and linked together well”* were encouraging and show that the intuition about user-selected querying was right in spite of the high number (1,090) of filter choices they had to deal with. This list, containing properties like flower colour, sepal length, height of plant etc. is gathered while initialising the interface by querying for all the properties of grass. However, the long list had drawbacks: comments like *“Hard to find the required filter in the list”* clearly show that the filtering interface needs some further thoughts.

Apart from an alphabetical order, participants suggested providing frequently used filters (*“Query box could contain a few of the more commonly used filter region, leaf size, synonyms”*) and to group them into categories and sub-categories (e.g. general characteristics (plant duration, sexuality, height etc.), region (Africa, Europe, Asia etc.), part of plant (anthers, spikelets, caryopsis etc.)).

Some participants suggested new visualisation features not foreseen during the design phase: *“Comparisons could be useful side by side visualisations? i.e. for one species distribution of annuals vs perennials. Could be very useful to show basic climate data on map”*. Other interesting suggestions include to overlay the geographical map with other imageries (e.g. street map, satellite and 3D imagery) or the use of a C-S-R triangle (Competitor, Stress tolerator, Ruderal) used by ecologists and botanists to show the performance of a plant respect to these categories [Gri74]. Data could then be plotted on the triangle that would become an alternative, topological view over the data.

Screen recordings of users interacting with the system and audio recordings were then analysed to understand how the system can be improved. Additionally, user interviews after the evaluation provided some guidance for possible improvements. A list of eighteen items were created that were indicative of the improvements required as well as the issues observed. The following list shows each item and an indication of the type of issue or request (FQ — Filter and Querying; P — Presentation of results, TE — Tasks and Experiment, V — Visualisation, FR — Feature Request) :

1. Different types of interactions with filters needed — if a slider is present, make text boxes available as well (FQ)
2. Provide a clearer indication of what filters have been applied (FQ)
3. Enable local queries to be transferred to build global queries — clicking on pie sections should create follow-up queries to help select further data (FQ)

4. Users experienced difficulty in adding filters — searching through long list of filters, manually browsing filters etc. (FQ)
5. Autocomplete is helpful, but causes confusion when users cause spelling mistakes (which is possible in a biological dataset) and autocomplete does not show results any more (FQ)
6. Provide a mechanism for clearing all filters (e.g. a clear button) (FQ)
7. Separate the (long) list of filters into smaller, more manageable list of filters distributed among different categories (FQ)
8. Users found it difficult to understand when and how a query was being built, and when to submit the query to be processed (FQ)
9. Some suggestions were provided for the best filters — e.g. Region, shape, density, duration, sexuality, year, synonyms and links (FQ)
10. When presenting all details about instances, provide the details in a categorised list or table (P)
11. Provide help and explanation of the data (e.g. abbreviations) (P)
12. Some users felt the guided tasks were not very relevant, but variations of them could be useful for users working with the particular dataset (TE)
13. Some users were not accustomed to using a similar operating system, and experienced difficulty in interacting with the system (TE)
14. Including more data would be helpful, in order to explain some of the data being presented. For e.g. presenting climate, soil and environmental data would explain why certain types of grasses are found in certain places (TE)
15. Some visualisations were not used at all (e.g. as a user mentioned, “*We didnt use the Bar-charts*” some charts are more helpful than others, and a personal preference can act at this stage) (V)
16. Some new types of visualisation such as scatterplots and venn diagrams would also be an interesting addition (V)
17. Provide means to compare and contrast data (FR)
18. It would be helpful to create drop-down lists to re-plot the charts on the basis of a different category (FR)

5.6.1.6 *Discussions, Key Findings and Limitations*

The eighteen items were grouped into five categories, each category represent a part of the interface or the evaluation. FQ (Filter and Querying) indicates all the observations, issues and suggestions related to the filtering interface and building user queries. As can be observed, this category contained the highest number of items (9). Users provided some suggestions about the presentation of the data (P), such as presenting instance-related data on tables or searchable lists and display helpful information on items that

can cause confusion among users. Some observation and suggestions involved the user tasks, dataset used for the experiment and the system used for the evaluation. Few comments regarding new visualisations and the usage of some of the visualisations were also observed. Users also suggested a two additional features that could improve the system — provide some ways to compare two instances, as well as compare two visualisations. They also suggested a drop-down list to re-plot charts like pie chart and bar charts on a different category.

The interest generated among users was significant, and post-experiment discussions noted that such interfaces are beneficial to end users and using the right dataset would be very helpful for users to conduct daily tasks. The users also noted without the system, it would take a significant amount of effort in order to perform the same tasks, as it involves navigating and searching through several sets of data and systems in order to assimilate all the information. Indeed, previous data processing and gathering stages are to be credited for constructing the final dataset and *.views.* was only responsible for presenting the information. However, the users appreciated the approach taken to systematically present data in multiple widgets simultaneously.

Providing a strong foundation that establishes the positive feeling of participants, the evaluation was a positive start to completing the first iteration of the user centered design approach. However, this evaluation had a major limitation – the analysis of the data was from a qualitative perspective, where user opinions and feedback were given a higher priority. At this stage of development, assessing what users feel about the system was an appropriate analysis, as this provided with interesting insights and directions for improving the system.

5.6.2 *Focus Group Sessions with open tasks*

Following the evaluation with students, *.views.* was then modified to include new features, bug fixes and cosmetic changes. Features that were added were enabling users to re-plot visualisation widgets on the basis of the variables that they select (a feature identified by users as essential), mechanisms to improve transferring of queries and result sets and improved backend to better handle different types of queries as well as larger results. Cosmetic changes include modifying the look-and-feel of the user interface to provide improved readability, adding domain-specific labels to help domain experts understand the interface better, removing certain pre-defined filters (that were pointed out as unhelpful by students) and improved interaction mechanisms. Six months following the evaluation, another evaluation was conducted with computer scientists and biologists with open tasks.

5.6.2.1 *Method*

A usability test was conducted in pairs of participants (a computer scientist and a biologist) to provoke a natural discussion between the subjects. The experiment was conducted in a comparative setting, where another system, K-Search [BCC⁺08] was also evaluated similarly²⁸. The pairs of participants, where each user was from a different background were encouraged to think-aloud while conducting the evaluation. User

²⁸ The K-Search system was developed further by K-Now, and a modified version was used for the evaluation

feedback was also collected as questionnaires and a focus group session following the completion of the evaluations. Participants were provided with open tasks and were encouraged to try the system out with queries they can formulate from their experience with the domain and similar datasets.

5.6.2.2 Procedure and Setup

A dataset, consisting of grass phylogenetic data and their global distribution was loaded on *.views.* and K-Search. The dataset is described in Section 4.1.4.2. The evaluation was conducted as a part of a full-day activity in the GrassPortal project on November, 2010. Following an initial group meeting, a description of the project and the interface was provided to the attendees. A group introduction was initially provided to the participants. A brief video demonstrating the K-Search and *.views.* interface and their key elements were projected, and a narration of how a dataset can be explored was provided alongside.

The study was conducted in three sessions of two experts working in pairs — a computer scientist and a biologist. This pairing was instrumental to understand how each expert looks at and interprets the visualisations and to foster discussion among the experts. No prescriptive task was given: participants were invited to try out tasks and queries that they would perform in their daily activities. User interactions and conversations were logged and recorded and a user satisfaction questionnaire was collected. Finally, all 6 participants discussed their comments and suggestions as a part of a focus group.

Recognising expensive expertise time to be a significant constraint, three pairs of participants were organised in a conference room, with parallel sessions running at the same time. Each session involved evaluating one system at a time. The pairs of participants were again introduced to each system, following the initial group introduction. An example was shown to the users that exemplified how the system could be used. Users were provided with a simple task to familiarise themselves with the system. Following this, one of the subject in each group was given control of the system to perform tasks they felt were appropriate to their daily activities. The pair shared the screen and control was swapped mid-way, providing the other participant with the chance to drive the exploration. Subjects were encouraged to discuss their tasks and their experience of the system as they navigate through it. Following completion of exploration of the system by all the participants, they were requested to fill a questionnaire specific to the system.

Each session lasted 30–40 minutes, and the mouse control was swapped mid-way. Following evaluation of one system, the other system was evaluated similarly. Upon completion of the evaluation of the second system, participants were provided with the questionnaire for the second system. After the responses for the questionnaires were provided, a final meeting was organised where the participants were asked to share their experience interacting with the two systems. Participants were also encouraged to share their comments on the different query and result presentation approaches that are employed by the two different systems.

5.6.2.3 Participants

Six participants were invited to participate in the evaluation: three computer scientists and three biologists. All participants were members of the Grassportal Project, and were conversant with search systems and were familiar with semantic technologies. Expertise of the users ranged from being familiar with the domain concepts and datasets (for computer scientists) to highly experts (for biologists).

5.6.2.4 Data Collection and Analysis

Back-end search and interaction logs provided quantitative data. During the evaluation, users' doubts and comments, as well as observations were noted down with the respective times. Additionally, audio and screen recordings of the entire evaluation session were also collected. While timing information and logs were available for qualitative analysis, the data collected was analysed from a higher level to identify possible interaction issues. This was mainly due to the lower number of users, since a full quantitative analysis would not be statistically significant. Instead, manual inspection of the recordings was conducted to understand if users appeared to need any new features (e.g. attempts to drag a pie chart section on the filtering interface to add the section as a value to the respective global filter) and identify issues with the user interaction.

User feedback in the form of questionnaire responses (B), focus group session and discussions during the evaluation session formed subjective data. An analysis was conducted to understand how the system scored in different criteria as judged by the users. Positive and negative comments regarding different aspects of the system were also identified to understand how the system performed.

5.6.2.5 Results

The observations show a marked improvement from the previous focus group with students. *views* was judged easy to use (83%), reliable (83%), fast (83%), stimulating (87%) and satisfying (87%). Figure 45 shows the user-satisfaction questionnaire responses of domain experts and computer scientists (Right) as compared to the responses of students.

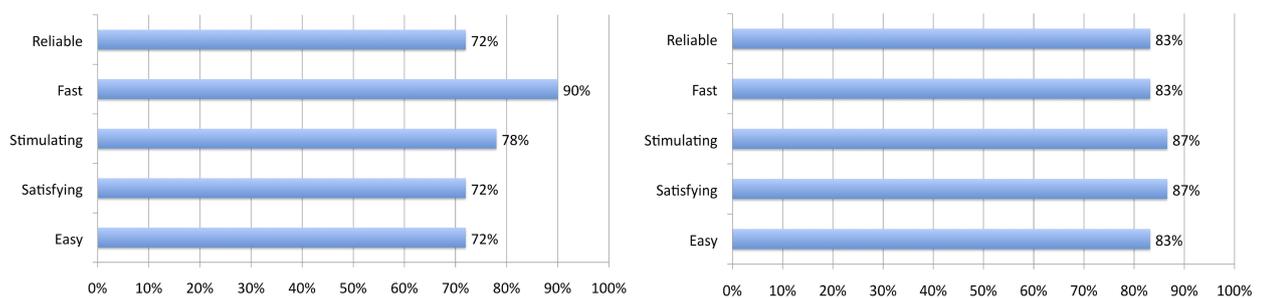


Figure 45: User-satisfaction questionnaire responses for students (Left) and domain and computer science experts (Right) show the overall improvement of the modified system.

Apart from the general consensus indicating a marked improvement in the user experience of the software, it is important to assess how individual experts with different expertise appreciated the system. The users rated the system on several criteria (ten) on

a 5-point Likert scale, which were then analysed. In the table 3, the odd numbered users (Users 1,3,5) were computer science experts, whereas the even numbered users (Users 2,4,6) were biologists. It was observed that in general, computer science experts found difficulty with the filters and interpreting some of the visualisations.

		Overall Reaction			Learning		Browsing Results			System	
		Ease of use	Satisfaction	Stimulating	Difficulty	Exploration by trial and error	Filter Manipulation	Browse results	Interpretation of Results	Speed	Reliability
Computer Science Experts	User 1	4	4	4	4	4	3	4	4	4	4
	User 3	3	4	4	4	4	3	5	3	4	4
	User 5	4	4	5	4	4	4	4	3	5	5
Biologists	User 2	4	4	3	4	4	4	4	3	3	4
	User 4	5	5	5	5	4	4	4	4	4	5
	User 6	5	5	5	5	5	5	5	5	5	5
	Avg.	4.16	4.33	4.33	4.33	4.16	3.83	4.33	3.66	4.16	4.5

Table 3: User-satisfaction responses for domain experts on a 5-point Likert scale.

Biologists found the system easier to learn compared to computer science experts, which could explain their higher level of satisfaction with the filters and the system in general. This is likely to be due to the partial familiarity with the data as some of the features used are common across the discipline.

In the follow-up group discussions, users commented positively on the intuitiveness and the general look-and-feel of the system. Users also appreciated the way in which they can “quickly see how data is distributed” and “got straight to where I needed to be”. Though users seemed to have difficulty in querying the interface, some appreciated the ability to “click on the menu to see all the possibilities” instead of a taxonomic view, while others disliked the drop-down list approach. Comments like “too much time lost scrolling through all morphological characteristics” and “Character list should be hierarchical so that it is easier to navigate” indicate that there is some re-thinking required regarding the filtering interface. Few users mentioned that they would like to see more data, for example, “Lack of specimen date information (Collection dates)”. “The current version only lists Accepted Names” and “The taxonomic data was not clear in that will the final system include both Accepted Names and Synonyms” indicate the users would like to perform disambiguation tasks like relate several species to each other.

Some users found the large number of available widgets was not always useful (“too many widgets at first, which get you a bit lost”) and showed explicit preferences (“tag clouds seem to be less useful than other features” and “geographical map not useful”) that could be incorporated in user profiles.

5.6.2.6 *Discussions, Key Findings and Limitations*

The self-selected tasks showed biologists could relate the tool to their daily work, however they expressed the need for more datasets to be visualised. They appreciated the ability to visualise a particular facet of the data in an aggregate visualisation, and then swap the view to a completely different facet. This was a feature that was added in the improved version of the system, an outcome of the first focus group session with students. Participants saw such visualisation approaches as a step forward from traditional search engines, as they could be empowered to find patterns or distributions in very little time.

The experts could associate the system to relate with their daily tasks and in general, they were excited about having a new way of exploring datasets to address their information needs. Users appreciated the benefit of having a visual approach toward exploring data, as opposed to a form and text driven approach adopted by typical search systems such as K-Search. At the same time, interaction issues identified in the evaluations posed challenges in filtering large datasets to reach the data of interest. All the users agreed that *.views.* was a step forward from traditional search engines and provides means for users to easily find patterns or distributions within the data.

This evaluation had a few factors that could be seen as limitations – low number of users, subjective evaluation and open tasks. On the one hand, these factors make the evaluation differ from evaluations typically observed in the semantic web community (where the focus is mostly on evaluating recall, precision etc.). On the other hand, these factors formed the basis of a highly focussed evaluation aimed at understanding the perception of technical as well as domain experts from different perspectives. At the time of the evaluation, a highly focussed approach involving lower number of expert users who could highlight important features and potential issues seemed appropriate for this stage of development. The introduction of open tasks also helped in understanding how well users could relate to the system with their daily activities and interests – additionally, it also helped users try different features and identify features that may be of use.

Overall the two formative evaluations were very positive. Comments like *“There is a huge amount of information available and after a while playing with the system it is rather intuitive”* and *“Clear layout, easy to understand and use”* indicate that the dashboard approach holds much potential. A rather interesting comment from an expert *“Clear colour presentation, gives pretty pictures very fast!”* indicates the proximity of this approach to the ideal goal of an interface developer — efficiently provide aesthetically pleasing visualisations to communicate essential information to the user. Often, users are interested in gathering a high level understanding of large datasets, rather than looking at individual data instances. A few comments such as *“It was on occasion rather ‘hard to use’ (though this may have just been inexperience)”* show that there is probably room for improvement in the intuitiveness of some parts, like the filter selection. However it should be noted that some of the difficulties could come from the data itself, for example the long list of filters (1,090!) the user has to go through to select the interesting item, is an essential part of the data structure. This evaluation clearly showed how this kind of details have to be considered beforehand if a generic tool to visualise Linked Data has to be provided.

5.6.3 Understanding public endpoints and System Requirements

As Linked Data become available from different sources, a visualisation tool should be able to take different sets and visualise them without too much tuning, ideally without any tuning. Porting a system onto another setting might not be straightforward and the technical implications of using a different infrastructure have to be understood in full as certain aspects of the interaction, e.g. system reaction time, must be kept between accepted thresholds to assure users acceptability. For example, a system is perceived as interactive if the response time is under 2 seconds, but for direct manipulation of data the reaction time must be under 2 milliseconds. The portability of *views* was tested on different domains and data sources: [SQL](#) databases, [RDF](#) triple stores and [SPARQL](#) endpoints. To better understand the limitations developers would face while building user-interfaces for open Linked Data, a realistic and large dataset was used: DBpedia contains almost three and a half million resources, stored in over a billion [RDF](#) triples (version 3.5.1, released Apr 28, 2010). This provided an excellent use-case for the research. Before starting implementation of the system on [SPARQL](#) endpoints, several tests were performed on large [SQL](#) databases. These tests had indicated that in order to provide a fluent interaction with the data via a user-interface, there are certain compromises that are required. For example, dragging a slider to continuously query the database would cause the system to slow down, as it has to continuously send queries and parse the results. Instead, sending queries only when the user finishes dragging the slider (indicated by a release of the slider handle) would make a significant improvement on the system.

The system evaluation was composed of two parts both logged and time-stamped:

1. Querying the endpoints and retrieving results from the filtering interface;
2. Visualising the result sets into 5 widgets textual results, geographical map, pie chart, bar chart, tag cloud.

The setup consisted of four cases based on the number of results returned 100, 600, 1100 and 2200. Sample queries like “Select all the public Universities in the United Kingdom”, or “Select all the places in United Kingdom” were passed from the interface to the backend. While developing the system, it was noted that some visualisations would often be significantly (and unpredictably) slower than others. This hinted at several possible causes, such as:

- Network traffic,
- Endpoint delays,
- Backend processing,
- Browser delays

Four background tasks were identified which could be responsible for a delay. The four individual tasks were measured: time to transfer queries to the backend (local network), time to execute query (endpoint performance), time to parse results (backend performance) and convert to [JSON](#) (backend performance), and time to transfer [JSON](#) objects (local network) to the frontend. The frontend was evaluated by timing the

performance (rendering time) of each visualisation widget. Figure 46 shows the relative response times for the DBpedia endpoint (increase in the result size maps: 100=1, 600=2, 1100=3, 2200=4). The time taken for the backend to process the query and send the results to the frontend varied from 123.78ms to 6.9s, with the query execution time varying from 98ms to 6.85s. For most of the cases (60%), the time taken for executing the query took more than 70% of the backend processing time. In order to see if there were computational bottlenecks or patterns that could be optimised, data was normalised to highlight the proportion among the different phases. Figure 46 plots the distribution of the 4 tasks and show several interesting points:

- The overall time taken by the backend is highly dependent on two factors — the query execution time; or the time taken to transfer the results to the frontend. These two tasks consume the maximum amount of time.
- The overall time taken by the backend is highly variable.
- The time taken for transferring the query to the backend and the time for converting the results to JSON objects is negligible compared to the other two.

A major concern is the query execution time, the variation of which is alarming and cannot be controlled. While it this could be attributed to high server load or the way queries are distributed, this is an important aspect that user interface developers need to take into consideration. The system tests show that though the query execution phase often takes a lot of time to complete, there are other phases in the backend processing that can be significantly improved. More investigation is needed to understand the causes of the delays in transferring the result objects to the frontend and further optimise this step.

This high variability in the query processing stage is in contrast to the performance achieved by traditional databases. In a similar experiment with a MySQL database, similar query results were retrieved and the backend performance was analysed. The overall backend processing time varied between 0.00026ms and 6.48ms. Though SQL data stores can be expected to be faster, the relative time taken by the query processing stage has been consistent, consuming most of the entire backend processing time. Figure 47 illustrates this variability where the system is significantly quicker, but the backend behaviour is similar across the examples — the query execution step takes the longest time.

5.6.4 Limitations

This study has one major limitation: the experiment was conducted on a public Linked Data endpoint, a system beyond the control of the experimental setting. This, while being a constraint poses an interesting question to the Semantic Web community – DBpedia is one of the most heavily used endpoints, and serves as an excellent example of a public endpoint with a large amount of information served in structured format. However, the issues observed in querying such an important resource indicates the potential issues that can be expected to arise out of a large organisational information infrastructure. Hence, despite being a limitation, the evaluation serves to help understand the potential issues that can arise out of interacting with very large scale semantic data. The

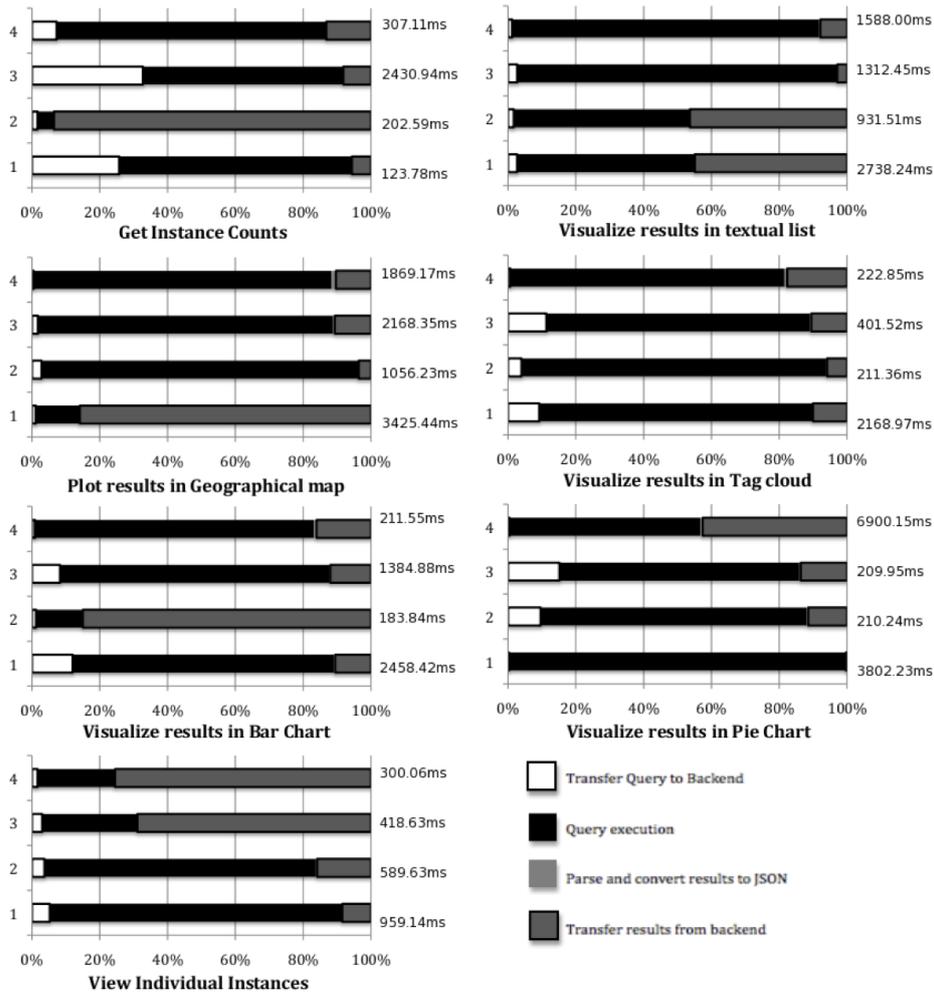


Figure 46: Plots showing the extremely high variation of query execution phase (98ms to 6.85s) in the backend processing. The size of the results (1 – 2200 results returned, 2 – 1100 results returned, 3 – 600 results returned and 4 – 100 results returned) are shown as individual bars on the y-axis, and the differently shaded x axis bars show the time taken to perform individual functions to retrieve the respective results. The plots show the relative times in different parts of the system

direct comparison with a MySQL database also is another limitation, but it is important to note that the different absolute times for processing the query request was not considered as important as the unpredictability associated with a public Linked Data endpoint.

5.6.5 Evaluation within an Organisation

An internal evaluation was conducted within Rolls Royce with aerospace engineers and managers to understand the applicability and usability of the system in a realistic scenario. The evaluation was conducted by an evaluation coordinator within the Samulet project, within the Knowledge Management team at Rolls Royce. Overall, 8 users tested

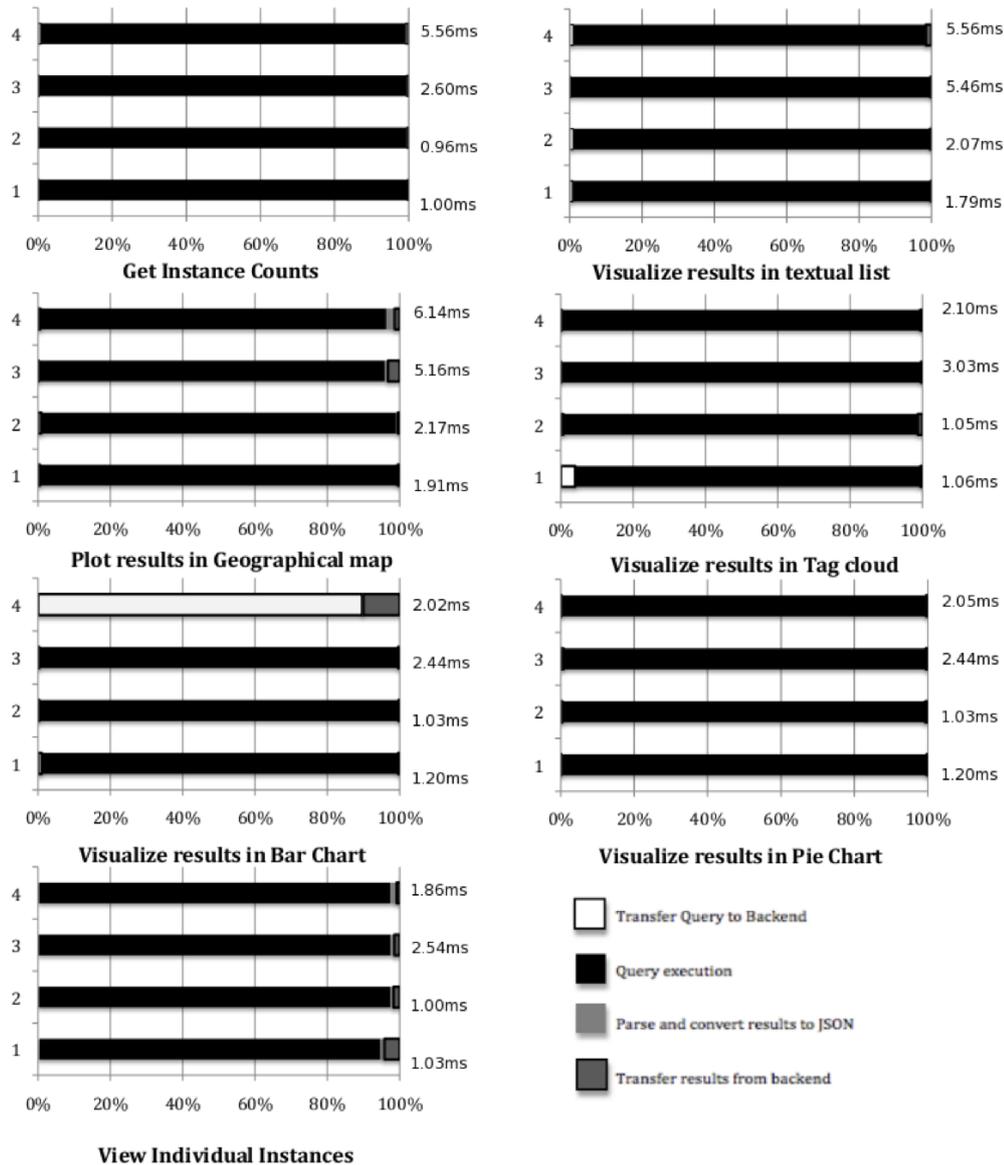


Figure 47: A traditional database being used as the backend — as expected, most of the times the query processing stage takes the longest times, however this is a consistent behaviour.

the system, which was setup with two different use cases and datasets: the first use case relates to exploring cost reduction ideas (5 users) in the manufacturing unit, while the second relates to analysing Concessions data (3 users). The datasets and use cases are described in more details in Sections 4.1.4.1 and 4.1.3.1. Two similar versions of *.views* were used, with a slightly different alignment of the visualisation widgets being used for each use case. Figure 48 shows the *.views* installation for the Cost Reduction use case. The installation for Concessions use case was very similar, with the only exception being the filters were placed on the left hand side of the screen as opposed to the top. The configuration of the two use cases were decided as a result of initial consultation with the respective user groups and data owners. It can be noted that, being deployed

within the context of aerospace engineering, the central focus is the engine map. The filters were pre-determined while deploying the solution, by identifying the filters of interest as a part of requirements gathering.

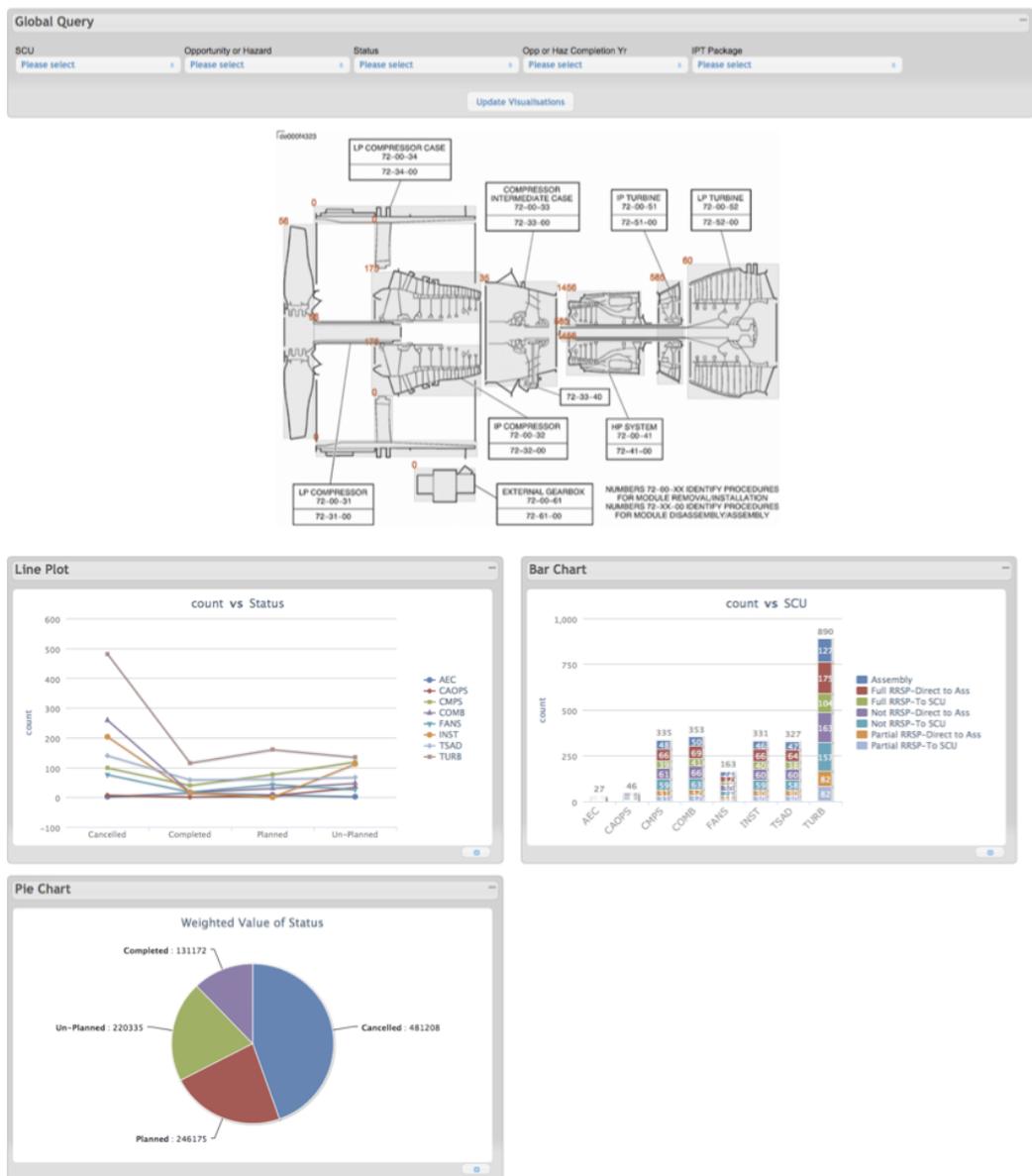


Figure 48: .views. deployed for Cost Reduction use case

Owing to organisational requirements, high cost of expert user’s time and security restrictions the number of users is highly limited, and hence eight highly expert users (who understand the domain, tasks and datasets very well) were considered to be a significant number. Though the evaluation was an internal one and the outcome not made available in publicly shared dissemination media, a brief summary of the evaluations are provided in this section. The evaluation sessions started with an initial introduction to the interface and how to construct filter queries, rebuild visualisations and view re-

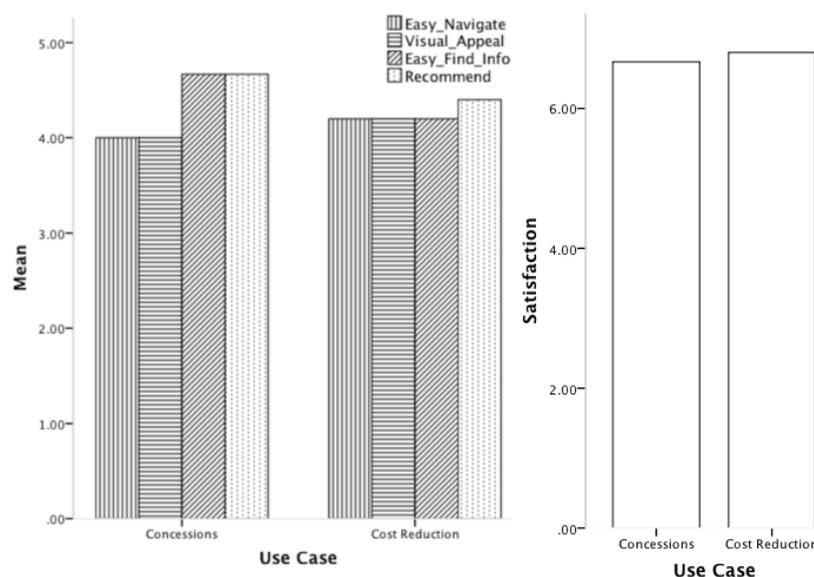


Figure 49: Evaluation Results

sults. The system was then handed over to the users, who then performed several tasks which were specific to the dataset and use case.

The results as shown in Figure 49 show that overall, users were very satisfied with the system (a mean score of 6.67 and 6.8 on a 7 point likert scale for the Concessions and Cost Reductions Use cases). Judging on a 5 point Likert scale (5 representing the most positive score, and 1 representing the least), users also found the system easy to navigate (a mean of 4 and 4.20), visually appealing (4.0 and 4.20) and easy to find information (4.667 and 4.20). Users also noted that they would recommend the system to their colleagues for exploring their own datasets.

5.6.5.1 Limitations

Though the number of users evaluating the system may not be considered statistically significant, the users appreciated the system overall. A further round of internal evaluation at a later date judged the readiness of the system as a TRL 6, which indicates the system is ready to be taken to a fully demonstrable system (for the two use cases)²⁹.

The evaluation within a Knowledge Management setting has been a low-scale one and only related to two use cases. A much larger evaluation with users within the information infrastructure of a typical large organisation, used as a part of several use cases is needed to validate and verify the effectiveness of *views*. from an organisational knowledge management context. Initial feedback from users and knowledge managers have been highly positive, and the potential of such a solution has been observed to a great extent. The system, having scored a TRL 6 rating (which is significantly high for a research solution) is highly encouraging. A need for adoption of such visual approaches in future versions of existing enterprise search solution was identified, which stemmed out from the development of the system.

²⁹ http://www.rolls-royce.com/about/technology/research_programmes/introduction/

The focus groups with users and system evaluations were extremely helpful in understanding what standard available technologies provide to developers, aligned against what users expected. Several features such as autocompletion of queries and dynamic queries were disabled, even though users expressed their desire to use such features. The evaluation within the organisation also demonstrated successfully the value of such an approach to end users. Though several other communities within the organisation were keen on employing the approach to explore their own (as well as unknown) datasets, this could not be pursued further owing to time and project restrictions. These studies are important in identifying future directions of research as well as realistically evaluating the capabilities of the system. Chapter 8, Section 7.1.1 reflects upon these findings and proposes solutions that can help alleviate some of the problems that developers might encounter while working with public Linked Data endpoints.

5.7 GENERIC MECHANISMS OF INTERACTION

The need for *views* to be a generic solution, independent from datasets and domains needed a unique way of formalising interpretation of user actions. As discussed in Section 5.3.2, several processes are involved:

1. Capture the interaction
2. Identify the target content of the interaction
3. Identify the context of the interaction
4. Interpret user action
5. Interpret user's information intent
6. Identify relevant query template
7. Generate SPARQL query

As can be expected, multiple visualisations have multiple ways to capture interactions, and hence, different ways of explaining the intentions behind a user action. Tables 4 and 5 illustrate how these processes are followed in the present implementation of *views*. The tables show the type of interaction a user performs, and on the basis of the target of the interaction and the context surrounding the interaction, an inference is made on what the user wants to do. Based on the inference, a template query is selected. The template query is then populated with the relevant information that is provided by the target and context, which then constitutes to formulate a complete SPARQL query. The following subsections discuss the different user interactions in more details from the perspective of global filters and local filters.

5.7.1 Global Interactions

The first type of interaction occurs when the user selects a broad subset of the data to explore – this is performed using the global filters. Interacting with the global filters make selections of the data that affect all the visualisation widgets on the dashboard.

Hence, the interactions are referred to as ‘global interactions’. Global filters are set on the basis of the data type, and therefore, impacts the type of interaction a user can have with them. For example, a data type property that expects a numeric field as a valid value can be presented as a slider or range slider. On the other hand, a data type property that represents a date field can be presented as a date input widget. When the data type can not be ascertained, a plain text entry field can be presented to the users.

The initialisation process (1. Init) involves the user entering the web address for the installation or reloading an existing interface. This process initially sends two main SPARQL queries to the backend - initiate exploration and display counts. The results of the first query are parsed to populate a list of filters, while the results of the second indicate how many instances exist in the dataset (this is displayed on a search button, with the text ‘Results Available’).

The user starts exploration by identifying a filter of interest (2. Select) from the drop-down list of filters. This creates a filter with an appropriate interface (based on the data type of the filter), which the user can interact with by entering their restrictions. As a part of this process, another query is executed to count the number of instances that are relevant to the filter term, which is then rendered on the search button. The user then interacts with the filters to indicate the relevant instances of interest via either text search fields, sliders, check boxes or date widgets (3. Enter text, 4. Click, click and drag, enter). These are then interpreted into SPARQL queries to display the number of relevant instances on the search button. The user, upon clicking the search button (5. Click) triggers several SPARQL queries. These queries are dependent on the type of visualisation widgets being presently used. Example queries, for which are shown in Table 5.

Type	Target	Context	Intent	Query Template	SPARQL
1. Init	All Widgets	Throughout interface	Initiate Exploration	Build filter list Display counts	<pre>SELECT DISTINCT ?s ?property ?value. } ORDER BY (?property). SELECT COUNT (DISTINCT ?s) WHERE { ?s ?property ?value. } ORDER BY (?property).</pre>
2. Select	Drop-down list item	Filter list	Select query facet for global query	count all instances containing relevant facet values	<pre>SELECT COUNT(DISTINCT ?s) WHERE { ?s ?property ?value. ?s <propertyX> ?valueX. }</pre>
3. Enter text	Text box (value entered ‘text term’)	Global filter query item (propertyX)	Work only with instances which have value ‘text term’ for propertyX	count all instances containing relevant facet values	<pre>SELECT COUNT(DISTINCT ?s) WHERE { ?s ?property ?value. ?s <propertyX> ?propertyXValue. FILTER regex(?propertyXValue,"i", "text term") }</pre>
4. Click, click and drag	checkbox, drop-down list item, date selector, slider (value set to ‘XValue’)	Global filter query item (propertyX)	Work only with instances which have value ‘XValue’ for propertyX (and match other global query restrictions)	count all instances containing relevant facet values	<pre>SELECT COUNT(DISTINCT ?s) WHERE { ?s ?property ?value. ?s <propertyX> ?propertyXValue. FILTER (?propertyXValue = XValue) }</pre>

5. Click	Search button	Global filters updated	Work only with instances which match global filter constraints	Re-plot all visualisation widgets	All relevant aggregate/instance queries to load visualisation widgets
----------	---------------	------------------------	--	-----------------------------------	---

Table 4: Global User Interactions translated to SPARQL queries

5.7.2 Local Interactions

Interactions within the scope of individual visualisation widgets or visual elements constitute Local Interactions. Upon the visualisations being rendered, the user has two main options – re-plot individual visualisations to present a different facet of the data or indicate individual data points or sections within visualisations to view the data. For certain visualisations (e.g. spatial maps), the user has the ability to zoom-in or out (4. Click or Scroll) from the level of view (e.g. zooming into a specific area of the engine). The Table 5 shows the way these interactions are translated in *.views*. If the user chooses to re-plot a widget (3. Select), the new facet is then set as the context for the widget and a new SPARQL query is created. The query is executed, and the results are parsed to rebuild the visualisation. Upon selecting individual sections (1. Click, 2. Click, 5. Click) of aggregate visualisations (e.g. pie charts), a SPARQL query is created to look for instances that match the condition where the context (facet property) contains the value clicked (target). As a result of executing the query, the user is provided with a popup that shows a list of instances. Clicking on any instance (6. Click) loads all the details of the instance on the popup, following another SPARQL query execution. The instances are presented as a paginated list, which can be browsed by the user if required (7. Click).

Type	Target	Context	Intent	Query Template	SPARQL
1. Click	Aggregate Section (propertyValueX)	1-D Aggregate Charts, aggregated by one facet (propertyX)	More info on a collection of instances	List all instances that match the value of facet	SELECT DISTINCT ?instance WHERE { global query triple(s). ?instance <facetCategory> ?facetCategoryValue. }LIMIT numPerPage
2. Click	Aggregate Section or data point (propertyValueX, propertyValueY)	2D Aggregate Charts, Aggregated by one facet (propertyX), grouped by one category (propertyY) or scatterplot	More info on a collection of instances	List all instances that match the value of facet and the relevant group	SELECT DISTINCT ?instance WHERE { global query triple(s). ?instance <propertyX> ?propertyValueX. ?instance <propertyY> ?propertyValueY. }LIMIT numPerPage
3. Select	Drop-down list item	Aggregate charts, line/scatter plots	Visualise data in another perspective	Re-plot a group of instances with respect to another facet	SELECT DISTINCT ?newCategoryValue COUNT (?instance) AS ?count WHERE { ?instance <newCategory> ?newCategoryValue. }ORDER BY DESC(?count)
4. Click or Scroll	Aggregate Section (Value X)	Spatial Map (layouts of physical objects i.e. engine)	Drill-down or roll-up to the next level of specificity or genericity	Re-plot all instances on the basis of a higher level of detail	SELECT DISTINCT ?spatialMapCategoryValue COUNT (?instance) AS ?count WHERE { ?instance <spatialMapCategory> ?spatialMapCategoryValue. FILTER (?spatialLevel = X). }ORDER BY DESC(?count)

5. Click	Specific location (Value X)	Spatial Map (geographical)	More info on a collection of instances	List all instances based on individual location (X)	<pre>SELECT DISTINCT ?instance WHERE { ?instance <geoCategory> ?geoCategoryValue. }LIMIT numPerPage</pre>
6. Click	Item on list of instances	List of instances	More info on an instance (or collection of instances)	List all information on an individual instance	<pre>SELECT * WHERE { <item> ?property ?value. UNION ?isvalueof ?property <item>. }</pre>
7. Click	pagination	List of instances	Inspect next/previous set of instances	List all instances that match global filters and are on the relevant page	<pre>SELECT DISTINCT ?instance WHERE { ?instance <Category> ?CategoryValue. }LIMIT numPerPage OFFSET pageNum*numPerPage</pre>

Table 5: Local User Interactions translated to SPARQL queries

As can be observed from the Table 4 and 5, several rules govern how a user interaction is interpreted, based on the context of the interaction. These interpretations are translated to formal SPARQL queries, via different query templates. While these rules are developed specifically for the present implementation of *.views.*, the behaviour they exhibit are generic, drawn from observation of other systems and user interviews. For example, the action of clicking on a marker in a geographic map resulting in a popup detailing the object clicked being presented to the user is an expected outcome, which is observed in several other systems such as Google Maps, Bing Maps and so on. It is to be noted, however, that these interpretation and rules are flexible, and can be modified based on a different set of requirements if needed. This generic approach of isolating user interactions, interpretations and results from the datasets, domains and user types helps establishing the flexibility of the solution.

5.8 SUMMARY

User interfaces in the Semantic Web have been mostly constrained, either in the context of domain-specificity, user expertise, user tasks or even datasets. The work involved with the development and design of *.views.* was an effort toward standardising a generic visualising mechanism for Linked Data and Semantic Web. It was also an effort toward understanding how users react to multiple visualisations when presenting highly domain specific data, in a generic manner. The positive feedback was encouraging and reinforced the possibility of the development of such frameworks. While domain experts appreciated a generic interface, they also expressed the desire of integrating domain-specific visualisations that they are accustomed to. For example, aerospace engineers expressed the need for engine maps and factory maps (which was later developed and incorporated in the system, as discussed in [MVL⁺12]) and enriched CAD diagrams. Similar to aerospace engineers, biologists expressed the need for a similar visualisation to describe the anatomy of plants or organs, and highlighting areas of interest based on the data. Most users commented on the quantity of the data, and in spite of being excited about the possibilities of visualising datasets by making use of simple querying and filtering, users were slightly disappointed with the lack of specific datasets or data instances. The users, however appreciated the capabilities of the system in dealing with

unknown datasets and several discussions ensued toward providing students with access to the system and various datasets. The second focus group session with biologists and computer scientist was very interesting, as users could immediately relate *views* to their daily tasks and could understand how such an interface would benefit them and the community.

The evaluation with public Linked Data endpoints provided unexpected results, as analysis revealed that the performance of the endpoints are not predictable and can result in users waiting for extended periods. One might argue that this study on DBpedia is not representative of all the Linked Data endpoints, that are available to developers. However, the study is focussed in understanding how developers can make use of available systems in order to make interactive visualisations using semantic data. DBpedia is one of the largest and most popular Linked Data endpoints, and has been considered to be one of the best examples of the Linked Data endeavour since its inception. Additionally, the DBpedia paper [BLK⁺09] had been awarded the most cited paper on the Journal of Web Semantics during the period 2006-10³⁰. While the experiment is focussed at a highly used Linked Data endpoint and other endpoints may not reflect this behaviour, it is important to be aware of the constraints that developers may face in dealing with public endpoints. A similar constraint can also be observed within large organisations, as massive data repositories can have multiple concurrent users and a situation may arise where several parallel queries to the organisational endpoint can severely limit the performance of a system. It can also be argued that developers may consider using a local endpoint to host large datasets, and all interactions would be only with privately hosted Linked Data endpoints. While such solutions are one of the few ways of mitigating the issues concerning public endpoints, it is an expensive one and several developers and smaller organisations may not have the necessary resources to take such an approach.

The first research question, (R1 – *How can visualisation interfaces provide effective means of exploring large scale distributed semantic data ?*) is tackled in this chapter. The chapter provides evidence to validate that by combining a multi-visualisation approach, users can effectively explore semantic data. Revisiting the two main factors highlighted in Section 5.1 highlights that while users appreciate a multiple visualisation dashboard approach and can easily understand large datasets from multiple perspectives, existing technological infrastructure in the Semantic Web raises several issues from a user experience as well as technological constraints point of view. The user experience is adversely affected as a result of technological limitations as several highly desired features had to be disabled such as auto completion and dynamic queries. From a technical implication, a limitation is in the amount of complex queries that can be generated in parallel. For example, the number of simultaneous widgets are limited since the framework would not seamlessly support a large number of queries. This chapter also addresses the second research question (R2 – *How can visualisation interfaces help explore semantic data in a generic manner ?*) and presents a mechanism that can be used to generically explore semantic data by translating and interpreting high level user interactions into lower level intents, and subsequently into formal queries. The enabling rules and templates for such are presented in the Section 5.3.2.

³⁰ <http://www.wiwiiss.fu-berlin.de/en/institute/pwo/bizer/news/JWSMostCitedArticleAward.html>

Whilst several researchers have benchmarked triple stores and a healthy competition between the various triple stores exist, it is also important to investigate the performances of such stores from the perspective of end users — the developer and user communities. Providing accurate and precise results, query evaluation and execution performances are highly significant and necessary for the database community, but there is a need for explicit studies to understand how such data stores can realistically support interactive Linked Data visualisations. In spite of a very well received interface for visualising query results, the focus group sessions highlighted the various shortcomings of the filtering interface, where users found it very difficult to navigate large list of filters. However, in comparison to another form-based query approach [BCC⁺08], users preferred a list-like interface similar to *.views.*. This raised the need for more investigation in other types of query approaches. The users appeared to be most excited about the visual approach of presenting query results, which was highly appreciated by all users. The unanimous appreciation of a highly visual approach together with a colourful layout and highly interactive mechanisms in exploring semantic result sets raised the need for an explicit focus on aesthetic design. The work discussed in the next section explores a different mechanism of exploring and querying semantic data, as a follow-up step from the need generated out of the user evaluations of the *.views.* system.

AFFECTIVE GRAPHS — EXPLORING AND QUERYING SEMANTIC DATA

6.1 INTRODUCTION

Starting with the lessons learned from the evaluation of *.views.* and the observations and feedback of the users, the second of the two-fold solution approach is discussed in this chapter. Development of the *.views.* solution and subsequent evaluations with potential users and domain experts identified several keypoints that are of interest to the next part of the thesis —

- Publicly available endpoints may be unpredictable and retrieving results may cause unexpected delays;
- A highly visual and interactive approach toward exploring datasets is highly appreciated by end users and was observed to stimulate a lot of interest;
- Representing data instances into recognisable and familiar visualisations can help users gather a good understanding of their data;
- A greater stress on expressing Information Need (query mechanism) is required to facilitate users express their queries.

The weakest feature of *.views.* was identified as the query mechanism, while the most appreciated quality was the visual approach of presenting results. The form-based query mechanism was not appreciated due to the large number of filters users had to browse and the difficulty in understanding how the filters are related to the concepts. Users were also pleased with the highly visual and interactive approach, and the interest generated a need to investigate a similar visual approach for querying semantic datasets. However, in order to provide a well-designed approach that is pleasing to users as well as provide essential functionality, an explicit attention to aesthetic design and Visual Analytic techniques is needed. This was the main motivation for the following piece of work: to provide a highly visual approach for querying semantic data that can stimulate an aesthetic response from users. The solution for querying semantic data presented in Chapter 4 discuss the proposal for an interactive interface, providing basic visualisations to guide users to progressively explore the dataset and discover more information with interactions.

Such a visual and interactive approach toward developing a solution for querying large datasets requires a multidisciplinary outlook. The nature of semantic data, being graphical introduces further challenges, as apart from visualising data instances, classes and concepts, their properties and relations are also essential. Research done by the Visual Analytics community is essential in understanding how large graphs can be visually represented, in a generic manner. Interaction design can identify ways to interact with such visual objects in order to translate user needs and intents to formal queries. A highly visual approach also calls for the need to investigate how aesthetic design can

be incorporated with Visual Analytics solutions since an attention to aesthetic design can provide a stimulating interface, make users more receptive and eventually improve the performance of the user. Semantic Web benefits from a characteristic of highly networked data instances, each well defined by self-explanatory schema. This type of data introduces new challenges and opportunities and an explicit study of aesthetics for Semantic Web applications is much needed. Thus, a strong understanding of the Semantic Web and its principles is needed in order to fully appreciate the possibilities offered by semantic data, as well as how a generic approach can be formulated.

The interest, in this chapter is only on a query approach, rather than a complete system that includes querying and visualising results. User evaluations with *.views* was instrumental in proving that the approach of multiple visualisations and dynamic querying works on semantic data, and it is possible to build generic interfaces that facilitate visualisation of results. This chapter does not encapsulate the visualisation of results, but instead provides a very basic tabular view of results. The main intention is that the querying interface will “plug-in” as a widget on the *.views* interface, thereby creating a concrete interface that enable exploration and querying of semantic data in a visual manner, and the results of queries will be visualised as multiple visual widgets. Hence, the rest of the chapter only discusses visualisation in the context of exploring data to build queries, instead of means to display query results.

The final solution, *Affective Graphs* was evaluated four times with several users with varying expertise and experience levels. This chapter discusses the development and evaluation of the system. Section 6.2 identifies a set of best principles for developing aesthetic interfaces for the Semantic Web. Section 6.4 discusses the approach employed by *Affective Graphs* and what are the main principles behind the final solution and the query approach. Section 6.5 introduces the final system, along with a scenario of use. The section 6.6 discusses the design rationale and discuss the several design decisions taken during the process of developing the system. Section 6.7 discuss how the system is built and the architecture of *Affective Graphs*, and how queries are translated from user interactions. Section 6.9 discusses the various evaluations that were conducted in order to understand how well the system performs as well as how pleasing the users judged the system. The chapter concludes with a summary.

6.2 PRINCIPLES OF AESTHETICS FOR SEMANTIC DATA

While aesthetics has not been considered explicitly in the Semantic Web and very few Visual Analytic systems can be identified that have been developed specifically for the Semantic Web community, it is possible to learn from such fields and formulate an approach to help developers and designers build interactive systems. The findings from aesthetics and Visual Analytics as well as the principles of Semantic Web can be aligned together to form a set of guidelines and principles for the community to consider while developing such solutions. This section proposes a set of design principles that are specific to the Semantic Web and can aid in the design and development processes. The guidelines are divided into two sections — general aesthetic principles involve the overall design and layout of the interface in general (Table 1); node-link representation principles involve the design of node-link graphs for representing Linked Data (Table 2).

Design Principle	Description	Proposed By
1. Use words, numbers and drawing to convey information	Graphics alone are not always enough to convey the significance of a piece of information — numbers, narratives, explanations can aid in better communication	[Tuf86]
2. Well balanced, proportioned and symmetrical design	Interface should be arranged so that optically larger and smaller objects balance each other in a symmetrical manner; Interface should be well proportioned (i.e. golden ratio, greater horizontal length etc.)	[Tuf86, Kim06, SH07, NTB03, SH07, BRSG07, FDPL05, Eico3]
3. Rhythm, unity in design	Interface should be designed from multiple visual elements, coherently constituting a pleasing layout, with regular patterns of visual changes to make the appearance exciting	[MG05, NTB03]
4. Different weights of lines to represent different information	Contrasting lines indicate different meanings — weights can be associated with values or types of data	[Tuf86]
5. Simple, consistent and stable figures	Complex visual representations require greater interpretation and add to the cognitive burden of users; Adhering to semantic web principles and standards require a consistent representation of data elements across domains and application areas	[BRSG07, BPS05]
6. Using variations of color, shape, size, intensity to present trends, interesting patterns, anomalies or represent similarity, physical connection	Color, hue, size, shape etc are visual clues that we can quickly spot, thereby making it easier to observe patterns, anomalies etc.	[HBE92, SH07, BRSG07]
7. Minimalist design, reduce visual clutter	Interface should be minimalistic, and contain as little data-free visual elements as possible	[NM90, Tuf86, SH07, BBD09, CS96]
8. Balance in harmony and typicality	Typical solutions require little effort, but at the cost of being a mundane solution — balance in variety and typicality is important	[SH07, KG10, Kim06, MG05]
9. Maintain consistency in visual representations, interaction mechanisms and standards	Visual representations (color, shape, hue etc) should be consistent across all domains; interaction mechanisms should be familiar to users and standardised (e.g. right-click should present context menus etc.)	[NM90, MG05]
10. Follow visual information seeking principles with minimal cognitive burden on users	Provide mechanisms to overview, navigate, filter and access data instances on demand, whilst ensuring minimum cognitive load and changes to the mental map	[RLN07, Shn96, Tuf86, Tuf90]

Table 6: General Principles for Aesthetic Linked Data Visualisation

The principles listed have been selected based on their applicability to Semantic Web applications, however, several other well established principles in interface and graphic design should also be considered while developing an application. It is important to note that whilst the existence of such principles is necessary, the applicability of the principles can often be at the discretion of the designer and developer.

6.3 EVALUATING AESTHETICS OF SEMANTIC WEB APPLICATIONS

Gaining a familiarity with existing Semantic Web tools, and principles of aesthetics, a question arises — “*What role has aesthetics played in the Semantic Web so far?*”. This was the motivation in conducting the first evaluation — the starting point of the research was to understand how well existing Semantic Web tools align with each other with respect to aesthetic properties. Semantic Web and Linked Data Interfaces have traditionally been designed and evaluated performance and reliability, with few evaluations focussing on usability [KBF07, LDS⁺05]. In addition to a greater stress on usability and

Design Principle	Description	Proposed By
11. Separate representations of hierarchical and non-hierarchical relations	Differentiating between hierarchical and non-hierarchical relations helps users navigate along or across graphs	[Eico3]
12. Reduce overlapping nodes		[Eico3]
13. Center parents or children	The parent node should be located as close as possible to the median position of the child nodes	[Eico3]
14. Cluster nodes based on semantics	The position of nodes should be based on their semantics so that nodes that are adjacent to each other have a significance	[Eico3, BRSG07]
15. Avoid edge crossings or overlaps	Every edge should be as visible and readable as possible and spaced apart from nodes	[Eico3, BRSG07]
16. Uniform and minimal edge bends	Minimal angles on the edges to help users follow links	[BRSG07]
17. Even distribution of nodes	Well distributed and evenly spaced nodes, but ensuring compactness	[BRSG07]
18. Maintain aspect ratio, symmetry	Maintaining symmetry within the layout as well as in the overall interface; aspect ratio of the graph should match the container (interface, screen, page etc.)	[Eico3, BRSG07]
19. Minimise total graph area	Compact layout but ensuring readability	[BRSG07]

Table 7: Principles for Aesthetic Node-Link Representations

user experience¹, a lot of consideration also needs to be paid to aesthetics while designing products [jor98]. Evaluating the aesthetic properties of Semantic Web-based user interfaces is a step forward in the direction of establishing a research area fundamentally focussed on the development of aesthetically pleasing interfaces for the Semantic Web. The first, relatively low cost evaluation aimed at objectively evaluating interface aesthetic properties as it can provide a simple and inexpensive way of assessing various aesthetic properties of systems. The experiment was based on the model provided by Ngo [NSA00, NTB03], where different metrics of an interface are computed, on the basis of the layout of visual objects within the visual space. These metrics have also been previously used in identifying most aesthetically pleasing layouts of websites from a set of candidate designs [ZTG11a, ZTG11b]. An extensive survey of the literature did not identify any existing work in Semantic Web research where such metrics have been used in evaluating aesthetic properties of interfaces. The metrics have also been reported to be highly correlated with subjective scores from users [ZTG11a, PHJR11, PCK05, AL11]. Another reason for considering interface layout metrics, as pointed out by [PHJR11] is the belief that the placement of visual objects in interfaces can be a strong predictor of aesthetic appeal and perceived usability. It is important to note that this study involved only the interface layout from a general aesthetic point of view and factors such as colour, styling, typography or individual visualisations were not a part of this study, but will be explored as part of a future work.

6.3.1 Experiment design

The evaluation was based on eight of the thirteen metrics provided by Ngo [NTB03, NSA00]. Five most important properties identified by Zain et al in their similar evalua-

¹ A significant lack of attention to usability and user experience has been a concern to the community. Indeed, David R. Karger's talk (http://videlectures.net/eswc2013_karger_semantic/) stressing the need for designing semantic web solutions with explicit attention to users bears testimony to the urgent need for such solutions

tion exercise with Ngo's measures are initially considered [ZTG11a, ZTG11b] - **Balance**, **Equilibrium**, **Symmetry**, **Sequence** and **Rhythm**. Balance is a measure of how the visual elements with different optical weights (larger objects are perceived as 'heavier') are distributed in the interface; Equilibrium indicates how centered the layout appears to be; Symmetry indicates how well replicated elements are on either side of the horizontal and vertical axes at the center of the interface; Sequence is a measure of how well objects are arranged in the interface, with respect to the movement of the eye; and Rhythm indicates the variation of visual objects in order to make an interface exciting. Two other metrics were identified from the design principles that would be important for the evaluation — **Cohesion** and **Unity**. Cohesion is a measure of the similarity of aspect ratios of the visual elements in an interface; and Unity is a measure of the extent to which the elements seem to belong together. The final property, Order and Complexity is defined as the sum of all the properties. The metrics not considered in this study are simplicity, regularity, homogeneity, economy and density. Certain visualisations such as graph-based ones can affect how simple or complex a user might interpret the interface. Other visualisations such as scatterplots or maps can also affect the interpretation of the general economy, regularity, homogeneity and density of an interface. These metrics will be investigated in more details in the future when there is a greater understanding of the implications of complex visualisations and graphs on these factors. All the scores range between 0 and 1, where 0 indicates highly negative assessment, while 1 indicates a highly positive assessment. These metrics are defined in [NSA00, NTB03] and the relevant formulae are provided in Appendix A. This evaluation will be revisited in Section 6.9.5, in an analysis that attempts to understand how the proposed solution performs as compared to the existing interfaces.

The first step in evaluating the metrics was understanding which tools would be good candidates for the comparative study. The survey of the literature identified ten well-known Semantic Web tools that have existed over the past few years: mSpace [sWRS06], PowerAqua [LMU06], K-Search [BCC⁺08], Sig.ma [TCC⁺10], Tabulator [BLCC⁺06], DBpediaFacet², Semantic Crystal, NLP Reduce, Ginseng and Querix [KBF07, KB10]. While some of the tools ranging from browsers and search systems (e.g. K-Search, PowerAqua, mSpace, Sig.ma) to standalone interfaces (e.g. Semantic Crystal, NLP Reduce, Querix, Ginseng) are clearly research prototypes, the intention is to also understand how they compare with the rest.

6.3.1.1 Computing Interface Aesthetics

A standalone java-based application (Aesthetics Calculator) was developed, which was fed screen shots of the interface layouts (Figure 50). A user then manually marked up the areas that contain visual objects by using mouse gestures like click and drag. Each of the manual annotations were then stored locally and their dimensions calculated. The application then aggregates the different measures as the visual objects are marked up, based on the formulae provided by Ngo. The screenshots of the ten systems are either obtained from local installations, publication material, website images or screenshots in user manuals. These measures are then collected and compared against each other. As this process involves human annotations (markups), each interface is annotated three

² DBpedia Faceted Search, <http://wiki.dbpedia.org/FacetedSearch>

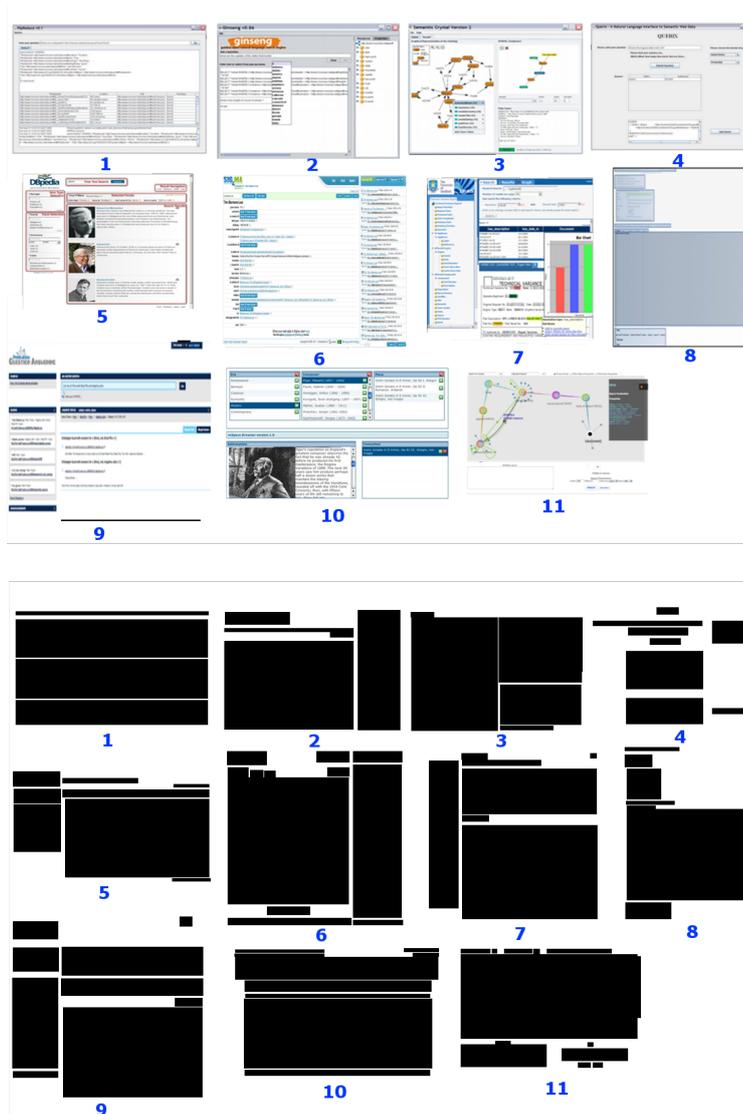


Figure 50: Layouts used to compute interface aesthetic metrics. The layouts were obtained from the interfaces directly or from the relevant papers, publication materials, websites or user manuals. The layouts shown are of ten well-known Semantic Web interfaces (screenshots of the system are shown on the top, while images at the bottom show the respective markups): 1. NLP Reduce, 2. Ginseng, 3. Semantic Crystal, 4. Querix, 5. DBpediaFacet, 6. Sig.ma, 7. KSearch, 8. Tabulator, 9. PowerAqua, 10. mSpace and 11. Affective Graphs (later discussed in Section 6.9.5)

times by one user and the mean is then computed and compared against the others. The following describes how the different measures were computed.

Figure 51 shows an example screen, where the layout has been equally divided into four quadrants – UL (Upper Left), UR (Upper Right), LL (Lower Left), LR (Lower Right). In this example, four elements are shown in different quadrants of the screen, numbered 1-4, with areas a_1 - a_4 respectively. Upon loading a new screenshot, the user starts marking up the different areas in the interface. As shown in the Figure 51 (Right), the user needs to select a visual area and click-and-drag to highlight the boundaries of

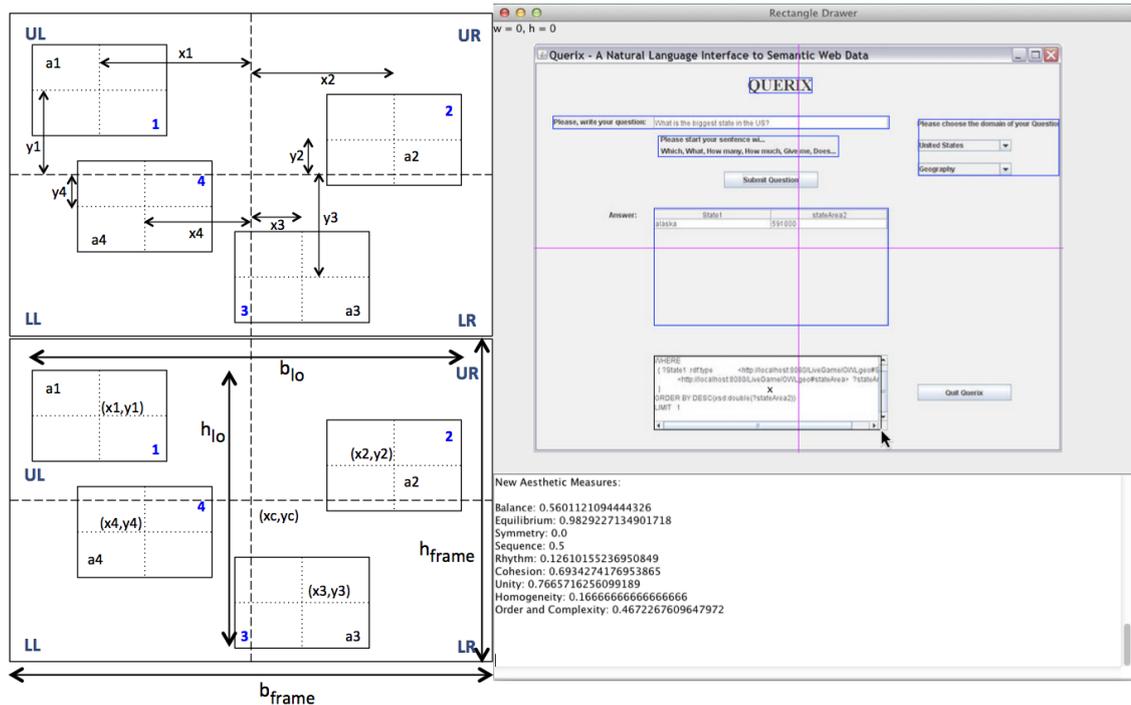


Figure 51: Calculating Interface Metrics. Left: Illustration of how different measures were calculated. Image recreated, inspired from [NSA00]; Right: The Java application, Aesthetics Calculator shows a user left clicking-and-dragging the mouse to mark up a visual area. Upon releasing the mouse button, the new aesthetic measures are calculated and shown at the bottom

the area. Upon releasing the mouse handle, the new metrics are calculated and shown at the bottom of the interface - with every subsequent markups, Aesthetics Calculator computes the new layout aesthetics, until all the sections have been marked up. The user can then finally take note of the metrics and proceed with further analysis. The rest of the section discusses how the metrics were calculated – this is also discussed in more details in Ngo’s papers [NSA00, NTB03].

CALCULATING BALANCE The first event that occurs after a section has been marked up is the dimensions of the new area, as well as the displacements from the imaginary central axes are calculated. Using these measures, the first metric, Balance is computed.

w_L (Weight Left), w_R (weight Right), w_T (weight Top) and w_B (weight Bottom) are then calculated as follows:

$$w_L = a_1 x_1 + a_4 x_4 \quad (1)$$

$$w_R = a_2 x_2 + a_3 x_3 \quad (2)$$

$$w_T = a_1 x_1 + a_2 x_2 \quad (3)$$

$$w_B = a_3 x_3 + a_4 x_4 \quad (4)$$

A more generalised way to calculate the measures is as follows:

$$w_j = \sum_i^{n_j} a_{ij} d_{ij} \quad j = L, R, T, B \quad (5)$$

BM_{vertical} and $BM_{\text{horizontal}}$ can then be computed as follows:

$$BM_{\text{vertical}} = \frac{w_L - w_R}{\max(|w_L|, |w_R|)} \quad (6)$$

$$BM_{\text{horizontal}} = \frac{w_T - w_B}{\max(|w_T|, |w_B|)} \quad (7)$$

The final measure of Balance can then be calculated by the following formula:

$$BM = 1 - \frac{|BM_{\text{vertical}}| + |BM_{\text{horizontal}}|}{2} \in [0, 1] \quad (8)$$

CALCULATING EQUILIBRIUM Equilibrium measures stability, by comparing the central points of each visual object with the central point of the interface. For the present implementation, the central point (x_i, y_i) of each object, i as well as the central point of the interface (x_c, y_c) was calculated, and then the following formula was applied to calculate the Equilibrium along the x (EM_x) and y (EM_y) axes:

$$EM_x = \frac{2 \sum_i^n a_i (x_i - x_c)}{nb_{\text{frame}} \sum_i^n a_i} \quad (9)$$

$$EM_y = \frac{2 \sum_i^n a_i (y_i - y_c)}{nh_{\text{frame}} \sum_i^n a_i} \quad (10)$$

where a_i indicates the area of the object i , b_{frame} and h_{frame} indicate the breadth and height of the frame and n indicates the number of visual objects in the frame. Finally, when EM_x and EM_y are computed, the measure of equilibrium can be calculated from the following formula:

$$EM = 1 - \frac{|EM_x| + |EM_y|}{2} \in [0, 1] \quad (11)$$

CALCULATING SYMMETRY Measure of Symmetry, or axial duplication is done from three perspectives: vertical, horizontal and radial. Hence, three measures of symmetry are calculated initially, SYM_{vertical} , $SYM_{\text{horizontal}}$, and SYM_{radial} considering all the quadrants (UL, UR, LL, LR) according to the formula:

$$SYM_{vertical} = \frac{|X'_{UL} - X'_{UR}| + |X'_{LL} - X'_{LR}| + |Y'_{UL} - Y'_{UR}| + |Y'_{LL} - Y'_{LR}| + |H'_{UL} - H'_{UR}| + |H'_{LL} - H'_{LR}| + |B'_{UL} - B'_{UR}| + |B'_{LL} - B'_{LR}| + |\Theta'_{UL} - \Theta'_{UR}| + |\Theta'_{LL} - \Theta'_{LR}| + |R'_{UL} - R'_{UR}| + |R'_{LL} - R'_{LR}|}{12} \quad (12)$$

$$SYM_{horizontal} = \frac{|X'_{UL} - X'_{LL}| + |X'_{UR} - X'_{LR}| + |Y'_{UL} - Y'_{LL}| + |Y'_{UR} - Y'_{LR}| + |H'_{UL} - H'_{LL}| + |H'_{UR} - H'_{LR}| + |B'_{UL} - B'_{LL}| + |B'_{UR} - B'_{LR}| + |\Theta'_{UL} - \Theta'_{LL}| + |\Theta'_{UR} - \Theta'_{LR}| + |R'_{UL} - R'_{LL}| + |R'_{UR} - R'_{LR}|}{12} \quad (13)$$

$$SYM_{radial} = \frac{|X'_{UL} - X'_{LR}| + |X'_{UR} - X'_{LL}| + |Y'_{UL} - Y'_{LR}| + |Y'_{UR} - Y'_{LL}| + |H'_{UL} - H'_{LR}| + |H'_{UR} - H'_{LL}| + |B'_{UL} - B'_{LR}| + |B'_{UR} - B'_{LL}| + |\Theta'_{UL} - \Theta'_{LR}| + |\Theta'_{UR} - \Theta'_{LL}| + |R'_{UL} - R'_{LR}| + |R'_{UR} - R'_{LL}|}{12} \quad (14)$$

where $X'_j, Y'_j, H'_j, B'_j, \Theta'_j, R'_j$ are the normalised values of

$$X_j = \sum_i^{n_j} |x_{ij} - x_c| \quad (15)$$

$$Y_j = \sum_i^{n_j} |y_{ij} - y_c| \quad (16)$$

$$H_j = \sum_i^{n_j} h_{ij} \quad (17)$$

$$B_j = \sum_i^{n_j} b_{ij} \quad (18)$$

$$\Theta_j = \sum_i^{n_j} \left| \frac{y_{ij} - y_c}{x_{ij} - x_c} \right| \quad (19)$$

$$R_j = \sum_i^{n_j} \sqrt{(x_{ij} - x_c)^2 + (y_{ij} - y_c)^2} \quad (20)$$

$$j=UL,UR,LL,LR \quad (21)$$

The values are normalised with respect to its maximum absolute value across the quadrants. Finally, the measure of Symmetry can be computed by the formula:

$$SYM = 1 - \frac{|SYM_{vertical}| + |SYM_{horizontal}| + |SYM_{radial}|}{3} \in [0, 1] \quad (22)$$

CALCULATING SEQUENCE The measure of arrangement of visual objects in the manner that facilitates the movement of the eyes is sequence. In this context, sequence related to a typical Westernised reading pattern (left to right), as did Ngo's formula. The

first step in calculating sequence is to weigh (visually, based on area) every quadrant based on the visual objects located in the relevant quadrant. This is done by the following formula:

$$w_j = q_j \sum_i^{n_j} a_{ij}, j = UL, UR, LL, LR \quad (23)$$

$$w = w_{UL}, w_{UR}, w_{LL}, w_{LR} \quad (24)$$

Each quadrant is then assigned a score, v_j , which is then finally used to calculate the sequence measure. For the heaviest quadrant, a score of 4 is given, 2nd heaviest is scored 3, and subsequently down to 1 for the lightest quadrant. The final score is calculated as:

$$SQM = 1 - \frac{\sum_{j=UL,UR,LL,LR} |q_j - v_j|}{8} \in [0, 1] \quad (25)$$

MEASURING RHYTHM Rhythm was measured by combining rhythm scores from three perspectives - horizontal, vertical and area. Rhythm measures how regular patterns of changes in the visual elements in terms of arrangement, dimension, number and form of elements. In order to compute the rhythm components, the following needs to be calculated first:

$$X_j = \sum_i^{n_j} |x_{ij} - x_c| \quad (26)$$

$$Y_j = \sum_i^{n_j} |y_{ij} - y_c| \quad (27)$$

$$A_j = \sum_i^{n_j} a_{ij} \quad (28)$$

where $j = UL, UR, LL, LR$. Similar to the measures of equilibrium, (x_i, y_i) is the center of the visual object, i and x_c, y_c is the center of the frame. The total number of objects are n_j

These scores are then used to compute the three components of rhythm:

$$\begin{aligned} & |X'_{UL} - X'_{UR}| + |X'_{UL} - X'_{LR}| \\ & + |X'_{UL} - X'_{LL}| + |X'_{UR} - X'_{LR}| \\ & + |X'_{UR} - X'_{LL}| + |X'_{LR} - X'_{LL}| \\ \text{RHM}_x = & \frac{\quad}{6} \end{aligned} \quad (29)$$

$$\begin{aligned} & |Y'_{UL} - Y'_{UR}| + |Y'_{UL} - Y'_{LR}| \\ & + |Y'_{UL} - Y'_{LL}| + |Y'_{UR} - Y'_{LR}| \\ & + |Y'_{UR} - Y'_{LL}| + |Y'_{LR} - Y'_{LL}| \\ \text{RHM}_y = & \frac{\quad}{6} \end{aligned} \quad (30)$$

$$\begin{aligned} & |A'_{UL} - A'_{UR}| + |A'_{UL} - A'_{LR}| \\ & + |A'_{UL} - A'_{LL}| + |A'_{UR} - A'_{LR}| \\ & + |A'_{UR} - A'_{LL}| + |A'_{LR} - A'_{LL}| \\ \text{RHM}_{\text{area}} = & \frac{\quad}{6} \end{aligned} \quad (31)$$

Rhythm is then finally measured according to:

$$\text{RHM} = 1 - \frac{|\text{RHM}_x| + |\text{RHM}_y| + |\text{RHM}_{\text{area}}|}{3} \in [0, 1] \quad (32)$$

MEASURING COHESION Cohesion is a measure of how regular aspect ratios of visual objects are throughout the frame. This was computed by investigating the aspect ratio of two factors: layout and frame. While the frame indicates the entire interface, the layout indicates the bounds which encapsulate all of the visual objects that are present in the frame. CM_{fl} is initially calculated for the relative measure of the ratios of the layout and screen:

$$\text{CM}_{fl} = \text{c}_{if} \text{ if } \text{c}_{if} \leq 1, \text{ else } \frac{1}{\text{c}_n} \quad (33)$$

$$\text{c}_{fl} = \frac{h_{\text{layout}}/b_{\text{layout}}}{h_{\text{frame}}/b_{\text{frame}}} \quad (34)$$

$$(35)$$

CM_{lo} is then computed to find the relative measure of the ratios of the objects and layout:

$$\text{CM}_{lo} = \frac{\sum_i^n f_i}{n} \quad (36)$$

$$f_i = t_i \text{ if } t_i \leq 1, \text{ else } \frac{1}{t_i} \quad (37)$$

$$t_i = \frac{h_i/b_i}{h_{\text{layout}}/b_{\text{layout}}} \quad (38)$$

The final score for cohesion is calculated by

$$\text{CM} = \frac{|\text{CM}_{fl}| + |\text{CM}_{lo}|}{2} \in [0, 1] \quad (39)$$

where $h_{\text{layout}}, b_{\text{layout}}$ is the height and width of the layout (within the bounding box of all the visual elements), while $h_{\text{frame}}, b_{\text{frame}}$ is the height and width of the entire frame.

MEASURING UNITY Unity signifies coherence, where visual elements appear to belong together, as one thing – similar sized objects, using less free space between objects, larger margin etc. contribute toward a high unity score. Unity is computed by initially computing the unity of form (U_{form}), which indicates the extent the elements relate with respect to size. This is done by the following formula:

$$U_{\text{form}} = 1 - \frac{n_{\text{size}} - 1}{n} \quad (40)$$

Unity of space (U_{space}) is then computed, which indicates the space left by the margin area of the interface and the space between visual objects. This is calculated by the following formula:

$$U_{\text{space}} = 1 - \frac{a_{\text{layout}} - \sum_i^n a_i}{a_{\text{frame}} - \sum_i^n a_i} \quad (41)$$

a_i, a_{layout} and a_{frame} indicate the area of a given object, i , area of the layout and the frame. n_{size} indicates the number of sizes used and n , the number of objects in the frame.

MEASURING HOMOGENEITY Homogeneity measures the evenness of the distribution of visual objects among the quadrants. It is calculated initially by measuring the number of ways a group of the visual objects (n) can be arranged in the four quadrants:

$$W = \frac{n!}{n_{\text{UL}}!n_{\text{UR}}!n_{\text{LL}}!n_{\text{LR}}!} \quad (42)$$

$n_{\text{UL}}, n_{\text{UR}}, n_{\text{LL}}, n_{\text{LR}}$ are the number of objects on the different quadrants and n is the total number of objects. The next score is calculated which indicates the maximum score W can possibly attain, by equally distributing the objects in different quadrants:

$$W_{\text{max}} = \frac{n!}{(\frac{n}{4}!)^4} \quad (43)$$

With the two scores (W, W_{max}) computed, the final homogeneity measure is calculated by:

$$\text{HM} = \frac{W}{W_{\text{max}}} \in [0, 1] \quad (44)$$

MEASURING ORDER AND COMPLEXITY Order and Complexity is the measure that brings all the scores together, as a final score that can then be computed to ascertain the overall aesthetic value of the interface. This is computed by averaging all the other measures, hence

$$OM = \frac{\sum_i^C a_i M_i}{C} \in [0, 1] \quad (45)$$

where C is the complexity of the layout, interpreted as the number of measures considered in the aesthetic evaluation (here, 8). a_i is the weight associated with each measure. This is presently set as a constant, 1. Ngo notes that the weights can be fine tuned by using more sophisticated techniques such as genetic algorithms.

The details of how these factors can be measured can also be found in Ngo's papers [NSA00, NTB03]. This illustration was provided to describe how they were computed in this evaluation, clarifying certain aspects such as terminology, methods, layout measures etc. that posed some challenges while conducting and setting up the evaluations.

6.3.2 Results

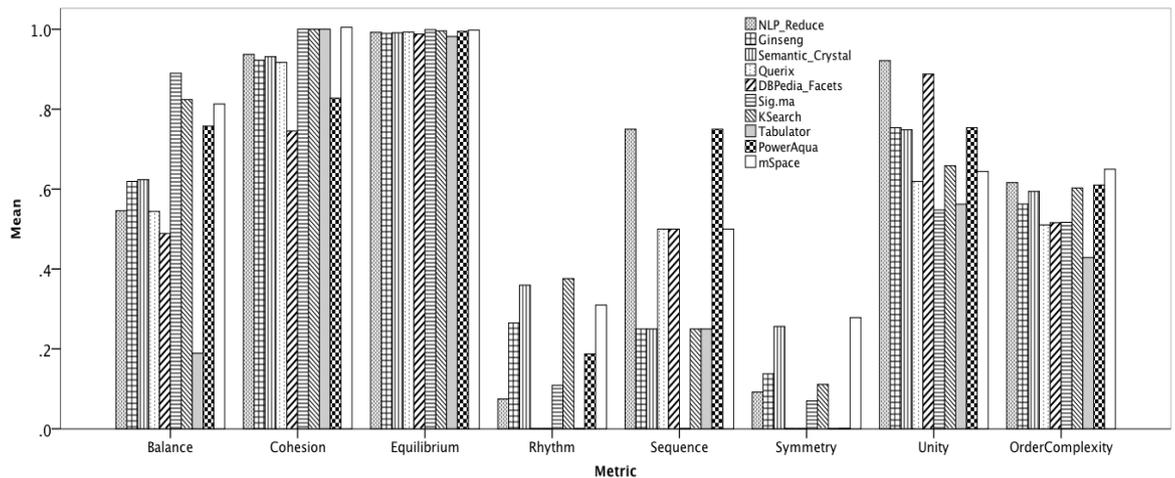


Figure 52: Comparative evaluation of seven aesthetic metrics with ten Semantic Web tools. The eighth metric, Order and Complexity is computed as a sum of the others

The scores were plotted as shown in the Figure 52. As can be seen, most of the interfaces performed well on equilibrium. Cohesion scores for DBPedia Facet and PowerAqua were comparatively lower, though all systems scored relatively high. Sig.ma was observed to be the most balanced system, followed by K-Search and mSpace. In general, all systems scored relatively low in Rhythm and Symmetry. Querix, DBpedia Facets and Tabulator scored the least in Rhythm, while Querix, DBpedia Facets, Tabulator and PowerAqua scored the least in Symmetry.

Overall, the best performing tool was mSpace, with a Order and Complexity score of 0.65, followed by NLP Reduce and PowerAqua with scores of 0.62 and 0.61. The

lowest scoring tool was Tabulator, with a score of 0.43. It can be observed that all the interfaces scored between low to medium, in terms of their overall aesthetic properties. This provided the starting point for the research, where the first step was to identify the need to investigate aesthetics and design tools with an explicit attention to aesthetics. It also demonstrated that out of ten well known Semantic Web tools, most interfaces do not score highly on aesthetic properties related to screen design and object positioning. Few other factors such as overall color palette, typography and iconography also have significant impact on aesthetics [Tuf86]. While these factors may be easily evaluated from a subjective point of view, objective analysis in a comparative setting is difficult. This is particularly due to the relative absence of benchmark scores to ascertain the relative positive or negative scores of interfaces in comparison to other.

6.3.3 *Limitations*

The most significant limitation of this evaluation is the focus on interface layouts – other factors such as colour, texture, shape and so on also need to be considered in order to comprehensively assess aesthetic properties of system. However, as Ngo anticipated, this task is significantly more difficult as it introduces far more variability in order to be considered as an objective analysis. Furthermore, this evaluation is only considered as a preliminary study on the attention to aesthetics in the design of existing Semantic Web interfaces. Restricting the evaluation to ten interfaces is also another limitation – a larger study that covered a lot more interfaces would provide a more representative study of all the interfaces that have been designed. However, at the time of this evaluation, such a study had not been conducted in the semantic web, and hence, it was considered most appropriate to conduct a preliminary comparative study with some well known interfaces. A further evaluation at a later stage with a larger number of interfaces and greater number of criteria could be an excellent continuation of this evaluation.

6.4 EXPLORING AND QUERYING SEMANTIC DATA — APPROACH

As it could be expected, representing Linked Data in an abstract way leads to graph representation as Semantic Data is essentially multiple data instances connected with links to constitute a highly connected and directional graph. Hence, the starting point for the proposed design is a node-link graph, where nodes represent concepts and links represent relations the concepts share. Node-link graphs, large ones particularly, are notoriously difficult structures to handle and understand. The challenge here is to present large sets of data, but preserving their links and hierarchical structure. The following requirements are in addition to the ones identified in Section 4.2.

The most important requirement was to facilitate the user to **explore** unknown (and known) datasets. In addition, users must be able to **query** for specific information using a visual approach, which is one of the key requirements following the evaluation of *views*. An important requirement for the design was to put a lot of emphasis on the **aesthetic quality** of the interface. The need for an explicit focus on aesthetics was identified as a result of users appreciating a highly visual and interactive interface, using a well defined structure and colourful visualisations. The approach had to be a **generic**

one, in order to ensure any Semantic dataset can be consumed. The proposed solution is divided into the following four major functions to address these requirements:

1. **Making the underlying schema of data apparent to users; highlighting further details such as context, relations and statistics.** Experiments and focus groups with users highlighted the need for a statistical representation of the data that is available. While ontologies provide a formal specification of the domain, the data itself is what the users are mostly interested in. Additionally, showing the statistics of the data would reduce cases where users would be creating complex queries only to find information that is not present in the dataset. Visualising the ontology(schema) as well as the data at the same time would be an ideal way to communicate the presence of information as well as how information is related.
2. **Support data driven exploration via statistics by making use of standard statistical presentation techniques.** Visualising entire ontologies and data instances can be a useful way of presenting the data as well as the domain, however at the expense of increasing cognitive burden and exhausting screen space — this generates a need for users to access concepts and their data ‘on interest’, rather than showing everything that is available.
3. **Provide access to individual data instances at all times.** While the initial interest was in presenting statistics with ontological concepts and properties, previous discussions with users and Semantic Web experts throughout our iterative design process resulted in the need for ways to provide easy access to data — this also supports an exploratory browsing paradigm, where users can always reach data instances when needed.
4. **Support highly specific Information Needs by introducing flexible user interactions.** The beauty of Semantic Web is in making available high quality, dynamic and precise information that is highly inter-connected: such information can be exploited from interfaces that can support building complex queries to precisely answer highly specific questions.

The identified features were aligned with the design principles listed in the previous section. Following analysis of the design principles, low-fidelity mock ups were built in order to understand how users would interact with visualisations in an intuitive manner.

6.5 AFFECTIVE GRAPHS — A SCENARIO OF USE

Figure 53 shows a screenshot of the final implemented system, *Affective Graphs*. The system was built as a result of several re-designs and prototypes, with constant inputs from users and evaluation feedback throughout the implementation. Section marked A shows the interactive node-link representation of the underlying data. The image shows a user exploring the latest DBpedia³ dataset, presently viewing lakes (the node on focus is at the bottom right corner). Users can gain a large amount of information

³ <http://dbpedia.org>, as on 20.05.2013

The screenshot displays the AffectiveGraphs interface with the following sections:

- Section A:** A node-link graph where the central node is 'lake(41400)'. It is connected to 'body of water(136032)', 'natural place(218240)', 'place(2580236)', 'agent(4126616)', and 'person(3270004)'. The 'place(2580236)' node is further connected to 'Thing', 'birthplace 3075036 resources', and 'agent(4126616)'. The 'agent(4126616)' node is connected to 'person(3270004)'. The 'person(3270004)' node is connected to 'birthplace 3075036 resources'.
- Section B:** A panel titled 'lake' showing 'Query Constraints' and 'Properties'. The properties listed are:
 - inflow (11352) --> River
 - outflow (11108) --> River
 - shoreLength (4884) --> kilometre
 - volume (4364) --> cubicMetre
 - areaOfCatchment (3476) --> double
 - > squareKilometre
 - Island (1044) --> Island
 - frozen (612) --> string
- Section C:** A control panel with 'Visual Query' tabs, 'Search for concept', 'Highlight Property', 'Freeze Nodes', and 'Hide Data Properties'.
- Section D:** A text area for the SPARQL query.
- Section E:** A panel for query parameters including 'Fields to return:', 'Query Parameters:', 'Limit: 1000', 'Offset: 0', 'Order by: select', and 'Type: ASC'. It also includes 'Search' and 'Clear Query' buttons.

Figure 53: A screenshot of *AffectiveGraphs*, where the node currently on focus is 'Lake': Section A contains the interactive node-link representation of the data, Section B contains contextual information relevant to the concept currently being explored (here, Lake), Section C contains search elements and controls the visual rendering of the node-link graph, Section D shows the SPARQL query being generated for search, Section E contains advanced features to modify the query.

just by observation and interacting with the visualisation. The following list presents some information that a user can quickly find while interacting with the system:

- The dataset contains information regarding places, persons, work, species, organisations, transportation, films and so on. This is observed by selecting the ‘owl:Thing’⁴ node and hovering over the pie sections.
- The subclass hierarchy is immediately apparent to the users — lakes are types of bodies of water, which are natural places and persons are types of agents etc. This is observed by following the triangle-shaped hierarchical edges. The colour of the edges indicate the respective pie-sections they originated from. The base of the triangle indicates the parent node, and the apex indicates the child node, essentially representing an “arrow” pointing toward the lower hierarchical concept.
- There are 41k lakes, 2.5m places, 3.27m persons, 218k natural places in the dataset etc. This can be found from the labels of the nodes as well as tooltips provided on pie sections.
- The amount of information on agents is the most, followed by places (hovering over the pie sections reveal the subclasses as agent, place, work, species etc. in the order of instance counts). This can be observed from the positioning and size of the pie chart sections — the subclasses with the greatest number of instances are the biggest in size as well as positioned at the bottom left of the pie charts. The sections are organised in a descending order of number of instances, following a clockwise direction.
- Lakes is a “leaf-node”, indicating that there are no subclasses of the concept found in the dataset. Leaf nodes are represented as circular objects containing no distribution of subclasses.
- The information regarding the birthplace of 3.07m persons are available in the dataset. This is found by hovering over the ‘birthplace’ relation connecting persons and places.
- There are five relations in persons that are linked to the same concept — *parent* (87k instances), *spouse* (100k instances), *influencedBy* (99k instances), *influenced* (49k instances) and *child* (43k instances). This can be seen as four curved loops originating from and ending in the person node — hovering on the edges reveal the relations and the number of instances.
- Three data properties exist in the lakes concept — *areaOfCatchment* (3.4k instances), *frozen* (612 instances) and *shoreLength* (4.8k instances). This is shown as three edges originating out from the Lake, hovering over the edges show the relation and the number of instances. Similarly, the data properties for person and place can also be investigated.
- Three object properties connect persons with places — *death place* (799k instances), *birth place* (3.07m instances) and *resting place* (52k instances). This is found by hovering over the three curves joining the person node with place node.

⁴ The first node that users can see is the *owl:Thing* node that encompasses all the data classes described within the dataset.

- The distribution of all the subclasses of Natural Places (Body of water, Mountain, Mountain range, Lunar crater, Cave), Places (Populated place, Architectural structure, Natural place, Protected area etc.), Body of Water (Stream, Lake), Agents (Person, Organisation) and Persons (Athlete, Artist, Officeholder, Politician etc.) can also be investigated. This can be done by selecting the respective nodes and hovering over the different pie sections.

In summary, a very high level understanding of 2.5m places, 218k natural places, 136k bodies of water, 4.1m agents and 3.27m persons as well as the broader content of the entire dataset can be very quickly gathered by observing the visualisation and interacting with it. Not only is the hierarchical structure of the dataset made apparent, the links and properties shared by concepts are also made available to the user.

Section B shows the contextual information of the node presently in focus (here, Lake — indicated by a larger font and more prominent node). By investigating the content on the right panel, users can understand that the data properties are shoreLength, areaOf-Catchment and frozen (data properties are coloured blue, and object properties are green following the standards set by Protégé). In addition, the number of instances connected with these relations are presented. The object properties are also presented, along with the other concepts the relations are connected to. Clicking on the concepts on the panel trigger new nodes to be formed, and the relations to be visualised.

Section C provides a mechanism for users to search for a specific concept, if the user prefers to search for any known concepts. Once a required concept has been found, a node is then added to the visualisation (without the user having to manually select pie sections). In addition to a concept search box, a property search box is also provided in this section — users can start typing a property name (or [URI](#)) that they are interested in, and the interface highlights the property of interest in the visualisation. This is an easy way of quickly spotting any property that the users are interested in. There are three other controls provided in the section C which control the force-directed layout and toggle the visibility of the object and data properties.

Section E provides advanced features for customising the [SPARQL](#) query formed while performing a search — such as, selecting the variables to be returned, limiting search results and so on. Section D displays the final [SPARQL](#) query formed, in case the user wishes to edit the query before searching. This is a requirement that is most often desired by expert users, who prefer fine-tuning their queries after having created a basic query.

6.6 DESIGN RATIONALE

Directly abstracting Linked Data leads to a node-link representation: this was the starting point of the design. The final interface was developed as a result of a set of several design decisions. The first phase was to understand how to design a node-link representation that can provide additional information about the underlying data as well as adhere to the aesthetic Principles defined in Section 6.2.

6.6.1 Consistency in Visual Representations

The first step was to develop a **consistent** representation for concepts that provided more information about the data in the concept. In the design, concepts are represented as circular nodes, while the properties have been represented as edges. In order to present information about a concept, the circular design of the nodes have been modified to show a pie-chart of the distribution of the subclasses of the concept. The pie sections are sensitive — clicking on each section triggers a new concept node to be created, which contains information about the respective subclass. Pie charts visualise data in a small area and provide a wealth of information that, if displayed as a graph would be confusing and difficult to grasp. The new node created is similar where distribution of its subclasses are rendered as another pie chart on the new node (Principle 5, 9 suggest consistent visual representations). If the new concept contains no subclasses, the node would be represented as a circular blank node (without a pie chart). The new node is connected with the originating node using a hierarchical edge that signifies a *rdfs:subClassOf* relation.

This design decision was on visualising the distribution of the subclasses — a pie chart was chosen as the most preferred representation due to three main factors: users are familiar with pie charts and can quickly assess the relative amount of data each section contains; pie-charts are able to convey statistical information within a small and regular area better than a table specially when conserving space is important (Principle 19, 7 suggest a minimalist approach); a circular depiction of nodes in a node-link graph is an organic geometric representation that most users are familiar with.

6.6.2 Representing Semantic Concepts

The pie-chart **representation** itself provided the next design challenge — the aim was to provide a pie-chart with regions easily distinguishable from one another. The use of colours and textures in order to distinguish pie sections is a standard process — however, the task is further complicated as there are multiple pie charts in the layout. Initial efforts at using a standard bank of 20 unique manually selected colours were unhelpful and caused confusion in the users as standard colours seemed to indicate certain commonality among the similar coloured-sections. Furthermore, a repetitive standard colour palette would reduce the variety in the design (Principle 8 suggests introducing variety in the design). The system was then changed to generate random colours in a HSB (or HSV) scale, varying only the hue values to keep the saturation and brightness consistent as well as provide a greater range of colours. The HSB scale was preferred to an RGB scale as the former provides greater flexibility in varying colours (here, hue) by keeping the saturation and brightness constant, which is not as easily achievable in the latter. Moreover, HSB scale is more natural and user friendly way since it replicates the way we “observe” colours⁵. Figure 54 explains this with a simple example — the figure on the left was produced from a set of randomly generated HSB colours with brightness and saturation fixed. Though the different regions in the two pie charts are easily distinguishable, the RGB pie chart contains sections of unequal brightness. RGB

⁵ <http://tools.medialab.sciences-po.fr/iwanthue/theory.php>

colours can also achieve the same results as the HSB pie chart, but with more complex methods.

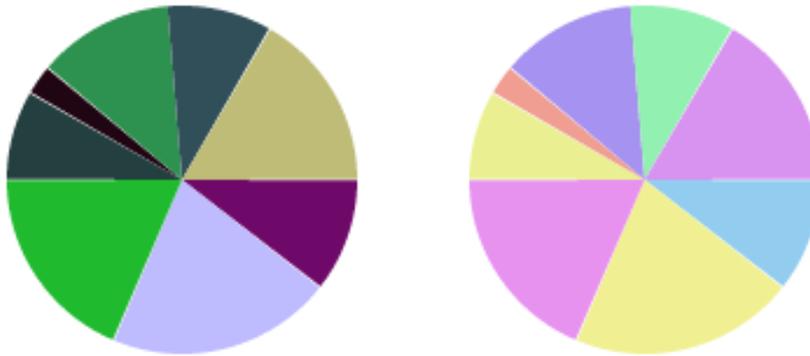


Figure 54: Comparison of randomly generated pie-chart colours for RGB (left) and HSB (right). Though the colours on the right are completely random, constant values of brightness and saturation generate a more pleasing chart. Images created in <http://sketch.processing.org/>

Though randomly generated, there is often a high chance of creating duplicate colours (that may have different colour codes, but are nearly indistinguishable by the human eye). Similar colours can also be wrongly interpreted as the same (or similar) concepts. This was significantly reduced by generating random colours that are different (based on a threshold) from the set of colours already generated by providing a look-up service for the set of colours already generated. The pie sections are further distinguished with the use of borders and interactive events (such as the respective sections are extended when the user hovers over them)

6.6.3 Representing Semantic Relations

The third design challenge was the **representation of relations** (properties) — an important consideration was the different types of relations that may exist within the data set. Three types of relations were identified — data properties, object properties and hierarchical properties. Hierarchical properties are a kind of object properties as they essentially describe the relations between a parent object and its child. However, in this representation hierarchical properties have been isolated from object properties (Principle 11 suggests separating hierarchical relations from others). Data properties are presented as free edges — where an edge is connected to the node at one end, and a circular object at the other. The positioning of data properties is random, but ensuring that the edges are not overlapped (Principle 15 suggests minimal overlaps of edges). The circular object is the sensitive end of an edge, providing interactions with the users (hovering on the object highlights the edge and provides a tool-tip display, clicking activates query mechanisms etc). The different edge representations are illustrated in the Figure 55

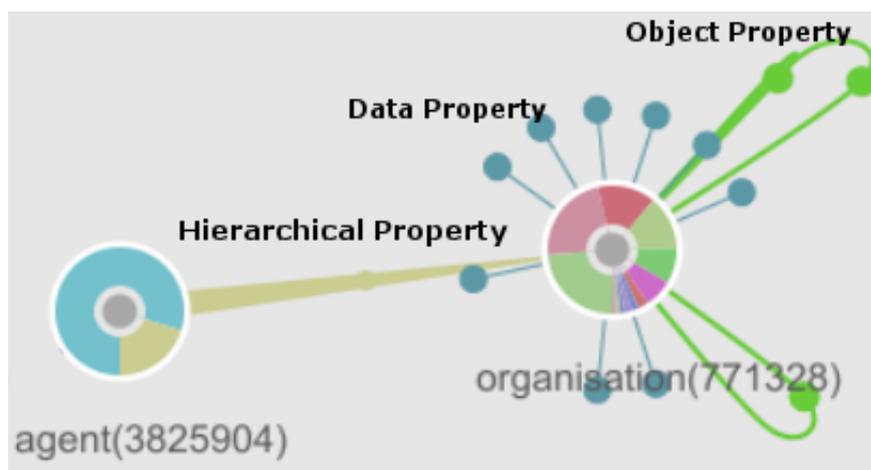


Figure 55: Different visual representations of edges bear different ontological significance — hierarchical edges are represented as triangles showing the parent and child node, data properties are represented as satellite objects and object properties are represented as curves

Object properties are represented as curves — this is due to the preference of users to curves as opposed to straight lines [BNo6, BW98] and also the lack of acute angles and edge bends (Principle 15 suggests avoiding edge bends). Cubic Bézier curves were chosen to render the edges as they provide a greater control and flexibility [Yam88]. Additionally, the presence of straight lines as connections between objects can make it difficult to follow links. Figure 56 (i) shows an example scenario where the same nodes have been connected using straight edges (left) and curved edges (right). Interactive mechanisms on the curved edges also make it easier for users to follow how the nodes are linked in case of larger graphs. Hovering on the sensitive sections of the edges highlight and increase their thickness, making them easier to spot as shown in Figure 56 (ii) (Principle 4 suggests the use of lines with different thickness to add variety; Principle 6 suggests the use of variations in colour to help users spot elements). The edges of the node currently in focus are also shaded in a darker colour to help users find the edges connected to the current node (Principle 6 suggests using variety in colours).

The representations of hierarchical and non-hierarchical relations are different, as each can be used as a different interaction mechanism (Principle 11 suggests a different representation of the two). While non-hierarchical relations describe the different properties of objects (such as birthplace, age, date of birth etc.), hierarchical relations contribute more toward explaining how the data is structured. Hence, the design decision was to provide edges with greater weights for representing hierarchical relations. Furthermore, in order to identify the children of a node, the hierarchical edges are represented as triangles signifying directionality, as shown in Figure 55. Inspired from the design of Protégé⁶, the data properties are coloured blue while the object properties are coloured green (Principle 5 and 9 suggest maintaining consistency and such colour schemes are well established within the Semantic Web community). This design helps

⁶ Protégé is one of the most widely used frameworks for modelling ontologies and knowledge systems, with a wide user community and <http://protege.stanford.edu/>

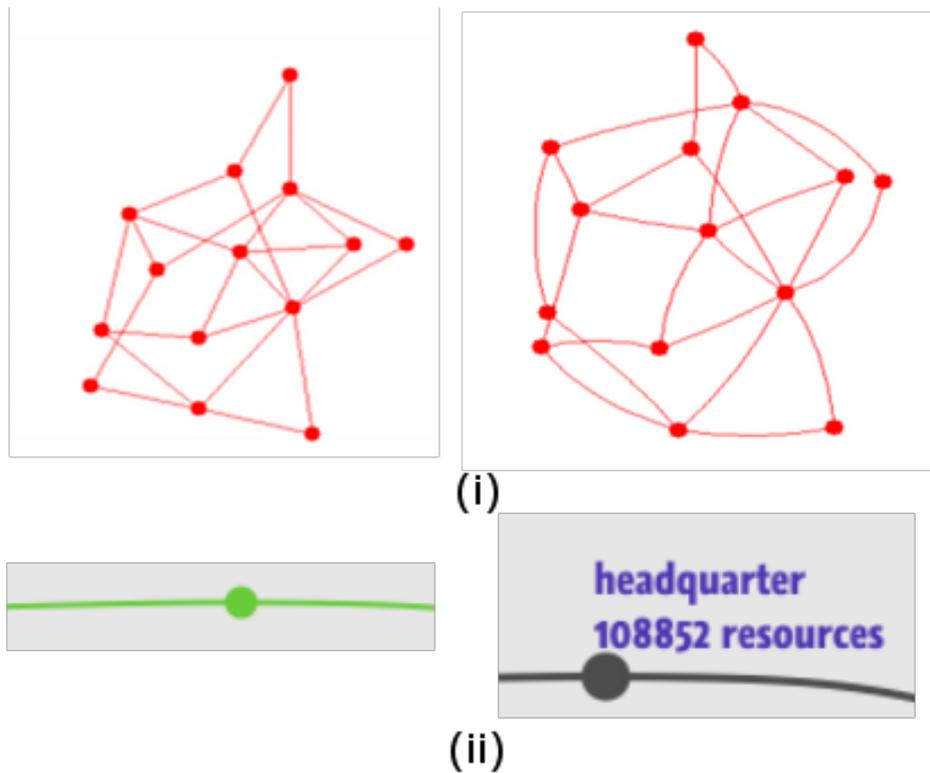


Figure 56: (i) Comparison of Straight Edges with curved Edges. Image from [FT04], (ii) Visual response to hovering over an object property

users quickly identify which edges are data and object relations, as well as hierarchical edges (Principle 4 suggests using differently weighted lines to signify different meaning and principle 6 suggests varying visual properties to enable users to quickly spot features).

6.6.4 Layout

The next design challenge was **layout** of the node-link graph. Initial attempts at automated layouts helped in balancing the representation effectively — nodes were arranged in a force-directed layout, based on the Processing simulation library, developed by Bernstein⁷. However, as the number of nodes increased, the graph grew far more compact than required. Furthermore, the force-directed layout made users disoriented and complicated the interaction as objects kept floating around the graphical space to optimally position them. Two changes were then made for the final design — a change in the layout algorithm and a change in the spring configurations.

The force-directed layout is active only during the initial 5 seconds of a node being created — this allows for enough time for the node to be rendered while positioning itself in an approximate enough position for a good layout. However, the user has the

⁷ Jeffrey Traer Bernstein's physics simulation library is a standard library available for use with Processingjs applications, <http://murderandcreate.com/physics/>

ability to click-and-drag the node to wherever they desire, without being restricted by the force-direction (Principle 9 suggests maintaining consistency in interaction mechanisms — this helps users learn the system more quickly as this is a familiar interaction present in several graph visualisations; Principle 8 suggests a balance in typicality — while representation of nodes as pie charts and introduction of several types of edges with different semantics is a novel addition, familiar interaction mechanisms help users adjust to a new setting). This provides flexibility and freedom to the users as they can layout their graphical exploration in any manner they please, but at the same time provides a rough approximation for a newly created node to ensure readability is maintained. The users however, have an option to disable the feature, which would cause the force-direction to be active at all times.

The second change involves modifying the contribution of different types of edges toward the force-direction as well as the overall display of the graph. In the final design, the hierarchical edges (subclass relations) are made the only type of edges that contribute directly toward the layout. This makes the layout more balanced and well-spread, instead of a highly compressed layout due to object properties exerting forces between many more nodes (Principle 12 suggests minimal overlapping nodes — the nature of the graph being relatively more spread-out ensures that the minimal number of nodes are overlapped; The balance provided by the hierarchical and non-hierarchical edges in contributing toward the layout ensures that Principle 19 is adhered to, without compromising on readability). The non-hierarchical edges do not contribute directly toward the layout, and are just links that visually connect concepts and interact with users.

6.6.5 *Designing Interactions*

Another design challenge was to introduce **interactions** in the visualisation. Initial prototypes were designed to make it quicker for the user to perform functions. For example, the nodes supported right-clicks to add concepts to queries, using a control key and right clicking on nodes would hide them and so on. However, this caused confusion among users, resulting in frustration and requiring assistance constantly. A re-design of the interface introduced similar interaction mechanisms as seen in other graph visualisation tools and interfaces such as Google maps (Principle 8 and 9 recommends using some familiar solutions to reduce the effort required for users in learning the tool). The interface allows drag and drop actions to reposition nodes as users prefer. Right-click on nodes and edges (Figure 60) provide users with context menu, enabling them to add concepts or relations to queries, hide nodes and so on.

6.6.6 *Designing an Integrated solution*

The final design challenge was to incorporate the visualisation into a complete interface, where all the visual elements are in **unison**. User studies and Visual Analytic principles such as Table 1 showed the need for an interface that provides features for advanced users. In addition, the need for representing the underlying formal query was also raised. In order to balance the layout, the two new visual elements (advanced and formal query display) were positioned below the graph interface. A contextual display

that provides more information about the current topic of exploration is also placed on the right. The positioning of various visual objects have been made to provide a well-balanced and symmetric layout. Consideration was also made to arrange the objects based on a sequence — objects should be positioned in a layout that facilitates movements of the eye (most readers start reading from the top left and move to the bottom right) [NSA00, NTB03].

Every subsequent iteration in the UCD methodology resulted in user evaluations, focus groups or interview sessions, incrementally providing insights to help build a continuously improving version of the system. Several users from different communities such as academia, research, aerospace engineering and Knowledge Management had been involved in the process, and their comments, feedback and suggestions were extremely helpful in developing the final design of the interface.

6.7 ARCHITECTURE

The basic architecture of Affective Graphs fits with the general solution architecture as shown in Figure 31, where the two sub-solutions were presented. The two sections, querying semantic data (top) and visualising semantic data (bottom) are used in conjunction to provide the bridge between users and their data. The logical architecture of Affective Graphs (Figure 57) is an extension of both the visualisation and querying sections of the figure, expanding both on ‘Query Builder’ and ‘Result Interpreter’ modules.

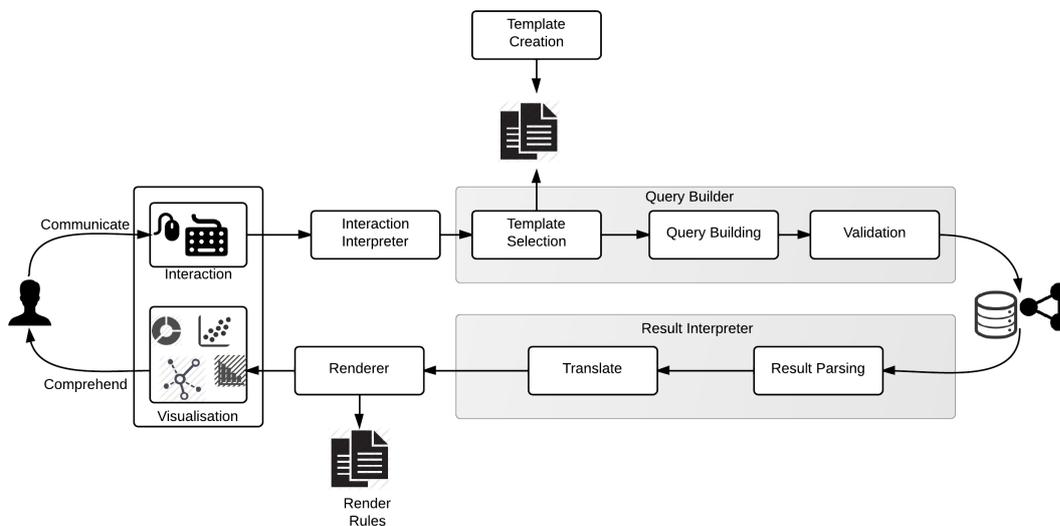


Figure 57: Architecture

It can be observed here that in comparison to the architecture of *.views*. (where Query Builder was represented as a single module), the Query Builder has several stages of processing involved. Owing to the greater complexity of all the involved processes, the module is expanded into constituent sub processes. In contrast to *.views*. (where the prime focus of the approach is a strong visualisation paradigm), the Result Interpreter

lacks an additional Interpretation stage, as in Affective Graphs, there is no need to abstract physical entities from result sets to determine which visualisation should be presenting a result. Hence, the Result Interpreter module is slightly less complex as compared to *.views.*. As with *.views.*, an interaction cycle with the user initiates a set of queries, which are then transferred to the semantic store. The result sets retrieved from the semantic store are then parsed, and ultimately are translated into visual representations, which are observed by the end users. Since each of the sub processes constitute a rather complex execution, they are identified as logical building blocks of the final solution.

The Interaction Interpreter interprets user actions and gestures to identify what a user's intent was and which visual item was the subject of the action. Based on this information, the Template Selection stage determines a query template. This is done by using a pre-built dictionary that maps several interactions on objects to an example template query. For e.g., a right-click on a concept followed by a "add-to-query" action is interpreted as a query for instances of the concept — hence, the respective query template would be selected. The selected query template is then passed to the Query Building stage, which converts the template to a formalised query — this involves joining the query with previous sub-queries (if the need arises, e.g. when a user selects multiple concepts and properties to join them). This stage also involves creation of local variables to be used in the formal query. The next stage, Validation has two main functions: rationalising variable names, and ensuring that the query is well-formed (parenthesis are matched, query triples and filters are well formed etc.). The outcome of the validation is a formal query, ready to be executed in triple stores.

Results from the triple store are then passed to the Result Interpreter. The primary function of Result Interpreter is to build data objects that can be interpreted by the Renderer, to visualise concepts and their distribution within a graphical framework. The first stage in the Result Interpreter is a Result Parser, which is responsible for parsing the result sets and understanding the distribution of data. The next stage is a Translator, that converts this information into data structures that can be passed to the Renderer. Based on basic rules, the renderer then converts the data structures into visual items.

From a technical implementation standpoint, the system is composed of two sub systems: the front end (right block, Figure 58) provides the visualisation, interactions, advanced controls and filters, the backend (left block, Figure 58) deals solely with querying endpoints using SPARQL generated during the previous processes. Every user-interaction with the various frontend modules results in SPARQL queries being generated in the SPARQL generator, which interprets these actions based on the methods discussed later in this section. The queries are then transferred to the PHP backend. The Query Engine in the backend interprets these queries and make any modifications in the query such as variable names, adding prefixes etc. The Data selector then checks a local cache to see if the same query had been used previously. Owing to the unpredictability (and at times, unavailability) of SPARQL endpoints as discovered in the previous chapter, a local cache (a MySQL database) was built that stores the responses of the queries that have been used to query the endpoint. This was a step taken to address the issue of unpredictable behaviour of endpoints, as highlighted during the evaluation of *.views.*

If the query's response had been previously recorded, and if the data provided by the endpoint has not been updated since, the previous response is gathered from the

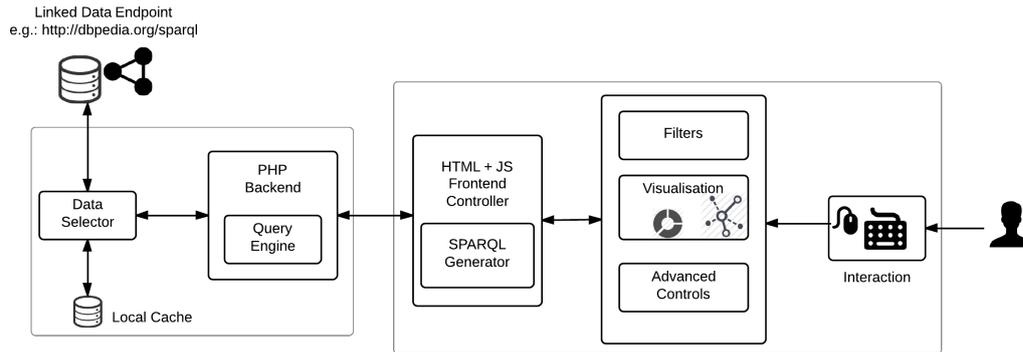


Figure 58: Architecture

cache. On the unavailability of any cached result for the query, the public endpoint is queried and the result is stored in the cache to be fetched at a later stage. The result is then converted to a [JSON](#) object and returned to the front end. The frontend interprets the object and renders the data into the visualisation as required.

The first step in implementing the solution is to understand how to construct automated queries based on user interactions. Users interact with visual objects in the web interface, triggering calls to the server, passing the [URIs](#) of the entities they represent; the server constructs the queries from templates. This process is slightly more complicated, as [SPARQL](#) query requires the usage of variables. This is obtained by continuously maintaining a catalogue of the entities being queried for and constructing variables built out of the [URIs](#). For every node being built (as well as during initialisation), the interface sends three requests to the server:

- A *subclass request* to list all the subclasses of the concept along with the respective counts of the number of instances within the domain
- A *domain property request* to list all the properties (along with number of instances) that have the concept as its domain.
- A *range property request* to list all the properties (along with number of instances) that have the concept as its range.

These requests are translated into formal [SPARQL](#) queries using templates such as the following subclass request query:

```
1. SELECT distinct ?subClass count (?x) as ?count ?label
2. WHERE {
3.   ?x a ?subClass.
4.   ?subClass rdfs:subClassOf dbp:Place.
5.   ?subClass rdfs:label ?label.
6.   FILTER langMatches( lang(?label), "EN" )
7. }order by desc(?count)
```

As can be seen from the query, the backend queries the endpoint for all the subclasses of a class that is currently being visualised (the 'Place' concept, as seen on line 4). The filter directs the endpoint to return only labels that are in english. The response from the query would then be converted into [JSON](#) in the backend, and then returned to the frontend. A sample response is as follows:

```
[{"subClass type":"uri",
 "subClass":"http://dbpedia.org/ontology/PopulatedPlace",
 "count type":"literal",
 "count":"360296",
 "count datatype":"http://www.w3.org/2001/XMLSchema#integer",
 "label type":"literal",
 "label":"populated place",
 "label lang":"en"},
 ...,
 {"subClass type":"uri",
 "subClass":"http://dbpedia.org/ontology/Monument",
 "count type":"literal",
 "count":"4",
 "count datatype":"http://www.w3.org/2001/XMLSchema#integer",
 "label type":"literal",
 "label":"monument",
 "label lang":"en"}]
```

The response provides the frontend with the subclass and the number of instances that are types of the subclass. This is then parsed by javascript and processing modules to build the pie chart and create the pie sections. Each pie section is built as an interactive element, listening to mouse gestures and responding accordingly.

Similarly, a sample range property request is as follows:

```
1. PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
2. SELECT distinct ?prop count(?instance)
   as ?count ?domain
3. WHERE {
4.   ?prop rdfs:range dbp:Stream.
5.   ?prop rdfs:domain ?domain.
6.   ?instance ?prop ?obj.
7. }order by desc (?count)
```

The query retrieves all the properties that have ‘Stream’ as its range (line 4). Along with the results, the query also requests for the domains of the properties as well as the number of instances. A similar response is returned for the domain and range property requests. The returned objects are properties and the number of instances. The properties are then classified into data type and object type properties.

The graphical space of *AffectiveGraphs* is a particle system, that simulates gravity, drag and apply forces between particles. Particles are objects that can move around within the particle system, based on the forces acting upon them. Spring forces and attractive forces act on the particles — springs ensure the connected particles are always maintained at a minimum distance from each other, while attraction can be positive or negative (repulsion). Unlike non-hierarchical relations, hierarchical ones are used as springs, thereby contributing toward the final layout. The semantic interpretation of a spring is a ‘rdfs:subClassOf’ (hierarchical) relation in our current implementation of *Affective Graphs*. However, this can be changed to any other relation that is deemed most appropriate for a particular dataset/task. A non hierarchical edge, on the other hand can have two semantic interpretations: an object edge that connects semantic concepts (like an object property connects two semantic concepts), represented by a green bezier curve between two nodes (edge C in Figure 59) or a loop that connects one node to itself (edge D in Figure 59); a data edge that emerges from a node and is represented by a blue straight line (edge B in Figure 59).

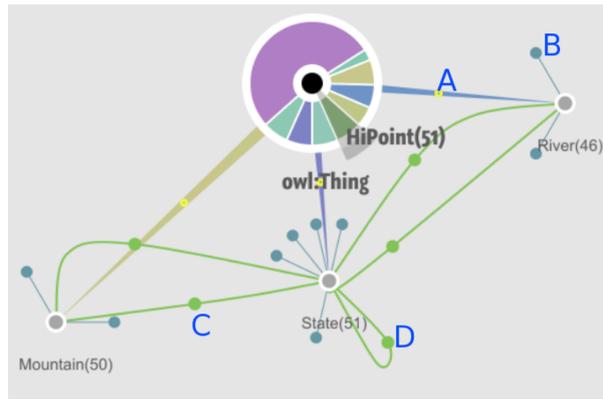


Figure 59: Different visual representations of nodes and edges bear different ontological significance

6.7.1 Query Mechanism

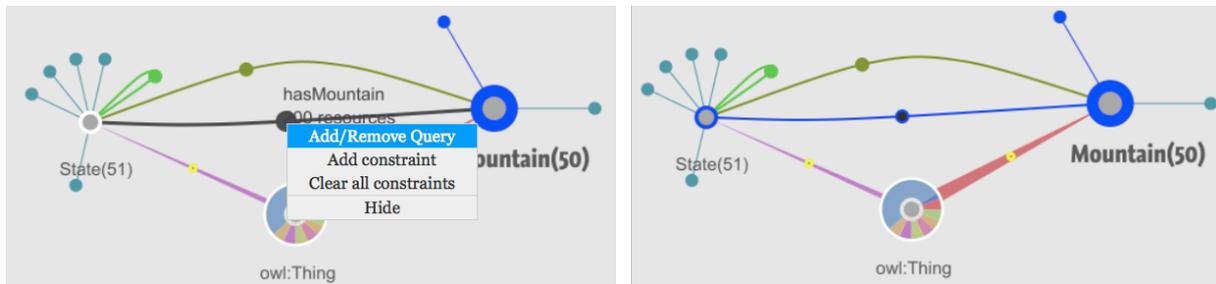


Figure 60: Visually building queries

The approach was to exploit the inherent feature in a semantic dataset where concepts are connected to themselves and other concepts with relations. Such a visual approach toward presenting, exploring and querying data stems from the belief that Semantic Web data is fundamentally highly visual and graphical and the approach toward consumption of such data could be more interactive by presenting to users the data as it was conceptualised by data providers at the time of creation. This makes construction of complex queries significantly easier — just by right clicking on nodes and edges and selecting ‘Add/Remove Query’ from the context menu to set/remove a query term, as shown in Figure 60.

The figure shows a screenshot of Affective Graphs configured to visualise the geographical dataset from the Mooney Natural Language Learning Data⁸. The left side of the image shows a user right-clicking on a non-weighted object edge (an object property, `hasMountain`) that connects `State` and `Mountain` to load the context-menu. Upon selecting ‘Add/Remove Query’ from the menu, the concepts `State` and `Mountain` are highlighted in blue as well as the edge `hasMountain`. The highlighting is a visual feedback that communicates that the system has accepted the user’s query and has built a

⁸ <http://www.cs.utexas.edu/users/ml/nldata.html>

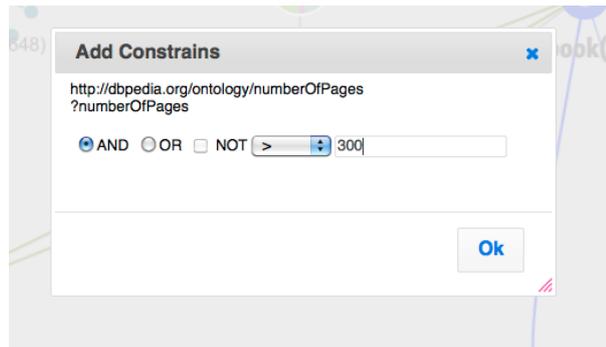


Figure 61: adding constraints

corresponding query. Right hand side of the Figure 60 shows Affective Graphs showing the query that was built.

If a concept was selected as a query item, then the interface highlights only the concept and interprets the action as the user is interested in looking at the instances that are types of the particular concept. Affective Graphs attempts to understand what concept or properties are being selected and associate internal query variables to the selections, partially building/re-building a query after every subsequent action. Each partial query is a small fraction of the final SPARQL query that represents what the user actions are interpreted as. The following is an example of a SPARQL query generated as a result of the user interactions as shown in Figure 60

```
@prefix mooney:<http://www.mooney.net/geo#>.
1. SELECT * WHERE {
2.   ?State a mooney:State.
3.   OPTIONAL { ?State rdfs:label ?State_Label}.
4.   ?Mountain a mooney:Mountain.
5.   OPTIONAL { ?Mountain rdfs:label ?Mountain_Label}.
6.   ?State mooney:hasMountain ?Mountain.
7.   ?Mountain mooney:height ?height.
8. }
```

The SPARQL query thus generated consists of two parts: partial query directly built by the user (lines 6 and 7) and partial query prepared by Affective Graphs (lines 2-5). While the user-built partial query is a direct representation of the selection made by the user, the partial queries are built out of the concepts that are selected — the final query looks for instances and their instances that are types of the selected concepts. These instances are ‘joined’ by the properties that are selected. Labels are used to render the results in a user-friendly way, separating the content from its formalised representation.

Often, users may have highly specific Information Need that they would like to query for such as the birth date or birth place of Elton John or a generic query such as the height of all mountains and rivers within a state that contains the character sequence ‘miss’ within their names. Queries such as these (FILTER queries) can be constructed in Affective Graphs using constraints — users can right click a node or concept that is set as a query and select ‘Add Constraint’ from the context menu, which would load a dialog that prompts for constraints. Figure 61 shows the user entering a constraint that sets the number of pages in a book authored by a writer to be greater than 300.

Users can also select if this constraint would be set as a negation constraint, as well as an OR query. The data type of property would dictate the type of constraint — if the data type is a string literal, then the filters being applied would be a regular expression filter. Instead, if the data type would be numeric, then comparisons would be possible. After a constraint is set, Affective Graphs would communicate the user of the new action by setting the respective query concept or property in a darker shade.

The following [SPARQL](#) query represents the interaction as shown in the figure:

```
@prefix dbont:<http://dbpedia.org/ontology/>.
1. SELECT * WHERE {
2.   ?writer a dbont:Writer.
3.   ?writer rdfs:label ?writer_Label.
4.   ?book a dbont:Book.
5.   ?book rdfs:label ?book_Label.
6.   ?writer dbont:notableWork ?book.
7.   ?writer dbont:birthName ?birthName.
8.   ?book dbont:numberOfPages ?numberOfPages.
9.   FILTER ( (?numberOfPages > 300))
10. }
11.
12. }
```

Setting concepts and properties as query items would result in the creation of partial queries on lines 2-10 and setting constraints would result in the creation of partial query on line 11. Multiple constraints would generate multiple lines of FILTER queries. Once the user has completed identifying the concepts and properties of interest, the relevant [SPARQL](#) query is displayed at the bottom of the screen (Section D, Figure 53). It can be often useful for users to configure their queries and only select concepts that they are interested in — e.g. though a query may contain multiple concepts and properties, it could be possible that in the end, a user is only interested in one particular concept and uses the rest as means of constructing logical joins to arrive at the resulting concept. Pagination and limiting result sets could also be a useful feature when dealing with large result sets. Users can click on the ‘Search’ button (Section E, Figure 53) to get results, which would be presented at the bottom of the screen.

The interpretation of the user actions and the subsequent translations to SPARQL queries are presented in more details in Section 6.10.

6.7.2 *User Interactions and Contextual Information Presentation*

Being a graphical interface, Affective Graphs supports mouse interactions such as left and right clicks, drag and hover as well as pre-configured keyboard short-cuts. Users navigate through the graphs by using conventional techniques such as left click on pie sections to create new nodes, hover on the sections to see concept labels, right click to load a context menu, drag nodes to reposition them to a more convenient location.

The right hand side of the Affective Graphs interface shows contextual information (i.e. Context Section) about the concept currently in focus, as well as the query being built (marked as section B in Figure 53). This context section is presented as an overlay on the graphical element, which can be easily hidden if the user wishes to. In addition to the constraints already applied in the query, this section presents a list of all the properties (object and data type) that are associated with the current concept in focus, along with indicating which other concepts are connected to the current concept via the object properties.

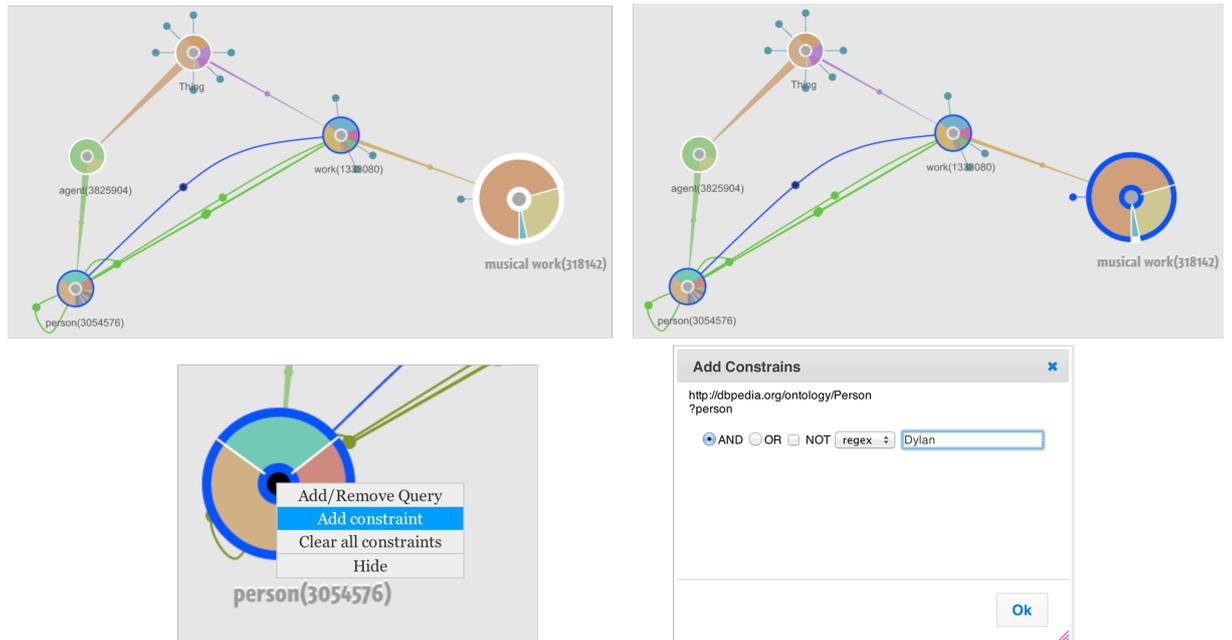


Figure 62: Illustration to describe how query formulation takes place. Left — the user starts by adding a property to query; Middle — another property is added to the query; Right — a constraint is added to the query

6.7.3 Query Formulation

A feature deemed essential by end users during focus group and interview sessions was the ability to build highly precise queries, rather than just explore datasets by browsing activities. Mouse gestures such as right clicks and selection of menu items can enable users to identify concepts and properties of interest. The ability of Affective Graphs to visually abstract from the schema of the dataset can help users to directly query for concepts and properties. Users just ‘selects’ items of interest, and each selection triggers a call to a function that builds the query. The query formulation consists of four parts: concept identification (create partial SPARQL query that creates the variables for concepts and sets their types, creates internal maps of variable names and concepts), property identification (aligns variables and concepts with the selected properties and builds partial triples for the SPARQL query), rationalisation (match variables and check for errors) and constraint addition (looks up map for variable names based on concepts, and adds filter terms).

Figure 62 illustrates how a user builds a query — the Information Need is to find what are the pieces of work (musical Work) that was written by Bob Dylan, and their record date. The dataset being used is a DBpedia dataset, available from <http://dbpedia.org/sparql>. The user initially explores the Person concept, reached via the Agent concept. The user then loads the Work concept, and observes a property connecting the two — ‘writer’. The user then right-clicks on the property and adds it to query. Following this action, the Person and Work nodes are highlighted in blue, so is the property the user had clicked on. This is illustrated in the figure, left most image.

At this stage, the system already builds a [SPARQL](#) query at the bottom of the visual interface, in the [SPARQL Query](#) box (Section D, Figure 53). As a result of the first step of query formulation (Concept Identification), the two concepts (Work and Person) are identified (by referring to the domain and range definition of the property clicked by the user). Variable names are constructed out of the concept URIs by extracting the final fragment of the URI. These names are stored in maps along the URIs. The first part of the [SPARQL](#) query is thus formed, which results in:

```
@prefix dbont:<http://dbpedia.org/ontology/>.
1. SELECT * WHERE {
2. ?person a <http://dbpedia.org/ontology/Person>.
3. OPTIONAL {?person <http://www.w3.org/2000/01/rdf-schema#label> ?person_Label.
4. FILTER langMatches( lang(?person_Label), "EN")}.
5. ?work a <http://dbpedia.org/ontology/Work>.
6. OPTIONAL {?work <http://www.w3.org/2000/01/rdf-schema#label> ?work_Label.
7. FILTER langMatches( lang(?work_Label), "EN")}.
}limit 1000 offset 0
```

As can be observed from the query, the concept variables are defined as types of the concept, and an optional label (lines 3,6) for the concept value is also requested. The same is applied to Work. Labels provide readable results, and ensure english terms are returned (lines 4,7). The second step of the query formulation is property identification, which is a similar task as the first — the properties selected by users as query properties are aligned with their domain and ranges. The domain, property and range of each property is then formed in an internal chain-like structure that ensures that a continuous linkage is established in the query. The variables (created in the previous stage) are then used to build the next set of triples:

```
...
8. ?work <http://dbpedia.org/ontology/writer> ?person.
...
```

The next step of the query formulation (Rationalisation) looks for possible conflicts in the query. Since at this stage, the query is fairly simple and there are no conflicts, this stage causes no significant change in the [SPARQL](#) query. The next step, constraint addition also does not have any significant effect on the result at this moment (since the user has only selected one property). The user then selects the next property of interest (middle, Figure 62), `recordDate`, after exploring the `MusicalWork` node. This property indicates the user is interested in the pieces of work that are also musical work and have a `recordDate`. The concept identification adds another concept to the internal mappings — `MusicalWork`, and creates a variable name for it. Similar triples like line 3,6 and 4,7 are built at this stage, for the concept `MusicalWork`, in addition to the already existing ones. The property identification step adds another triple to the query,

```
...
9. OPTIONAL {?musical_work <http://dbpedia.org/ontology/recordDate> ?recordDate}.
...
```

Selection of the `recordDate` property indicated that a datatype property was selected. The present implementation is configured so that datatype properties are not added into the query as an AND selection (a preliminary analysis of the dataset highlighted that data may be sparse, and an OR query for this dataset would be better, since the result sets would be larger). This can easily be changed if desired otherwise.

The next step, Rationalisation identifies a potential problem in the [SPARQL](#) query — this is because, there is no complete linkage between the three concepts (Person, Work and Musical Work). In the rationalise step, Affective Graphs identifies that Musical Work is a subclass of Work and the user explicitly queried for a subclass’s property. Hence, all instances of Work are replaced by Musical Work. As can be observed, this stage builds the query by making certain assumptions (e.g. if a lower hierarchical concept is selected, it is inferred that any references to any of its parents refers to the lower concept). However, this step can be further expanded to include any other type of rule ⁹. The previous query is now modified to

```
1. SELECT * WHERE {
2. ?person a <http://dbpedia.org/ontology/Person>.
3. OPTIONAL {?person <http://www.w3.org/2000/01/rdf-schema#label> ?person_Label.
4. FILTER langMatches( lang(?person_Label), "EN")}.
5. ?musical_work a <http://dbpedia.org/ontology/MusicalWork>.
6. OPTIONAL {?musical_work <http://www.w3.org/2000/01/rdf-schema#label> ?musical_work_Label.
7. FILTER langMatches( lang(?musical_work_Label), "EN")}.
8. ?musical_work <http://dbpedia.org/ontology/writer> ?person.
9. OPTIONAL {?musical_work <http://dbpedia.org/ontology/recordDate> ?recordDate}.
}LIMIT 1000 OFFSET 0
```

The last step is ineffective at this stage, since the user has not yet expressed the need for a restrictive query.

In the final step (right, Figure 62) requires the user to express a constraint — the user right clicks on the ‘Person’ node, and selects the add constraint to show the constraint popup menu. The user then types “Dylan” in the search box, and accepts the new changes. The first three steps (Concept Identification, Property Identification and Rationalisation). The last step, Constraint addition now adds the final part of the [SPARQL](#) query,

```
...
10. FILTER ( regex(?person, "dylan", "i" ))
```

6.7.4 Result Presentation

Presentation of results is a challenging process that can have multiple solutions, based on different motivations such as user preference, expertise, application framework, domain and so on. The solutions that are considered are mostly visualisation of result sets as charts, graphs, maps and so on by incorporating basic visualisations. However, in the current implementation results are presented in a sortable table, improved from a standard endpoint presentation as [HTML](#) tables. As noted earlier, that the presentation of the results is not an integral part of Affective Graphs as the system is to provide users with an interactive and highly visual way of exploring and querying unknown Linked Data.

⁹ Though the possibility is certainly provided, this feature needs careful thought before any new rules can be added

6.7.5 Tools Used

Affective Graphs was built using a client-server architecture. A web-based interface was developed using [HTML](#) and javascript. The visualisation is built using [Processingjs](#)¹⁰. [CSS](#)¹¹ and [jqueryUI](#)¹² are responsible for styling the interface, while [jquery](#)¹³ handles the interaction with the server. The backend consists of [PHP](#)¹⁴ scripts, using [ARC2](#)¹⁵ to interact with Linked Data endpoints.

6.8 CONFORMANCE TO REQUIREMENTS

The following table (Table 8) shows how the functional and non-functional requirements are addressed in the design for Affective Graphs. As can be observed from the table, the FR7 has not been addressed by Affective Graphs. This is owing to the primary function of Affective Graphs as a solution for querying and exploring large semantic datasets. Though the fact that several visualisations (in a scenario where multiple concepts are loaded) are presented together might contribute toward simultaneous views, the requirement is left blank as it is fulfilled by the `.views.` approach.

6.9 EVALUATION

As a user-centered development process, several sessions of discussions, focus groups, and evaluations with users shaped the final interface for Affective Graphs. Changes were functional as well as enhancements (such as adding contextual menu items, tooltips, search boxes etc.) after each session with users, hence the interface has significantly evolved since its inception. Four significant user evaluations are discussed henceforth:

- Evaluation 1: Formative Evaluation with Domain Experts
- Evaluation 2: A user evaluation with experts and casual users to understand how the tool performs compared to other tools.
- Evaluation 3: A user evaluation with Semantic Web experts to understand how well users perceive the system with increased exposure to the tool.
- Evaluation 4: A user evaluation with experts and casual users to understand how a natural language approach can aid users build queries for very large datasets.
- Evaluation 5: An objective evaluation of aesthetic properties of the system, compared with existing tools.

¹⁰ <http://processingjs.org/js>

¹¹ <http://www.w3schools.com/css/>

¹² <http://jqueryui.com/>

¹³ <http://jquery.com/>

¹⁴ <http://php.net/>

¹⁵ <https://github.com/semsol/arc2/wiki>

Requirement	Querying Semantic Data
FR1. Ability to visualise	Visual approach that provides statistical information of underlying data and describes how the data is structured
FR2. Express Information Need	Users can make use of interactions to identify the concepts and relations of interest and build explicit queries
FR3. Facilitate querying	Interface designed to enable users query by using simple mouse gestures
FR4. Unknown data	Generic mechanism to explore and build queries, using Semantic Web principles
FR5. Scalable	Multiple visualisations of aggregates can provide an overview of different facets of large data sets
FR6. Visual Summaries	Presenting aggregate data in visualisations
FR7. Simultaneous views	
FR8. Advanced querying mechanisms	Advanced user controls to facilitate experts fine-tune queries
FR9. Knowledge Management	Applying basic visualisations and interaction paradigms catering to all user needs and expertise
NFR1. Domain independent	Visualisations and means for querying and translating user actions are highly generic
NFR2. Intuitive	Visualisations and interactions are inspired from standard designs and interaction mechanisms
NFR3. Generic	Familiar interactions and visualisations make caters to all types of users (additionally, provides flexibility for advanced users)
NFR4. Aesthetically Pleasing	Explicitly built using aesthetic design principles
NFR5. Simplistic Query	Visual and interactive mechanisms require users to use simple mouse gestures to identify items of elements to query
NFR6. Familiar Representations	Familiar graph visualisations with basic pie charts present semantic data
NFR7. Consistent Representations	Representations of concepts and properties are consistent with other Semantic Web tools and throughout the design
NFR8. Interaction Mechanisms	Familiar interaction mechanisms as expected from other similar systems
NFR9. Provenance	Each instance can be drilled-down to their URIs, to be resolved to a further information page provided by data owners
NFR10. Shortcuts	key-combinations provide shortcuts for repetitive tasks (e.g. add into a query) or special features (e.g. hide a concept)

Table 8: Aligning final interface design with functional and non-functional requirements

6.9.1 *Formative Evaluation with Domain Experts*

The first prototype of the Affective Graphs system was built to understand how to introduce a visual query paradigm to enable exploration of Linked Data. As a part of a [UCD](#) process, the evaluation was conducted during the early stages of development of the system. The main goals of this evaluation were to determine:

1. The value of graphical approaches to visualising domain data and support statistics-driven interactions to support real-world tasks.
2. The overall perception of a highly visual and interactive experience in exploring domain data.
3. To identify interaction and design issues that may affect user experience early in the development process.

Though Affective Graphs was still in a nascent stage where several facilities were not made available, it was important to evaluate the core approach of presenting a highly visual and interactive interface for exploring data. A preliminary evaluation at this stage thus seemed appropriate as an user-centered approach methodology involved user feedback during the design and development stages.

This evaluation was conducted with the help of Vitaveska Lanfranchi as a part of the SAMULET project.

6.9.1.1 *Method*

Assessing initial reaction to Affective Graphs was one of the primary goals for conducting the evaluation. A formative evaluation with domain experts in the field of Aerospace Engineering was conducted. Users were provided with a brief overview of the data as well as the interface, following which the evaluation commenced. This evaluation focussed on understanding the usability of Affective Graphs, as well as identifying what are the possible interaction issues that can affect the user experience.

6.9.1.2 *Procedure and Setup*

An aerospace engineering dataset consisting of manufacturing specifications on different components was formalised and uploaded to a local Virtuoso¹⁶ installation. The dataset (Technical Variance) discussed the instances where there was a need for engine components to be manufactured according to a variance in the technical specifications.

The evaluation was done over a day in June, 2011, with each of the six users being involved for about 40 minutes at a time. Participants were initially introduced to the project and the purpose of the evaluation followed by a presentation of the Affective Graphs interface. A 2 minute 30 second introductory video along with a running commentary was presented that guides the participants through the interface and describes how to perform tasks. Users were then asked to follow a guided task in order to familiarise themselves with the system (and the dataset, since none of the users had previously used the dataset) and then use their understanding of the system and data to

¹⁶ <http://virtuoso.openlinksw.com/>

explore the interface by designing their own tasks. An example guided task is provided in Appendix D.

Notes were taken while conducting the evaluation to record the time when a subject was confused or asked for help. The kind of help requested as well as specific questions were noted during evaluation to be analysed at a later stage. Upon completion of the experiment, subjects were interviewed for 10 minutes and were asked for suggestions or ideas to help improving the system. Each participant was asked qualitative questions designed to highlight the usability and acceptability of the solution, using the questionnaire shown in Appendix D. All the participants were finally invited to a focus group where the results were discussed and possible improvement suggestions were collected.

6.9.1.3 *Participants*

Six professionals from the aerospace industry (4 men, 2 women) were recruited on the basis of acquaintance to have a hands-on session with the Affective Graphs interface. One participant was aged less than 25, two aged between 25-34, and two between 45-54. All of the participants were proficient in internet applications and experienced with search tools. In terms of domain and technical abilities, all of the users can be classified as experts.

6.9.1.4 *Data Collection and Analysis*

Users were encouraged to 'talk-aloud' while evaluating the tool to understand how they process the visualisation and their thought processes. Audio and on-screen activities were also recorded to be analysed at a later stage. Additional data collected from the evaluation were logs of back end queries with timestamps, evaluation questionnaire answers, screen and audio recording of users interacting with the tool.

The data collected was analysed in two perspectives. Qualitative feedback from the questionnaires, comments during the think-aloud study and focus group discussions provided insights for the first analysis, which was aimed at understanding how users felt using the system and the approach. Comments and questionnaire responses were grouped into different criteria to formulate a better understanding of how the users felt about the system. Interaction logs, observations and audio-video recordings, on the other hand provided key insights into how users were interacting with the system and the issues they encountered during their interaction. The following section discusses the results of the evaluation.

6.9.1.5 *Results*

The analysis of the feedback from the users involved grouping the responses in several criteria: overall satisfaction, effectiveness, usability and learnability. Overall, the system was judged to be pleasant and participants had a positive experience working with the interface. Users noted that it was easy to learn and explore new features with Affective Graphs. The system was also stimulating to users and the concept of visual interaction with data was quite new and was well appreciated. Comments from users like "*Seems clear what you are selecting; basic concept is good*", "*easy to use interface*", "*goes a long way to assessing data quickly*", "*very pleasant*", "*good UI, intuitive and interactive*" and "*very clear + user friendly*" were very encouraging and indicated that the approach of visually

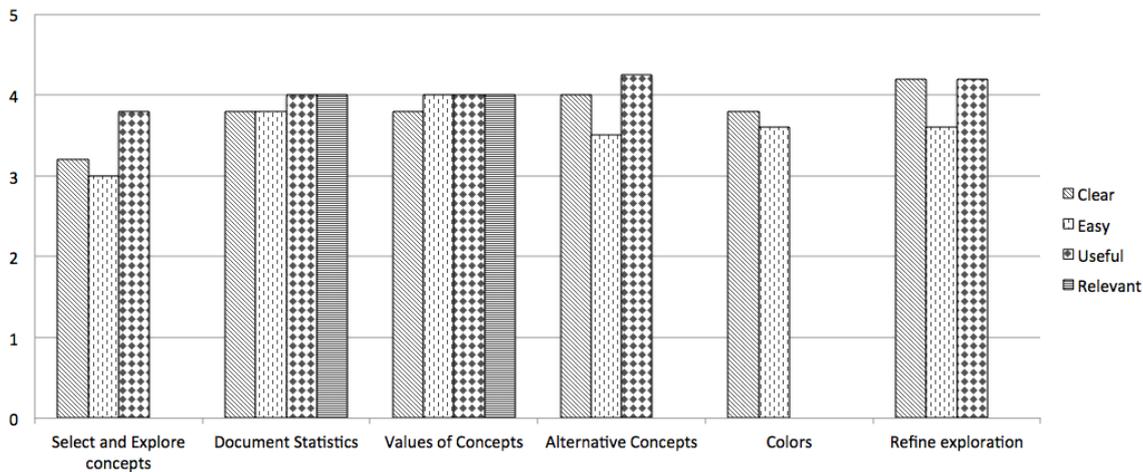


Figure 63: Results of the Formative Evaluations with domain and Knowledge Management experts

representing data elements in such a way was appreciated by users. Figure 63 shows how participants judged different features of the interface such as exploration of concepts, colours, document statistics etc. based on four different criteria — clarity, ease, usefulness, relevance.

Though the system was positively accepted by the participants, users found some of the visual representations difficult to use — comments like *“More labels on the graphs to clear it up and links to graphs would be good”*, *“movement of concept discs”*, *“force directed layout kept things moving when I wanted it to be still”*, *“text on the node not extremely clear”* indicate that there are two concerns in the presentation of graphs: layout and text. The force directed layout, though very helpful in arranging visual elements in a well-balanced way was often a hindrance to the user, thereby causing discomfort and distraction. The text labels were too small in size to be readable and needed to provide more contextual information to the users. In spite of being presented with graphs, users would prefer reading more text (labels) to explain them rather than select individual nodes to find an explanation of the data contained within it. A further comment on providing more contextual information *“not easy to see the results user has a very narrow window on the data at any time”* established the need for more contextual information for the concept in focus as well as the concepts not in focus. Users would want context in addition to their train of thoughts, but would also like to see the big picture (what are the things one is not currently focussing on). Although certain interaction issues were identified, all the users found the interface and interaction mechanisms engaging, exciting and would like to have a similar interface as a part of their daily tasks. Overall, while the evaluation was positive, three areas were established that required more attention: **results presentation** (there was a need for visually enhancing the result sets), **concept selection** (selecting concepts from pie charts was identified to be an exhaustive task and users preferred to also have the ability of text searches) and **layout mechanism** (constantly moving nodes and edges confused and disoriented users).

During the evaluation, it was observed that users required help or suggestions in order to proceed with the task. The kind of help would mostly be technical, but there

User ID	Re-Align Nodes	Visual Search	Help Requested	User Confusion	Delete Nodes	Toggle Context-window	Comments/Suggestions
1	16	6	4	1	2	6	"freeze" nodes, sort properties alphabetically, highlight most-interesting graps as suggestions
2	6	2	3	0	0	11	Show friendly texts on screen, position text on fixed place holders, provide a hovering context menu that is visible by default
3	4	6	0	3	1	10	
4	4	3	2	2	0	4	Use random colours instead of a colour palette
5	3	4	3	0	0	1	

Table 9: A manual inspection of screen recordings show different exploratory behavior and uncover several issues

were a few times when users needed explanation of the data being observed. Users also seemed to be confused at times while performing their tasks. The context window (Section B in Figure 53) on the right was set as hidden by default . This required users to click and set it to visible, but due to individual preferences, users kept hiding the window when viewing the graph and setting it to visible when required. Subsequent analysis of the video and audio recordings for five of the users highlighted several issues in the interface (the screen and audio capture for the sixth participant was not possible at the time). Though five users can be a low number, it can be argued that their observations were highly valuable as they were experts in Knowledge Management as well as the aerospace domain. Moreover, for the first exposure to users in an iterative user centred design process, a low number of users are helpful in identifying a considerable amount of issues¹⁷. Furthermore, the focus group with the five participants at the end of the sessions were highly productive and discussions ensured that all participants could present their views on different aspects of the system as well as provide suggestions. The screen interactions were analysed to identify different types of events during the user's interaction with Affective Graphs: re-aligning nodes, looking for specific visual objects, requesting help, being confused, nodes being deleted to create space and toggling of context window. These events were then manually counted to understand how often they occurred during a 10 minute interaction with the users as shown in Table 9. The table also shows the suggestions or comments provided by the users during the evaluation, some of which were not recorded by the users in the feedback.

¹⁷ Though there has been a lot of debate regarding validity and applicability of the popular notion of five users being sufficient to discover 80% of usability issues, using five experts who understand the data, domain and the technology as the first set of users in an iterative design process is motivated from a purely practical perspective — the limited availability of knowledge workers proficient in search systems and the high expense of a knowledge worker's time limited the number of users as well as the amount of time they could be engaged.

A skim through the table indicates how different users who have different behavior helps identify certain issues. Each of the users highlighted different (as well as common) issues that needed attention. The discussions in the focus group session at the end of the day also collated the issues and several common issues were discovered. User 1 had a particular approach of interacting with the interface — the table shows that the highest number of re-alignments were performed by this user. The screen recordings indicate that the user preferred objects to be aligned from left — right and the node presently in focus positioned at the center of the screen. Any action such as adding or deleting a node caused the layout to change, which caused the user to continuously re-position the nodes to their preferred orientation. None of the other users had any specific preferences, and during the evaluations it seemed they were satisfied with the positions of the nodes unless they drifted toward the corners where it became difficult to read the corresponding labels (when mentioned in the focus group session, however all users noted the discomfort in adjusting to a constantly-updating layout).

User 2 and 3 had a particularly high number of toggles for displaying the context-window. The screen recordings indicated that these users preferred to have the visualisation spread across the screen while exploring the visualisations, and only when needed did they choose to display the context window. This was in contrast to user 5 and 4, where they preferred to leave the window in place (essentially, hovering in the right hand side of the screen) unless the nodes were hidden by the windows. Some users required some help in interacting with the system — the doubts formed the basis of a discussion where the users provided certain suggestions and comments. Help was requested on the layout, colour palette used, edge representations and deleting nodes. These comments were collected and a newer version of the system was developed later to address the highlighted issues. It was also noted that users were having difficulty spotting visual objects from a layout which had several objects — this e.g. users 1, 3 and 5. These events are counted whenever the user is seen hovering over the pie sections or reading through lists of properties several times or over a long period of time.

Users were required to design their own task based on their interests and understanding of the system and the data as a part of the evaluation. This was to understand the motivations behind a user selecting different concepts and choosing their tasks — while most users chose tasks based on their roles or professional interests, some tasks were aimed at identifying organisational patterns and trends. Perhaps the most interesting motivations for one user (User 4) was to choose the task and topics on the basis of colours. This highlighted an important observation: it could be possible that users can be influenced by visual characteristics like colour and saturation while engaged in an open exploratory task. It also fuelled the need for a re-design of the interface with a greater **emphasis on aesthetic** properties of Affective Graphs.

6.9.1.6 *Discussions, Key Findings and Limitations*

Overall, the positive appeal of a visually pleasing interface was well appreciated by users. There were a few interaction issues that were identified by the participants as well as identified by observations during the evaluation. Upon uncovering the issues, it was clear where the next focus of development would be:

- A need for a better layout as users were confused by a constantly re-orienting layout.

- A better design for the contextual information window as users could be frustrated by constantly hiding/showing the window
- A better design of the nodes, with specific attention to color palettes as sometimes users were confused by the colors where a green color could indicate a positive set of data, while red a negative set.

Though the findings were important and necessary at the initial stages of the development, the evaluations have a few drawbacks. It would have been more desirable to have a larger set of users involved. However, at the time of designing the evaluation, owing to the expenses involved in engaging expert users, it was decided that the lower number of users would be appropriate enough to provide initial inputs. This evaluation also had its limitations in terms of the type of data collected — there was a greater stress on collecting subjective feedback to understand the user perception from a qualitative standpoint. This was however due to the lower number of users, which would not have a high statistical significance. Furthermore, an quantitative evaluation would be more useful when the system is at a later stage of development.

6.9.2 *User Evaluation with Casual and Expert users*

The first evaluation saw an initial prototype of the Affective Graphs interface evaluated in a Knowledge Management scenario, with few highly expert users in a basic setting, where it was important to understand user opinions as well as initial feedback. However, it is important to verify if the approach of the system can be used to perform fact finding tasks, which users of Linked Data engage themselves with. There were two main objectives for the next evaluation:

1. How does Affective Graphs perform in comparison to other systems employing different query approaches ?
2. How does an aesthetically designed interface affect the user's perception of the system as a whole ?
3. How differently do casual users and experts behave with the system?

In order to answer the first question, other systems were identified which employ different querying mechanisms (such as natural language, form based and graph based) and all the tools were evaluated together with the same questions and dataset. Since the evaluation was conducted on all systems, a larger discussion on the comparative performance of each system is discussed in the evaluation paper by [EWC12b]. The scope of the discussion, and the analysis in this thesis is limited mostly to what pertains to Affective Graphs.

6.9.2.1 *Method*

A independent comparative evaluation was conducted as a part of the second evaluation campaign in the SEALS project ¹⁸ during June, 2012. An independent test leader identified five systems (Semantic Crystal, K-Search, Ginseng, Affective Graphs and NLP

¹⁸ <http://www.seals-project.eu/seals-evaluation-campaigns>

Reduce) employing different querying techniques (visual, natural language, form based) to be evaluated under the same conditions by users of varied expertise.

6.9.2.2 Procedure and Setup

Subjects were provided with each system at a time to answer five questions, where questions and the systems were selected randomly to avoid any preconceived biases, frustration from other systems or the impact of learning from other systems. The questions were presented at random, one at a time by a evaluation controller developed by the SEALS project¹⁹. Before starting evaluating a system, the subject was provided with a short (5-10 minute) demonstration explaining the query approach adopted by the tool as well as how to formulate a query. Following the demonstration, subjects proceeded with the evaluation where they were required to answer five questions.

A geographical dataset within the Mooney Natural Language Learning data was identified as an appropriate dataset for this evaluation. The reason for this choice is due to the familiarity of the domain with all users - the geography ontology consists of 9 classes, 11 data properties, 17 object properties and 697 instances. Five questions carefully chosen from the Mooney Natural Language Learning Data²⁰ were presented to the users, which needed to be answered by interacting with the systems. The questions were of different complexities:

- A query involving one concept and one relation (*e.g. All capitals of states in the USA*);
- A query involving two concepts and two relations (*e.g. Cities in states through which a river Mississippi runs*);
- A query involving a comparison (*e.g. States that have a city named Columbia with a city population of over 50,000*);
- A query that involves superlative comparison (*e.g. Lakes that are present in a state with the highest point*); and
- A negation query (*e.g. Rivers do not traverse the state with the capital Nashville*)

The evaluation was conducted by a test leader who was independent from the development of the tools, to avoid any bias during the evaluation. The tools were explained in the same way, by the same test leader to avoid any unfair bias toward a particular tool.

Overall, the duration of an evaluation session with each subject evaluating all the systems took 60-90 minutes. Participants were provided with questionnaires at the end of evaluating each system, and an informal discussion ensued after all the systems were completed.

¹⁹ <http://www.seals-project.eu/>

²⁰ <http://www.cs.utexas.edu/users/ml/nldata.html>, consisting of 877 natural language questions for the geography data

6.9.2.3 *Participants*

Twenty subjects (12 females, 8 males) aged between 19-46, with a mean age of 30 years were recruited to evaluate the systems via email. The subjects were grouped into two categories: expert and casual. Expert users had moderate to advanced knowledge and experience with Semantic Web and ontologies, while casual users had little or no knowledge. Casual users were recruited from the staff and student population of the University of Sheffield, while the expert users were identified from the Organisations, Information and Knowledge (OAK) group, at the University of Sheffield.

6.9.2.4 *Data Collection and Analysis*

As discussed previously, Elbedweihy conducted a broader analysis that spans across all the systems to understand how the different approaches are used by expert and casual users. Their analysis and subsequent discussions were reported in their paper [EWC12b]. This thesis, however presents discussions from the point of view of Affective Graphs, following a similar (also analysing input time, success rate and number of query iterations), independent analysis of the data shared by the SEALS project.

Quantitative data was collected during the evaluation in the form of interaction, back-end processing and query logs. The input time was obtained by computing from the interaction logs recorded by the evaluation controller as well as Affective Graphs. Users indicated when they are satisfied with the results obtained, which provided the number of query iterations as well as success rate. Qualitative data was also collected for future analysis, that included user satisfaction questionnaires, evaluator observations, discussions and feedback session at the end of the evaluation. The subjects were presented with three questionnaires – a System Usability Scale questionnaire, a demographics questionnaire and an extended questionnaire, as shown in Appendix E.

The statistical package, SPSS²¹ was used to analyse the data shared by the SEALS project. The analysis was conducted in two ways – to understand how long users took to formulate a satisfactory query, and to understand how many iterations of queries users had to undergo to reach the right answers. These measures were compared against different query approaches, and aligned with the user type. Quantitative data was analysed firstly by computing SUS scores, and compare the scores with other systems as well as among user types. Qualitative feedback was analysed by grouping positive and negative comments according to different criteria of the system. The following section provides the result of the evaluation in details.

6.9.2.5 *Results*

As discussed earlier, the results for the two user types are discussed in two ways: how long it takes for a user to formulate a satisfactory query (query input time) and how many times they executed their queries to perform their tasks (number of attempts). An attempt is classified as the individual formulating a query and clicking “Search” to indicate they are satisfied with their query and are requesting the results for the query. Figure 64 shows three boxplots – SUS score, query input time and number of attempts for each tool clustered by the user type.

²¹ <http://www-01.ibm.com/software/uk/analytics/spss/>

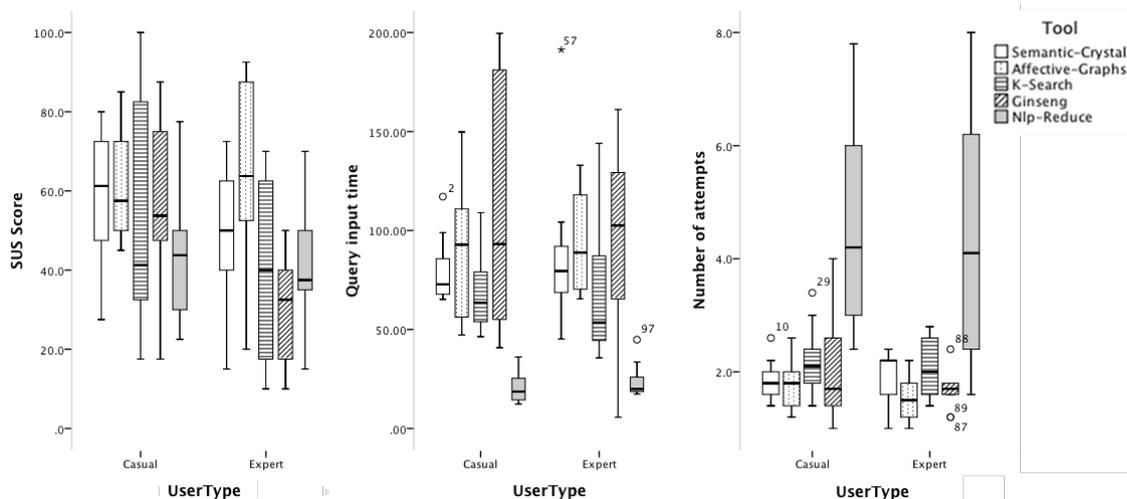


Figure 64: Evaluation results for SUS scores, query input time and number of attempts for Affective Graphs: the users have been grouped into two types — experts and casual

Analysing the query input time (middle boxplot) shows a trend that was expected — NLP Reduce took the least time to formulate queries, while Semantic Crystal and Affective Graphs took considerably longer. The only exception being Ginseng, where users were frustrated by a restrictive natural language interface (this is described in more details in [EWC12b]). Users took relatively longer to formulate queries using the graphical approach of Affective Graphs. While conducting the experiment, it was observed that this was mostly due to the fact that users found themselves engaged with the system, and were interested in exploring different features of the interface. The number of attempts also provided an interesting insight — users took the most number of attempts in retrieving satisfactory results using NLP Reduce. Affective Graphs scored among the least for both experts and casuals. Given their prior knowledge in Semantic Web formalisms, experts took the least number of attempts using Affective Graphs as they can relate to a graphical representation of graph data.

Combining the query input time and number of attempts reveals the most interesting observation — casual and expert users took a significantly large amount of time in order to formulate highly precise queries, thereby being able to answer their Information Need satisfactorily. This finding is key to the purpose of Affective Graphs, as users can exploit the graphical highly approach to express themselves more precisely to their satisfaction.

The users highly appreciated the system and felt excited about an interactive and intuitive system presenting information in a pleasing way. One of the most encouraging comments from one of the participants, *“it is interesting that when you use a colourful and interactive system, you do not mind trying several times to get an answer as it is a playful and enjoyable experience”* clearly identifies with the focus and aim of an aesthetic experience — to help users comprehend, query and explore unknown data and provide a pleasant, exciting and enjoyable experience. Another interesting comment *“we have been exposed to natural language querying tools like Google and yahoo for a long time and hence find ourselves more comfortable with such systems, but had I been introduced to such graphical techniques, I*

would probably choose them over traditional natural language systems” shows that a user’s pre-disposition and prior experience with natural language interfaces can influence the acceptability of a different solution. However, if the experience of using such an interface is pleasant and enjoyable, there is a greater likelihood of the system being accepted by user communities.

Users were provided with a questionnaire consisting of *System Usability Scale (SUS)* questions (Appendix E), which are standard questions for determining a user’s perception toward several aspects of a system. Affective Graphs scored the highest (60.0) in average overall usability score (SUS) when both the user types are combined (compared to 55 scored by Semantic Crystal, 41.25 by K-Search and 40.0 by Ginseng and NLP Reduce). Interestingly, expert users like the system more than casual users, possibly explained by the familiarity and prior knowledge of Semantic Web formalisms and graphical approaches to representing and conceptualising data.

While there were users who disliked a completely visual approach toward exploring data, most of the users liked this approach, and would prefer to frequently use the system as a part of their daily analytical tasks. Most users also felt the system was easy to use, though the experts seemed to be more comfortable with the system — the experts have prior knowledge of semantic formalisms and have a better understanding of the ontological concepts and visual representations of properties and classes. Similar to ease of use, most often experts found the system easier to learn owing to their knowledge and expertise. Often repeated as comments were that both the casual and expert users found the interface fun, playful and enjoyable to use overall. This is extremely encouraging since this shows that it is possible to interact with Linked Data in a manner which does not involve highly formal and structured ways of querying.

In addition to the questionnaire responses, users were asked qualitative questions that attempt to understand the positive and negative features of the system. The responses of the questionnaires were collated and grouped into different categories:

1. Affective Graphs Interface
2. Visual Query Mechanism
3. Result presentation
4. Others

This discussion is driven by the positive and negative comments regarding these aspects of the system. The evaluation with the users has been highly satisfying, where users confirmed the visual approach and appreciated the different features that promise to make interacting with Linked Data an enjoyable, fun and exciting experience for users. Comments such as *“once I got the hang of it, it made much more sense and was easy to use”*, *“Bit of a learning curve but after that it was quite easy and intuitive to use”*, *“Easiest to define queries out of the ones I’ve used”* and *“this system was simpler to use than I expected”* show that the users had an initial impression of the system to be difficult to use. However, with a little experience and learning, the intuitiveness and ease of use was apparent. Learnability is a feature that is extremely important specially with new approaches toward consuming Linked Data.

Comments such as *“The graph visualisation worked well graphics were intuitive and easy to use and combine I liked to see the links between the concepts it made it easier to understand”*,

“The nice user interface made for a more pleasant search experience, and the animations made it clear which concepts were connect”, “friendly interface, fun” and “The graphic interface is really intuitive and easy to use Visual appearance of system was modern and interesting” reflects the positive feeling that users had after using the system. Upon asking how the system could be improved, several suggestions came up, such as displaying the entire ontology at one go and hide reverse relations (e.g. `hasMountain` and `isMountainOf`). While hiding reverse relations could help reducing the number of edges, it would only be beneficial if there are reverse relations existing in the ontologies (a few datasets do not have any reverse relations defined) and the same data is reflected on a reverse relation (it could be possible that the reverse relations are not populated synchronously), thereby possibly increasing the chances of a user missing information. Showing the entire ontology to a user can also have a negative effect of increasing cognitive burden on users by showing them information that is not relevant to their interests. The current approach has been to present information to the user only if they explicitly convey their interest on specific concepts and relations.

Users appreciated the visual query approach, and liked the interactive mechanisms involved in the querying process. Comments like:

- *“The query generation is intuitive and simple to use. It hides all complexity of the underlying query language. You dont need to think in advance the order of the elements that have to be taken into account in the query and add them any time.”*
- *“I liked to see how the links and circles activated when I added them to my question. That made me realise what I was actually been query, that also gave me and idea of the coverage of my question”*

show that the users appreciated the visual communication of the query being applied and how the querying mechanism attempts to ‘hide’ the complexity involved in building a [SPARQL](#) query. However, users felt that several things could be improved in the system in two main areas: automatically linking query concept and properties (where smaller queries can be linked to construct a single larger query, automatically selecting concepts as query elements when a property is selected), a more varied colour palette to prominently highlight filter constraints. Users also felt that there could be more on-screen help to guide the users in building queries and there seemed to be some cognitive gap among users while converting a natural language question (tasks) into a representative visual query.

Results presentation was by far the weakest aspect of the system — other systems such as K-Search and Semantic Crystal scored relatively higher in this category. There are three reasons for this: a highly visual and interactive mechanism of querying and exploring data generated an expectation of a similar representation for the result sets (comments such as *“better presentation of query results”, “perhaps a more graphical approach to the answers like when creating the select statement would help”* indicate that the users were slightly disappointed with a textual result set), lack of enrichment of result sets (comments such as *“I’d like to have ‘move over’ function that brings up a summary of each result”* imply that some level of processing on the results would be helpful) and experimental constraints required users to create queries in a specific manner where they were requested to specify the variable names as a part of the query, thereby resulting in a set of URIs returned rather than labels (*“the results were too SW”*). As previously highlighted, it

is also to be noted that Affective Graphs is a tool that was built specifically for querying and exploring Linked Data. A different interface has been built for rendering result sets as presented in the previous chapter, which will be integrated with Affective Graphs at a later stage.

One of the user's comment, "*It would probably look OK as a search system in a Science Fiction B-Movie*" is encouraging and valued to the approach — users seemed to be excited and stimulated by using the system and the role of an aesthetic design was the key to this.

6.9.2.6 *Discussions, Key Findings and Limitations*

Overall, the comparative evaluation showed that there was a general appreciation of the visual approach adopted by Affective Graphs. The attention to aesthetics during development of the solution helped in formulating a pleasant solution to users – the high SUS scores indicate users had a preference for the system. In general, the time taken to formulate a query with an NLP approach was lesser than Affective Graphs, but the number of attempts users needed to reach a satisfactory answer was among the least. This indicated that though users needed longer times to formulate queries, but could successfully reach the data of interest without many retries. The longer time required for formulating queries could also be as a result of users playing with the system, trying several features, satisfying their curiosity.

The evaluation also noticed that experts could relate to Affective Graphs more, and, as a result were more satisfied with it (hence, the highest SUS scores). This was due to both their familiarity with semantic web concepts, as well as their methodological approach toward constructing the queries. Both types of users could build complex and accurate queries without the need for multiple iterations of querying.

Users expressed the need for gaining more familiarity with the system. This formulated the next step of development – understand how usage in the longer term evolves. The evaluation also identified the need for modifying several features in the system such as adding a context menu, standardising interaction mechanisms, general bug fixes and backend performance improvements.

The evaluation has one major limitation: several systems have been used to exemplify how the different query approaches appealed to the users. However, it is to be noted that the user feedback often relies heavily on implementation issues and design choices taken by the developers. Personal preferences, look-and-feel and other subtle criteria may bias user opinions. Considering the overall collective perception of users toward positively or negatively assessing different approaches were similar among the users, the evaluation indicated that the findings about individual systems could be generic enough to be representative of query approaches. A larger evaluation with a greater sample of users and variety of user interfaces, randomly assigned to users could be an excellent way to investigate this further.

6.9.3 *Effect of prolonged use*

This evaluation was part of a joint work with Khadija Elbedweihi and Vitaveska Lanfranchi.

The motivating factors for conducting this evaluation were two-fold:

1. an evaluation at an earlier development cycle introduced several questions that were highly interesting and required a more in-depth study of how users interact with and respond to the system in a longer term.
2. as a scheduled evaluation to estimate how newly added features and modifications were perceived by users.

The most common feedback from the previous evaluation was that the users enjoyed interacting with the system and, in spite of being perceived as slightly complex, users found the system highly stimulating and engaging. This was largely credited to the ‘playful’, attractive and interactive nature of the system. The users, however mentioned that they would like more training and opportunity to learn the system in order to exploit the full potential of the system. This, along with the appreciation of the aesthetic appeal of the tool seeded the next stage of development for Affective Graphs, where the interest was in understanding how learning the system would affect the use of the system. More specifically, to understand:

1. How easy (time and effort required) it is to learn how to use Affective Graphs to perform tasks of different complexity and conduct exploratory search tasks ?
2. What is the effect of learning on performing tasks ?
3. How does learning affect the aesthetic perception of the system ?

Learnability, used interchangeably with *ease of learning* is an important criterion of usability that focuses on the ease of learning how to use a system or an interface. [Sha86] describes learnability as the relation of performance and efficiency to training and frequency of use. [Nie93] discusses how learnability can be measured in terms of the time required for a user to be able to perform certain tasks successfully or reach a specified level of proficiency. A similar definition is given by [Shn98] as “the time it takes members of the user community to learn how to use the commands relevant to a set of tasks”. [TAo8] argues that measuring usability in a one-time evaluation might be misleading since the use of some applications/systems requires frequency and therefore assessing learnability would be essential.

Learnability has received a significant amount of focus in the literature, some of which focused on assessing learnability as a usability criterion while others investigated how it is affected by different factors (such as interface design). While some of this work focused on *initial learnability* (referring to the initial performance with the system), others looked at *extended learnability* (referring to the change in performance over time) [GFAo9]. For example, [HEE⁺02] studied the learnability of two hypermedia authoring tools (HATs) as perceived by academics. Subjects’ answers to a set of Likert scale-based questions and their feedback, which was recorded during the sessions, were used to investigate learnability issues. In [Paroo], learnability of two different methods of interaction with databases was compared using similar measures which are based on subjective data (such as questionnaires and users’ feedback). [Jen05] assessed the learnability of searching two university Web sites by asking students of the first university to search the other site and vice versa. In contrast to the previous studies, efficiency-based measures, including success rate (number of tasks performed correctly) and the time required to perform the tasks, were used to assess learnability.

Additionally, [RM83, WJLW85, DBW89, HEE⁺02] showed that learnability and usability are congruent.

Despite this attention, both IR and Semantic Search evaluations focused either on performance-oriented aspects (such as precision and recall) [HHM⁺10, BSdV10] or assessed usability in terms of efficiency, effectiveness and satisfaction, leaving aside learnability and memorability [KBF07, EWC⁺12a].

6.9.3.1 Method

A summative evaluation consisting of multiple sessions, each with a set of guided tasks in a controlled laboratory setting was conducted with expert users. The evaluations were conducted over a period of two weeks during October, 2012, each user evaluating the system every consecutive day in three sessions. Observations during the experiments were noted down, along with the times to be analysed later on. Minor events and interesting observations such as users being disoriented, confused, trying new features were noted while the experiment proceeded, as well as post-evaluation.

6.9.3.2 Procedure and Setup

A dataset, consisting of information regarding papers presented in conferences and workshops in the area of Semantic Web²² was uploaded to a local Virtuoso installation and made available for Affective Graphs to query. There were four main motivations for choosing the dataset, given the group of expert users —

- Semantic Web experts are familiar with the dataset;
- Users are familiar with Semantic Web concepts;
- Users have a good understanding of scientific publishing;
- Availability of real-world query logs²³.

The logs of the user queries for the dataset were then analysed to understand the different types of requests made by users. Following an analysis, the following four types of most frequently used queries were identified:

1. Simple Task (ST): $C_n A_n F_n$;
 $n = 1$
 Simple queries that comprises only one concept and one attribute but also a filter or a restriction value applied to the attribute. E.g. *Find the people with first name 'Knud'*
2. Multiple Attributes Task (MAT): $C_n A_m$;
 $n = 1, m \geq 1$
 Increased number of attributes without a filter. E.g. *List the name, page and homepage of organisations*

²² Semantic Web Dog Food dataset, <http://data.semanticweb.org/>

²³ The USEWOD 2012 dataset (<http://data.semanticweb.org/usewod/2012/challenge.html>) consisted of two years of real-world query logs from 12/2008 till 12/2010

3. Multiple Concepts Task (MCT): $C_n R_m$;
 $n \geq 1, m \geq 1$
 Searching across multiple concepts, similar to breadth search. E.g. *List all the people who have given keynote talks*
4. Complex Task (CT): $C_n A_m F_o R_p$;
 $n > 1, m, o, p \geq 1$
 Include all the four components: concepts with relations linking them, attributes/properties of the concepts as well as filters restricting the values of the attributes. E.g. *Find the page and homepage of each person whose status is 'Academia' and was a chair of a session event and find its location.*

where 'C' = Concept, 'A' = Attribute, 'F' = Filter, 'R' = Relation.

The three sessions were spread over three consecutive days. On the first session, subjects were initially introduced to the experiment and its goals, followed by a 5 minute presentation and explanation of the system. The second session started with a similar 5 minute presentation, with special focus to how the system can be efficiently used and a few shortcuts. Following a 5-10 minutes hands-on practice session, users were then given control of the system and were asked to perform four tasks, one of each type. In addition to the fact-finding tasks, users were also asked to perform an exploratory task, where the real answer to the task was not known, and could depend on the user's interpretation of the question. These exploratory tasks were asked on the first and third sessions. These tasks were included as part of the first and third sessions, since it exploratory tasks were expected to take longer. The tasks were presented one at a time by the evaluation controller developed by the SEALS project. The set of tasks are provided in Appendix G, where in every session, a random task from each category was selected. Notes were taken while conducting the evaluation to record the time when a subject was confused or asked for help. The kind of help requested as well as specific questions were noted during evaluation to be analysed at a later stage.

6.9.3.3 Participants

Ten expert users (8 men, 2 women) aged between 22-38 (mean of 31 years) were then asked to perform a given set of search tasks using the interface in a controlled laboratory setting over three one-hour sessions. The users were either researchers or software developers highly proficient in Semantic Web technologies as well as conversant with scientific publishing domain.

6.9.3.4 Data Collection and Analysis

As discussed previously, the most common ways in literature to measure learnability were either based on objective data by comparing users' performance/efficiency over time or subjectively using learnability questions such as "I found this interface easy to learn". To allow for deeper analysis, both objective and subjective data covering the experiment results were collected. Input time, success rate and number of query iterations provided objective data, while responses from three questionnaires (System Usability Scale [Bro96], Extended and Aesthetics) provided the qualitative data. The input time

was obtained by computing from the interaction logs recorded by the evaluation controller as well as Affective Graphs. Users indicated when they are satisfied with the results obtained, which provided the number of query iterations as well as success rate.

Questionnaires were filled in at the end of every session as presented in Appendix F. The System Usability Scale questionnaire included questions that attempt to understand the affect of learnability on usability. The extended questionnaire included more specific questions on learnability, remembering features and so on. Aesthetics questionnaire included questions on what the user's perception of various aesthetic properties of the system. Additionally, users were also presented with open-ended questions to gather further details about how their experience was.

The statistical package, SPSS was used to analyse the data. The analysis was conducted in two ways – to understand how long users took to formulate a satisfactory query and to understand how many iterations of queries users had to undergo to reach the right answers. The measures for each of the three sessions were compared to understand how the behaviour of users vary with learning the system. Additionally, the measures were also grouped into the type of queries to understand how the users interacted with the systems while performing tasks of varied complexity. Qualitative feedback was analysed by grouping positive and negative comments according to different criteria of the system. The following section provides the result of the evaluation in details.

6.9.3.5 Results

Though a considerable amount data was collected from the evaluation, analysis and discussions will focus on two major aspects: how users performed their tasks and how they perceived the interface. Additionally, the feedback users provided in terms of responses to open-ended questions will also be discussed. Following the evaluation, two main observations were investigated: *input time* and *number of attempts*. Here, input time is defined as the amount of time taken to compose a satisfying query i.e. time taken from the time the user starts the task till the search is executed. Number of attempts is defined as the number of queries executed by users to complete their tasks. These measures were analysed to understand how the behaviour of users was affected as they learnt the system.

Objective data, in the form of query logs were collected, which indicated two key features — how quickly do users perform tasks after training, how many attempts at completing the task do the users need before they are satisfied with the results. An evaluation controller collected user interaction events and the logs were later analysed. The logs were grouped into the three sessions and the data was compared. This is shown in Figure 65.

EFFICIENCY AND EFFECTIVENESS The figure on the left shows a boxplot of the distribution of the number of attempts required to perform the five types of tasks in the different sessions. During the evaluation, two types of behavior among users were observed, based on the type of task. Users behaved similarly when they were faced with simpler tasks (ST, MAT, MCT) and their behavior changed as the tasks became more complex. Initially, users needed a few attempts at solving simple tasks. Complex tasks required a few more attempts. The number of attempts required for the simple tasks started reducing over the sessions as users gained more familiarity with the tool.

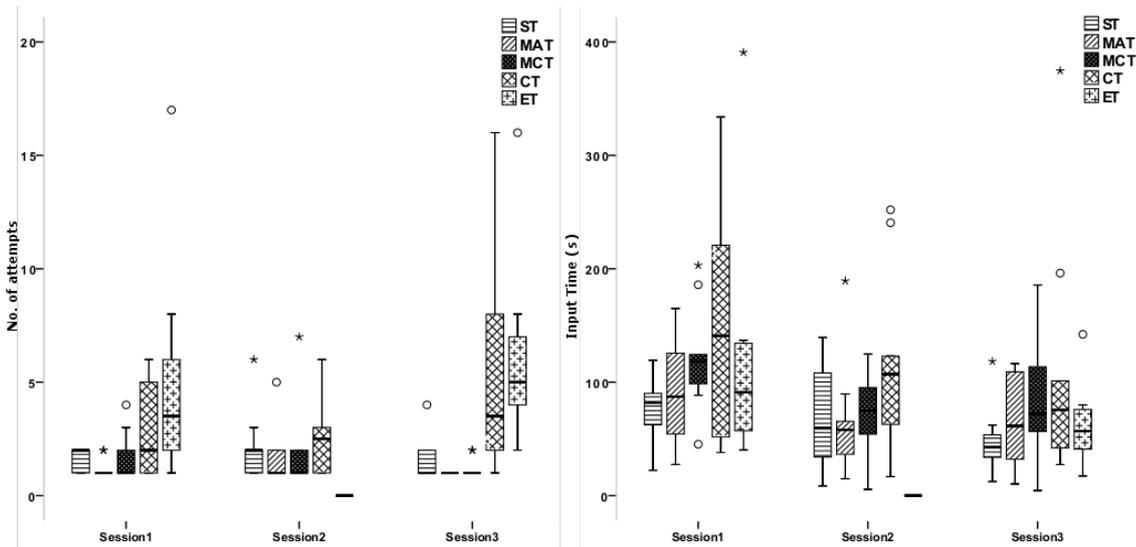


Figure 65: Analysis of performance of users with five types of tasks: ST: Simple, MAT: Multiple Attribute, MCT: Multiple Concept, CT: Complex, ET: Exploratory. Figure on the left shows a box-plot of the number of attempts required to solve a task satisfactorily, grouped into sessions and task types. Figure on the right shows a box-plot of the input time grouped into sessions and task types. Images from [MPE⁺14]

This was an expected result, as users are more comfortable with a new interface with time, and gain more expertise interacting with it. Furthermore, users started trying new techniques and features during the second session, which increased the level of comfort and helped users adapt to Affective Graphs more.

An unexpected observation was the change in behavior while performing slightly complex tasks- users seemed to require more attempts in order to perform the tasks during the latter sessions. This was surprising, as the users were expected to find such tasks easier with more time and familiarity. The (median) number of attempts for complex task (CT) increased from 2 in session 1 to 2.5 in session 2 and 3.5 in session 3. The (median) number of attempts for exploratory tasks (ET) increased from 3.5 in session 1 to 5 in session 3. This clearly showed a change in the behavior and approach toward solving complex tasks.

The figure on the right shows a boxplot of the distribution of the amount of time required by users to formulate their queries to solve the tasks (input time). In general, a significant decrease of input time was observed from an overall average of 106.48s in the first session and 72.72s in the second session to 66.85s in the final session. All of the types of tasks have shown a steady reduction in the input time. Observations during the evaluation sessions credited this to the increased comfort and acquaintance with the system and its features with more time and familiarity with the systems. While the relatively simpler tasks (ST, MAT and MCT) have seen a general reduction in the times, the complex and exploratory tasks are of greater interest to analysis owing to the more complex nature of the tasks.

The user's performance in the complex tasks (CT and ET) appear highly interesting. The reduction in time is significant — from a median of 140.86s in the first session and 107.01s in the second to 75.57s in the final session for the complex tasks (CT) and 91.16s

in the first session to 56.87s in the final for the exploratory tasks (ET). This supports the earlier observation where the number of attempts increased with more familiarity with the tool — greater familiarity and comfort with Affective Graphs helped users try several things more quickly as users found it easier to formulate queries. This was also observed during the evaluations: initially, users were carefully building long queries, connecting multiple concepts. This technique gradually changed to a different one in the second session, where users tried short bursts of queries, gradually building up to form a longer one. Users could use the outcome of the short queries to quickly formulate a longer query, which was well-informed and driven by the results of short queries. Let us consider the second exploratory task in Appendix G: the task requires the user to identify persons who are experts in Knowledge Management. There could be several approaches toward solving this task — users could look at all persons that have organised tutorials that are associated with Knowledge Management, or all persons who have several publications on the topic. The ultimate goal of this task is to connect multiple concepts (either proceedings, tutorials or workshop events) by corresponding relations, and identify people. The approach followed in session 1 was to select all the relevant relations, set constraints and connect the concepts in the very first attempt at one go — this would not provide acceptable results for many queries, thereby making the user try re-building the entire query after every attempt. Upon realising this repetitive process, most users gradually shifted their approach toward building short queries (to find all the authors, tutorial authors etc.) and investigate the results to build a final query that was more certain to provide useful answers.

USER SATISFACTION Three questionnaires were presented to users at the end of every session — a general usability questionnaire (ten 5-point Likert scale questions for computing SUS scores [Bro96]), an extended questionnaire (five 5-point Likert scale questions and 2 open questions, related to understanding learnability, ease of use and remembering features) and an aesthetics questionnaire (fifteen 5-point Likert Scale questions, related to understanding various aesthetic features of the interface). Finally, two open-ended questions were asked at the end of the third session, which aimed at understanding what users liked/disliked in the system, and how that changed over the sessions (as a result of learning and increased familiarity)

Overall, the system was appreciated by users, and the high SUS scores indicate the positive response. The high SUS score of 76.25 in the initial session indicated the first reactions of the users to the system — users seemed to be excited about the interactive and ‘playful’ environment, which evoked an overall positive response. The second session noted a significant improvement in the SUS scores (82.5) — this was more due to the users being satisfied about successfully trying new features. The final session shows a marginally decreased SUS score (81.25), but still significantly higher than the first. This can be explained by users being more accustomed to the system, and having learnt new features over the sessions, the initial excitement of discovering new features had normalised by the third session. Users appreciated the visual approach adopted by Affective Graphs, as shown in the answers when asked what the users liked about the system:

- *“The visual aspect of the system makes it less tedious to perform searches. I also found it easier to define the relations and constraints on the relations than in other systems”*

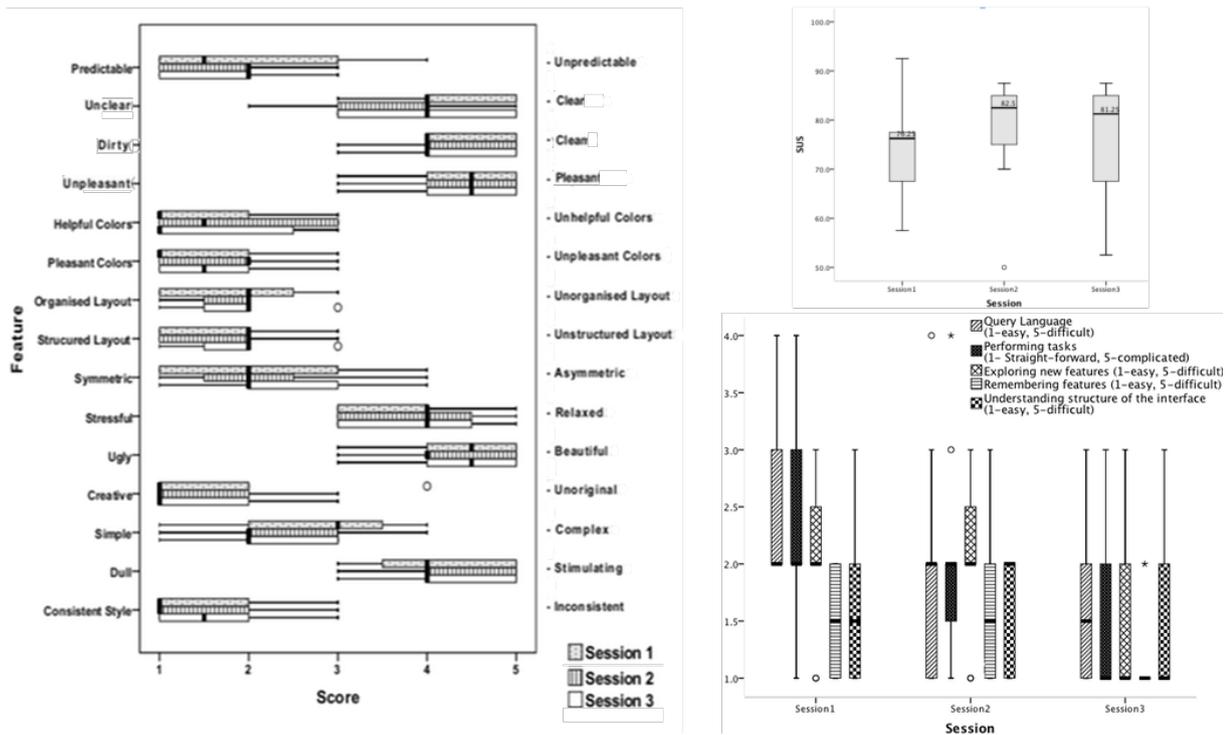


Figure 66: Results of the User Evaluations. Left — Aesthetics questionnaire responses, Right Top — SUS scores, Right Bottom — Extended questionnaire responses. Image from [MPE⁺14]

- *“interface is visually appealing: responsive, colourful, professional feedback is good nice to explore the structure of the underlying data without clutter”*
- *“the graphical interface makes it very easy to use. the exploratory nature is very good, it is easy to learn by exploring”*
- *“This interface eases a lot the use of SPARQL and makes simple querying an RDF dataset”*
- *“Overall though, I liked how queries could be built using the system to retrieve precise facts. It is much easier to use a system like this than having to build queries by hand”*

Affective Graphs was judged by users using 5-point Likert scales, values ranging from 1 to 5, 1 being the most positive, while 5 being the most negative. The system was judged moderately easy to use and understand (Figure 66, bottom right) initially with a median score of 2 in the initial two sessions. The system was also initially judged to be moderately straightforward. With more familiarity and experience with the tool, users felt more confident and the median scores reduced to 1 from 2 in the first two sessions. Users gradually felt more comfortable exploring new features by trial and error, and this also reflected in the user’s subjective scores — A median of 2 in the first two sessions reduced to 1 in the third. Overall, users also did not face any issues remembering features throughout the experiment, and by the third session, they appeared to remember features without any effort. A few comments also noted how the users perception and experience of the system changed over time:

- *“Ability to use the system effectively and my confidence/speed in using the system grew over time.”*
- *“The system became easier to use and understand over time.”*
- *“With practise, the tool became easier to use. I found satisfactory solutions to the questions much quicker in the second and third sessions than I did in the first.”*
- *“At first the system seems complicated to use. After using it a few times I found it a lot easier to construct the queries.”*
- *“I also got more confidence with using the system to progressively build queries instead of trying to get everything in place at once”*

Users were also initially confused by few features and visual elements on the screen — for example, the context window on the right (Section B, Figure 53) was initially perceived as an element which was not helpful, but with more familiarity the use of the window was more evident and users could eventually perform their tasks much better. User comments such as *“(I) thought having to use the right hand context box to find out possible onward links to other concepts was a pain. got used to this and in the last session was using this box all the time to find suitable associated concepts.”* and *“With time I learned how to use the pane on the right more effectively to identify potential relationships I could explore between an object already on the canvas and other objects.”* show that there is much scope for improvement in such areas, but the value of the content on the window is highly useful if the users gain more familiarity with the tool.

The continual insistence on an aesthetically pleasing experience as one of the key design requirements had significant benefits on the users. Several users appreciated the interface, and during informal discussions showed a lot of interest in how the tool was built. Comments such as:

- *“Nice UI, clear design to see results”*
- *“easy to use; intuitive; friendly good way to visualise the structure and data. Good to see possible links from selected concepts”*
- *“interface is visually appealing: responsive, colourful, professional feedback is good nice to explore the structure of the underlying data without clutter”*
- *“Its a great search tool!”*
- *“UI nice and is good to use, faster to use than typing SPARQL.”*
- *“(Liked) The layout and the connections between the different sets. As it was easy to see where the connection between sets were.”*
- *“(Liked) the highlighting tool when searching for particular entities was well designed, and helped with finding the correct query”*

show the overall pleasant experience users had while interacting with Affective Graphs. The response to the interface is directly an outcome of the design decisions while developing the system. The Figure 66 shows how the users perceived different aesthetic properties of the interface over time. There were no significant changes observed in the

user's perception of aesthetic the properties. The median scores of 9 of the 15 properties remained constant throughout the sessions. 4 of the properties had a minor variance in the sessions. Most of the users found the interface to be creative, beautiful and stimulating. The colours presented to the users were pleasant and helpful in general. Users initially found the system relatively complex, but with more learning, the system appeared to be simpler to use.

6.9.3.6 *Discussions, Key Findings and Limitations*

The multi session learnability evaluation with experts provided a lot of interesting observations as well as indicated areas of improvement. Apart from minor interface changes (adjusting size of context window, radius of nodes and weights of nodes and springs), the most significant improvement was needed in clearer visual feedback of nodes with constraints applied.

The observed change in search behaviour was a very interesting result of the evaluation – users seemed to change their search approach from carefully constructing queries while taking a significant amount of time to short bursts of increasingly evolving queries. The next outcome was that the SUS scores peaked at the second session, and appeared to reduce marginally to 81.25. This indicated that as the users gained familiarity with the system and successfully trying out new features, they developed a greater satisfaction toward the system. The third outcome was that there wasn't a significant change in the aesthetic perception of the system over the sessions. Out of 15, 9 aesthetic properties remained the same. The final outcome came as a part of observations and log analysis – users seemed to extensively use the search feature to quickly identify relevant concepts. This paved the way toward the next stage of development: to explore a hybrid approach that exploits a natural language input system to enable users access the concepts of interest quickly.

The evaluation had one major limitation – the number of sessions could be higher to gather a greater understanding of the users, their search behaviour and their assessment of the system usability scores. The SUS scores appeared to peak at the second session, while drop down marginally after the third. A larger number of sessions could investigate how the SUS scores are affected in a much longer term. It would also be interesting to observe how users judge the aesthetic criteria of the system with a much greater familiarity over a larger number of sessions. Preliminary results showed that no significant changes have been observed after three sessions, but it would be an interesting study to understand how this varies over a much longer period of time.

6.9.4 *Hybrid Approach Evaluation*

This evaluation was a part of a joint work with Khadija Elbedweihy and is reported in [EMWC14].

One of the main observations during the previous user evaluations was aimed at understanding how users make use of the additional features provided with the interface. In particular, specific attention was paid to how users made use of the search functionality provided in the Section C (Figure 53). Most users were observed to heavily use the concept search, which is used to look-up an index of all the concepts to find and load any relevant concept, and the high dependence of the user on this function motivated

the next stage of research. Once the relevant concepts were loaded, users visually investigated their connections and properties to build their queries. For example, a task that requires a user to find information related to authors writing a paper on a particular topic, users preferred to conduct two concept searches (Author and Publication) instead of browsing through the pie-charts of concepts to look for the relevant concepts. At the same time, users liked the idea that once they reached the concepts, they could see how they were connected and that could help them build precise queries and apply constraints.

Despite evaluating with small datasets, it was observed that users preferred a text-search to find relevant concepts instead of visual look-ups. This raised the need for further investigation into a more text-driven approach to initiate the exploration process specifically for larger datasets, as visually exploring a large number of pie charts is expected to be a longer and more exhaustive process (in cases where a specific Information Need exists). A hybrid approach was thus developed, which plugged-in a natural language component to the Affective Graphs interface. The idea of the hybrid approach is to provide users with a convenient starting point if they have a highly specific task at hand. Users can enter keywords or natural language questions to reach the relevant concept and properties. Affective Graphs then tries to interpret the query and build a corresponding interpreted visual query, which can then be fine-tuned by the user as desired.

6.9.4.1 *Integration with Natural Language*

An effective natural language (NL) system is highly complex and suffers from several possible issues such as applicability in multiple domains, adaptability, accuracy etc. Owing to possible inaccurate, imprecise and incomplete responses, users may not always reach the information they look for. However, using such a system to provide an initial set of concepts can be extremely helpful and convenient to users without prior knowledge of the domain or dataset. This approach attempts to exploit a NL approach to quickly provide seed concepts, visualising which can enable users to build a highly complex, specific and fine-tuned query. The interest in this phase of research, was in understanding how a text-driven approach could help users quickly reach the concepts they are interested in, and further fine-tune their queries once they visualise the relevant concepts.

In this implementation, a natural language system indexed with the DBpedia dataset²⁴ [EWC] was integrated with Affective Graphs. In a typical Knowledge Management scenario, most enterprise search systems can be plugged-in with minor configurations to provide relevant concepts and properties. Highly specialised services (for e.g. [BC10] in the aerospace domain or the Biportal annotation service²⁵ in the biomedical domain) are capable of identifying domain-specific entities, concepts and relations in text. Such systems can be employed as a service layer to transform a natural language query to its constituent concepts and relations in domain-specific organisational infrastructure. The discussion in this thesis would focus on how the natural language system was incorporated within Affective Graphs, instead of its implementation and performance.

²⁴ <http://dbpedia.org>

²⁵ <http://biportal.bioontology.org/annotator>

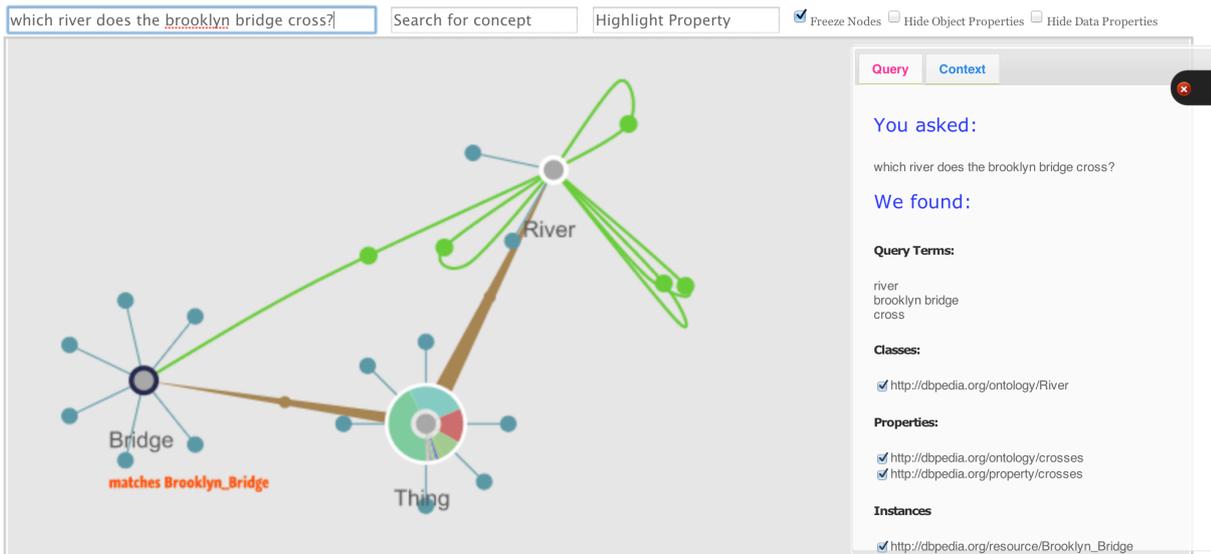


Figure 67: Incorporating a natural language search box to provide means for users to enter questions and keyword based queries (top). In this example, the user has just entered a query, and Affective Graphs renders the identified concepts and attempted to build a query from the retrieved terms. The queries are analysed to identify concepts which are then subsequently loaded in the visualisation, as well as displayed on the right Query panel

The main design challenge was in understanding how to incorporate a natural language search bar with the system. Preliminary suggestions of an initial Google-like search screen where users typed their queries on a simple text box seemed to be intuitively challenging, as the result from the text search would not be a textual list of results, but a graphical visualisation of the search concepts. Hence, the final design was to modify the Section C of Figure 53 to incorporate a search box as shown in Figure 67, top.

In addition to a natural language input box, the Affective Graphs interface was also modified to provide feedback of what concepts and properties the natural language system could identify. The context window was updated (Figure 67, right) to incorporate a query panel which presented all the concepts, query terms, properties and instances that were identified in the query. In the example shown in the Figure 67, the user enters a text query ("which river does the brooklyn bridge cross"), which results in several terms (river, brooklyn bridge, cross) being identified. The response from the NL component provides further semantic information for these terms and their values, such as river is a class, brooklyn bridge is an instance of the bridge class and cross is a property that connects river and bridge. The response from the NL service is a JSON object as follows:

```
{
  "terms": [
    {"queryTerm": "river", "type": "Concept", "uri":
      ["http://dbpedia.org/ontology/River"]}
  ],
  {"queryTerm": "brooklyn bridge", "type": "Instance", "uri":
    ["http://dbpedia.org/resource/Brooklyn_Bridge"],
    "ontologyType": "http://dbpedia.org/ontology/Bridge"},
  {"queryTerm": "cross", "type": "Property", "uri":
    ["http://dbpedia.org/ontology/crosses*&http://dbpedia.org/ontology/Bridge*&http://dbpedia.org/ontology/River",
```

```
"http://dbpedia.org/property/crosses*&*http://dbpedia.org/ontology/Bridge*&*http://dbpedia.org/ontology/River"] ]
}
}
```

As can be observed, the **JSON** object provides three types of query terms- instances, concepts and properties. For instance, the query terms *river* and *crosses* are matched to the concept `dbo:River` and the properties `dbo:crosses` and `dbp:crosses` respectively. The property description also includes its domain and range. The **JSON** object, upon being received by Affective Graphs, is parsed. The concepts (classes) are analysed first, and all the unexplored concepts are loaded. The next step analyses the instances, and attempts to build partial queries following each instance identified. The relevant concepts are first added to query, and the instances are then added as constraints of the concept. The final processing occurs with the properties — all the domain and range concepts are first loaded, following which each identified property is added to the query. The final result of the processing provides users with a visual representation of the concepts and all their properties, the relevant entities highlighted as queries. This provides users with the information that there are several properties that may be relevant. Affective Graphs already pre-builds the visual query (by adding `bridge` to query, and subsequently adding ‘brooklyn bridge’ as a constraint) as much as possible, allowing the user to complete the query without requiring the user to conduct extensive visual look-ups. Items on the query panel on the right can also be selected/de-selected to modify the visual query as required. For example, if the **NL** system returns several possible properties, the user can de-select the irrelevant ones from the right panel.

The next process of fine-tuning the query (such as adding any other concept, property or constraint) remains the same. Finally, once the visual query is built, the user can view the formal query and then search for answers. The results are then returned in a tabular format like previous versions.

The motivation for the following evaluation were two-fold:

1. Previous evaluations observed users exploiting the concept-based search feature to a significant extent for a smaller dataset. A hybrid approach seemed to be a good opportunity to combine the two approaches to develop easier means to reach the concepts of interest for larger datasets.
2. As a scheduled evaluation to evaluate how the system performs with a large dataset.

Though the hypothesis that a hybrid approach can be effective in helping users build visual queries, the applicability and implementation of the same may not be as straightforward since two approaches are complete opposite in terms of their interactions (as presented in Figure 15 and 16, where a typical natural language approach and highly visual approach lie on two different spectra of the formality continuum, visual abstraction and interactivity scales). While natural language systems are designed to be a very quick and efficient system, where users can try several queries until they reach their result, graphical and visual approaches require more patience and attention to formulate complex queries in a highly structured and logical manner. Hence, it is important to understand how the two approaches can be combined together to help the user cognitively conduct their tasks without being overburdened by the prominence of one approach over the other. It is also important to understand if the user’s perception of the

aesthetic appeal of Affective Graphs is preserved when introduced with a slightly different interaction mechanism as well as a much larger dataset. A much larger dataset can provide significantly more challenging conditions for users to overcome, and though the interface has not undergone significant changes, the change in interactivity may affect a user's perception of the system. Thus the following evaluation was planned to understand:

1. How easy (in terms of time and effort required) it is to use the hybrid approach to perform tasks of different complexity ?
2. Can the approach support large scale datasets?
3. How does the hybrid approach on a large dataset affect a user's perception of the aesthetic appeal of the interface?
4. How differently do casual users and experts behave with the system?

6.9.4.2 *Method*

A formal evaluation with expert and casual users was conducted with Affective Graphs in a controlled laboratory setting. The evaluation was conducted over a period of 7 days, in the month of October, 2013. Observations during the experiments were noted down, along with the times to be analysed later on. Minor events and interesting observations such as users being disoriented, confused, trying new features were noted while the experiment proceeded, as well as post-evaluation.

6.9.4.3 *Procedure and Setup*

Affective Graphs, coupled with the NL component was connected to the live installation of DBpedia, available from <http://dbpedia.org/sparql>. In addition to the live endpoint, another endpoint was made available by the QALD workshop²⁶, which was loaded with the same data. This endpoint was used as an alternate endpoint if the primary live endpoint failed at any time during the evaluation. There were three primary motivations for choosing DBpedia: it is a domain-independent datasets and all users were familiar with Wikipedia²⁷ as a knowledge source; it is one of the largest semantic datasets available as Linked Open Data; and the QALD workshop already made available a bank of questions that search systems can be benchmarked against.

Following an analysis of the QALD questions, four questions were selected which resembled a combination of features as the MCT and CT tasks identified in Section 6.9.3.1. The questions for the evaluation were as follows:

1. Q1. How did Bruce Carver die?
2. Q2. List all the songs released by Bruce Springsteen between 1980 and 1990.
3. Q3. When was Capcom founded?
4. Q4. Who was the wife of President Lincoln?

²⁶ <http://greententacle.techfak.uni-bielefeld.de/~cunger/qald/index.php?x=task1&q=3>

²⁷ http://en.wikipedia.org/wiki/Main_Page

5. Q5. List the cities in Alaska with more than ten thousand inhabitants

ST and MAT were not considered as they were relatively simpler to perform and most NL systems would be effective at solving the tasks by themselves. For example, the birthdate of a person is a simple query to be answered by very basic systems. Hence, for the context of this evaluation, these queries were not considered.

The evaluations were structured as follows — the participants would be initially briefed on search approaches, providing examples from the system. A sample query then would be used to explain how Affective Graph's visual approach can be used to build queries. A further example query would be used to explain how the natural language interface interprets a text query and provides semantic concepts. The users would then be asked to complete the query to build a final query and execute the search. Once users were comfortable with the system, the evaluation controller developed by the SEALS project would be initialised, which provided questions in a randomised order and kept track of user performance in the background. Most users took typically between 10 and 20 minutes to solve all the five tasks, while the total time for the evaluation session lasted typically between 30 and 40 minutes.

6.9.4.4 *Participants*

24 participants in total were involved in evaluating the system. Participants comprised of 12 casual users and 12 experts, aged between 18 and 53, with a mean age of 31 years. Casual users were drawn from a pool of staff and student population at the University of Sheffield. Expert users from the Organisations, Information and Knowledge group at the University of Sheffield and K-Now²⁸, a software development company were invited to participate in the evaluation. Experts had knowledge and experience working with Semantic Web technologies and standard Semantic datasets.

6.9.4.5 *Data Collection and Analysis*

Both objective and subjective data was collected to allow deeper analysis from a qualitative and quantitative perspective. Interaction logs from Affective Graphs as well as the evaluation controller provided quantitative data, which were later analysed to provide input time. Users indicated when they were satisfied with the results obtained, analysing which, the success rate and number of query iterations were obtained.

Qualitative data was also collected for future analysis, that included user satisfaction questionnaires, evaluator observations, discussions and feedback session at the end of the evaluation. The subjects were presented with three questionnaires – a System Usability Scale questionnaire, a demographics questionnaire and an extended questionnaire, as shown in Appendix F.

The statistical package, SPSS was used to analyse the data. The analysis was conducted in two ways – to understand how long users took to formulate a satisfactory query and to understand how many iterations of queries users had to undergo to reach satisfactory answers. The subjective data was analysed to understand: how users appreciate or criticise the approach, how can the system be improved, general comments on how such approaches can be incorporated in standard knowledge practices of the

²⁸ <http://www.k-now.co.uk/>

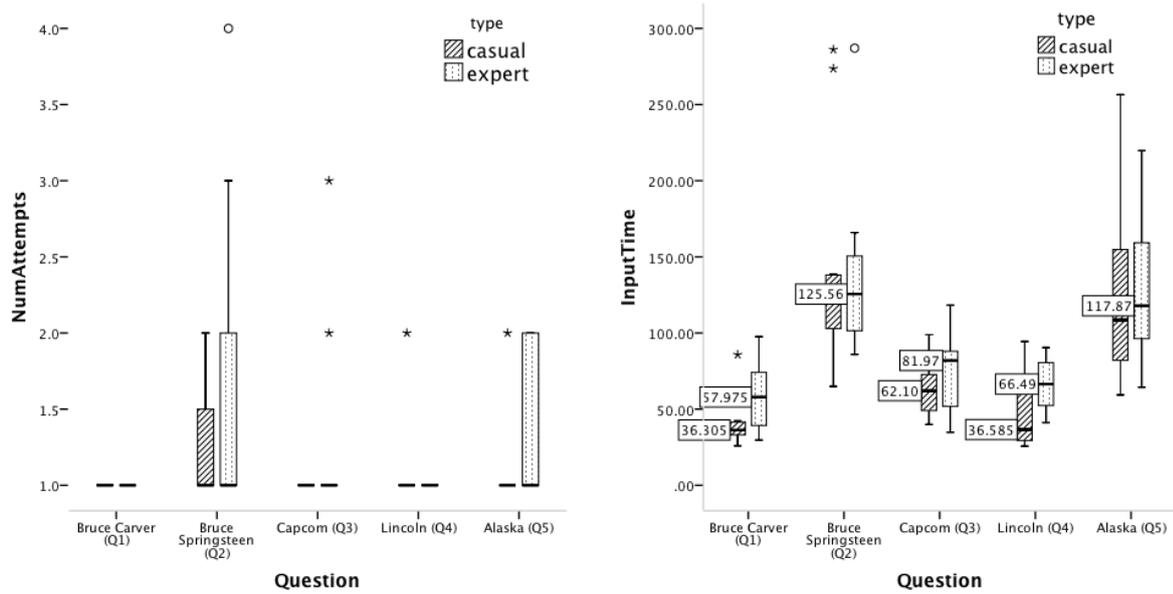


Figure 68: Analysis of hybrid search approach, grouped by experts and casual users. Figure on the left shows the number of attempts for each query, where a few queries required more than one attempts. Figure on the right shows the input time taken to complete formulating a query.

users. The analysis was conducted by grouping the data by query as well as user type. Additionally, the aesthetics questionnaire was analysed to assess how the users judged the aesthetic appeal of the system.

6.9.4.6 Results

Overall, all the users could perform all tasks presented to them, and were satisfied with the results. As with the previous evaluation, the discussion will involve two main findings: Efficiency and Effectiveness, and User Satisfaction.

EFFICIENCY AND EFFECTIVENESS Figure 68, left presents a boxplot of the number of attempts for every question, grouped by the user type. The figure shows that most users could find their answers in one attempt for all the questions, apart from a few outliers. The two questions with a greater variation were Q5 and Q2. These questions required the users to manually enter constraints as a way of fine-tuning queries. Most users could find the results in one attempt, while five (1 casual, 4 experts) required two attempts to answer the query on Alaska (Q5). It was observed that the greater number of experts requiring more than one attempt to answer the query was due to trying out different natural language queries. Some experts tried different variations of the query (e.g. “Alaskan cities 1000 people”, “city Alaska inhabitants” or “cities Alaskan population”). A few users also added a different constraint to other properties to formulate the query — for e.g. instead of the ‘populationTotal’ property, they used ‘populationUrban’ or ‘populationDensity’ (Figure 69, Left). Some users also had difficulty in understanding some of the names of properties in the dataset — e.g. in the query Q5, the concept ‘City’ is connected to the concept ‘Administrative Region (Alaska)’ with three properties

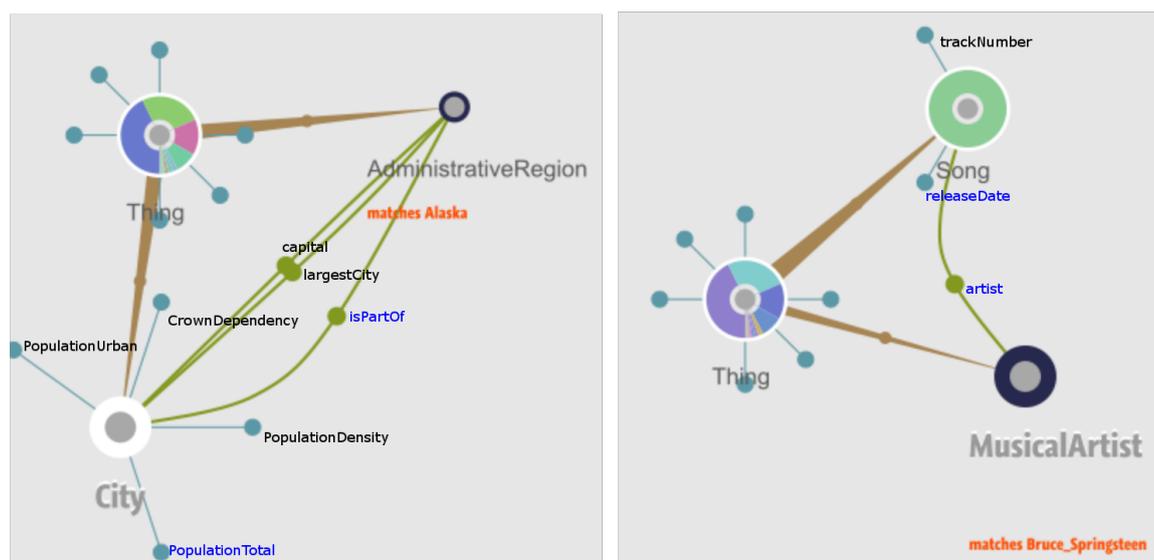


Figure 69: Two queries provided some challenges for users — Left, Q5 required the two properties `isPartOf` and `populationTotal` (marked in blue) to be added to query with a single constraint on `populationTotal`. Right, Q2 required `artist` and `releaseDate` (marked in blue) to be added to query with multiple constraints added to `releaseDate`

`'isPartOf'`, `'largestCity'` and `'capital'`. While the property `'isPartOf'` would provide the required answer, users did not find the property name intuitive enough to select it at first, resulting in a few re-attempts.

Q2 showed the maximum variation in the number of attempts. Three casual and expert users took two attempts, while two other experts took 3 and 4 attempts respectively. The formulation of the constraint presented users with a challenge — this was observed to be related to the implementation. Q2 requires users to identify a relation (`artist`) between the concepts `'Song'` and `'MusicalArtist'`, as well as add two date constraints on the property `'releaseDate'` to create a range filter of 1980 — 1990 (Figure 69, Right). The implementation required users to add the two constraints sequentially, by repeating a few steps. Further on, the date constraints required users to select from a calendar display the dates starting from 1/1/1980 to 31/12/1990. This process caused difficulty among users as despite being presented with a pop-up calendar, users preferred to enter `'1980'` and `'1990'` in the date input field. This was identified by most users as a difficult step, and few users suggested removing the calendar UI element and replace it with a simple text input box using simple Javascript validations. Users also suggested to enable a feature where multiple constraints can be added at one time, instead of a series of sequential steps.

The other three tasks were performed with a median of one attempt, while a few outliers do exist, as shown in the figure. This was due to a few interaction issues and the users identifying different properties than the ones required. For e.g. Q3 required users to identify properties `'foundingDate'` or `'foundingYear'` as query properties. However, some users selected `'foundedBy'` and `'founded'` properties. Upon receiving irrelevant results, users re-tried the queries with the right ones.

Input time, or the time taken by a user to compose a satisfying query can also provide a good indication of how well the system can perform. Figure 68, right show how

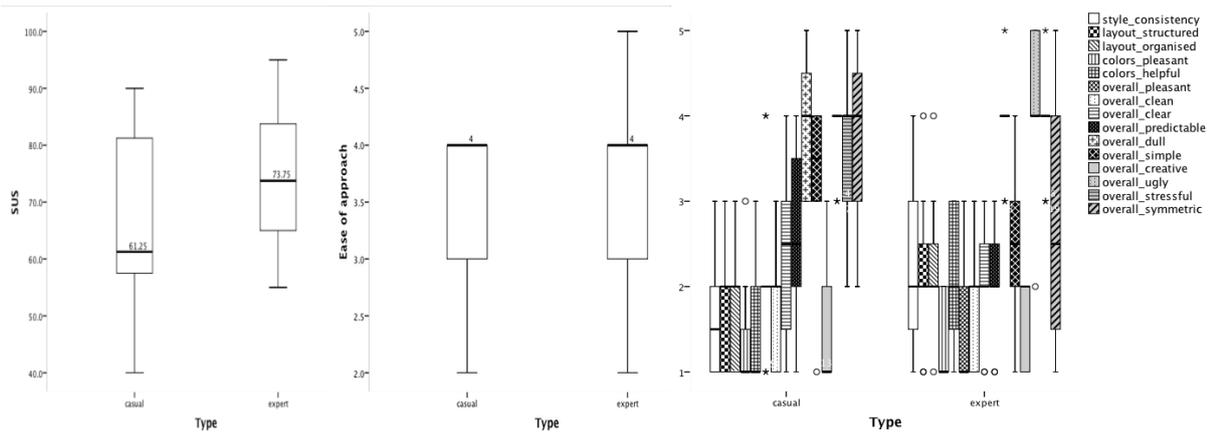


Figure 70: Qualitative data analysis — Figure on the left shows a boxplot of SUS scores grouped into expert and casual users; Middle figure shows a boxplot of overall ease of approach, grouped into expert and casual users; Right figure shows the aesthetics scores provided by users

the two types of users perform with the different questions. The maximum input times recorded were 287 and 286 seconds for the Q2 question, by an expert and a casual user respectively. The minimum time recorded by a casual user was 26 seconds for the Q4 query, while the minimum time for an expert was 30 seconds for the Q1 query. On an average, it can be observed that casuals (median input time of 76.88) were generally quicker than experts (median input time of 94.48) in answering all the questions. Such a finding was unexpected, as it was expected that experts (given their understanding of Semantic Web concepts and querying techniques) would be quicker at grasping the concepts and therefore formulating the queries. One of the reasons for this unexpected result could be attributed to an observation made during the evaluations: given a better understanding of natural language systems and formal query techniques, the experts tended to investigate various technicalities within the system. For example, some experts manually looked up the SPARQL query thereby generated, trying to understand how the query was being built.

Another reason was that experts, given a better understanding of query formalisms, were inclined to follow a more systematic approach and follow every step meticulously, re-checking every step once a query was built before executing it. Casuals, however, with a relatively less experience with formal queries tended to follow a more casual approach, and therefore were quicker with their querying. The final reason that was observed was that experts were trying several variations of the natural language query as input to the NL system, which resulted in several re-attempts as well.

USER SATISFACTION Three questionnaires were presented to the users at the end of the evaluation session similar to Section 6.9.3.1- a general usability questionnaire (ten 5-point Likert questions for computing SUS scores), an extended questionnaire (a single Likert question on the difficulty of the approach and two open-ended questions for the positive and negative experience of the hybrid approach) and an aesthetics questionnaire (fifteen 5-point Likert questions).

Figure 70, left shows a boxplot of the SUS scores provided by the users, grouped by the type of users. Overall, it was observed that experts liked the system considerably more than casuals. A median score of 73.75 and mean of 75 from the experts show that the system was well appreciated. Casual users also rated the system highly with a median score of 61.25 and mean of 64.58. While a direct comparison with the previous evaluations cannot be drawn due to completely different experimental setups, these results were encouraging. In the first evaluation with casual and experts, casual users rated the tool as 55, while experts rated it as 63.75. Most users were excited with the playful and interactive nature of the system (2 experts and 1 casual user provided very high SUS scores of over 90).

Several users (both experts and casual users) seemed to be interested about the system and inquired when and how will the system be made available online. The users also noted that such systems would be important for the research they conduct as well as their daily tasks — for e.g. a politics student mentioned her need for such a system to understand how different countries and their legislative policies are framed and how they are interlinked. Another user from the music studies department appreciated the system and expressed the need for such an approach while looking for albums or records of musicians based on different genres.

The comments mentioned on the feedback questionnaire for the open ended questions were very positive toward the hybrid approach as well as the visual approach in general. Comments such as:

- *“It was useful to have all the relevant entities and classes preloaded onto the diagram, and they were usually the right ones.”*
- *“I liked that the system automatically identified the main concepts from the query, so that the exploration process was faster.”*
- *“I think it allows easy exploration of non-specific / vague queries, whilst simultaneously allowing you to head directly to your target if you DO know what it is. Pretty cool, and flexible.”*
- *“Providing the NL query first was very quick and user friendly. Most of the time, the system provided the correct query or very close to it — only need a little bit of tweaking. This made it fast to formulate queries.”*
- *“I like the hybrid approach because it was a visual representation of the search I was carrying out, allowing me to see every step of my search and how my search can be changed to find different search results.”*
- *“It was very visual and you get a good sense of the types of questions that you can ask from the data even if you do not know the data.”*
- *“I appreciated the link between visual data and the ability to be able to short cut to get to the data i want”*

showed they appreciated the approach where their natural language query was interpreted into semantic concepts and entities and subsequently represented as visual objects. Comments such as *“easy to type natural language, I like interacting with the visual*

side.” and “Graphical representation of the relationships between different concepts was interesting” show that the users also appreciated the visual approach of querying.

Comments such as “Good as a transition from the text based query approach I am used to. Best of both worlds. My preferred learning style is visual so this may have helped the ease of understanding visualisation of data. The nodes and arcs show data connections well. The options to refine searches were obvious and well set out. This approach reduces the many pages of results from a standard eg Google search.” show that some users appreciated the approach and the system as a whole, but also a personal preference and choice plays an important role in user appreciation of the approach. Contrary to the user, however, another user mentioned a preference to a text-driven approach though she appreciates a visual approach: “Whilst the graph was utterly gorgeous, it seemed to distract and make a targeted search (such as the ones given in the tasks) take longer than it needed to. I suspect that personally, unless I was looking for inspiration / brainstorming, I’d usually prefer to use a more text-based format most of the time. That said, I am not sure if I’m a typical user — for example, I detest news in un-necessary graphical format and won’t watch a video if there’s a text-based news article which reflects the same thing”.

Three users in particular (2 casual users and 1 expert user) were more critical of the system with SUS scores ranging between 40 and 50. Two users (an expert and a casual user) mentioned that they felt the step of entering a text query in the NL search box and then visualising it to refine the query and finally obtaining results seemed like an additional step compared to a Google-like approach they are more accustomed to: one user mentioned the following as a negative point of the approach — “It seemed an extra step to get to your answer rather than just typing in a search and it appearing in results.”. The same user, however commented about the hybrid approach: “It is unusual. I wouldn’t like to do this purely from the graphics. I liked the use of logic to get to the answer.”. Upon reflection, perhaps a solution to the contradictory views would be to directly provide answers to queries when the user enters a search. For cases where a satisfactory answer could not be provided, a more visual approach would be followed via Affective Graphs. A simple redesign, however would not be the ideal solution, since a seamless transition with a smooth flow between the two approaches is desired, which needs a considerable amount of insight and research.

Negative comments provided by users could suggest possible improvements and criticisms could help understand what are the issues with the approach. These comments were grouped into two categories — approach (what users did not like in the hybrid approach) and implementation (what users did not like with the existing implementation). A familiarity with a Google-like keyword-based approach meant that some users (specially casual users) already had a pre-conceived notion regarding search, and a deviation from the same raised a few concerns among users (as noted previously). In this regard, several discussions ensued where participants were explained how semantic search is different and listing cases where typical keyword-based approaches would not provide sufficient results. The existing bias toward an established approach was observed in few users, who compared the need for semantic search approaches with existing search systems.

Some users (mostly experts) also noted a few drawbacks of the NL system, where several variants of questions were tried during the evaluation:

- “It doesn’t always match the terms you are looking for.”

- *“The free-text component sometimes doesn’t recognise words although they are an exact match.”*
- *“It was quite complicated to build from the start by yourself if it didn’t bring up matches from the initial search question.”*
- *“Need several steps to get my search. Sometimes can not parse my query correctly.”*
- *“sometimes not clear what the system was picking up and why.”*

These comments highlight some drawbacks of free NL systems that may exist and are always a concern for NLP developers — they do not always pick up the concepts that are of interest. However, while the performance and development of the NL component is out of scope for this thesis, its applicability in a hybrid approach, coupled with a visual approach was evaluated positively with experts and casual users alike. It is expected that with a continuously evolving and improving NL component, these comments would be reduced and a better starting point for the users can be achieved.

Some implementation issues were highlighted by the users, which were not observed in earlier evaluations (primarily due to (a) the type of queries being evaluated this time — i.e. multiple constraints on properties and (b) new features introduced in the system):

- *“adding constraints is a bit labourious as you need to open the panel again for adding each one, also it would be good to have an edit button rather than just remove one”*
- *“would have been nice to see the visualised query update in real time as concepts etc were ticked / unticked on the right hand side. (query panel)”*
- *“the add constraint option was not entirely clear i.e. when looking at different dates it was tricky to know what the symbols meant.”*
- *“consider a spelling correction that picks up from possible values in dbpedia for example”*
- *“consider clarifying the query results with the user, i.e. when performing the search for lincoln you could point out there is more than 1 match for lincoln, which one would you like (kind of like the BBC weather forecast)”*

These comments provide invaluable feedback and insight into how the system can be improved as well as highlight issues that exist within the present implementation of the system. The comments regarding adding constraints note that the users experienced some difficulty in performing these tasks and they are reflected in the quantitative data analysis where a greater input time and number of attempts were observed for few users. Two users also mentioned the need for a few learning sessions and noted that a few features may be difficult to memorise. Users also experienced some difficulty with the experimental conditions. For e.g. the evaluations were conducted on a macbook, and some users being windows users, found it difficult to use the mousepad (though an external mouse was provided). One user also noted that a bigger screen would be helpful to perform the tasks specially due to the visual approach.

The aesthetics questionnaire contained the same questions as with the previous evaluation — requiring users to judge the system on four criteria: style, overall aesthetics, layout and colours. Figure 70, Right shows how the two types of users judged the system on these criteria. Casual and expert users noted that the style of the system was

consistent, scoring a median 1.5 and 2 (1 being the most positive and 5 being the most negative). In terms of layout, most users noted the system was well structured and organised (a median score of 2 for both types, where 1 being the most structured and organised, 5 being the least). Two users, among the experts noted that the high score (as an outlier) marked the layout as not well organised, since they preferred the ‘Search’ button to be on the right hand panel, instead of the bottom as it would reduce continuous scrolling. Both types of users found the colours in the system pleasant, scoring 1 (1 being most positive and 5 being least). The colours were also judged to be helpful by both type of users (casual users scored a median of 1, while experts scored 2).

The system was judged positively in terms of overall aesthetics- users found the system clean, scoring it 2 (experts and casuals). Experts scored the system clean (2), while casuals provided an average score (2.5). Both types of users found the system relatively complex, with casuals (3.5) scoring the system more complex than experts (2.5), 1 denoting a simple system, 5 denoting a complex system. Casual found the system very creative (median score of 1, where 1 denotes creative and 5 denotes unimaginative), and experts scored the system highly on creativity (median score of 2). Both users felt that the system is beautiful, scoring a median of 4 (1 denotes ugly, 5 denotes beautiful). Both users also agreed that using the system was exciting, scoring a median of 4 (1 denoting stressful, 5 denoting exciting).

The greatest variation was observed in the scores of symmetry — while casuals seemed to agree that the system was mostly asymmetric, experts felt that the system inclined to a more symmetric one. Most users had a doubt on this question, and mentioned they feel the system is mostly symmetric, but in a positive manner. Most users felt that the asymmetry in the system helped it be more effective, and they would not suggest any changes that could affect the symmetry. Though the aesthetics are highly subjective, the feedback and comments from users showed that they enjoyed the experience of working with the system and felt it was a creative and beautiful solution.

6.9.4.7 *Discussions, Key Findings and Limitations*

The evaluation highlighted several key findings: the most important being, the observation that most users were able to retrieve satisfactory answers at the first attempt. This was, in spite of a very large dataset like DBpedia being queried. The addition of a natural language component assisted users in reaching their concepts of interest quickly, and hence, users could start building queries without requiring to follow several concepts. The next finding, perhaps expected to an extent was that experts generally took longer than casual users to formulate a satisfactory query. This can be explained by the general curiosity of experts, who could relate the approach to their conceptual understanding of semantic querying. Experts were also observed to be more careful and meticulous in formulating their queries, hence required longer query input times. Some experts took time to validate complex SPARQL queries by reading through the queries manually to understand how the system gradually builds the queries. Overall, experts liked the system more than casual users, which can also be explained by the familiarity of experts with semantic web principles. Some users commented on the need for an additional step after a Google-like approach of a natural language querying. Overall, all users found the querying experience a enjoyable, stimulating and creative approach.

A few minor interaction issues were also identified – most users experienced some difficulty setting date filters using the date selection widget, where users were expected to select a date. However, some users expressed that instead, a textual way to enter dates in either MM-DD-YYYY (or similar) formats would be a better approach. Some users were also slightly frustrated by the way constraints are created: presently, multiple constraints can be added on a concept/property by selecting “add constraint” from the context menu drawn by right-clicking on the visual item multiple times. However, users suggested to enhance the ‘add constraint’ dialog so that multiple constraints can be added at one time.

A major limitation of this study was that the evaluation considered the hybrid approach as an isolated query approach. A larger scale comparison with both of the two approaches (natural language and graphical) is essential as a part of a later evaluation. This is to help validate how the potential benefits of the two approaches can be translated in a hybrid manner, compared with each other in similar experimental conditions. A further evaluation with a larger variety of tasks (exploratory, fact-finding etc.) would be helpful in understanding which approach is most appropriate for the type of task. However, at the time of development, this evaluation was deemed the most appropriate since it was important to understand the first reactions from users, while performing a smaller set of tasks.

6.9.5 *Evaluating Aesthetic Properties of Affective Graphs*

Several rounds of re-design after consecutive user-feedback resulted in a version of Affective Graphs that was relatively mature and ready to be evaluated with a final set of users. In order to understand the aesthetic properties of the interface, the same metrics as previously used to compare existing Semantic Web interfaces (see Section 6.3) are used to compare the systems with Affective Graphs. This evaluation was aimed at answering two major questions:

- How does the system compare with respect to the system that was judged to be the most aesthetically pleasing tool?
- How does the system compare with the highest score achieved by any tool for the individual categories?

Section 6.3 showed that the Semantic Web interfaces that were earlier analysed had relatively low scores, with the highest score obtained by any tool being 0.65 out of a maximum possible score of 1. The eight measures of aesthetic properties were calculated for Affective Graphs and compared with two other sets of scores — the tool which scored the highest in the previous experiment, mSpace. Additionally, Affective Graphs was compared with the highest scores obtained by any tool in each of the individual criteria. The experiment identified the tools which obtained the highest scores for different categories — balance (Sig.ma), Equilibrium (Sig.ma), Symmetry (mSpace), Sequence (PowerAqua, NLP Reduce), Rhythm (K-Search), Cohesion (mSpace) and Unity (NLP Reduce).

Similar to the previous experiment, the scores for each metric are calculated three times. However, in this experiment, the highest scores obtained by each tool, since the best scores are of interest in this evaluation. Figure 71 shows a comparative plot of

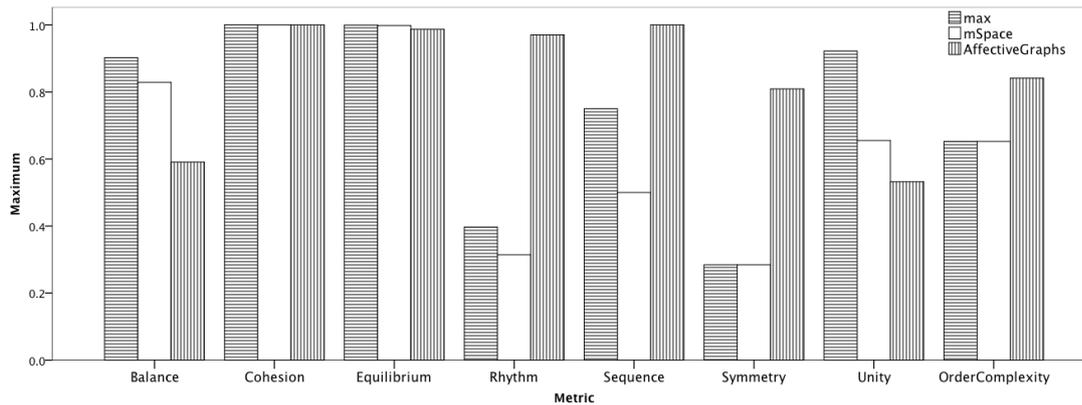


Figure 71: Comparison of Affective Graphs with the highest scoring tool and the maximum scores in each aesthetic metric

the best scores obtained by any tool, best scoring tool and Affective Graphs. The figure shows that Affective Graphs scored the highest in four out of the seven categories (Rhythm, Sequence, Symmetry and Cohesion). The system scored lesser than the other two in balance and unity. Equilibrium scores are marginally lower than the other two, with Affective Graphs scoring 0.987 as compared to 0.999 by the highest scoring tool (Sig.ma).

Overall, as can be seen from the graph, Affective Graphs scored the highest, significantly higher than the best scores obtained by the Semantic Web tools (an order and complexity score of 0.8404 as compared to 0.6523). Whilst these scores, are by no means conclusive in deciding the most aesthetically pleasing interface, the positive results serve as a good indication toward developing a more pleasant experience for users. The intent, in this evaluation was not to judge an interface as the most aesthetically pleasing one, but to explore an alternative way of objectively evaluating Semantic Web interfaces and assessing how the implemented system scores with respect to existing systems. This is important as the system was designed considering aesthetics as one of the most important features.

6.9.5.1 Limitations

An objective evaluation as the one explored can only provide an indication of the aesthetic properties of a system. However, the truest reflection on the aesthetic appeal of an interface can only be provided by a subjective judgement of the users. Personal preferences, bias, societal impact, learning experiences and other factors influence a user's judgement and preference for a particular interface. This makes it an extremely difficult task to assess a real user's perception to an interface. While the significant role of users in determining the pleasurable quality of an interface is acknowledged, early objective analysis of the aesthetic principles is helpful and can provide a starting point for development. As previously discussed, interface layout is only one of the several factors that contribute toward the aesthetic appeal of an interface. However, other factors such as colour and texture are considerably difficult to formulate in order to ascertain an objective value [NTB03].

6.10 GENERIC MECHANISMS OF INTERACTION

The requirement of Affective Graphs to be a generic solution, independent from datasets and domains needed a unique way of translating user actions. Representing concepts and properties in a typical manner (as traditional graph and network representations) provided the first step in doing so. However, the approach also needed to address requirements such as querying, exploration and providing overviews while being a highly interactive and visual one. This resulted in the development of several rules and templates, that can translate low level user interactions to intents and then finally to SPARQL queries. this section presents the rules/templates involved in these processes – while an overview of the processes have been provided earlier in Section 6.7, this section presents the generic templates used for the processes. The rules are presented in the Table 5

Type	Target	Context	Intent	Query Template	SPARQL
1. Init	Owl:Thing node	Initial Node	Initiate Interaction with 3 initiating queries	subclass request domain request range request	<pre> SELECT distinct ?subClass count (?x) as ?count ?label WHERE { ?x a ?subClass. ?subClass rdfs:subClassOf owl:Thing. ?subClass rdfs:label ?label. FILTER langMatches(lang(?label), "EN") }order by desc(?count) SELECT distinct ?prop count(?instance) as ?count ?domain WHERE { ?prop rdfs:range owl:Thing. ?prop rdfs:domain ?domain. ?instance ?prop ?obj. }order by desc (?count) SELECT distinct ?prop count(?instance) as ?count ?range WHERE { ?prop rdfs:domain owl:Thing. ?prop rdfs:range ?range. ?instance ?prop ?obj. }order by desc (?count) </pre>

2. Click	Section (subclass X) of a node	Parent Node	Explore new section with 3 queries	subclass request domain request range request	<pre>SELECT distinct ?subClass count (?x) as ?count ?label WHERE { ?x a ?subClass. ?subClass rdfs:subClassOf X. ?subClass rdfs:label ?label. FILTER langMatches(lang(?label), "EN") }order by desc(?count) SELECT distinct ?prop count(?instance) as ?count ?domain WHERE { ?prop rdfs:range X. ?prop rdfs:domain ?domain. ?instance ?prop ?obj. }order by desc (?count) SELECT distinct ?prop count(?instance) as ?count ?range WHERE { ?prop rdfs:domain X. ?prop rdfs:range ?range. ?instance ?prop ?obj. }order by desc (?count)</pre>
3. Right Click – add to query	Node X, representing a concept	Node	Indicate node X is of interest in the query	Add node concept to query	<pre>SELECT * WHERE { ?XConceptValue a <XConcept>. OPTIONAL {?XConceptValue rdfs:label ?label. FILTER langMatches(lang(?label), "EN")}. }</pre>
4. Right Click - add constraint, enter filter restriction ("filterTerm")	Node X, representing a concept	Node	Apply a filter restriction on node	Add filter constraint to concept	<pre>SELECT * WHERE { ?XConceptValue a <XConcept>. OPTIONAL {?XConceptValue rdfs:label ?label. FILTER langMatches(lang(?label), "EN")}. FILTER (regex(?XConceptValue, "filterTerm", "i")) }</pre>
5. Right Click - add to query	Object Property X	Property X connecting Node A and Node B	Indicate property X is of interest in the query	Add connected nodes (A and B) to query Add property X to query	<pre>SELECT * WHERE { ?ConceptAValue a <ConceptA>. OPTIONAL {?ConceptAValue rdfs:label ?ConceptALabel. FILTER langMatches(lang(?ConceptALabel), "EN")} . ?ConceptBValue a <ConceptB>. OPTIONAL {?ConceptBValue rdfs:label ?ConceptBLabel. FILTER langMatches(lang(?ConceptBLabel), "EN")}. ?ConceptAValue <PropertyX> ?ConceptBValue. }</pre>

6. Right Click - add to query	Data Property X	Property X connecting Node A	Indicate property X is of interest in the query	Add connected node A to query Add property X to query	<pre>SELECT * WHERE { ?ConceptAValue a <ConceptA>. OPTIONAL {?ConceptAValue rdfs:label ?ConceptALabel. FILTER langMatches(lang(?ConceptAValue), "EN")}. OPTIONAL {?ConceptAValue <PropertyX> ?PropertyXValue}. }</pre>
7. Right Click - add constraint, enter text filter restriction ("text filter")	Data Property X	Property X connecting Node A	Apply a text filter restriction on property	Add filter constraint to property	<pre>SELECT * WHERE { ?ConceptAValue a <ConceptA>. OPTIONAL {?ConceptAValue rdfs:label ?ConceptALabel. FILTER langMatches(lang(?ConceptAValue), "EN")}. OPTIONAL {?ConceptAValue <PropertyX> ?PropertyXValue}. FILTER regex(?PropertyXValue,"i","text filter"). }</pre>
8. Right Click - add constraint, enter numeric filter restriction (numericfilter)	Data Property X	Property X connecting Node A	Apply a numeric filter restriction on property	Add filter constraint to property	<pre>SELECT * WHERE { ?ConceptAValue a <ConceptA>. OPTIONAL {?ConceptAValue rdfs:label ?ConceptALabel. FILTER langMatches(lang(?ConceptAValue), "EN")}. OPTIONAL ?ConceptAValue <PropertyX> ?PropertyXValue}. FILTER (?PropertyXValue = numericfilter). }</pre>
9. Right Click - add constraint, enter date filter restriction	Data Property X	Property X connecting Node A	Apply a date filter restriction on property	Add filter constraint to property	<pre>SELECT * WHERE { ?ConceptAValue a <ConceptA>. OPTIONAL {?ConceptAValue rdfs:label ?ConceptALabel. FILTER langMatches(lang(?ConceptAValue), "EN")}. OPTIONAL {?ConceptAValue <PropertyX> ?PropertyXValue}. FILTER ((?completionDate = 'YYYY-MM-DD'``xsd:date)) }</pre>
10. Click	'Search' Button	A satisfactory query created, filters applied	Execute the query	List instances matching the query	<pre>SELECT * WHERE { <current SPARQL query> }LIMIT 1000</pre>
11. Text entry	Text search' 'text	Concept Search field	Search for a known concept	Concept Search	<pre>SELECT ?conceptValue WHERE { ?conceptValue a ?concept. FILTER regex(?conceptValue,"i","text search"). }LIMIT 10</pre>

Table 10: User Interactions translated to SPARQL queries in Affective Graphs

As discussed in Section 6.7, a new node created results in three SPARQL queries being triggered - a subclass request, a range request and a domain request. The same occurs during initialisation (1. Init, when the user chooses to reload the page or load the page for the first time), since the initial step requires a new node to be created. Following the initialisation step, a user can choose to either proceed exploration by clicking on different sections of the initial node (2. Click, which results in the same queries being triggered for the new node) or perform a search for relevant concepts (11, which results in a popup dialog listing the relevant concepts identified). Selecting an item from the popup dialog results in another node being created, and thus the three queries being triggered.

Following the creation of a new node, and during the exploration of a dataset, the user can build precise queries by interacting with the nodes and links. The user can select a specific node/property (3,5,6) as a query object by right-clicking on the visual object and selecting “add to query” from the context menu. This will generate the relevant SPARQL query, based on the concept/property selected as well as connect to any previous query the user may have connected. If the user has any specific information need, then the user can add a filter constraint on the query by right-clicking and selecting “add constraint” (4,7,8,9) from the context menu. This results in the addition of the necessary filters on the SPARQL query. Upon being satisfied with the query thereby built, the user can then click the ‘Search’ button (10) to finalise and execute the SPARQL query.

6.11 SUPPORTING USER NEEDS IN SEMANTIC WEB

Unlike *.view.*, lessons learned from the development of Affective Graphs were rather on a higher level, in essence establishing and confirming two main hypotheses:

- Attention to aesthetics is essential, and aesthetic designs can be achieved by aligning functionalities with best principles of design
- The role of users is central to ensuring the development of a well-received system

Initial experiments with Linked Data interfaces and aesthetic measures has highlighted the need for explicit attention to aesthetics while designing interfaces for the Semantic Web. Starting from principles developed by the Human-Computer Interaction (HCI) and the Visual Analytics communities, a set of best principles were proposed that could be used to develop Semantic Web applications. These principles were used to design and build an interface that facilitates exploratory browsing of Linked Data. The approach was validated by several evaluations, with users from varied domains and expertise levels.

The role of users in a UCD approach is paramount. Users need to be consulted and should be made a part of the solution. This, in addition to bringing a sense of belonging can help users feel responsible for the solution. While there are several ways of involving users, personal interviews and focus group sessions have provided the most insightful results, and there is a greater need for such types of evaluations within the community. It is important to understand the affective appeal of systems on users, and such sessions can provide insights not easily achievable with a formal user evaluation. Involving users from different domain also ensured the final solution being developed is a generic one and domain-independent.

Tech-Users	Lay-Users
T1. Intuitive Navigation through LD structures	L1. Intuitive navigation through the large amounts of complex, multi-dimensional data
T2. Data exploration to understand content and structure	L2. Exploratory knowledge discovery
T3. Data exploration to identify links across and within datasets	L3. Support for basic to advanced querying, to support filtering and IR in order to cater to experts as well as casual users
T4. Data exploration to identify errors, noise and anomalies in content and syntax	L4. Detailed analysis of ROIs
T5. Advanced querying using formal query syntax	L5. Publication/syndication
T6. Publication/syndication, verification and validation of new data and links	L6. Data extraction for reuse
T7. Data extraction for reuse	L7. Presentation of the results of analysis to different audiences.

Table 11: Design guidelines proposed by Dadzie and Rowe [DR11] visual information presentation catering to two types of users — lay and tech

This section also revisits the literature, and from a higher level attempts to understand how Affective Graphs can support a typical Semantic Web user needs. [DR11] provided a comprehensive review of Linked Data visualisation approaches and categorised their design guidelines based on the perspectives of a tech user and a lay-user. Upon reflection, it is useful to look back at these guidelines and align them with the features provided in the Affective Graphs interface in order to understand how it fits with the greater expectations of the Semantic Web community. The guidelines proposed by the authors are shown in Table 11 .

One of the main guidelines proposed by the authors for both lay and tech users is the need for an intuitive interface that facilitates browsing of large complex multi-dimensional data (L1, T1). User evaluations and focus groups have been highly positive and indicate that users had a good experience with the tool while exploring data. The exploratory tasks focussed on understanding how well users can explore an unknown dataset to find relevant information. Users found the interactive and visual approach stimulating and were willing to explore data in a playful manner, thereby finding answers to their tasks (L2, T2). There was also a noted change in behaviour as users started becoming more confident with the system with increasing familiarity and practice. Affective Graphs also makes it apparent to the user how different concepts are hierarchically related and what are the common relationships they share (L3,T3). Exploring nodes in focus provided users with more information about the node, which helped users gather an understanding of a concept without actively searching for it's content (L4).

Although not discussed within the scope of this thesis (mostly due to the minor simplistic implementation details), Affective Graphs also has features for exporting data from query results (T7, L6, L7). The SPARQL queries are stored in the system, and if a user is interested in the result sets, they can export the results in a file (which is the same action as a URL call, with a parameter to return the response as XML, JSON, HTML, RDF, CSV, etc.) . This feature was disabled during the evaluations as it was not a focus of our experiments. Since Affective Graphs is not meant to be a standalone system, and will be integrated with another visualisation framework, which makes it possible to simultaneously visualise result sets in multiple facets (L7). One of the most useful features identified by Dadzie and Rowe is the ability to query for specific instances of

data within a dataset. Affective Graphs helps users build queries in a highly visual and interactive manner. We believe this would be of immense help to lay users as they would not be trained in formal query syntax. Their interactions with visual elements would generate queries which would enable them to answer specific Information Needs (L₄, L₃). The interactions can also serve as a starting point for advanced users, who can then directly edit the [SPARQL](#) query thereby generated or modify various query parameters such as limit the number of results, order results and so on (T₅).

6.12 SUMMARY

This chapter presented the second of the two proposed solutions, focussing on querying large semantic datasets by employing a highly visual and interactive approach. This chapter addressed and provided evidence for the third research question (R₃ – *How can visualisation interfaces be designed to be visually pleasing?*). Starting from the results of *.views.*, this chapter motivated the need for a greater emphasis of a querying system. Visual Analytic techniques and aesthetic design principles highlighted from a study of the literature were aligned against Semantic Web principles to provide a guide toward aesthetically developed Semantic Web exploration solution. [VA](#) techniques provide insight into how visualisations can be employed to present large multi dimensional datasets. Employing aesthetic design ensures that the experience of using such interfaces is an enjoyable and pleasant one, thereby making users more receptive. These principles were used as guidelines to develop the Affective Graphs solution, which was then positively evaluated in several rounds of user evaluations.

Several design decisions were also made as a result of lessons learnt from evaluations and user studies. The design principles also provided guidelines to develop the solution. These decisions can be employed into many other solutions that employ similar mechanisms — for e.g. employing a HSB colour scheme for an automated colouring palette, fine-tuning and providing users with greater control of automated graphical layouts, employing standard and familiar user interactions. The generic mechanism of interactively exploring semantic datasets partially (along with *.views.*) addresses the second research question (R₂ – *How can visualisation interfaces help explore semantic data in a generic manner ?*), by making use of interpretation/translation rules to convert high level user interactions to low level intents, and subsequently to formal queries.

The applicability of Affective Graphs within an organisational [KM](#) perspective is significant — increasing amounts of datasets are continuously made available to all users within organisations, and there is a need for users to access and comprehend large volumes of unknown datasets quickly. Affective Graphs has been developed as an exploratory solution that can help users explore an unknown data space, identify concepts and topics of interest and build highly directed queries. Users can quickly plug-in to large, previously unknown datasets, and easily assess their content as well as structure. Incorporating natural language (generic as well as purpose-built for specific domains) systems can also help users reach their data of interest efficiently if they have a particular Information Need. These features of Affective Graphs can help organisations consume semantic data in an efficient, scalable and generic manner.

This chapter also discusses how several user communities from various domains have contributed toward evaluating the solution, and providing ideas, suggestions and feed-

back that have been invaluable in developing a more refined solution. Five evaluations were conducted that provided user feedback in different stages of the development. The first evaluation with domain experts provided the first attempt at presenting a visual and interactive solution — while the system was clearly not a refined one and contained several interaction issues, the users appreciated the approach and provided helpful insights into addressing some of the issues. It also helped reaffirm the positive implication of a visual and interactive approach, and the enjoyable and stimulating experience noted by users provided a promising direction.

The next three user evaluations presented users with a more refined interface, with fewer interaction issues. However, with every evaluation, there were several new issues discovered. This establishes the notion of the importance of user evaluations — where even in the fourth evaluation, new interaction or design issues were identified. Varying user types, tasks and experimental settings, new issues can be discovered which continuously need addressing. The final evaluation, comparing established [SW](#) systems revisited an earlier evaluation where the best performers of the systems in each aesthetic criteria were compared with Affective Graphs. An overall positive score in this evaluation showed that, in an objective setting, Affective Graphs is well placed as a more aesthetically pleasing solution. It is to be noted that this evaluation was not a final judgement on how much more aesthetically pleasing the solution is in comparison to standard tools in the community, but a direction toward exploring other qualities of an interface. This is important in the [SW](#) and [KM](#) communities, as there is a lack in adopting aesthetic design in developing solutions.

The nature of the proposed solution itself, being very interactive and visual, perhaps accounted for users who were highly engaged with the system — interacting with the system to build complex and fine tuned queries, explore unknown data, discover new information are all part of a user's Information Need. Providing users with mechanisms to perform these tasks in an interactive, systematic and yet playful manner created an enjoyable experience for the users. The adoption of [UCD](#) and greater engagement of users while developing [SW](#) tools and solutions is extremely urgent and highly in need. In fact, David Karger's keynote at ESWC 2013, titled 'A Semantic Web for End Users' argued for a greater involvement of users within the community. There is a need for a better application of usability methods, and my experience while developing the solution re-established the role of users as paramount in the design, development and implementation of a solution. Despite Kaufmann's findings in [[KBf07](#)], it was observed that the view-based highly graphical solution provided very high satisfaction scores and performed well in comparison to other approaches. I believe it is the combined approach of employing [UCD](#) as well as developing solutions with explicit attention to aesthetics as well as a highly visual and interactive approach provided a highly stimulating and enjoyable interface to users.

During the evaluations, users from many domains expressed their intent of using the system for their own datasets, and several plans are in-place for future exploration into different areas and possible features. The evaluations have also paved the way for establishing a new visual paradigm of browsing and exploring unknown datasets, which until now was only possible by writing directed queries, manually inspecting instances or viewing pre-built mashups. The following chapter discusses findings from the two systems and presents details regarding possible future work.

Part IV

CONCLUSIONS AND FUTURE WORK

DISCUSSIONS

Following several iterations of design, implementation, evaluations and follow-up analysis, the *.views.* and Affective Graphs solutions had matured to developed systems, continuously guided by user feedback, comments and suggestions. The user centered design process formed the backbone of the development of the two solutions, and ensured the final solutions were well accepted by various user communities, from different domains. This chapter presents a high level discussion on the novel elements of the two solutions. Additionally, this chapter reports the lessons learned from developing *.views.*, which eventually influenced how Affective Graphs was developed and designed. I believe that these findings are highly generic and significantly relate to most solutions semantic web. The next section presents the lessons learned and novel contributions of the first solution, *.views.*

7.1 VISUALISING SEMANTIC RESULT SETS: A REFLECTION

The development of the first solution, *.views.* was inspired by several other systems and approaches developed in the Semantic Web. Systems such as Tabulator [BLCC⁺06] seeded ideas for the core of the system – a multi-visualisation approach, where different facets of the data would be presented in different visual paradigms. Separating visual regions in the interface to communicate a specific type of information resulted in the creation of visualisation widgets. This was inspired by solutions such as Sig.ma¹, mSpace [sWRS06] and Museum Finland [HMS⁺05]. Form based systems such as K-Search [BCC⁺08] inspired the approach toward dynamic forms for generating global queries to help users filter large volumes of data, using dynamic queries. The dashboard layout that brings together various elements was inspired from the designs of several websites typically used in daily tasks as well as typical Business Intelligence solutions such as IBM Cognos², SAP BI³ etc. Techniques presented by [Few06, Tuf86, Tuf90] also helped in the design and layout of the dashboard approach.

The design of the approach was built upon Shneiderman’s classic paradigm of Information Seeking [Shn96] (“Overview first, Zoom/Filter, Details on demand”), where a high level view of the data is presented to users by employing a multi-visualisation approach. Using dynamic querying [AWS92, WS92, GR94], users can then filter the dataset to reach items of interest, while visualising the data in different perspectives. Upon reflection of issues arising during the development of *.views.*, the most concerning was the performance of a public linked data endpoint. This caused the backend to be unreliable and unpredictable, and therefore posed significant technical challenges while developing the solution. The following section discusses how some of these issues were mitigated by suggesting a set of guidelines.

¹ <http://Sig.ma>, <http://sig.ma/>, Last accessed 25/06/12

² from <http://www-01.ibm.com/software/analytics/cognos/solutions.html>

³ <http://www.sap.com/pc/analytics/business-intelligence/software/overview/index.html>

7.1.1 *Lessons Learned and Guidelines*

To better understand the constraints for an effective and generic approach to the user-centred visualisation of Linked Data, both user and system aspects have to be taken into account. Therefore both a formative user evaluation and a testing of the backend performance were needed to better understand pitfalls and potential. This section pulls together the results of the two evaluations and discusses which points have to be taken into account when designing user interfaces to Linked Data.

The focus group sessions have been instrumental in identifying how user needs and expectations for exploring unknown datasets and domains can be generalised — transforming an intuition into a concrete approach. The discussions with domain experts, computer science professionals and students have indicated that such a generic approach is much appreciated, specially when data is provided to users unfamiliar with Semantic Web technologies. Majority of the focus group participants were not conversant with Semantic Web principles and query languages — in order to be accepted by a wide audience, there must be a complete separation of users from raw semantic data. By providing multiple means of querying the underlying data (global filters and local filters), users can be equipped with different interaction mechanisms to explore their datasets. However, being completely unaware that each subsequent interactive step results in SPARQL queries being sent to the backend. This transparency is important to a user — as end user's interest lies in understanding their data — not by writing complex scripts and queries, but by quick, seamless and fluid navigational and interaction techniques. However, by being able to drill down into individual data instances, the users will always have access to their data.

Apart from understanding the potential impact of the system architecture on user expectations and experience, the approach has also been to understand how a generic visualisation framework can be provided. The dashboard approach has been applied over the years to cater to different requirements for different domains. *.views.* is capable of porting to several backends — traditional databases as well as Linked Data or triplestores. The flexibility offered by such systems seems to be ideal for a Semantic Web visualisation framework — developers can contribute by creating 'add-on' visualisation widgets that cater to specific domains or data types; users can select which visualisation widgets they prefer to use and in which order; users can select sections of the visualisations within individual widgets to further investigate areas of interest; data owners can use such framework to quickly explore their own datasets, shared data or even organisational data. It is also the familiarity of the users with systems following the dashboard approach that has influenced our design choices — the users would not need to be trained in using dashboards as they almost unknowingly use such systems daily (BBC, igoogle, etc.).

Initial intuition before the system evaluation was that the response from publicly available SPARQL endpoints would be quite slow as compared to querying a local database; the system tests proved that the real problem is the inconsistent time lags of the query execution phase, a far greater challenge, as it cannot be fully controlled or estimated. Reflecting on the implications for the user interface and the interaction, these basic guidelines can be considered:

7.1.1.1 *Instance Counts*

Providing instance counts (while entering textual queries, slider actions or selecting checkboxes etc.) should be handled with care. It was often found that due to the delay (in processing of a previous query) in the backend, the results being passed to the frontend (thereby parsed and rendered in the frontend) overwrite the recent results. This happens more often while obtaining instance counts, as the interaction itself expects several quick responses from the backend. For the present implementation, timeouts of 200ms from the frontend were introduced so that it would prevent delayed results from previous queries over-writing recent results. Another suggestion would be to add user interventions before sending queries to the backend e.g. pressing enter to retrieve instance counts instead of the queries being triggered after a “KeyUp” or “OnChange” event.

7.1.1.2 *Dynamic Querying*

Real-time querying via generic filters (e.g. dragging sliders to update visualisations dynamically) could cause the system severe delays as queries are continuously fired creating a backing of unresolved requests. To overcome this problem three solutions are possible:

- Provide user interventions before passing queries to the backend;
- Prevent continuous dynamic querying by providing only discrete interface items (i.e. remove elements like sliders);
- Cache the result of a query and use the dynamic filtering on the retrieved set, but in this case only restrictions of the set will be possible.

The present implementation focussed on the first solution mainly due to the fact that it does not pose any restrictions on the interactivity of the querying interface.

7.1.1.3 *Automatic Suggestions*

Automatic suggestions were disabled, as responses from the backend were often late and could overwrite more recent suggestions, as explained in a previous point. As the backend processing was time-exhaustive, it was decided to disable searching using regular expressions. Instead, queries are presently performed using URIs. Doing so does not put any demand on the endpoint to process regular expressions, thereby making the backend respond quickly.

7.1.1.4 *Aggregate Queries*

When large result sets were returned from the backend, there was a significant delay in loading up aggregate visualisations pie chart, bar chart and tag clouds. There are two solutions to this that can be perceived:

- Limit the response to display the top few results, or
- Provide the results progressively.

For the present implementation, the first approach was chosen as it provides a comparatively lesser amount of data (often the most important bit) to process. In this implementation, the users however have the option of the entire result set, if they choose to do so. Doing so improved the performance significantly to an average of 30.5 ms from 1059.6 ms. However, the inconsistency of the SPARQL endpoints still remain. Apart from reducing the time taken to process the queries (and results), this helped in providing a more readable visualisation. Figure 72 shows the improvement in the readability of the bar chart widget when this was done. The figure on the left shows the aggregate results before any limits applied. The figure on the right shows the improved readability achieved by applying limits.

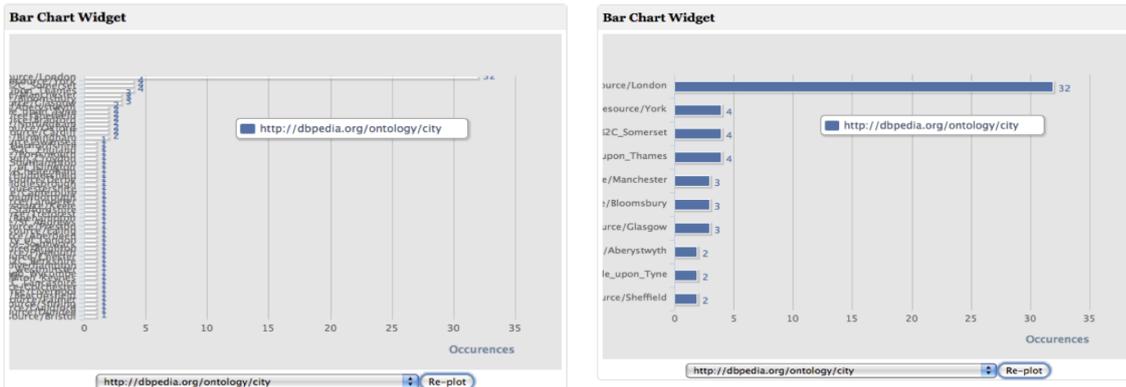


Figure 72: Improving readability (and speed) by limiting results for aggregate queries. Both figures show identical information (e.g., most resources are in London). However, the first widget is almost illegible.

7.1.1.5 Textual Results

As can be seen from Figure 46, the query execution time for the text result widget was high. Further on, more instance matches result in further delays since it takes more time to process large data objects. Textual result can then be presented on request in a separate overlapping layer, i.e., by providing only a window (as a page) of the entire result set (Figure 73). This solution reduces time and improves readability. Thereby, instead of sending one highly time consuming query, we modified the system to send several short queries when required.

The above mentioned guidelines are often adopted by software developers in order to cope with unreliable or slow databases. Several other software engineering approaches can be adopted for improving the system performance like caching results, interacting with local datasets, engineering SPARQL queries to provide quicker responses etc. However, the intention of evaluating the system limitations is to understand how good practices in information visualisation can be harmonically aligned with the current system infrastructure, keeping the user expectations in mind. The unreliability of the Linked Data endpoints may create issues (like delays, timeouts etc.) in user interfaces as well as deactivation of essential functionalities, which can adversely affect the user experience. In order to adhere to specific guidelines established by the information visualisation community, the backend infrastructure needs to provide a consistent performance. So-

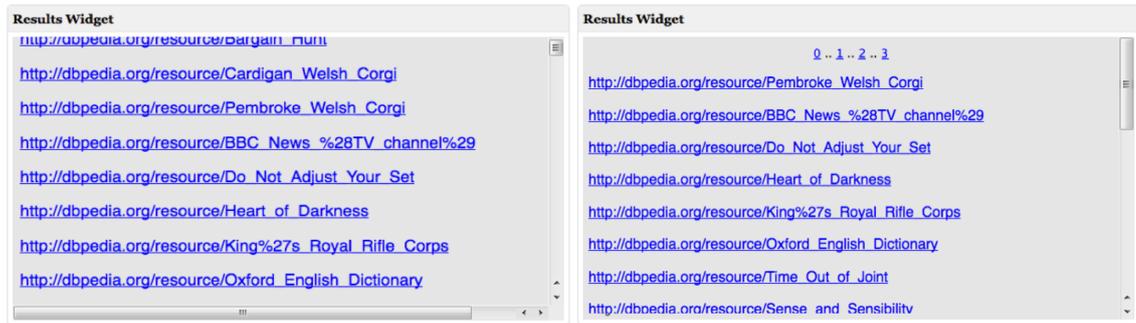


Figure 73: Improving readability for textual results by providing a page view of large result sets. Note the scroll bar for both figures indicating the number of textual results contained in the widget.

lution developers in the enterprise need to be aware of these limitations and guidelines while designing enterprise solutions. As discussed previously, while the evaluations have been conducted on a public SPARQL endpoint, large data repositories in large organisations can pose similar issues. The outcome of the quantitative evaluation in a public Linked Data endpoint setting poses an important technical challenge that can affect usability and user experience significantly. The design guidelines, design and development of the multi-visualisation approach, along with the outcome of the evaluations supports the first claim of the thesis, C1. The claim states that a multi-visualisation approach which provides simultaneous views of different facets of large result sets can provide effective means to explore large scale data. However, existing frameworks pose challenges that can restrict a fully interactive user experience.

7.1.2 Novelty in the solution

The need for a generic, visual and interactive solution steered the direction toward a visualisation-based, interaction driven approach. Adopting the approach for a semantic web infrastructure established a genericity, where the solution developed was not specific to an individual domain. Furthermore, the inclusion of user communities with various expertise ensured genericity of users. A significant engagement of users via user centered design processes and multiple iterations of design has been largely missing in the Semantic Web community.

While the iterative development process of *.views.* itself is unique to the community, several features and elements in the design of the interface can be identified which are novel to the semantic web. The features and elements of novelty and contributions can be grouped into the following five categories: Genericity, Interactivity, Filter and Querying, Visualisation and Development Guidelines. The following list describe these novelties, particularly from a system design and features point of view.

- Genericity of the *.views.* solution is firstly ensured by involving user communities to help guide key design and interface decisions have contributed to a generic system. A significant focus was paid in gathering an initial understanding of domains and users by conducting user interviews and focus groups. Studying

existing knowledge practices, systems and visualisations most used by user communities also identified several potential directions of development. This ensured the visualisation solutions that were adopted were most suited to the right user groups. However, adhering to strict semantic principles and adopting a rule based framework for exploring semantic resultsets ensured that the final solution would be a generic one, with the possibility of extending to specific domains by including custom widgets. From an implementation perspective, a semantic web framework contributed to a generic solution, owing to its self-descriptive, structured and highly formalised approach. Hence, the novelty of the *.views.* solution is in establishing a user-centered development approach within the Semantic Web involving domain and technology experts. While a standard practice in Information Retrieval and Human Computer Interaction research fields, such approaches are largely missing from Semantic Web research. This is a novelty, as a typical Semantic Web infrastructure has several characteristics (e.g. a very high level of formalisation, domain-independence, reasoning) that makes it a unique framework for development.

- *.views.* is an interaction-driven approach, where each user interaction is translated into user intent and finally, SPARQL queries. Translation rules and templates are domain and user independent, and rely on interpreting user interactions to information need and user intents. These rules are generic, and can be employed in other solutions to easily formulate how a click-and-drag operation on a slider can be quickly translated into a highly formalised SPARQL query. The translation from user interactions to user intent and subsequently to a formal query is the next novelty of the solution.
- Two types of querying mechanisms are employed in *.views.* – global and local queries. Global queries select a subset of the data, to be visualised by all the visualisation widgets. Local queries, on the other hand provide users with means to focus on specific facets of the data and explore multiple facets in a flexible manner by interacting with different visualisation widgets. This approach of combining different types of query paradigms in a generic manner for exploring semantic data is identified as a novel contribution of the solution.
- Multi-visualisation of semantic data is a common solution for exploring semantic data. The *.views.* solution approaches visualisation of data from a generic standpoint. While this is not a novel contribution and several systems have employed this technique to great success, it is worth mentioning that *.views.* provides domain and generic visualisations in a flexible manner. The benefit of the *.views.* approach is that it can be very easily plugged-in to a new dataset and used to explore datasets effectively and quickly, instead of spending enormous efforts to configure for new datasets/domains.
- As a part of the development of the *.views.* approach and subsequent performance evaluation, several crucial findings were observed. The most important finding was that typical public semantic data infrastructure struggles to provide a consistent performance, and hence impacts the predictability of solutions interacting with public endpoints. This resulted in the development of guidelines to help mit-

igate the performance issues, as discussed in Section 7.1.1. While some design decisions in the guidelines are often used in Information Retrieval and Human Computer Interaction fields, the semantic web community lacks precise guidelines or suggestions for developing interactive interfaces. These guidelines are hence, a novel contribution to the Semantic Web community.

views. was evaluated within an organisational framework with a low number of users. While the evaluation lacks in several aspects, and hence only indicates its applicability in a knowledge management environment, the positive feedback from users show that the solution is well appreciated.

7.2 EXPLORING AND QUERYING SEMANTIC DATA: A REFLECTION

The development of Affective Graphs was based on three foundational ideas: the need for more intuitive, usable, visual and interactive querying mechanisms, the need to visualise schema as well as data, and an aesthetic approach. As described previously, several systems and approaches have provided inspiration for different elements of the implementation such as design, layout, visualisations, colors, interactions and so on. View-based tools such as Semantic Crystal [KBF07] and IsaViz⁴ have inspired the visual approach of presenting Semantic Data as a node-link graph. Presenting statistical information about data instances related to semantic classes was inspired from the idea of providing such visualisations with visually encoded nodes and edges (size, colour, texture etc.) based on the number of data instances⁵ [HBO10]. The approach employed by faceted browsing systems such as [sWRS06] and Museum Finland [HMS⁺05] and mashups such as Sparks⁶ inspired the mechanism for driving interactive exploration via facets of the data.

7.2.1 Novelty in the solution

Affective Graphs aims at providing visual and interactive means for users to identify elements of interest, and subsequently formulate complex queries. Given the need for a scalable and generic solution, the challenge was to develop mechanisms that are intuitive, comfortable and effective. Affective Graphs was carefully developed to match best practices and design principles of aesthetic design and visual analytics for the Semantic Web. These principles were formulated by an extensive survey of the literature, and are proposed as a major contribution of the research. The novelty of the design of Affective Graphs is in adhering strictly to the design principles, and user evaluations indicate the validity of these principles and the aesthetic appeal of the interface.

Several iterations of design were involved in developing the final solution, starting from an initial basic prototype. Each iteration involved users completing the process as an evaluation, providing comments to feed-in to the next cycle of development. Hence, the first novelty in the solution is in the design process itself. A user centered design process, involving technical and domain expert as well as casual users encapsulated

4 IsaViz, <http://www.w3.org/2001/11/IsaViz/>

5 Such techniques are practiced in several approaches, especially in many visualisation tools such as Prefuse (<http://prefuse.org/>), Protovis (<http://mbostock.github.io/protovis/>), d3js.org, sigmajournal.org etc.

6 Sparks Prism, <http://sparksrdf.github.com/>

the entire development of Affective Graphs. Such development methods are novel in semantic web research, and the positive evaluations of the system validates the benefit of such an approach.

The design of Affective Graphs, particular to the decisions taken while development of the solution presents the next set of novelties. While these design decisions are discussed in more details in Section 6.6 driven by the requirements of the solution, this section summarises the novel elements in the design.

- Visual summaries of data – The need for a scalable solution motivated the approach toward presenting large collections of data in a structured manner. This needed a way to present users with visual summaries of data. However, the summaries must be presented in a manner that is easily comprehensible to end users, and is a domain independent one. Hence, this resulted in the decision of providing statistics of a subset of data as a pie chart, embedded on a node. The node indicates the present concept of interest, and the pie chart indicates how the instances of the concept are distributed among the subclasses of the concept. This provides an easy understanding of a large collection of data, by hierarchically observing the distribution of the underlying data instances. If the user is interested in pursuing a subclass of the data, then he/she can click on the relevant section to further explore the subclass. This way of embedding aggregate visualisations on graph views to summarise and preview underlying data is novel to the semantic web community.
- Aggregate-driven data exploration – The aggregate visualisation (pie chart) on a semantic concept, while providing summaries of data, also serve the purpose of driving user exploration. This provides users the ability to easily follow hierarchies to reach a specific subset of the data, without having to enter complex queries. While such features can be observed in faceted browsers (where users explore and drill-down to subsets of data), a visual paradigm of such exploratory mechanisms is a novel contribution to the semantic web community. The aggregates provide an excellent way of following Shneiderman’s well known design paradigm “*overview first, zoom and filter, then details-on-demand*” [Shn96]. The aggregate visualisations in a higher level provides overviews of the dataset. Where interested, users can click on individual sections of aggregates to zoom-in to a higher detail. Finally when the users wish, they can use simple gestures to formulate queries and reach individual data instances.
- Multi-path exploration – The graphical approach employed by Affective Graphs provides a highly visual and dynamic environment for users to explore data. The presentation of nodes and pie charts provides ways for users to follow multiple paths of exploration. Unlike faceted browsers, users are free to follow different paths to explore data from different perspectives. Typically, faceted browsers enable exploration from one concept to the next, one at a time. The user is only restricted to one path of exploration. However, Affective Graphs allows users to branch out their exploration to multiple paths, in parallel. This is a novel contribution that can help users easily and quickly span out to a larger subset of the data.

- Visual Communication of Semantic context – While the need of users are mostly in understanding data, and finding data instances of interest, how the data is organised and relates to different concepts can help users better drive their exploration. This needs the ability to quickly understand the semantic content and context of large data. Affective Graphs does this by clearly communicating the semantics of the objects users observe. Different types of visual objects represent different semantic properties: e.g. curved links represent non-hierarchical object relations, while triangular links represent a hierarchical relation. While the representation of the schema itself is not unique, the integration of the schema with content (pie chart) and context (different visual representation of semantic entities, and context window) summaries is novel.
- Visualising Data and Schema – Semantic Web tools have mostly visualised either data instances or ontologies (schema). However, the approach of providing visualisations of data schema enriched with statistics about the underlying data is a novelty introduced by Affective Graphs. Representation of semantic concepts as a pie chart, providing an overview of the underlying data presents users with a visual and interactive way of exploring the dataset, driven by the content.
- Design – Design of semantic elements within Affective Graphs was a significant contribution in itself. While the idea of visualising semantic data as node-link graphs is not new, how the visual elements have been represented (as pie charts, bezier curves, hierarchical and non-hierarchical edges) is. Visualising concepts and properties differently to communicate different semantic characteristics is a novel contribution, that has not been pursued in the Semantic Web community previously. Automatically visualising large subsets of data as pie charts with randomly generated pleasing color templates is also a design challenge that was addressed in Affective Graphs.
- Aesthetically Pleasing – Central to the design and development of Affective Graphs was the aesthetic appeal of the solution. This has not been considered in the Semantic Web community so far, and is a novel contribution of the solution that attempts to highlight the importance and significance of aesthetics.

Several novelties of the user interactions are also to be noted in Affective Graphs. This relates particularly to the ability to easily formulate precise complex queries by making use of typical natural user interactions. Moreover, in the final version of Affective Graphs (Hybrid), the ability to translate a natural language query to a visual representation, and then providing users the facility to fine-tune queries is also noted as a novel contribution. Translating user interactions to formal queries involves two significant tasks: interpret user interactions to identify information need, translate information need to structured queries. The novelty of Affective Graphs in user interactions are as follows:

- Translating user interactions to structured queries – The first part of translating user interactions into structured queries is achieved by interpreting a user action based on the context of the interaction. This is then used to identify which query template is most relevant to the interaction. The query template is populated with

concepts and properties that are related to the user interaction, and a final formal query is then generated. The process that is involved in such interpretation/translation is discussed in Section 6.10 and is noted as a novelty toward generically exploring and querying semantic datasets.

- Translating natural language to visual representations – The hybrid approach of integrating natural language with a graph based approach to query semantic data is a novelty in the semantic web. While other hybrid approaches (e.g. K-Search’s natural language + keyword search) have been proposed in the past, this approach is novel and aims at alleviating several issues arising out of using either graph based or natural language approach independently.

7.3 IMPACT ON THE STATE OF THE ART

This section discusses the ways how the work presented in this thesis extends and contributes toward the state of the art in four ways:

- Approaches employed to present and query Semantic data
- Extending techniques for data exploration
- Extending the concept of hybrid search (from semantic + keyword search to hybrid interaction mechanisms)
- Extending Semantic Web interface design by proposing guidelines and principles

It is to be noted that the contributions discussed in this section is specific to only the state of the art in the relevant areas. The general contributions of the thesis are presented from a higher level perspective in Sections 1.4, 8.1 and 8.2, and these contributions do not necessarily reflect the design, technical and implementation novelties of the solutions developed, discussed previously in Sections 7.1.2 and 7.1.2.

Chapter 3 highlighted the most relevant research for the work described in this thesis. Section 3.1 discussed various systems to exemplify different ways Semantic data can be queried and presented. The Figure 15 presented an extension of Kaufmann’s formality continuum [KBF07, KB10], which is now revisited to identify how the developed technologies (*.views.*, Affective Graphs and Hybrid Approach) align with the state of the art. Figure 74 shows that the positioning of the approach can be highly varied and exist mostly in the middle of the continuum. Interestingly, it can be noted that the interactive multi-visualisation approach as well as global and local filters provide users with various mechanisms of querying Semantic data. Hence, the positioning of *.views.* on the formality continuum is mostly varied, based on the type of visualisation widgets that are made available for the particular dataset, use case or user. The global queries contribute toward a form-based approach as employed by [BCC⁺08], while local queries and visualisations depend on the type of visualisation used: e.g. advanced visualisations such as graph-based or networks generate formal queries such as [KBF07]. At the same time, the multiple coordinated visualisation approach employed by *.views.* extend the interactive approach employed by Mashups such as Sparks ⁷.

⁷ Sparks Prism, <http://sparksrdf.github.com/>

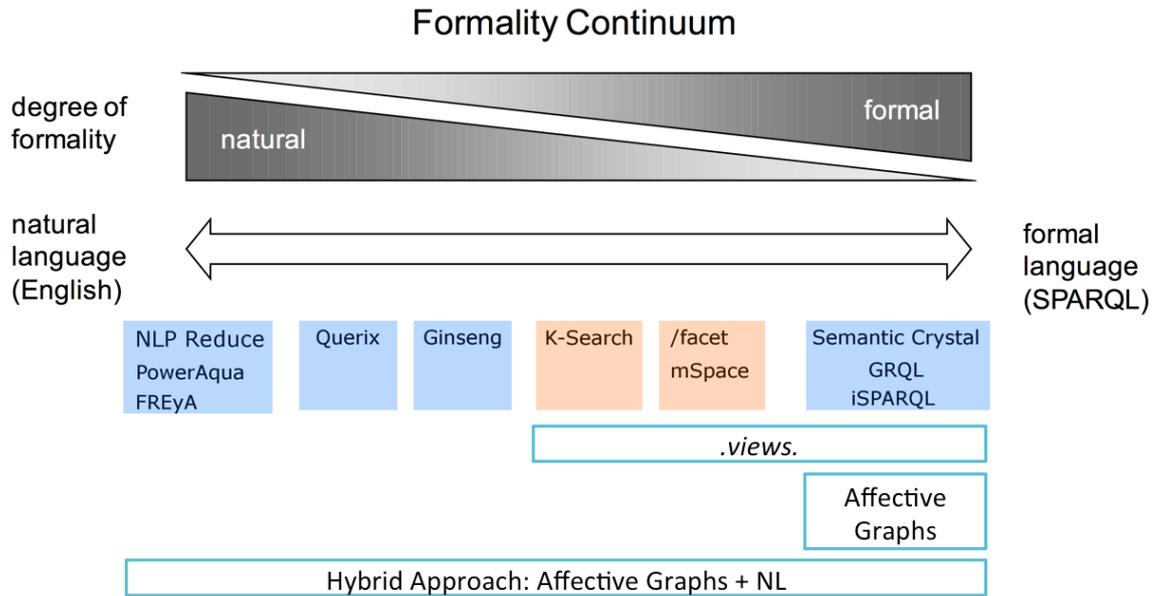


Figure 74: Positioning the approaches on an extension of the formality Continuum, illustrated by [KB10]. Image adapted from [KBF07]

Affective Graphs, on the other hand exist on the right extreme of the formality continuum. The highly graphical and systematic approach toward formulating queries is a more formal process, in spite of a highly interactive and visual approach. As a result, the system is placed as the most formal approach. The introduction of a natural language component widens the positioning of Affective Graphs (Hybrid Approach, Section 6.9.4), as a natural language system is located on the left extreme of the formality continuum. Several features within the system such as query window (Figure 67, Right window) or constraint configuration (Figure 61) to reconfigure query terms involves a form-based approach. Hence, the Hybrid Approach approach spreads throughout the formality continuum, and provides users with a variety of means for expressing their information needs. Most of the systems in the state of the art have distinct approaches and can be positioned in specific regions of the formality continuum, while the developed systems tend to provide users with a much wider range of interactivity and query mechanism.

Chapter 3 also discussed relevant state of the art in terms of interactivity and visual encoding and presented the relative positions of systems on a two-dimensional graph (Figure 16). *.views.* and Affective Graphs occupy significant positions on the scale – both systems are highly interactive and can facilitate the generation of complex queries from interactions. This interactivity is one of the strong points of the systems, and hence their high position on the interactivity scale. The task of positioning *.views.* on the visual abstraction scale is relatively more complex, given its variable and flexible framework to add/remove complex visualisations. Hence, the system has been positioned alongside Faceted Browsers and Mashups such as mSpace [sWRS06], CS Active Space [GAC⁺04], Tabulator [BLCC⁺06] and Sparks. With much greater amount of visual abstraction and visualisations, the Affective Graphs system has been placed as an advanced visualisation with high interactivity and maximum visual abstraction.

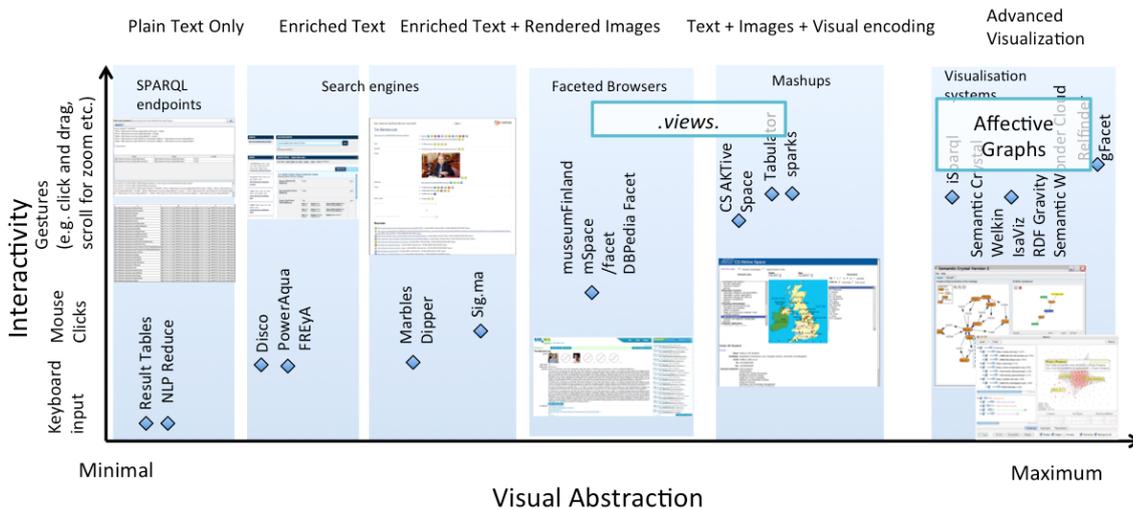


Figure 75: Relative positions of *.views.* and Affective Graphs on the Interactivity vs Visual Abstraction graph, as introduced in Section 3.1

The Affective Graphs approach employed several techniques to facilitate visual exploration of complex Semantic data – key to this discussion is the extension of faceted browsing to introduce a multi-faceted multi-path exploratory paradigm. Faceted browsing of Semantic data, as employed by [HvOHo6], DBPedia Faceted Search⁸ and mSpace [sWRSo6] provide means for users to drill-down to a subset of the data by following a series of progressively specialised sorted list of categories. While such mechanisms are simple to use, and provide users with quick access to the data they are interested in, they are limited in two main ways: single path exploration and generality and scalability of facets.

Figure 76 illustrates these limitations by taking examples from the literature. The screenshot on the top left shows the uni-directional column faceted approach employed by [sWRSo6]: a user clicks on the left most column to identify a topic of interest from the highest hierarchical concept. This triggers re-building of the columns on the right, based on the selection. Clicking on the next column on the right then rebuilds the following columns to the right. Hence, the user reaches a subset of the data of interest in the direction of left to right (as shown in the arrow). [WAso8] addressed this from a different point of view: in such directional column faceted approaches, no information is transferred from the columns on the right to the ones on the left. Once a user has selected items on the right, there is no indication of the associated categories of the items with respect to the columns on the left. Wilson et. al. proposed their Backward Highlighting approach to indicate which categories on the left columns are associated with selections made on the right columns. Figure 76, top right shows a scenario where a user clicks on an element on the right most column, and the corresponding elements in the left columns are highlighted. This approach, though does not provide multi-path navigation but is a step forward from a typical column faceted approach.

The approach employed by Affective Graphs is a multi-path faceted approach - where each node is represented as a facet, and the faceting function is a hierarchical category. The node-link based graphical approach presents users with the ability to focus on mul-

⁸ <http://wiki.dbpedia.org/FacetedSearch>

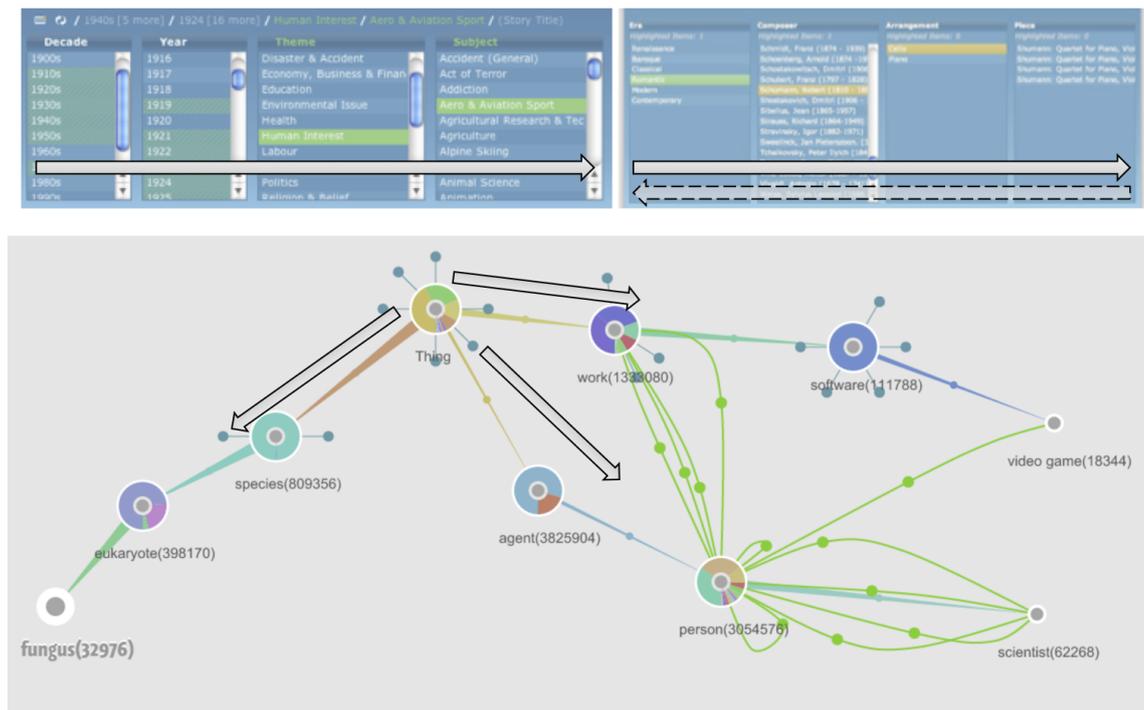


Figure 76: Single-path navigation: top left shows an approach where users progressively select subsets of the data from Left to Right (Image from [sWRS06]); top right shows an approach where users navigate from Left to Right, but relevant information related to the right columns is presented on the left columns (Image from [WAs08]); Multi-path navigation support of Affective Graphs enables exploration in multiple directions and provide means for users to discover connections between the paths

multiple paths while exploring the data. Figure 76 presents a screenshot of Affective Graphs, illustrating the multi-path navigation. As can be observed, the three arrows indicate three simultaneous paths of exploration that users can follow: left most, showing the user's exploration reaching 'fungus'; middle, showing the user's exploration reaching 'scientists'; and right most, showing the user's exploration reaching 'video game'. Any connections between the paths are immediately highlighted, enabling users to quickly build queries to find any information of interest - e.g. the connection ('GameArtist', visible on hovering on the link) between 'video game' and 'person'. The users can choose to hide any path as and when they desire, hence enabling multiple paths of simultaneous faceted navigation. Such approaches to exploring unknown data in a generic manner has not been explored in the past, and is a contribution toward the field.

The second limitation in most faceted browsers is generality and scalability – most systems such as [HMS⁺05, BHH⁺06] have pre-configured facets, which limits them being used in any other dataset or domain. For example, the Museum Finland system would need significant reconfiguration to be adapted to a biological dataset. facet [HvOH06] is an extension to the idea that facets can be specified at run-time, therefore providing users with some flexibility. However, facet as well as mSpace are both systems that are not scalable beyond few columns and would introduce cognitive burden on users if they encounter a large number of columns. The graph based approach that visualise entire columns as pie charts in organic shapes provide a large amount of information,

categorised and classified in an easy to understand manner. Affective Graphs addresses these issues by employing a generic approach (Section 6.10) to enable multi-path faceted navigation paradigm in a highly visual and interactive manner that enables scalable exploration of large datasets.

The next contribution toward the state of the art is in the hybrid approach employed by Affective Graphs. The notion of Hybrid Search in the Semantic Web has traditionally been in the task of searching itself, where the novelty is the combination of semantic search and keyword search to identify relevant documents and rank the results accordingly [BCC⁺08]. This contribution of the thesis is in extending the concept of hybrid search to encapsulate user interaction – Affective Graphs combines a natural language question answering approach with a view-based graphical interaction-driven approach. The approach applies the two interaction mechanisms and combines the strengths to negate the weaknesses of each approach. A natural language system, while extremely easy to use and highly scalable poses a few challenges, such as domain specificity and difficult to express complex queries. A view based approach, on the other hand, while can be generic and simple to formulate specific queries, is time consuming and requires considerable visual lookups from users [EMWC14]. Combining the natural language approach to such systems can provide users with a convenient starting point to visually explore a smaller subset of the data by quickly identifying and visualising relevant concepts following an initial concept search.

The final impact on the state of the art is the guidelines and principles developed as a result of implementing and designing the solutions. Extending on principles of aesthetic design, this thesis builds on several works from different contexts such as interface design, dashboard design, aesthetic UML diagrams, website design etc. The recommendations and principles provided in such works such as [Tuf86, Kim06, SH07, NTB03, SH07, BRSG07, FDPL05, Eic03, MG05, BPS05, HBE92, NM90, CS96, KG10, Shn96, Few06] have been combined together into a Semantic Web framework. Such a study (Sections 6.2) had not been conducted previously, and hence is a unique contribution to the Semantic Web community. Equally important are lessons learned from developing, designing and implementing the *views* solutions: typical issues encountered while developing Semantic Web solutions and how to address them are missing in the literature⁹. While some of such issues are well known by developers, they have been rarely mentioned in the literature, and thus a clear discussion of the practical issues encountered by developers is needed. The guidelines (Section 7.1.1) are hence another important contribution to the state of the art and developers of Semantic technologies are encouraged to consider these guidelines.

7.4 SUMMARY

Two themes of research were established early on: visualising semantic result sets (*views*.) and exploring and querying semantic data (Affective Graphs). While the first resulted in several novel approaches, the main contribution is seen as the development of design guidelines to help solution developers design semantic web solutions. The guidelines that help mitigate issues with semantic web infrastructure provide a significant con-

⁹ Though some techniques are employed by websites and visualisation systems, dissemination of the same has been lacking

tribution in the field. Some technical novelties also evolved from the development of *.views.*. In summary, the technical novelty in *.views.* is not in multi-visualisation of different facets of data, but in the process of providing a generic and flexible framework for the visualisation of large semantic data. The combination of rules and query templates to translate user interactions into formal queries is a contribution that can be developed further in the future and provides a good starting point for further development in the field. The final novel element in semantic web development is the research process itself, with a heavy consultation and engagement of user communities. While some of the novel aspects of the research are observed regularly in other fields of research (e.g. Information Retrieval, HCI etc.), a combination of these factors into the semantic web field is a novel contribution.

A set of novel elements for Affective Graphs were also identified from an implementation/design as well as impact on the state of the art point of view. The attention to aesthetics throughout the development of the solution is a key element in the solution design, and hence is a significant contribution to the field. User centered design, similar to the development of *.views.*, is a largely ignored design methodology in the semantic web, and, has been stressed throughout the development of Affective Graphs. The design of the system, and the visual representations of the different concepts, properties and instances has several elements of novelty, as do the mechanisms of interactions to formulate queries. The next chapter discusses how the research questions have been answered and provides evidence to validate the claims. The chapter also presents several directions that have been identified to continue the research in the near future.

CONCLUSIONS

The move towards employing semantic techniques within existing organisational frameworks raises a significant Knowledge Management challenge. The adoption of semantic search techniques and W3C standards into solutions provided by Enterprise software vendors (e.g. IBM, Oracle, SAS, Microsoft) paves the way forward to semantics-driven organisational repositories [Hor11]. The incorporation of semantics into web search solutions such as Google's Knowledge Graph or Facebook's Graph Search serves to position semantic technology as a key player in today's search systems. Large datasets and information sources are continuously added within organisational frameworks and massively scalable solutions are essential for an organisation's ability to reach useful and actionable information quickly to take informed decisions. The significant role of Visual Analytic solutions in tackling extremely large and complex datasets has been long established and many enterprise software vendors (e.g. IBM¹, SAS², Tableau³, Aduna⁴, Sinequa⁵) invest in such techniques for providing business intelligence and enterprise search solutions.

While this progress is encouraging and stimulates considerable research, there is a significant lack of generic visual interfaces for exploring large unknown semantic datasets today, and, as a matter of fact, only Semantic Web people could look into those repositories and even they, with difficulties. This thesis has explored how large semantic datasets can be consumed in a highly generic, visual and interactive manner by deconstructing the problem into two smaller problems: how can unknown semantic result sets be visualised in a generic, scalable and interactive manner; and how can large semantic datasets be explored and queried. The first solution proposes to employ multiple simultaneous generic visualisations to summarise multiple facets of the same result set, presenting each visualisation widget in an organised, well structured layout. The visualisation widgets are highly flexible and facets of the data being visualised are also customisable to provide users with a greater control of what they visualise. The second solution proposes to employ a graph-based visualisation that presents basic visualisations to summarise underlying content. The visualisations are connected with others using semantic relations, as found in the data.

The two solutions are highly generic approaches and provide users with a visual and interactive way of exploring and querying large unknown semantic data. Genericity of the approaches is maintained by how the approaches interpret user interactions and translate them into formal queries. Though the interpretation and translation of interactions are different in the two approaches, the same essence is preserved in the approaches. Low level user interactions (mouse gestures, clicks, keystrokes etc.) are interpreted to high level user intent, which are translated into Information Needs, rep-

1 <http://www-01.ibm.com/software/analytics/cognos/enterprise/>

2 <http://www.sas.com/software/visual-analytics/overview.html>

3 <http://www.tableausoftware.com/>

4 <http://www.aduna-software.com/node/26>

5 <http://www.sinequa.com/en/page/solutions/big-data.aspx>

resented as formal queries. The interpretation and translations are performed using basic rules and templates. Typical interaction mechanisms (adapted from well-known, existing systems) are employed to ensure all users are comfortable with each interaction paradigm as well as its translation into a formal query. Both approaches have been evaluated with users from different domains and with varied expertise, and several iterations of design ensured that users are presented with familiar interaction mechanisms, and genericity is preserved with respect to all domains, application areas and user types.

8.1 REVISITING THE RESEARCH QUESTIONS AND CLAIMS

Following the development of the solutions and the subsequent evaluations, the research questions are revisited. Several phases of research, covering different fields were conducted in order to address the primary research problem. The main research question, *how can large scale, multi-dimensional, distributed, heterogeneous semantic data be effectively explored in a generic, usable and pleasing manner within large organisations* was broken down into five consecutive questions. Reflections on each question and the respective claims are further discussed as follows:

R1. How can visualisation interfaces provide effective means of exploring large scale distributed semantic data ?

This thesis presents a multi-visualisation approach to effectively explore large scale semantic data, by visualising different facets of the dataset (Section 4.4.3.1). This provides users with simultaneous views over different aspects of the dataset, thereby providing a greater understanding of the dataset. Though the approach was appreciated by users, several issues were highlighted during the performance evaluations with public endpoints. This caused several features, many of which were very well received by users, to be disabled. A set of design recommendations are proposed, that can help solution developers address these issues while building highly scalable solutions.

The claim C1 answers this research question — *A multi-visualisation approach that provides simultaneous views of different facets of large datasets can provide effective means to explore large scale data. However, existing frameworks pose technical challenges that may restrict a fully interactive experience for users.* The claim consists of two aspects: the positive appeal of employing a multi-visualisation approach and the issues arising out of limitations of existing infrastructure. Users scored the system highly in terms of usability and overall impression, as shown in Sections 5.6.2.5 and 5.6.5. Section 5.6.3 provides evidence of performance issues in a Linked Data setting, where unpredictable behaviour of endpoints led to disabling several features as well as reducing the number of queries. As a result of such steps, systems are limited in the number of queries and their response times (R1.1), thereby negatively impacting user experience (R1.2). Certain design decisions can be taken (R1.3) in order to mitigate the issues arising out of unreliable backends, such as removal of features like instance counts, dynamic querying etc. (Section 7.1.1).

R2. How can visualisation interfaces help explore semantic data in a generic manner ?

A generic solution is desired so that the solution can be easily ported to other domains, datasets and user communities. Genericity is established by using rules and templates that convert lower level user interactions to higher level user intents, which are subsequently translated into formal queries. This ensures that data remains isolated from Information Needs — rules and templates govern how the two are connected.

Claim C2 addresses this research question — *Generic means for exploring semantic data can be developed by aligning user interactions with various combinations of generic and specific queries. This requires formalising the process of translating low level user interactions to semantic queries by interpreting user actions to high level intent.*

The rules and templates can be easily modified to adapt to different domain, application or user-specific needs. This process can help formalising (R2.1) the process of visualisation semantic data, since rules and templates can be formally developed to indicate how the solution would behave with different kinds of data. Applying query templates that can map user interactions within visualisations to user intents can provide ways to formalise the querying process, since they are governed by formal rules (R2.2).

The process of translating user interactions to formal queries and the subsequent translation of the query results to visualisation by exploiting rules and templates result in a combined and formalised approach that can help develop generic ways to query any semantic data (R2.3). The solutions discusses how user interactions are translated into formal queries as well as result sets into visual items in the Sections 5.3 and 6.7

These generic rules that provide a bridge between low level user interactions to high level user intents, and then subsequently to SPARQL queries are presented in Sections . The formal queries are built on top of previously accumulated queries and by analysing the content and context of each user interaction. The solutions have been developed and tested on multiple types of data (including conducting several user evaluations as discussed previously), ranging from open datasets such as <http://data.gov.uk> and <http://dbpedia.org> to domain specific datasets such as aerospace data (data made available by Rolls Royce plc in the Samulet project) and plant life sciences data (data made available by the Grassportal project). Genericity in terms of users is also validated by the wide range of technical and domain expertise of the participants in the user evaluations.

R3. *How can visualisation interfaces be designed to be visually pleasing?*

Claim C3 answers this research question: *It is possible to develop guidelines of aesthetic design for exploration of large scale semantic data by combining aesthetic design, Visual Analytics and Semantic Web principles. These guidelines can then be used to develop user interfaces to help explore large scale data in a generic, usable and pleasing manner*

An explicit attention to aesthetic design principles, combined with Semantic Web, Human Computer Interaction and Visual Analytic principles can provide a multi-disciplinary approach toward developing aesthetically pleasing solutions, as shown in the evaluations conducted in the Section 6.9. Past research in several fields (e.g. art and design, product design, architecture, film and photography etc.) has identi-

fied several guidelines such as the rule of thirds, unity in variety, scale, proportion and so on. These principles can be translated to a Semantic Web visualisation settings (R3.1), which can be used to develop interactive interfaces as was discussed in Section 6.2. The approach scored better than other well-established Semantic Web tools in terms of function (Section 6.9.2) as well as form (Section 6.3 where interface aesthetics metrics were compared).

Users directly assessed their aesthetic appeal to the Affective Graphs interface as a part of the user evaluations. Overall, users judged the interface highly on aesthetic qualities such as pleasant, beautiful, structured and stimulating. The long-term evaluations had little or no variability on a user's aesthetic perception of the approaches — this showed that users are most likely to appreciate the solution even in the longer term.

These principles need to be taken into consideration while developing the solution, in addition to involving user communities. While all the principles (such as the node-link principles) would clearly not be applicable for all proposed solutions, many can be easily addressed while designing the interface. Principles such as designing the layouts can be easily and less expensively addressed by providing several mock-ups and using objective measurements to select the best performing layout (as discussed in Section 6.3). A minimalist design is also mostly preferred since it would help reducing clutter and cognitive burden on the users. Consultations with users from the initial stages of the design and development processes can provide impressive results, and by extensively using a user centered design approach can also help the solution aesthetically pleasing.

The continuous engagement with different user communities within and outside organisational hierarchies ensure the development of highly usable solutions. Formal evaluations, interviews and focus group sessions conducted while developing the solutions as a part of iterative user centered design process ensured that usability issues are identified at the earliest (Sections 5.6 and 6.9).

Early identification of functional and non-functional requirements (Section 4.2) is also possible by a thorough understanding of domains and users. Studying the datasets and user needs can also provide hints to design solutions that can be verified with users. A careful attention to aesthetic design principles as well as continuously engaging users while developing solutions can help design more aesthetically pleasing solutions. Within an enterprise, users from all hierarchical levels (from managers to knowledge workers) should be involved in the process of designing and developing solutions.

Hence, generic, usable, effective and pleasing semantic interfaces can be developed by considering the following points: adhering to strict standards and protocols; iterative User-Centered Design; application of typical and familiar interaction and visual paradigms.

While the development of the two solutions have been focussed within a knowledge management scenario, with two real world use cases motivated by the need for easier and effective access to organisational information, the evaluation within this framework (Organisational Knowledge Management) have been limited. This is due to the expenses involved in recruiting highly experienced and skilled knowl-

edge workers. Though such users have been highly engaged in the development of the solutions (albeit in lower numbers), their inputs have been extremely important. Preliminary evaluations with lower number of users within large organisations, involving real datasets have been positive and indicate a high acceptability of the solutions. However, their performance within real world use cases, in live scenarios need to be evaluated in a much larger scale.

8.2 IMPACT

The prospect of a better management of information has prompted the adoption of a Semantic Web based framework in many organisations. Indeed, the field of Semantic Web is in it's infancy and much progress has been noted since its initial conception in 1999. A plethora of interfaces have been developed by the research community, and its adoption into the wider world has been growing over the years. The impact of the technologies developed, I believe are notable in the longer term: exploring a dataset in any domain and dataset of choice can be very valuable for several communities. A more immediate benefit has been noted in two areas: Knowledge Management and software development. The research conducted during the development of the *.views*. has been in collaboration with the aerospace industry, scientists and software developers as a part of research projects. In the Samulet project, *.views*. has been evaluated to a Technology Readiness Level (TRL) 6 at Rolls Royce for the two use cases it was evaluated for. TRL assessments are generally made to understand the fitness, deployability and scalability of a solution, and is assessed on a scale of 1–9, where TRL 1 indicates the initiation of an idea, and TRL 9 indicates a mature product in service⁶. A TRL assessment of 6 indicates that the solution has passed a full scale system evaluation. It is a significant achievement, I believe for a research prototype to be accepted as in a near production-ready state. The multiple visual approach embedded in search results have been highly appreciated within the organisation, and there is more investigation into understanding how such techniques can be incorporated in enterprise search systems for visualising search results.

The next impact was on steering a more visual approach within semantic search tools: a hybrid search system, K-Search was initially developed as a research prototype, which later on matured to be a software product that is employed in various organisations and domains. The ideas of multiple visualisations have been fed-in to the system and, in addition to results displayed in a text-based widget, there is now the possibility of visualising the results into different graphical widgets⁷. These are currently commercialised by K-Now⁸, a spin out company of the University of Sheffield, Department of Computer Science. The impact on the state of the art has been in extending techniques such as column faceted navigation and hybrid search. The faceted technique has been extended from a single path approach to a multi-path one, while the hybrid search has been extended from a semantic and keyword search to a unique interaction mechanism involving natural language and view based graphical approach. The thesis also

⁶ <http://www.publications.parliament.uk/pa/cm201213/cmselect/cmsctech/348/348we21.htm> discusses the assessments of technologies and capabilities at Rolls Royce

⁷ An example of the approach can be found at <http://grassportal.shef.ac.uk:8080/grassportal/>

⁸ <http://www.k-now.co.uk/k-now/>

proposed techniques, guidelines and principles for developing interactive visualisation systems for semantic web, which have been so far missing in the community. These contributions to the state of the art are discussed further in Section 7.3

A longer term, and with a potentially more far reaching impact, I feel is the explicit focus on aesthetic design while developing Semantic Web tools. Research so far in the semantic community has not considered the aesthetics of interfaces and the experiments conducted in the thesis show the benefits of an aesthetic interface. I hope this work encourages more research in the Semantic Web community, thereby paving the way for new stimulating interfaces and interaction mechanisms.

8.3 CONTINUING RESEARCH AND FUTURE WORK

Following from the research during my PhD and upon reflecting on the lessons learnt and the experience gained interacting with semantic data, I see several possibilities in continuing research. Most interesting among them are in three directions: expand the work in the thesis; investigate novel ways of visualising and interacting with multiple datasets; and investigate novel ways of visualising social media. In the following sections, I discuss how I propose to contribute toward these areas.

8.3.1 *A Combined Approach — a proposal*

Evaluations, focus groups, interviews and stimulating discussions with potential users were encouraging in establishing the validity of a highly visual approach in exploring semantic data. However, at this stage, research into visualising results of semantic queries and exploring and querying semantic data has been conducted in an isolated manner. While lessons learnt from the development and evaluation of one technique has fed into the development of the other, an exploration into a combined approach is necessary. It is to be noted that this, in spite of being an apparently simplistic task is far more complex in nature. The technical challenges in connecting the two systems are not as significant, as it requires the transferring of a SPARQL query from Affective Graphs to Points of View. However, the more fundamental challenge arises from understanding user behaviour and attitudes toward a new approach due to the following reasons:

ESTABLISHING AN APPROACH RATHER THAN SYSTEMS It is important to note that the approach of visual and interactive means toward exploring data is the next step of research. Technical implementation of such an approach may not necessarily imply physical integration of the two systems. What is more essential, is to conduct a deeper analysis of the two methods and explore user opinion on the new approach.

TRADITIONAL SYSTEMS VS A VISUAL APPROACH Traditional search engines and keyword-based systems have created a largely text-based approach toward interacting with search systems, which most users have gained familiarity with. A visual approach is a different paradigm, which relies on highly interactive and visual means of communicating user interest and machine response. It is important to gain a deeper understanding how the two systems need to be bridged to exploit them to their fullest.

CONTROLLING THE PROCESS There is also a need to understand how much control (on the interaction, visual representations, layouts and so on) do users want while interacting with a strictly visual mechanism for querying semantic data as well as visualising results. A new paradigm should not be at the expense of user control and freedom, but should provide users with more flexibility. More research is needed in this area, to better understand how to address this.

EXISTENCE OF BROWSER AND TEMPLATE BASED EXPLORATION FOR SEMANTIC DATA

The Semantic Web community have long been developing and exploring semantic browsers and template based visualisations. It is necessary to understand how a multiple visualisation system could complement existing and established techniques in a generic manner.

UNDERSTANDING MOST APPROPRIATE VISUALISATIONS? A research question that was already identified during the development of Points of View was how to know which visualisations are appropriate for a particular user, dataset, task or domain. There is a need to explore this question further in order to understand how a generic approach can be taken to develop a comprehensive approach toward exploiting the two techniques.

EFFECT ON AESTHETICS Affective Graphs is a solution that was designed specifically with an attention to aesthetics. Integrating the two solutions would affect the aesthetic appeal on the final system. It is very important to understand how the two approaches can be integrated to form an aesthetically pleasing interface. I believe this needs a ground-up approach to design the solution, by isolating every element to be integrated in an aesthetically pleasing interface.

The combination of the two approaches is not as straight-forward and there is a need for answering these questions. Following the User-Centred Design methodology, this piece of work would be a natural continuation of the two isolated pieces of work. The first step would be to follow an extensive consultation with end users to understand how they would like to interact with the final system and what their expectations could be from the system. At this stage, it is essential not to be limited by the two systems, but to focus on the methodology rather than the implementation of it. It is also important to understand how much control should be placed on the user's discretion on the level of visualisations being incorporated as well as how the logical flow of the system should proceed.

8.3.2 *Visualising Semantic Data*

Affective Graphs was built on the idea that it is possible to generically explore and query any semantic dataset from any domain in an aesthetically pleasing manner. However, the limitation of the system is its restriction to only one dataset at a time. The next extension of the research would be to explore how to explore multiple datasets at the same time — this is a highly challenging task as it can involve different domains and specificity at times. For example, a user exploring DBpedia can browse through different concepts of the dataset. However, upon reaching a subset of the data (such as species descriptions of flowering plants in the Figure 77), the user has a very limited view of

the data. This is due to a transition in the Information Need of the data. While the user initially browsed the top-level concepts of the data hierarchy, the user's Information Need was more generic. However, with a deeper exploration into the lower levels of the hierarchy, the Information Need became more specific. This continues till the point the dataset cannot provide any more data. In our example, the dataset only provides information till the user browsed plants. The next level of data (flowering plants) could only identify 'grape' as a type of flowering plant.

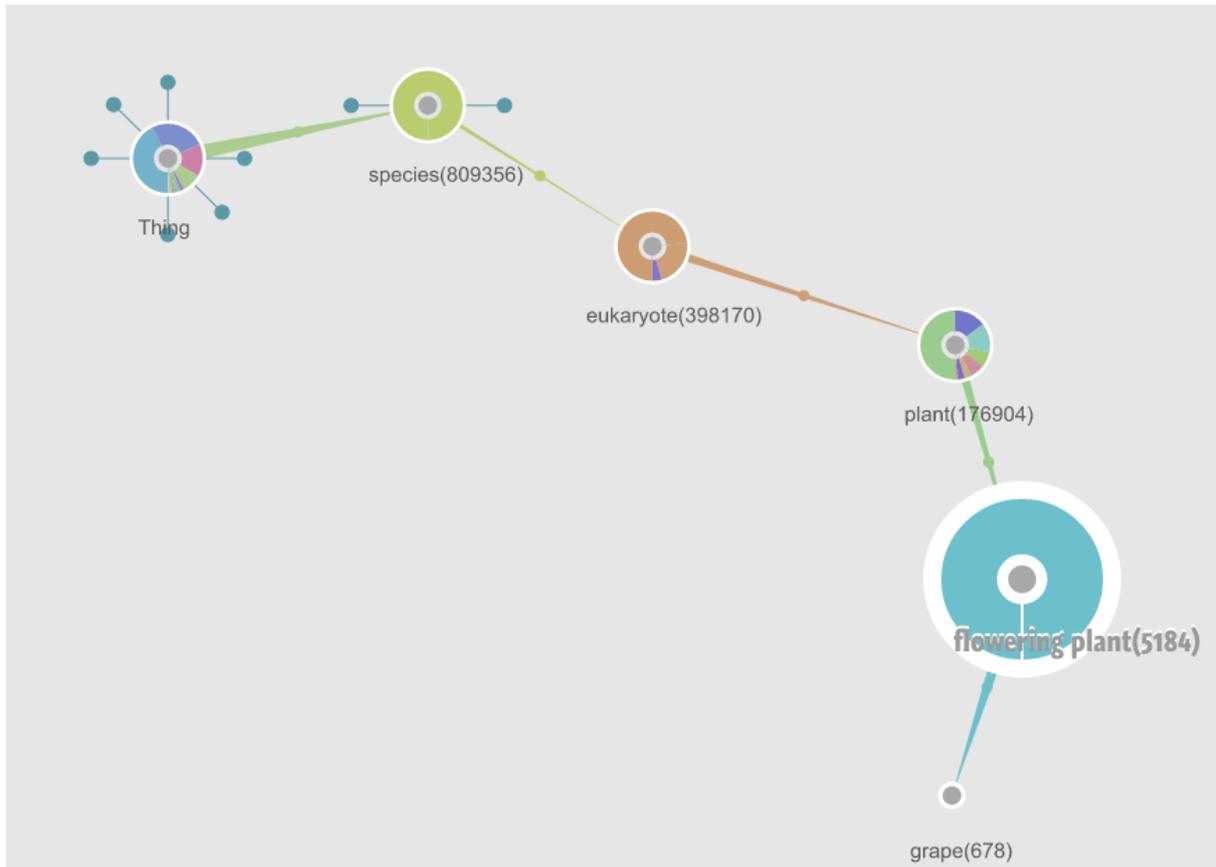


Figure 77: A transition in the Information Need from generic to highly specific results in the inability of a dataset in providing specific information

The same problem exists vice-versa — a highly specific dataset is unable to provide generic data. The next step in the research would be to exploit semantic networks and interLinked Datasets to provide mechanisms for users to “tunnel into” another dataset. This, of course presents a set of technological and cognitive challenges and raises a number of interesting questions:

HOW TO MAINTAIN CONTEXT OF USER EXPLORATION ? Navigating multiple datasets in a seamless manner requires the user to keep a track of their exploratory activity. This helps users to re-evaluate if they are on the right track or diverging from their goal. Furthermore, a record of their exploratory activity makes the user stay in focus and remain in context. It is important to understand how such information can be maintained and presented to the user.

HOW TO SCALE-UP TO EXTREMELY LARGE DATASETS ? Affective Graphs has been used and developed on datasets that have billions of triples. However, the constant generation of data and potential connections with other large datasets can easily require technologies to deal with trillions of triples. This is an interesting problem raising questions on two sides: how can the system scale up from a technical perspective?; and how can the system scale up from a cognitive perspective?

HOW COULD VISUALISATIONS BE USED TO REPRESENT ENTIRE DATASETS ? A proposed way to mitigate the problem of very large datasets is to build small visual representations that summarise the content of an entire dataset — for e.g., Affective Graphs uses pie charts to summarise the content of a concept (or a dataset). However, when navigating multiple datasets, new techniques of visualising several datasets should be investigated.

HOW CAN INTERACTION TECHNIQUES ALLOW SEAMLESS NAVIGATION OVER DATASETS ? A traditional browsing on the hypertext web requires users to click through links, and the browser displays new content on the page. It is important to understand how the same paradigm can be preserved, but implemented using visual means.

I believe that the benefits of a mechanism of navigating large datasets could be far-reaching: large semantic datasets in different domains and specialisations exist in different data stores, query-able on different endpoints. Research so far has only involved background processes to help integration or processing of data in a basic level, hidden from the view of end-users. However, the ability to seamlessly navigate large data stores could seed the next phase of research where users can directly engage with Semantic Data and browse the web of data.

8.3.3 *Visualising Social Media*

The vast amount of data currently available from Social Media (SM, i.e. Twitter, Flickr etc.) and LD is a mine of potentially invaluable information in many domains: medical science, education, public administration etc. This has paved the way for a new class of users that have very specific Information Needs that are often reflected in the available data but hard to satisfy. Such users are often highly knowledgeable of the domain but lack the technical skills to query the data with existing interfaces. Available tools to browse semantic data are often too complex, requiring the user to form semantic queries, and often textbased. On the other hand, existing interfaces presenting social media data are easier to use but often lack the added value of semantics: users are faced with a constant stream of new information and very few mechanisms to understand it.

The nature of inter-connectivity, self-description and hierarchical organisation is the very essence of Semantic Web. It enables machines and humans to understand and comprehend the data. Over the past year, I have been exploring social media from different perspectives and have found similarities among the two types of data, albeit, large differences between the two do exist. I believe that the first and greatest similarity in the two types of data is the interconnectedness in the data. Semantic Web describes how multiple concepts are linked with each other: in essence, how concepts are connected. Social media concerns with who interacts with whom: in essence, how people are connected. The next similarity is self-description: while Semantic Web uses vocabularies

and properties to describe a concept, social media uses person profiles and demographics to describe an individual or a community. The final similarity, I believe is how the two are connected: semantic data is hierarchical, with concepts being types and super classes of other concepts. Social media, on the other hand involve individuals sharing ideas, topics and concepts among each other. Discussions are based on topics and concepts, which are all semantically representable. This provides the building blocks for the next stage of research in this direction: how to represent social media using semantics, acknowledging the similarities between the two types of data.

Over the duration of my PhD, I have also been focussing on visualising social media data. Two main strands of visualisations have been explored: visualising content and visualising relationships and users. This has stemmed to the development of a few prototype systems such as [MCGL12](Context and Hierarchy Chain to visualise content), [CMC] (VisInfluence, to visualise relationships and users) and [GLMC11, LCG⁺11] (SimNET, to visualise relationships and users).

Initial user impressions on the working prototypes are positive, the evaluation of the same, in a realistic scenario is a challenging process and opens new directions for research. An evaluation of the systems during a real-life emergency situation (or even, a simulation) will be under highly stressful conditions. This requires careful thought and well designed experiments to understand how such evaluations can be best conducted. Furthermore, each of these systems are isolated pieces of developing work. Much research is necessary in order to understand how best to integrate such systems (or approaches) in a comprehensive system.

8.4 FINAL REMARKS

The research conducted over the past few years into making semantic data more consumable for end users has been highly enlightening and insightful. I have had the opportunity to closely work with end users and stakeholders during my research and have collaborated with researchers from several research fields. My experience within an industrial as well as an academic setting has been very special since I have had the opportunity to understand the expectations and limitations from both communities. Adopting a user-centred approach has made me better understand users and their expectations and re-iterating through the entire design process have been very exciting and highly stimulating.

I believe the beauty in technology can be delivered in simplicity and hiding the technology from end users to create a playful, stimulating and exciting environment can lead to more receptive and forgiving users. The word “forgiving” is a rather untechnical term that I use to express the fact that users will be willing to give a piece of technology another try if they are receptive and excited by the system. While it is important to ensure a system behaves as expected all the time, the unfortunate reality is that technology will fail at certain times. It is important to reduce the frustration of users and ensure they have a pleasant experience with the system. I believe that providing users with a pleasurable experience is the first step in doing so. Aesthetics and art are highly complex fields that have existed for centuries and are constantly evolving. Driven by cognitive perception, psychology, art and design, principles have been proposed to help guide designers to build better and more aesthetic products. My effort has been

to integrate principles from multiple disciplines for the Semantic Web to aid designers and developers build more pleasing solutions.

I feel that my research, spanning over the last four years have had several success stories and contributions. Preliminary research in understanding the principles of Semantic Web and an iterative user-centred approach toward developing a visualisation framework helped me understand the needs of users. The evaluations raised some interesting questions, highlighted key issues and helped seed the next phase of research. An exploration into the existing and well established Semantic Web tools and a literature survey highlighted that the attention to aesthetic design has been highly limited in the Semantic Web community. Following a comprehensive review of the literature across disciplines, I identified principles that relate to the field and used them to develop a visualisation solution for exploration and query of semantic datasets. The results from the evaluations have helped guide my thoughts toward possible future work, and I believe the impact of the continuing research are significant.

Overall, I feel that the greatest achievement of my thesis was incorporating a multidisciplinary approach toward making semantic data more accessible and consumable by users by empowering user communities steer design decisions. I hope this fosters a new direction in Semantic Web and Information Retrieval research and generates new, stimulating results in the near future.

Part V

APPENDIX

AESTHETIC METRICS

The following provides a description and the formulae to calculate the metrics as used in the evaluation. These definitions and formulae were proposed by Ngo.

Balance is defined as the distribution of optical weight in a picture, where optical weight is the perception that some objects appear heavier than others. Larger objects are heavier while smaller objects are lighter.

$$\text{Balance} = 1 - \frac{|\text{BM}_{\text{vertical}}| + |\text{BM}_{\text{horizontal}}|}{2} \in [0, 1] \quad (46)$$

Equilibrium represents stabilisation, a midway center of suspension. It can be defined as equal balance between opposing force, various visual objects are centers of forces. A layout is in equilibrium when its center coincides with the center of the frame.

$$\text{Equilibrium} = 1 - \frac{|\text{EM}_x| + |\text{EM}_y|}{2} \in [0, 1] \quad (47)$$

Symmetry denotes the balanced distribution of equivalent elements about a common line. Essentially representing axial duplication, symmetry defines how well a unit on one side of the center is replicated on the other side.

$$\text{Symmetry} = 1 - \frac{|\text{SYM}_v| + |\text{SYM}_h| + |\text{SYM}_r|}{3} \in [0, 1] \quad (48)$$

The measure of **sequence** relates to the way that visual objects are positioned in a layout with respect to the movement of the eye - heavier objects being on the top left, while lighter and smaller objects at the bottom right.

$$\text{Sequence} = 1 - \frac{\sum_{j=\text{UL,UR,LL,LR}} |q_j - v_j|}{8} \in [0, 1] \quad (49)$$

Rhythm relates to understand the variety in the arrangement, dimension, number and form of visual objects within a layout.

$$\text{Rhythm} = 1 - \frac{|\text{Rhythm}_x| + |\text{Rhythm}_y| + |\text{Rhythm}_{\text{Area}}|}{2} \in [0, 1] \quad (50)$$

Cohesion denotes how the aspect ratios of each visual element relates to the screen's width and height.

$$\text{Cohesion} = \frac{|\text{CM}_{\text{fl}}| + |\text{CM}_{\text{to}}|}{2} \in [0, 1] \quad (51)$$

Unity signifies coherence, where visual elements appear to belong together, seen together as one thing - similar sized objects, using less space between elements, larger margins and so on.

$$\text{Unity} = \frac{|\text{UM}_{\text{form}}| + |\text{UM}_{\text{space}}|}{2} \in [0, 1] \quad (52)$$

Order and Complexity is defined as the sum of all the above measures for a layout.

$$\text{Order_Complexity} = \frac{\sum_{i=1}^7 M_i}{7} \in [0, 1] \quad (53)$$

EVALUATION QUESTIONNAIRES - POINTS OF VIEW FOCUS GROUP SESSION WITH OPEN TASKS

Dear Participant,

Thank you for taking part in this evaluation for the GrassPortal Project. We will be looking at two interfaces. One is the official GrassPortal interface developed by Knowledge Now Ltd for the GrassPortal project. The other is an experimental interface for visual data browsing developed by Suvodeep Mazumdar as part of his PhD at the University of Sheffield. We will ask you to work in pairs testing first one system and then the other. You were asked to think about a query to work on before the meeting and this is what we will work on in the evaluation.

Your participation in today's activities and your opinions are very valuable to us and will help improve the system. Overall we expect the session to last about 3 hours with each person working for 80 to 90 minutes with breaks to rotate the pair of participants through three half hour slots. The work is scheduled as follows:

Group Introduction (10 mins)

- Explanation of what the tests involve
- Any questions
- Sign consent form
- Complete Person Profile

Test 1 Victoria K-now (30 mins)

- Introductory video (2mins)
- First participant tests system (10 mins)
- Second participant tests system (10 mins)
- Complete K-now system questionnaire (5 mins)

Test 2 Suvodeep Visualisation (30 mins)

- Introductory video (2mins)
- First participant tests system (10 mins)
- Second participant tests system (10 mins)
- Complete visualization system questionnaire (5 mins)

Group Discussion (45 mins)

- Tell us what you think.

13:00-13:10	Group Introduction	
13:10-13:40	Pair 1 - Evaluator 1	Pair 2 - Evaluator 2
13:40-14:10	Pair 2 - Evaluator 1	Pair 3 - Evaluator 2
14:10-14:40	Pair 3 - Evaluator 1	Pair 1 - Evaluator 2
14:45-15:00	Group Discussion	

Consent Form

Conditions for taking part in the Study

Your participation is voluntary and you may choose to withdraw at any time and for any or no reason.

If you withdraw from the study please inform the evaluators if you wish that the information you have provided be destroyed.

All information you provide will be anonymised; you will not be identifiable in any way.

The information derived from this study will be used only for the purposes of the GrassPortal project and any associated publications.

Name: _____

1. I understand that my participation is voluntary and that I am free to withdraw at any time without giving any reason.
2. I understand that my responses will be anonymised before analysis.
I give permission for members of the research team to have access to my anonymised responses.

(signature/initials)

Personal Profile ID _____

1. Age: < 25 25-34 35-44 45-54 > 54

2. Did you participate in the GrassPortal Requirements Meeting?

yes

no

3. Would you consider yourself to be? (tick all that apply)

Scientist / expert in biological and environmental science

IT Specialist / developer

Student

Other _____

4. Please briefly describe your role

5. Concerning the use of web based search systems for biological and environmental information, would you consider yourself to be?

Expert

Competent

Novice

5. Which kind of information do you use most? (tick all that apply)

Species data

Traits data

Environmental data

Genetic data

Other, please specify: _____

6. How often do you search for information to accomplish your job?

Daily

Many times a week

Once a week

Several times a month

Once a month

Less than once a month

I ask someone to do the search for me

7. Please briefly describe a query that you would like to search for today

GrassPortal (K-Search) User Satisfaction Questionnaire ID ____

1. How would you rate your overall impression of the system?

5	4	3	2	1
Excellent				Poor

5	4	3	2	1
Satisfying				Frustrating

2. How easy was it to learn how to use the system?

5	4	3	2	1
Very Easy				Very hard

3. How easy was it to formulate a query?

5	4	3	2	1
Very Easy				Very hard

4. How closely did the results of your query match your needs?

5	4	3	2	1
Perfectly				Not at all

5. How good was the interface at helping you understand what the results were?

5	4	3	2	1
Excellent				Poor

6. How would you rate the following features of the system?**Concepts**

5	4	3	2	1	n/u
Excellent				Poor	Not used

Grass Taxonomy

5	4	3	2	1	n/u
Excellent				Poor	Not used

Keyword Search

5	4	3	2	1	n/u
Excellent				Poor	Not used

Search Results

5	4	3	2	1	n/u
Excellent				Poor	Not used

Visualize Results

5	4	3	2	1	n/u
Excellent				Poor	Not used

Filter

5	4	3	2	1	n/u
Excellent				Poor	Not used

7. What did you like least about the system?**8. What did you like most about the system?**

GrassPortal (.view.) User Satisfaction Questionnaire ID ____

1. How would you rate your overall impression of the system?

5	4	3	2	1
Excellent				Poor

5	4	3	2	1
Satisfying				Frustrating

5	4	3	2	1
Stimulating				Dull

2. How easy was it to learn how to use the system?

5	4	3	2	1
Very Easy				Very hard

3. How easy was it to explore features by trial and error?

5	4	3	2	1
Very Easy				Very hard

4. How easy was it to formulate a query?

5	4	3	2	1
Very Easy				Very hard

5. How good was the interface at helping you understand what the results were?

5	4	3	2	1
Excellent				Poor

6. How would you rate the following features of the system?**Visualize Results**

5	4	3	2	1
Excellent				Poor

Filter

5	4	3	2	1
Excellent				Poor

System Speed

5	4	3	2	1
Excellent				Poor

System Reliability

5	4	3	2	1
Excellent				Poor

7. What did you like least about the system?**8. What did you like most about the system?****9. Any Comments/Suggestions ?**

EVALUATION QUESTIONNAIRES - POINTS OF VIEW FOCUS GROUP SESSION WITH GUIDED TASKS

Dear Participant,

Thank you for having accepted to take part in the evaluation of the grassPortal system. What you will try is part of the visualization system to display information respect to different dimensions and dynamically explore the distribution of data respect to specific values.

Your participation in today activities and your opinions are invaluable to us and will help improve the system. Overall we expect the session to last 30 to 40 minutes scheduled as follow:

1. **Introduction to the grassPortal project and its visualization tool 10 minutes**
 - short explanation of what is going on behind the scene
 - ask any question
2. **Information Exploration task: 20 minutes**
 - a written task is given to you to direct your activity
 - use grassPortal to try to accomplish the task
3. **User satisfaction questionnaire: 5 minutes**
 - fill in a questionnaire
4. **User experience interview: 5 minutes**
 - the researcher will ask you a few questions about your experience

You are not obliged to use the system for the whole time allocated. You decide when a task is over by informing the researcher.

During the two sessions a researcher will observe your activity and your interaction will be logged. Should you wish to take notes, please use the paper provided. All recording is anonymous and will remain completely confidential.

Should you have any questions, please do not hesitate to ask.
Thank you for participating,

The grassPortal evaluation team

Personal Profile

1. Age: <25 25-34 35-44 45-54 > 54

2. Position: _____

3. Are you familiar with graphical visualization systems? If yes, which one have you used?

Information exploration with multi-faceted views

You will be presented with 5 different types of visualizations – tag cloud, list results, pie chart, geoplot, and bar chart. In addition, you will be presented with a filtering interface (query box) which can be used to add any query that you feel is useful.

General Plots – finding out aggregates

1. How many grass species have **ID Ranges** value roughly higher than > 444000?
2. Working from 1, how many grasses have **names** containing 'cae'?
3. Clear all the filters, how many grasses are contained in the system ?
4. Add filter for **plant density**. How many grasses have **plant density** solitary or caespitose ?

Aggregate visualizations – Tag Cloud, Bar Chart, Pie Chart

5. Consider only the grasses which have solitary **plant density**. Visualize the grasses. Looking only at the tagCloud, which region has the highest distribution of grasses? Infer the name of the country if possible
6. Keeping the same filter values and looking at the pie chart and bar chart, what percent of the distribution is unknown?
7. Clear all the filters. Add a filter element **hiluim <shape>**. How many grasses have elliptic hiluim ? Visualize them. Looking only at the results widget, can you infer how many grasses are present?

Geographical visualization and Result lists

Remove all of your filters, clear all the existing filters.

8. Select a new filter, **anthers <color>**. Set your query to only look at anthers which are 'yellow' colored. Visualize them. Look at the geographical map. Control the zoom so that the entire world can be seen on the map. Can you name two areas (countries or continents) which have the greatest number of grass species?

9. With the same filter settings, zoom down to London. You should see a marker indicating that a grass with yellow anthers have been found in London. Click on the marker. Can you Name the grass and the Region ?

10. Click on the information window where it says 'click for details'. On the displayed dialog, scroll down to the bottom. Looking at the window, can you tell what is the plant duration of this particular grass?

11. Clear all the filters. Search for all the grasses which have their names containing 'chasmanthium'. Looking only at the map, can you tell how many of the grasses are found in Europe and how many in North America? Zoom down to Mexico. Name the grass which is found in Mexico (by clicking on the marker) and by looking at the details (click for details), can you find out the color(s) of its caryopsis ?

User Satisfaction Questionnaire

1. Overall reaction to the system

1	2	3	4	5
difficult				easy

1	2	3	4	5
frustrating				satisfying

1	2	3	4	5
dull				stimulating

2. Learning

1. Learning to operate the system

1	2	3	4	5
difficult				easy

2. Exploring new features by trial and error.....not applicable

1	2	3	4	5
difficult				easy

3. Performing tasks is straightforward

1	2	3	4	5
never				always

3. Task Flow

1. Understanding where to start with the task was

1	2	3	4	5
hard				easy

2. Choosing the step to do next was:

1	2	3	4	5
difficult				easy

4. Result Display

1. Manipulating the filters was

1	2	3	4	5
difficult				easy

2. Browsing the results in the visualization panel was

1	2	3	4	5
difficult				easy

3. Exploring the documents via the visualization panel was

1	2	3	4	5
difficult				easy

6. System Capabilities

1. System speed

1	2	3	4	5
slow				fast

2. System reliability

1	2	3	4	5
low				high

List the most negative aspect(s):

List the most positive aspect(s):

Do you have any further comment?

Dear Participant,

Thank you for having accepted to take part in the evaluation of the Visual Exploratory Tool (Affective Graphs) for the Samulet knowledge base.

This tool has been developed at the University of Sheffield as part of the Samulet research project, the final goal of which is to create an effective system for accessing and sharing technical information inside the organisation.

Please consider we are not evaluating a complete system but just collecting opinions, impressions and feelings on a radically new interaction paradigm, that will allow to explore complex knowledge bases in a visual way.

The tool aims at visually exploring and browsing a set of documents (in this case, technical Variances) by clicking on interesting items; for each clicked item the system reports the number of documents found and their distribution; you can refine your search by further clicking on sub-concepts or by exploring alternative paths. Please note that the system does not offer searching capabilities.

Your participation in today activities and your opinions are very valuable to us and will help improve the system. Overall we expect the session to last 30 to 40 minutes scheduled as follow:

1. **Introduction: 10 minutes**
 - Watch a 2 minutes video that introduces the system and highlight its capabilities
 - ask any questions and sign the consent form
 - fill in a background questionnaire
2. **Task completion (given): 10 minutes**
 - a written task is given to you to direct your exploration
 - ask the experimenter for assistance at any time
 - use the system to try to accomplish the task given
3. **Task completion (chosen) and system exploration: 10 minutes**
 - complete your chosen task
 - explore the system capabilities at your ease
4. **User satisfaction questionnaire: 10 minutes**
 - fill in a questionnaire
5. **Focus Group: 15 minutes**
 - A group discussion about the experience, suggested improvements and desired capabilities

You are not obliged to use the system for the whole time allocated. You decide when a task is over by informing the researcher.

During the session a researcher will observe your activity and your queries will be logged. Should you wish to take notes, please use the paper provided.

Should you have any questions, please do not hesitate to ask.

Thank you for participating,
The OAK Group, Sheffield University

Personal Profile

1. Age: <25 25-34 35-44 45-54 > 54

2. Position: _____

2. Please briefly describe your role

3. Please describe your responsibilities

4. Would you consider yourself to belong to?

- Design
- Service
- Design for Service
- Other _____

5. Which kind of information do you use most? (tick all that apply)

- Event Reports
 - Strip Reports
 - Technical Variances
 - Service bulletins
 - Modification bulletins
- (list continued on next page)

- Engine manuals
- Ground base system data
- Telemetry
- Air On Ground customer support data
- Maintenance data
- Customer support data
- CFBU (Customer Facing Business Unit) data
- OBU (Operational Business Unit) data
- Other, please specify: _____

6. Which online information do you usually access? (tick all that apply)

- Capability Intranet
- Engine Data Center
- SDM (Service Data Manager)
- InfoCentre
- People Pages
- Lessons Learnt Logs
- Community of Practice
- Invoices database
- Company Databases (e.g. Materials Database)
- Engineering Guides (e.g. MMMs, DRAs, HIPLs, GPGs)
- Other, please specify: _____

7. How often do you search for information to accomplish your job?

- Daily
- Many times a week
- Once a week
- Several times a month
- Once a month
- Less than once a month
- I ask someone to do the search for me

8. How often do you use Internet (mail, website surfing etc.)?

- never
- rarely
- sometimes
- often
- all the time

Practice Task

You are responsible for the performance of a manufacturing facility based at Derby. A meeting has been set up with the responsible of Technical Variance team in order to monitor Technical Variance requests.

For the meeting you have to produce slides to give an overview of the distribution of operators who have requested technical variances on a particular component ('nose cowl').

Your first step is to browse the available Technical Variances to find out how many documents have references to components (hasComponent). When you have found the desired subset of document, you want to analyse them by operator (hasOperator), to get an overview of the number of requests for each operator.

- **Task 2**

Please describe your task to the researcher.

Why did you choose this topic?

Have you done this exploration before?

User Satisfaction Questionnaire

1. Overall reaction to the system

1	2	3	4	5
difficult				easy

1	2	3	4	5
frustrating				satisfying

1	2	3	4	5
dull				stimulating

1. I am able to complete my work quickly using this system.

1	2	3	4	5
Strongly agree				Strongly disagree

2. I feel comfortable using this system.

1	2	3	4	5
Strongly agree				Strongly disagree

3. It is easy to find the information I need.

1	2	3	4	5
Strongly agree				Strongly disagree

4. The information provided with the system is easy to understand.

1	2	3	4	5
Strongly agree				Strongly disagree

5. The information is effective in helping me complete my work.

1	2	3	4	5
Strongly agree				Strongly disagree

6. The interface of this system is pleasant

1	2	3	4	5
Strongly agree				Strongly disagree

7. I like using the interface of this system.

1	2	3	4	5
Strongly agree				Strongly disagree

8. This system has all the functions and capabilities I expect it to have.

1	2	3	4	5
Strongly agree				Strongly disagree

2. Learning

1. Learning to operate the system

1	2	3	4	5
difficult				easy

2. Exploring new features by trial and error.....not applicable

1	2	3	4	5
difficult				easy

3. Performing tasks is straightforward

1	2	3	4	5
never				always

3. Visual Exploration

1. Selecting the concepts and exploring their distribution was

1	2	3	4	5
confusing				very clear

1	2	3	4	5
difficult				easy

1	2	3	4	5
Not useful at all				useful

2. The statistics on the number of documents were:

1	2	3	4	5
confusing				very clear

1	2	3	4	5
difficult				easy

1	2	3	4	5
Not useful at all				useful

1	2	3	4	5
Not relevant at all				Very relevant

3 The possible values of a concept were

1	2	3	4	5
confusing				very clear

1	2	3	4	5
difficult				easy

1	2	3	4	5
Not useful at all				useful

1	2	3	4	5
Not relevant at all				Very relevant

4. Choosing alternative concepts to explore was

1	2	3	4	5
confusing				very clear

1	2	3	4	5
difficult				easy

1	2	3	4	5
Not useful at all				useful

4. Using the right hand sidebar to refine your exploration was

1	2	3	4	5
confusing				very clear

1	2	3	4	5
difficult				easy

1	2	3	4	5
Not useful at all				useful

5. The colours on the graphs were

1	2	3	4	5
confusing				helpful

1	2	3	4	5
Not useful at all				useful

5. System Capabilities

1. System speed

1	2	3	4	5
difficult				easy

List the most negative aspect(s):

List the most positive aspect(s):

Interview Script

1. Should you describe what the system allows you to do to a colleague what would you say?

2. How do you feel about the level of control you had?

3. What is your opinion on having only a visual mean to explore a set of document?

4. What would you use this system for ?

4. Would you prefer using this system standalone or in combination with a search system?

5. Did you explore multiple concepts for one task?

6 Would you consider the use of this system for your job? How useful would it be?

7. Are there any other comments you would like to make about your experience?

AFFECTIVE GRAPHS EVALUATION QUESTIONNAIRES: CASUAL AND EXPERT USERS

SEALS System Usability Scale "SUS" Questionnaire

We are using the email you provided only as an identifier. We are not going to send you any emails, nor distribute it by any means.

* Required

Please choose the tool that you used from the list: *

K-Search

I think I would like to use the system frequently. *

1 2 3 4 5

Strongly Disagree Strongly Agree

I found the system unnecessarily complex. *

1 2 3 4 5

Strongly Disagree Strongly Agree

I thought the system was easy to use. *

1 2 3 4 5

Strongly Disagree Strongly Agree

I think I would need the support of a technical person to be able to use this system. *

1 2 3 4 5

Strongly Disagree Strongly Agree

I found the various functions in this system were well integrated. *

1 2 3 4 5

Strongly Disagree Strongly Agree

I thought there was too much inconsistency in this system. *

1 2 3 4 5

Strongly Disagree Strongly Agree

I would imagine that most people would learn to use this system very quickly. *

1 2 3 4 5

Strongly Disagree Strongly Agree

I found the system very tedious / troublesome to use. *

1 2 3 4 5

Strongly Disagree Strongly Agree

I felt very confident using the system. *

1 2 3 4 5

Strongly Disagree Strongly Agree

I needed to learn a lot of things before I could get going with this system. *

1 2 3 4 5

Strongly Disagree Strongly Agree

Submit

AFFECTIVE GRAPHS EVALUATION QUESTIONNAIRES: LEARNABILITY

Demographics Questionnaire

We are using the email you provided only as an identifier. We are not going to send you any emails, nor distribute it by any means.

* Required

Please enter your email address: *

Year of Birth: *

Gender: *

Profession: (In case you are a student, please state your subjects as well) *

My knowledge of formal query languages: (e.g.: SQL, RDQL, SPARQL, etc.) *

advanced	good	average	little	none
<input type="radio"/>				

My knowledge of the Semantic Web: *

advanced	good	average	little	none
<input type="radio"/>				

My knowledge of Ontologies: *

advanced	good	average	little	none
<input type="radio"/>				

My knowledge of visual interfaces (any kinds of interface similar to the one you just tested) *

advanced good average little none

The training session was helpful to you *

1 2 3 4 5

Strongly Agree Strongly Disagree

The printed training material was helpful to you *

1 2 3 4 5

Strongly Agree Strongly Disagree

What are the things that you liked about the system that you think they changed over sessions and by learning the tool and how did they change (positively or negatively) and why? *

What are the things that you didn't like about the system that you think they changed over sessions and by learning the tool and how did they change (positively or negatively) and why? *

Extended Questionnaire

We are using the email you provided only as an identifier. We are not going to send you any emails, nor distribute it by any means.

* Required

The system's query language was easy to understand and use. *

1 2 3 4 5

Strongly Agree Strongly Disagree

Tasks can be performed in a straight-forward manner *

1 2 3 4 5

Strongly Agree Strongly Disagree

Exploring new features by trial and error is: *

1 2 3 4 5

Easy Difficult

Remembering features and how to use them is: *

1 2 3 4 5

Easy Difficult

Understanding the structure of the interface is: *

1 2 3 4 5

Easy Difficult

What did you like about the system you have just tested and why? *

What things you didn't like about the system you have just tested and why? *

Assessing complexity of tasks

1. List the name, page and homepage of organizations.
2. List all the conference venues and their meeting rooms.
3. Give me the people with first name "Knud".
4. Give me the description and summary of keynote talks which took place at "www" conferences and the name of the presenter.

1. Give me the name, homepage and page of people who were workshop organizers for a workshop about "ontology matching".
2. Give me the proceedings whose title contains "Semantic Search".
3. List the programme committee members and the conference events they participated at.
4. List the name, familyName and status (e.g., academia, industry) of all people.

1. List all the people who have given keynote talks.
2. Give me the organizations whose name contains "Karlsruhe"
3. Give me the page and homepage of each person whose status is "Academia" and was a chair of a session event and give me its location.
4. List the location, homepage and summary of all tutorial events.

Please rank the following question categories (similar to the ones you saw) according to how easy they were for you to construct. *

Submit

System Usability Scale "SUS" Questionnaire

We are using the email you provided only as an identifier. We are not going to send you any emails, nor distribute it by any means.

* Required

I think I would like to use the system frequently. *

1 2 3 4 5

Strongly Disagree Strongly Agree

I found the system unnecessarily complex. *

1 2 3 4 5

Strongly Disagree Strongly Agree

I thought the system was easy to use. *

1 2 3 4 5

Strongly Disagree Strongly Agree

I think I would need the support of a technical person to be able to use this system. *

1 2 3 4 5

Strongly Disagree Strongly Agree

I found the various functions in this system were well integrated. *

1 2 3 4 5

Strongly Disagree Strongly Agree

I thought there was too much inconsistency in this system. *

1 2 3 4 5

Strongly Disagree Strongly Agree

I would imagine that most people would learn to use this system very quickly. *

1 2 3 4 5

Strongly Disagree Strongly Agree

I found the system very tedious / troublesome to use. *

1 2 3 4 5

Strongly Disagree Strongly Agree

I felt very confident using the system. *

1 2 3 4 5

Strongly Disagree Strongly Agree

I needed to learn a lot of things before I could get going with this system. *

1 2 3 4 5

Strongly Disagree Strongly Agree

Submit

Learnability Questionnaire on Aesthetics

* Required

The Layout of the system was *

1 2 3 4 5

Symmetric Asymmetric

*

1 2 3 4 5

Well Structured Unstructured

*

1 2 3 4 5

Well Organised Disorganised

The style of the system was *

1 2 3 4 5

Consistent Inconsistent

The colors used in Affective Graphs were *

1 2 3 4 5

Pleasant Unpleasant

*

1 2 3 4 5

Helpful Unhelpful

Overall, the system was *

1 2 3 4 5
Pleasant

*

1 2 3 4 5
clean

*

1 2 3 4 5
clear

*

1 2 3 4 5
Predictable Unpredictable

*

1 2 3 4 5
Dull Stimulating

*

1 2 3 4 5
Simple Complicated

*

1 2 3 4 5
Creative Unimaginative

*

*

1 2 3 4 5

Ugly Beautiful

*

1 2 3 4 5

Stressful Exciting

AFFECTIVE GRAPHS EVALUATION TASKS: LEARNABILITY

G.0.1 *Simple Tasks (ST)*

Find the people with first name 'Knud'.

Find the "inproceedings" whose title contains 'Semantic Search'.

Find the organizations whose name contains 'Karlsruhe'.

G.0.2 *Multiple Attributes Task (MAT)*

List the name, page and homepage of organizations.

List the name, familyName and status of all people.

List the location, homepage and summary of all tutorial events.

G.0.3 *Multiple concepts Task (MCT)*

List all the conference venues and their meeting rooms.

List the programme committee members and the conference events they participated at.

List all the people who have given keynote talks.

G.0.4 *Complex Task (CT)*

Find the description and summary of keynote talks which took place at 'WWW' conferences and the name of the presenter.

Find the name, homepage and page of people who were workshop organisers for a workshop about 'Ontology Matching'.

Find the page and homepage of each person whose status is 'Academia' and was a chair of a session event and find its location.

G.0.5 *Exploratory Task (ET)*

1. Imagine you are a young researcher, starting your career in 'Ontology alignment'. Since the organisation you are affiliated to conducts research in a different area, you do not have direct access to experts in your area of research. The only access to information is via Affective Graphs, which provides visual means to look for information. Using such systems, can you identify a few researchers in the area of your interest and why have you chosen them?

2. Imagine you are organising a day-long workshop on knowledge management in business at an organization. As a part of the workshop, there would be two tutorials from experts. Who are the experts you would choose and why?

BIBLIOGRAPHY

- [AAW07] Gennady Andrienko, Natalia Andrienko, and Stefan Wrobel. Visual analytics tools for analysis of movement data. *SIGKDD Explor. Newsl.*, 9(2):38–46, December 2007.
- [AB08] Ben Adida and Mark Birbeck. Rdfa primer: Bridging the human and data webs, October 2008.
- [Ack89] Russell L Ackoff. From data to wisdom. *Journal Of Applied Systems Analysis*, 16(1):3–9, 1989.
- [ACK04] Nikos Athanasis, Vassilis Christophides, and Dimitris Kotzinos. Generating on the fly queries for the semantic web: The ics-forth graphical rql interface (grql). In SheilaA. McIlraith, Dimitris Plexousakis, and Frank Harmelen, editors, *The Semantic Web â ISWC 2004*, volume 3298 of *Lecture Notes in Computer Science*, pages 486–501. Springer Berlin Heidelberg, 2004.
- [AG03] Elias M. Awad and Hassan M. Ghaziri. *Knowledge Management*. Prentice Hall, February 2003.
- [AKM09] Erick Antezana, Martin Kuiper, and Vladimir Mironov. Biological knowledge management: the emerging role of the semantic web technologies. *Briefings in Bioinformatics*, 10(4):392–407, 2009.
- [AL99] Maryam Alavi and Dorothy E. Leidner. Knowledge management systems: issues, challenges, and benefits. *Commun. AIS*, 1(2es), February 1999.
- [AL01] Maryam Alavi and Dorothy E. Leidner. Review: Knowledge management and knowledge management systems: Conceptual foundations and research issues. *MIS Quarterly*, 25(1):107–136, 2001.
- [AL11] Ahamed Altaboli and Yingzi Lin. Objective and subjective measures of visual aesthetics of website interface design: The two sides of the coin. In JulieA. Jacko, editor, *Human-Computer Interaction. Design and Development Approaches*, volume 6761 of *Lecture Notes in Computer Science*, pages 35–44. Springer Berlin Heidelberg, 2011.
- [ALB11] Maneesh Agrawala, Wilmot Li, and Floraine Berthouzoz. Design principles for visual communication. *Communications of the ACM*, 54(4):60–69, April 2011.
- [AMkP04] Chadia Abras, Diane Maloney-krichmar, and Jenny Preece. *User-Centered Design*. Sage Publications, 2004.

- [Atk73] Charles Atkin. Instrumental utilities and information seeking. In Peter Clarke, editor, *New Models for Mass Communication*, pages 205–242. SAGE Publications, Beverly Hills, 1973.
- [Avo4] Grigoris Antoniou and Frank van Harmelen. *A Semantic Web Primer*. MIT Press, Cambridge, MA, USA, 2004.
- [AWS92] Christopher Ahlberg, Christopher Williamson, and Ben Shneiderman. Dynamic queries for information exploration: An implementation and evaluation. ACM Press, 1992.
- [Bato3] Marcia Bates. Toward An Integrated Model of Information Seeking and Searching. *Graduate School of Education and Information Studies*, 2003.
- [BB08] C. Becker and C. Bizer. DBpedia Mobile-A Location-Aware Semantic Web Client. *Proceedings of the Semantic Web Challenge*, 2008.
- [BBD09] Fabian Beck, Michael Burch, and Stephan Diehl. Towards an aesthetic dimensions framework for dynamic graph visualisations. In Ebad Banissi, Liz J. Stuart, Theodor G. Wyeld, Mikael Jern, Gennady L. Andrienko, Nasrullah Memon, Reda Alhadj, Remo Aslak Burkhard, Georges G. Grinstein, Dennis P. Groth, Anna Ursyn, Jimmy Johansson, Camilla Forsell, Urska Cvek, Marjan Trutschl, Francis T. Marchese, Carsten Maple, Andrew J. Cowell, and Andrew Vande Moere, editors, *IV*, pages 592–597. IEEE Computer Society, 2009.
- [BC10] Jonathan Butters and Fabio Ciravegna. Authoring technical documents for effective retrieval. In *Proceedings of the 17th international conference on Knowledge engineering and management by the masses, EKAW'10*, pages 287–300, Berlin, Heidelberg, 2010. Springer-Verlag.
- [BCC⁺08] Ravish Bhagdev, Ajay Chakravarthy, Sam Chapman, Fabio Ciravegna, and Vita Lanfranchi. Creating and using organisational semantic webs in large networked organisations. In Amit Sheth, Steffen Staab, Mike Dean, Massimo Paolucci, Diana Maynard, Timothy Finin, and Krishnaprasad Thirunarayan, editors, *The Semantic Web - ISWC 2008*, volume 5318 of *Lecture Notes in Computer Science*, pages 723–736. Springer Berlin / Heidelberg, 2008.
- [BCCGP04] R. Benjamins, J. Contreras, O. Corcho, and A. Gómez — P. A. rez. Six challenges for the semantic web. *AISSIGSEMIS Bulletin*, 1(1): 24–25, 2004.
- [BGFMo8] Ann Blandford, Thomas R G Green, Dominic Furniss, and Stephann Makri. Evaluating system utility and conceptual fit using cassm. *International Journal of Human-Computer Studies*, 66(1071-5819):393–409, 2008.
- [BGG96] Martin B., Subramanian G.H., and Yaverbaum G.J. Benefits from expert systems: An exploratory investigation. *Expert Systems with Applications*, 11(1):53–58, 1996.

- [BH99] Hugh Beyer and Karen Holtzblatt. Contextual design. *interactions*, 6(1072-5520):32–42, 1999.
- [BHBL09] Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked Data - The Story So Far. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 2009.
- [BHH⁺06] M. Butler, D. Huynh, B. Hyde, R. Lee, and S. Mazzocchi. Longwell project page, 2006.
- [BL06] Tim Berners-Lee. Linked data-design issues, 2006.
- [BLCo8] Tim Berners-Lee and Dan Connolly. Notation3 (n3): A readable rdf syntax, 2008.
- [BLCC⁺06] Tim Berners-Lee, Yuhsin Chen, Lydia Chilton, Dan Connolly, Ruth Dhanaraj, James Hollenbach, Adam Lerer, and David Sheets. Tabulator: Exploring and analyzing linked data on the semantic web. In *Proceedings of the 3rd International Semantic Web User Interaction*, 2006.
- [BLHL01] Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. *Scientific American*, 284(0036-8733), May 2001.
- [BLK⁺09] Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. Dbpedia - a crystallization point for the web of data. *Web Semant.*, 7(3):154–165, September 2009.
- [BMT05] Elena Paslaru Bontas, Malgorzata Mochol, and Robert Tolksdorf. Case studies on ontology reuse. In *Proceedings of the 5th International Conference on Knowledge Management IKNOW05*, 2005.
- [BN06] M. Bar and M. Neta. Humans prefer curved visual objects. *Psychological science*, 17(8):645–648, 2006.
- [BP97] Uwe M. Borghoff and Remo Pareschi. Information technology for knowledge management. *J. UCS*, 3(8):835–842, 1997.
- [BPS05] Richard Brath, Mike Peters, and Robin Senior. Visualization for communication: The importance of aesthetic sizzle. In *Proceedings of the Ninth International Conference on Information Visualisation, IV '05*, pages 724–729, Washington, DC, USA, 2005. IEEE Computer Society.
- [Bro96] John Brooke. SUS: a quick and dirty usability scale. In P. W. Jordan, B. Thomas, B. A. Weerdmeester, and I. L. McClelland, editors, *Usability Evaluation in Industry*, pages 189–194. Taylor and Francis, 1996.
- [BRSG07] Chris Bennett, Jody Ryall, Leo Spalteholz, and Amy Gooch. The aesthetics of graph visualization. In *Computational Aesthetics*, pages 57–64, 2007.
- [BSdV10] K. Balog, P. Serdyukov, and A.P. de Vries. Overview of the TREC 2010 Entity Track. In *TREC 2010 Working Notes*, 2010.

- [BW98] Ulrik Brandes and Dorothea Wagner. Using graph layout to visualize train interconnection data. In Sue Whitesides, editor, *Graph Drawing*, volume 1547 of *Lecture Notes in Computer Science*, pages 44–56. Springer Berlin / Heidelberg, 1998. 10.1007/3-540-37623-2_4.
- [Caso2] Donald O. Case. *Looking for Information: A Survey of Research on Information Seeking, Needs, and Behavior (Library and Information Science)*. Academic Press, 1st edition, May 2002.
- [CCC⁺09] Timothy Chee, Lee-Kwun Chan, Min-Hooi Chuah, Chee-Sok Tan, Siew-Fan Wong, and William Yeoh. Business intelligence systems: state-of-the-art review and contemporary applications. In *Symposium on Progress in Information & Communication Technology*, pages 96–101, 2009.
- [CDM⁺04] Tiziana Catarci, Paolo Dongilli, Tania Di Mascio, Enrico Franconi, Giuseppe Santucci, and Sergio Tessaris. An ontology based visual tool for query formulation support. In *ECAI*, pages 308–312, 2004.
- [CDToo] C.W. Choo, B. Detlor, and D. Turnball. *Web work: Information seeking and knowledge work on the World Wide Web*. 2000.
- [CGQ08] Gong Cheng, Weiyi Ge, and Yuzhong Qu. Falcons: searching and browsing entities on the semantic web. In Jinpeng Huai, Robin Chen, Hsiao-Wuen Hon, Yunhao Liu, Wei-Ying Ma, Andrew Tomkins, and Xiaodong Zhang, editors, *WWW*, pages 1101–1102. ACM, 2008.
- [Cho99] Chun Wei Choo. Closing the cognitive gaps: How people process information. *Financial Times, Mastering Information Management Series*, 1999.
- [CHSo4] Philipp Cimiano, Siegfried Handschuh, and Steffen Staab. Towards the self-annotating web. In *Proceedings of the 13th international conference on World Wide Web, WWW '04*, pages 462–471, New York, NY, USA, 2004. ACM.
- [CK98] John V. Carlis and Joseph A. Konstan. Interactive visualization of serial periodic data. In *Proceedings of the 11th annual ACM symposium on User interface software and technology, UIST '98*, pages 29–38, New York, NY, USA, 1998. ACM.
- [CKI88] Bill Curtis, Herb Krasner, and Neil Iscoe. A field study of the software design process for large systems. *Communications of the ACM*, 31:1268–1287, 1988.
- [CKKS02] William F. Cody, Jeffrey T. Kreulen, Vikas Krishna, and W. Scott Spangler. The integration of business intelligence and knowledge management. *IBM Systems Journal*, 41(4):697–713, 2002.
- [CL90] Wesley M. Cohen and Daniel A. Levinthal. Absorptive Capacity: A New Perspective on Learning and Innovation. *Administrative Science Quarterly*, 35(1), 1990.

- [CM88] William C. Cleveland and Marylyn E. McGill. *Dynamic Graphics for Statistics*. CRC Press, Inc., Boca Raton, FL, USA, 1st edition, 1988.
- [CMo6] Nick Cawthon and Andrew Vande Moere. A conceptual model for evaluating aesthetic effect within the user experience of information visualization. In *Proceedings of the conference on Information Visualization, IV '06*, pages 374–382, Washington, DC, USA, 2006. IEEE Computer Society.
- [CMo7] Nick Cawthon and Andrew Vande Moere. The effect of aesthetic on the usability of data visualization. In *Proceedings of the 11th International Conference Information Visualization, IV '07*, pages 637–648, Washington, DC, USA, 2007. IEEE Computer Society.
- [CMC] Amparo E Cano, Suvodeep Mazumdar, and Fabio Ciravegna. Social influence analysis in microblogging platforms—a topic-sensitive based approach. *Semantic Web*.
- [Cono4] World Wide Web Consortium. Resource description framework (rdf) : concepts and abstract syntax, Feb 2004.
- [CS96] Paul Chandler and John Sweller. Cognitive load while learning to use a computer program. *Applied Cognitive Psychology*, 10(2):151–170, 1996.
- [CSGo7] Liming Chen, Nigel R. Shadbolt, and Carole A. Goble. A semantic web-based approach to knowledge management for grid applications. *IEEE Transactions on Knowledge and Data Engineering*, 19(2):283–296, 2007.
- [CSP10] Constantinos K. Coursaris, Sarah J. Swierenga, and Graham L. Pierce. Effects of aesthetics and playfulness on web usability - an empirical investigation. In Martin Santana, Jerry N. Luftman, and Ajay S. Vinze, editors, *AMCIS*, page 549. Association for Information Systems, 2010.
- [CWo4] D. Chaffey and S. Wood. *Business Information Management: Improving Performance Using Information Systems*. Financial Times Prentice Hall, 2004.
- [DAC11] Danica Damljanovic, Milan Agatonovic, and Hamish Cunningham. Freya: An interactive way of querying linked data using natural language. In Raul Garcia-Castro, Dieter Fensel, and Grigoris Antoniou, editors, *ESWC Workshops*, volume 7117 of *Lecture Notes in Computer Science*, pages 125–138. Springer, 2011.
- [DASHo6] Antonella De Angeli, Alistair Sutcliffe, and Jan Hartmann. Interaction, usability and aesthetics: what influences users' preferences? In *Proceedings of the 6th conference on Designing Interactive systems, DIS '06*, pages 271–280, New York, NY, USA, 2006. ACM.
- [Dau11] Frithjof Dau. *Semantic Technologies for Enterprises*, April 2011.
- [dav97] *Working Knowledge: How Organizations Manage What They Know*. Project Management Institute, December 1997.

- [DBW89] Fred D. Davis, Richard P. Bagozzi, and Paul R. Warshaw. User acceptance of computer technology: a comparison of two theoretical models. *Manage. Sci.*, 1989.
- [DDSC07] Mira Dontcheva, Steven M. Drucker, David Salesin, and Michael F. Cohen. Relations, cards, and search templates: user-guided web data integration and layout. In *Proceedings of the 20th annual ACM symposium on User interface software and technology, UIST '07*, pages 61–70, New York, NY, USA, 2007. ACM.
- [DEG⁺03] Stephen Dill, Nadav Eiron, David Gibson, Daniel Gruhl, R. Guha, Anant Jhingran, Tapas Kanungo, Sridhar Rajagopalan, Andrew Tomkins, John A. Tomlin, and Jason Y. Zien. Semtag and seeker: bootstrapping the semantic web via automated semantic annotation. In *Proceedings of the 12th international conference on World Wide Web, WWW '03*, pages 178–186, New York, NY, USA, 2003. ACM.
- [DFH03] John Davies, Dieter Fensel, and Frank van Harmelen, editors. *Towards the Semantic Web: Ontology-driven Knowledge Management*. John Wiley & Sons, Inc., New York, NY, USA, 2003.
- [DFKO02] Ying Ding, Dieter Fensel, Michel Klein, and Borys Omelayenko. The semantic web: Yet another hip? *Data Knowledge Engineering*, 41:205–227, 2002.
- [DLS07] John Davies, Miltiadis Lytras, and Amit P. Sheth. Guest editors' introduction: Semantic-web-based knowledge management. *IEEE Internet Computing*, 11:14–16, 2007.
- [DP97] Thomas H. Davenport and Laurence Prusak. *Information Ecology: Mastering the Information and Knowledge Environment*. Oxford University Press, 1st edition, 1997.
- [DR11] Aba-Sah Dadzie and Matthew Rowe. Approaches to visualising linked data: A survey. *Semantic Web*, 2(2):89–124, 2011.
- [DW04] John Davies and Richard Weeks. Quizrdf: Search technology for the semantic web. In *Proceedings of the Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS'04) - Track 4 - Volume 4, HICSS '04*, pages 40112–, Washington, DC, USA, 2004. IEEE Computer Society.
- [EB11] Micheline Elias and Anastasia Bezerianos. Exploration views: understanding dashboard creation and customization for visualization novices. In *Proceedings of the 13th IFIP TC 13 international conference on Human-computer interaction - Volume Part IV, INTERACT'11*, pages 274–291, Berlin, Heidelberg, 2011. Springer-Verlag.
- [Eico3] Holger Eichelberger. Nice class diagrams admit good design? In *Proceedings of the 2003 ACM symposium on Software visualization, SoftVis '03*, pages 159–ff, New York, NY, USA, 2003. ACM.

- [EMWC14] K. Elbedweihy, S. Mazumdar, S. N. Wrigley, and F. Ciravegna. Nl-graphs: A hybrid approach toward interactively querying semantic data. In *Proceedings of Extended Semantic Web Conference (ESWC)*, 2014.
- [ES93] K Anders Ericsson and Herbert A Simon. *Protocol Analysis - Rev'd Edition: Verbal Reports as Data*. Number 0262550237. The MIT Press, 1993.
- [EWC] K Elbedweihy, S.N Wrigley, and F Ciravegna. Bridging the gap between natural language queries and ontology concepts using word sense disambiguation. In *NLP DBpedia Workshop, 12th International Semantic Web Conference (ISWC2013), 21-25 October, Sydney, Australia*.
- [EWC⁺12a] K Elbedweihy, S N Wrigley, F Ciravegna, Dorothee Reinhard, and Abraham Bernstein. Evaluating semantic search systems to identify future directions of research. In *Second International Workshop on Evaluation of Semantic Technologies*, number 843 in CEUR Workshop Proceedings, pages 25–36, May 2012.
- [EWC12b] Khadija Elbedweihy, Stuart N. Wrigley, and Fabio Ciravegna. Evaluating semantic search query approaches with expert and casual users. In *Proceedings of the 11th international conference on The Semantic Web - Volume Part II, ISWC'12*, pages 274–286, Berlin, Heidelberg, 2012. Springer-Verlag.
- [Faio4] Andrea Weltzl Fairchild. Describing Aesthetic Experience: Creating a Model. *Canadian Journal of Education*, 16(3):267–280, 2004.
- [FDPL05] P. Fishwick, S. Diehl, J. Prophet, and J. Löwgren. Perspectives on aesthetic computing. *Leonardo*, 38(2):133–141, 2005.
- [Felo4] Susan Feldman. The high cost of not finding information. *KM World*, 13(3), 2004.
- [Few06] Stephen Few. *Information Dashboard Design: The Effective Visual Communication of Data*. Number 3900693099. O'Reilly Media, 2006.
- [FT04] Benjamin Finkel and Roberto Tamassia. Curvilinear graph drawing using the force-directed method. In *Proceedings of the 12th international conference on Graph Drawing, GD'04*, pages 448–453, Berlin, Heidelberg, 2004. Springer-Verlag.
- [FVHD⁺02] Dieter Fensel, Frank Van Harmelen, Ying Ding, Michel Klein, Hans Akkermans, Jeen Broekstra, Arjohn Kampman, Jos van der Meer, Y Sure, R Studer, et al. On-to-knowledge: Semantic web enabled knowledge management. *submitted to IEEE Computer*, 2002.
- [FW02] C. Fluit and J. Wester. Using visualization for information management tasks. In *Information Visualisation, 2002. Proceedings. Sixth International Conference on*, pages 447 – 454, 2002.

- [GAC⁺04] Hugh Glaser, Harith Alani, Sam Chapman, Alexiei Dingli, Nicholas Gibbins, Stephen Harris, m.c. schraefel, and Nigel Shadbolt. Cs active space: Building a semantic web application. In *Proceedings of First European Semantic Web Symposium (ESWS 2004)*, pages 417–432. Springer Verlag, 2004.
- [GFA09] Tovi Grossman, George Fitzmaurice, and Ramtin Attar. A survey of software learnability: metrics, methodologies and guidelines. In *Proceedings of the 27th international conference on Human factors in computing systems*, 2009.
- [GG86] T. Gilad and B. Gilad. Smr forum: Business intelligence - the quiet revolution. *Sloan Management Review*, 27(4):53–61, 1986.
- [GGB⁺05] Jan Gulliksen, Bengt GÅ¶ransson, Inger Boivie, Jenny Persson, Stefan Blomkvist, and Åsa Cajander. Key principles for user-centred systems design. In Ahmed Seffah, Jan Gulliksen, and MichelC. Desmarais, editors, *Human-Centered Software Engineering â Integrating Usability in the Software Development Lifecycle*, volume 8 of *Human-Computer Interaction Series*, pages 17–36. Springer Netherlands, 2005.
- [GGP⁺10] Roberto Garca, JuanManuel Gimeno, Ferran Perdrix, Rosa Gil, Marta Oliva, JuanMiguel Lopez, Afra Pascual, and Montserrat Sendın. Building a usable and accessible semantic web interaction platform. *World Wide Web*, 13(1-2):143–167, 2010.
- [GK86] Sumantra Ghoshal and Seok Ki Kim. Building effective intelligence systems for competitive advantage. *Sloan Management Review*, 28(1):49–58, Fall 1986.
- [GL85] John D. Gould and Clayton Lewis. Designing for usability: key principles and what designers think. *Commun. ACM*, 28(3):300–311, March 1985.
- [GLMC11] Anna Lisa Gentile, Vitaveska Lanfranchi, Suvodeep Mazumdar, and Fabio Ciravegna. Extracting semantic user networks from informal communication exchanges. In *The Semantic Web–ISWC 2011*, pages 209–224. Springer, 2011.
- [GMS01] Andrew H. Gold, Arvind Malhotra, and Albert H. Segars. Knowledge management: An organizational capabilities perspective. *J. Manage. Inf. Syst.*, 18(1):185–214, May 2001.
- [GR94] Jade Goldstein and Steven F. Roth. Using aggregation and dynamic queries for exploring large data sets. In *Proceedings of the SIGCHI conference on Human factors in computing systems: celebrating interdependence*, CHI '94, pages 23–29. ACM, 1994.
- [GRF09] Tera Marie Green, William Ribarsky, and Brian D. Fisher. Building and applying a human cognition model for visual analytics. *Information Visualization*, 8(1):1–13, 2009.

- [Gri74] J. P. Grime. Vegetation classification by reference to strategies. *Nature*, 250:26–31, 1974.
- [GSK⁺10] Michael Granitzer, Vedran Sabol, Wolfgang Kienreich, Dickson Lukose, and Kow Weng Onn. Visual Analyses on Linked Data â An Opportunity for both Fields. Technical report, Riga, Latvia, 2010.
- [HB11] Tom Heath and Christian Bizer. *Linked Data: Evolving the Web into a Global Data Space*. Synthesis Lectures on the Semantic Web. Morgan & Claypool Publishers, 2011.
- [HBE92] Christopher G. Healey, Kellogg S. Booth, and James T. Enns. Harnessing preattentive processes for multivariate data visualization. Technical report, Vancouver, BC, Canada, Canada, 1992.
- [HBE95] Christopher G. Healey, Kellogg S. Booth, and James T. Enns. Visualizing real-time multivariate data using preattentive processing. *ACM Trans. Model. Comput. Simul.*, 5(3):190–221, July 1995.
- [HBE96] Christopher G. Healey, Kellogg S. Booth, and James T. Enns. High-speed visual estimation using preattentive processing. *ACM Trans. Comput.-Hum. Interact.*, 3(2):107–135, June 1996.
- [HBO10] Jeffrey Heer, Michael Bostock, and Vadim Ogievetsky. A tour through the visualization zoo. *Communications of the ACM*, 53(10.1145/1743546.1743567):59–67, 2010.
- [HEE⁺02] Marti Hearst, Ame Elliott, Jennifer English, Rashmi Sinha, Kirsten Swearingen, and Ka-Ping Yee. Finding the flow in web site search. *Commun. ACM*, 2002.
- [Hek06] P. Hekkert. Design aesthetics: principles of pleasure in design. *Psychology Science*, 48(2):157–172, 2006.
- [Hen07] Shawn Lawton Henry. *Just ask: integrating accessibility throughout design*. Lulu. com, 2007.
- [HHM⁺10] Harry Halpin, Daniel M Herzig, Peter Mika, Roi Blanco, Jeffrey Pound, Henry S Thompson, and Thanh Tran Duc. Evaluating Ad-Hoc Object Retrieval. In *Proceedings of the 1st International Workshop on Evaluation of Semantic Technologies (IWEST)*, 2010.
- [HHU⁺11] Aidan Hogan, Andreas Harth, JÃErgen Umbrich, Sheila Kinsella, Axel Polleres, and Stefan Decker. Searching and browsing linked data with swse: The semantic web search engine. *J. Web Sem.*, 9(4):365–401, 2011.
- [HK09] David Huynh and David Karger. Parallax and Companion: Set-based Browsing for the Data Web. 2009.
- [HLTE11] Philipp Heim, Steffen Lohmann, Davaadorj Tsendragchaa, and Thomas Ertl. Semlens: Visual analysis of semantic data with scatter plots and semantic lenses. In *Proceedings of the 7th International Conference on Semantic*

- Systems (I-SEMANTICS 2011)*, pages 175–178, New York, NY, USA, 2011. ACM.
- [HMS⁺05] Eero Hyvönen, Eetu Mäkelä, Mirva Salminen, Arttu Valo, Kim Viljanen, Samppa Saarela, Miikka Junnila, and Suvi Kettula. Museumfinland-finnish museums on the semantic web. *Web Semant.*, 3:224–241, October 2005.
- [Hor11] Elisabeth Horwitt. *The semantic web gets down to business*, 2011.
- [HRRSo7] Richard Harper, Tom Rodden, Yvonne Rogers, and Abigail Sellen. *Being Human: Human-Computer Interaction in the Year 2020*. Microsoft Research Ltd, 2007.
- [Hua] Ming-Hui Huang. Designing website attributes to induce experiential encounters. *Computers in Human Behavior*, 19(4):425–442.
- [HvOHo6] Michiel Hildebrand, Jacco van Ossenbruggen, and Lynda Hardman. /facet: A Browser for Heterogeneous Semantic Web Repositories. In *ISWC*, 2006.
- [Ili13] Noah Iliinsky. Choosing visual properties for successful visualizations. Technical report, IBM, 2013.
- [Inso9] Alfred Inselberg. *Parallel Coordinates: Visual Multidimensional Geometry and Its Applications*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2009.
- [Jac97] Ray Jackendoff. *The architecture of the language faculty*. Number 28. MIT Press, 1997.
- [Jas11] Ashok Jashapara. *Knowledge Management: An Integrated Approach*. FT Press, 2nd revised edition edition, 2011.
- [Jen05] Judy Jeng. Usability assessment of academic digital libraries: Effectiveness, efficiency, satisfaction, and learnability. In *LIBRI: INTERNATIONAL JOURNAL OF LIBRARIES AND INFORMATION SERVICES*, 2005.
- [jor98] Human factors for pleasure in product use. *Applied Ergonomics*, 29(1):25–33, February 1998.
- [Joro2] Patrick W. Jordan. *Designing Pleasurable Products*. CRC, 1 edition, August 2002.
- [JV07] Leonard Jessup and Joseph Valacich. *Information Systems Today: Managing in the Digital World*. Prentice Hall Press, Upper Saddle River, NJ, USA, 3rd edition, 2007.
- [KB10] Esther Kaufmann and Abraham Bernstein. Evaluating the usability of natural language query languages and interfaces to semantic web knowledge bases. *Journal of Web Semantics*, 8(4):377–393, 2010.

- [KBF07] Esther Kaufmann, Abraham Bernstein, and Lorenz Fischer. NLP-Reduce: A "naive" but Domain-independent Natural Language Interface for Querying Ontologies. 2007.
- [KBH06] Robert Kosara, Fabian Bendix, and Helwig Hauser. Parallel sets: Interactive exploration and visual analysis of categorical data. *Transactions on Visualization and Computer Graphics*, 12:558–568, 2006.
- [KG10] Minu Kumar and Nitika Garg. Aesthetic principles and cognitive emotion appraisals: How much of the beauty lies in the eye of the beholder? *Journal of Consumer Psychology*, 20(4):485–494, October 2010.
- [Kim06] Nanyoung Kim. A history of design theory in art education. *Journal of Aesthetic Education*, 40(2):12–28, 2006.
- [KMS⁺08] Daniel A. Keim, Florian Mansmann, Jörn Schneidewind, Jim Thomas, and Hartmut Ziegler. Visual data mining. chapter Visual Analytics: Scope and Challenges, pages 76–90. Springer-Verlag, Berlin, Heidelberg, 2008.
- [KMSZ06] Daniel A Keim, Florian Mansmann, Jörn Schneidewind, and Hartmut Ziegler. Challenges in visual data analysis. In *International Conference on Information Visualisation*. IEEE Computer Society, 2006.
- [KNS⁺06] Daniel A. Keim, Tilo Nietzsche, Norman Schelwies, Jörn Schneidewind, Tobias Schreck, and Hartmut Ziegler. A spectral visualization system for analyzing financial time series data. In *EuroVis*, pages 195–202, 2006.
- [Kot06] Munir Kotadia. Business intelligence lies beyond it: Dresner, 2006.
- [Kri83] James Krikelas. Information-Seeking Behavior: Patterns and Concepts. *Drexel Library Quarterly, (Foundations of Library Practice)*, 19(2):5–20, 1983.
- [KSB⁺09] Martin Krzywinski, Jacqueline Schein, İnanç Birol, Joseph Connors, Randy Gascoyne, Doug Horsman, Steven J Jones, and Marco A Marra. Circos: An information aesthetic for comparative genomics. *Genome research*, 19(1549-5469):1639–45, 2009.
- [Lan09] A. Lang. Aesthetics in information visualization. *Trends in Information Visualization*, page 8, 2009.
- [LBOA04] Helmut Leder, Benno Belke, Andries Oeberst, and Dorothee Augustin. A model of aesthetic appreciation and aesthetic judgments. *British Journal of Psychology*, 95(4):489+, 2004.
- [LCG⁺11] Vitaveska Lanfranchi, Rodrigo Fontenele Carvalho, Anna Lisa Gentile, Suvodeep Mazumdar, and Fabio Ciravegna. A semantic knowledge management framework for informal communication exchanges. In *Demo Track, 10th International Semantic Web Conference (ISWC 2011)*, 2011.

- [LDS⁺05] Holger Lausen, Ying Ding, Michael Stollberg, Dieter Fensel, Ruben Lara Hernandez, and Sung-Kook Han. Semantic web portals: state-of-the-art survey. *Journal of Knowledge Management*, 9(5):40–49, 2005.
- [LMo7] Andrea Lau and Vande Andrew Moere. Towards a model of information aesthetics in information visualization. In *International Conference on Information Visualisation*, 1550–6037, pages 87–92. IEEE Computer Society, 2007.
- [LMUo6] Vanessa Lopez, Enrico Motta, and Victoria S. Uren. Poweraqua: Fishing the semantic web. In *ESWC*, pages 393–410, 2006.
- [LTo4] Talia Lavie and Noam Tractinsky. Assessing dimensions of perceived visual aesthetics of web sites. *International journal of human-computer studies*, 60(3):269–298, 2004.
- [LUMo6] Yuanguai Lei, Victoria Uren, and Enrico Motta. Semsearch: A search engine for the semantic web. In *Proc. 5th International Conference on Knowledge Engineering and Knowledge Management Managing Knowledge in a World of Networks, Lect. Notes in Comp. Sci., Springer, Podebrady, Czech Republic*, pages 238–245. Springer-Verlag, 2006.
- [LWT05] Cathy S Lin, Sheng Wu, and Ray J Tsai. Integrating perceived playfulness into expectation-confirmation model for web portal context. *Information & Management*, 42(5):683–693, 2005.
- [Mar97] Gary Marchionini. *Information Seeking in Electronic Environments (Cambridge Series on Human-Computer Interaction)*. Cambridge University Press, March 1997.
- [Maro6] Gary Marchionini. Exploratory search: from finding to understanding. *Commun. ACM*, 49(4):41–46, April 2006.
- [MCGL12] Suvoodeep Mazumdar, Fabio Ciravegna, Anna Lisa Gentile, and Vitaveska Lanfranchi. Visualising context and hierarchy in social media. *International Workshop on Intelligent Exploration of Semantic Data (IESDâ2012) at EKAW, 2012*, 2012.
- [MG05] John David McClurg-Gevenese. The principles of design. *Design in Theory and Practice*, 2005.
- [MHo4] Deborah L. McGuinness and Frank Harmelen. Owl web ontology language overview, May 2004.
- [MI96] R. Mizoguchi and M Ikeda. Towards Ontology Engineering. Technical Report AI-TR-96-1, The Institute of Scientific and Industrial Research, Osaka University, 1996.
- [MKo1] Ji-Won Moon and Young-Gul Kim. Extending the tam for a world-wide-web context. *Inf. Manage.*, 38(4):217–230, February 2001.

- [MMME₁₁] A. Malik, R. Maciejewski, B. Maule, and DS Ebert. A visual analytics process for maritime resource allocation and risk assessment. In *Visual Analytics Science and Technology (VAST), 2011 IEEE Conference on*, pages 221–230. IEEE, 2011.
- [MO] Enrico Motta and Francesco Osborne. Making sense of research with explore. In Birte Glimm and David Huynh, editors, *International Semantic Web Conference (Posters Demos)*, CEUR Workshop Proceedings. CEUR-WS.org.
- [MPCar] Suvodeep Mazumdar, Daniela Petrelli, and Fabio Ciravegna. Exploring user and system requirements of linked data visualization through a visual dashboard approach. *Semantic web : interoperability, usability, applicability*, November 2013 (to appear).
- [MPE⁺₁₄] Suvodeep Mazumdar, Daniela Petrelli, Khadija Elbedweihi, Vitaveska Lanfranchi, and Fabio Ciravegna. Affective graphs: The visual appeal of linked data. *Semantic Web – Interoperability, Usability, Applicability*, 2014.
- [MPWG₁₂] Kim Marriott, Helen Purchase, Michael Wybrow, and Cagatay Goncu. Memorability of visual features in network diagrams. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2477–2485, 2012.
- [MVL⁺₁₂] Suvodeep Mazumdar, Andrea Varga, Vita Lanfranchi, Daniela Petrelli, and Fabio Ciravegna. A knowledge dashboard for manufacturing industries. In Raúl Garc a-Castro, Dieter Fensel, and Grigoris Antoniou, editors, *The Semantic Web: ESWC 2011 Workshops*, volume 7117 of *Lecture Notes in Computer Science*, pages 112–124. Springer Berlin Heidelberg, 2012.
- [MW₀₄] John McCarthy and Peter Wright. Technology as experience. *interactions*, 11(5):42–43, September 2004.
- [NAEW₀₆] Miguel Baptista Nunes, Fenio Annansingh, Barry Eaglestone, and Richard Wakefield. Knowledge management issues in knowledge-intensive SMEs. *Journal of Documentation*, 62(1):101–119+, 2006.
- [ND₈₆] Donald A. Norman and Stephen W. Draper. *User Centered System Design; New Perspectives on Human-Computer Interaction*. L. Erlbaum Associates Inc., Hillsdale, NJ, USA, 1986.
- [Nie₉₃] Jakob Nielsen. *Usability Engineering*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.
- [NM₉₀] Jakob Nielsen and Rolf Molich. Heuristic evaluation of user interfaces. In *CHI '90: Proceedings of the SIGCHI conference on Human factors in computing systems*, 0-201-50932-6, pages 249–256. ACM, 1990.
- [Nm₀₁] Natalya F. Noy and Deborah L. mcguinness. *Ontology Development 101: A Guide to Creating Your First Ontology*. Online, 2001.

- [Nono8] Ikujiro Nonaka. *The Knowledge-Creating Company (Harvard Business Review Classics) (Harvard Business Review Classics)*. Harvard Business School Press, December 2008.
- [Noro4] Donald A. Norman. *Emotional Design: Why We Love (Or Hate) Everyday Things*. Basic Books, January 2004.
- [NSA00] D.C.L. Ngo, A. Samsudin, and R. Abdullah. Aesthetic measures for assessing graphic screens. *J. INF. SCI. ENG*, 16(1):97–116, 2000.
- [NTBo3] D.C.L. Ngo, L.S. Teo, and J.G. Byrne. Modelling interface aesthetics. *Information Sciences*, 152:25–46, 2003.
- [Obe13] Daniel Oberle. How ontologies benefit enterprise applications. *Semantic Web: Interoperability Usability Applicability*, 2013.
- [ora] *Oracle Business Intelligence Answers, Delivers, and Interactive Dashboards User Guide*.
- [OSGms08] Alisdair Owens, Andy Seaborne, Nick Gibbins, and mc schraefel. Clustered TDB: A clustered triple store for jena. Nonpeerreviewed, Electronics and Computer Science, University of Southampton, November 2008.
- [PAG09] Jorge Pérez, Marcelo Arenas, and Claudio Gutierrez. Semantics and complexity of sparql. *ACM Trans. Database Syst.*, 34(3):16:1–16:45, September 2009.
- [Par00] Soyeon Park. Usability, user preferences, effectiveness, and user behaviors when searching individual and integrated full-text databases: implications for digital libraries. *J. Am. Soc. Inf. Sci.*, 2000.
- [PCK05] Su-e Park, Dongsung Choi, and Jinwoo Kim. Visualizing e-brand personality: Exploratory studies on visual attributes and e-brand personalities in korea. *International Journal of Human-Computer Interaction*, 19(1):7–34, 2005.
- [PHJR11] Helen C Purchase, John Hamer, Adrian Jamieson, and Oran Ryan. Investigating objective measures of web page aesthetics and usability. In *Australasian User Interface Conference*, pages 19–28, 2011.
- [Pin99] Helena Sofia Pinto. Towards ontology reuse. In *Proceedings of AAAI99's Workshop on Ontologies and Problem Solving Methods: Lessons Learned and Future Trends*, pages 7.1–7.12. AAAI, 1999.
- [PKM⁺03] Borislav Popov, Atanas Kiryakov, Dimitar Manov, Angel Kirilov, and Ognyanoff M. Goranov. Towards Semantic Web Information Extraction. In *In proceedings of ISWC (Sundial Resort, volume 20, 2003)*.
- [PLK05] Virpi Pirttimäki, Antti Länqvist, and Antti Karjaluoto. Measurement of business intelligence in a finnish telecommunications company. In Dan Remenyi, editor, *ECKM*, pages 451–458. Academic Conferences Limited, Reading, UK, 2005.

- [PMDC09] Daniela Petrelli, Suvodeep Mazumdar, Aba-Sah Dadzie, and Fabio Ciravegna. Multi visualization and dynamic query for effective exploration of semantic data. In *In proceedings of International Semantic Web Conference*, volume 5823, pages 505–520. Springer Berlin / Heidelberg, 2009.
- [Pow03] D. J. Power. *A Brief History of Decision Support Systems*, 2003.
- [PSP⁺13] DennisB. Propst, SarahJ. Swierenga, GrahamL. Pierce, Eunseong Jeong, and ConstantinosK. Coursaris. From the ground-up: Role of usability and aesthetics evaluation in creating a knowledge-based website for the u.s. army corps of engineers. In Aaron Marcus, editor, *Design, User Experience, and Usability. Web, Mobile, and Product Design*, volume 8015 of *Lecture Notes in Computer Science*, pages 274–283. Springer Berlin Heidelberg, 2013.
- [QD99] Edward J. Quigley and Anthony Debons. Interrogative theory of information and knowledge. In *Proceedings of the 1999 ACM SIGCPR conference on Computer personnel research, SIGCPR '99*, pages 4–10, New York, NY, USA, 1999. ACM.
- [QLo8] Bastian Quilitz and Ulf Leser. Querying distributed rdf data sources with sparql. In *Proceedings of the 5th European Semantic Web Conference on The Semantic Web: Research and Applications, ESWC'08*, pages 524–538, Berlin, Heidelberg, 2008. Springer-Verlag.
- [RATS10] Mirizzi Roberto, Ragone Azzurra, Di Noia Tommaso, and Eugenio Di Sciascio. Semantic wonder cloud: exploratory search in dbpedia. In *In proceedings of the 2nd International Workshop on Semantic Web Information Management (SWIM 2010)*, 2010.
- [RC04] Mariano Rico and Pablo Castells. Jacinta: a mediator agent for human-agent interaction in semantic web services. In *Proc. The Semantic Web-International Semantic Web Conference (ISWC)*, 2004.
- [RLNo7] R. Rosenholtz, Y. Li, and L. Nakano. Measuring visual clutter. *Journal of Vision*, 7(2), 2007.
- [RM83] Teresa L. Roberts and Thomas P. Moran. The evaluation of text editors: methodology and empirical results. *Commun. ACM*, 1983.
- [RM10] Matthew Rowe and Suvodeep Mazumdar. Integrating and interpreting social data from heterogeneous sources. In *Proceedings of Linked User Profiles and Applications on the Social Semantic Web, Extended Semantic Web Conference 2010*, 2010.
- [Rob10] Mark A. Robinson. An empirical analysis of engineers' information behaviors. *J. Am. Soc. Inf. Sci. Technol.*, 61(4):640–658, April 2010.
- [Row07] Jennifer Rowley. The wisdom hierarchy: representations of the dikw hierarchy. *J. Inf. Sci.*, 33(2):163–180, April 2007.

- [RRH00] Stephanie Rosenbaum, Janice Anne Rohn, and Judee Humburg. A toolkit for strategic usability: results from workshops, panels, and surveys. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, CHI '00, pages 337–344, New York, NY, USA, 2000. ACM.
- [RSP02] Y. Rogers, H. Sharp, and J. Preece. *Interaction Design: Beyond Human-Computer Interaction*. John Wiley and Sons Ltd, 2002.
- [RSSH09] James Richardson, Kurt Schlegel, R. L Sallam, and Bill Hostmann. Magic quadrant for business intelligence platforms. Technical report, Gartner, 2009.
- [Rub94] Jeffrey Rubin. *Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests*. John Wiley & Sons, Inc., New York, NY, USA, 1994.
- [Rud09] Olivia Parr Rud. *Business intelligence success factors: tools for aligning your business in the global economy*, volume 18. John Wiley & Sons, 2009.
- [SA00] Guus T. Schreiber and Hans Akkermans. *Knowledge engineering and management: the CommonKADS methodology*. MIT Press, Cambridge, MA, USA, 2000.
- [SAN07] Anna Stavrianou, Periklis Andritsos, and Nicolas Nicoloyannis. Overview and semantic issues of text mining. *SIGMOD Rec.*, 36(3):23–34, September 2007.
- [SB92] Manojit Sarkar and Marc H. Brown. Graphical fisheye views of graphs. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '92, pages 83–91, New York, NY, USA, 1992. ACM.
- [SBB⁺02] Ivan A. Sag, Timothy Baldwin, Francis Bond, Ann A. Copestake, and Dan Flickinger. Multiword expressions: A pain in the neck for nlp. In *Proceedings of the Third International Conference on Computational Linguistics and Intelligent Text Processing*, CICLing '02, pages 1–15, London, UK, UK, 2002. Springer-Verlag.
- [SBC97] Ben Shneiderman, Don Byrd, and W. B Croft. Clarifying search: A user-interface framework for text searches. Technical report, 1997.
- [SC06] Amanda Spink and Charles Cole. Human information behavior: Integrating diverse approaches and information use. *J. Am. Soc. Inf. Sci. Technol.*, 57(1):25–35, January 2006.
- [SH07] H.N.J. Schifferstein and P. Hekkert. *Product Experience*, chapter Product Aesthetics. Elsevier Science, 2007.
- [Sha86] Brian Shackel. Ergonomics in design for usability. In *People & computers: Designing for usability. Proceedings of the second conference of the BCS HCI specialist group*. Cambridge University Press., 1986.

- [Shn96] Ben Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *VL '96: Proceedings of the 1996 IEEE Symposium on Visual Languages*. IEEE Computer Society, 1996.
- [Shn98] Ben Shneiderman. *Designing the User Interface - Strategies for Effective Human-Computer Interaction*. Addison-Wesley Longman, Reading, MA, 3rd edition, 1998.
- [Sho13] Arisa Shollo. *The Role of Business Intelligence in Organizational Decision-making*. Copenhagen Business School Copenhagen Business School, Institut for IT-Ledelse Department of IT Management, 2013.
- [SLHo3] T Skog, S Ljungblad, and L.E. Holmquist. Between aesthetics and utility: designing ambient information visualizations. *IEEE Symposium on Information Visualization (IEEE Cat. No.03TH8714)*, pages 233–240, 2003.
- [SMo7] M. Stefaner and B. Muller. Elastic lists for facet browsers. In *Database and Expert Systems Applications, 2007. DEXA'07. 18th International Workshop on*, pages 217–221. IEEE, 2007.
- [SP10] Igor Svetel and Milica Pejanovic. The role of the semantic web for knowledge management in the construction industry. *Informatica (Slovenia)*, 34(3):331–336, 2010.
- [Spl08] Andrea Splendiani. Rdfscape: Semantic web meets systems biology. *BMC Bioinformatics*, 9(Suppl 4):S6, 2008.
- [SRN11] Magnus Stuhr, Dumitru Roman, and David Norheim. Lodwheel - javascript-based visualization of rdf data. In Olaf Hartig, Andreas Harth, and Juan Sequeda, editors, *COLD*, volume 782 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2011.
- [SWM⁺04] Norbert Schwarz, Piotr Winkielman, Bruce Morton, John Skowronski, and Bruce Whittlesea. Processing fluency and aesthetic pleasure: Is beauty in the perceiver's processing experience. *Social Psychology Review*, 8:364–382, 2004.
- [sWRS06] m.c. schraefel, Max Wilson, Alistair Russell, and Daniel A. Smith. mspace: improving information access to multimedia domains with multimodal exploratory search. *Commun. ACM*, 49:47–49, April 2006.
- [TAo8] T. Tullis and W. Albert. *Measuring the User Experience: Collecting, Analyzing, and Presenting Usability Metrics*. Elsevier/Morgan Kaufmann, 2008.
- [TC05] James J Thomas and Kristin A Cook. *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. National Visualization and Analytics Ctr, 2005.
- [TC06] James J Thomas and Kristin A Cook. A visual analytics agenda. *IEEE Computer Graphics and Applications (2006)*, 26:10–13, 2006.

- [TCC⁺10] Giovanni Tummarello, Richard Cyganiak, Michele Catasta, Szymon Danielczyk, Renaud Delbru, and Stefan Decker. Sig.ma: Live views on the web of data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 8(4):355 – 364, 2010.
- [TDOo8] Giovanni Tummarello, Renaud Delbru, and Eyal Oren. Sindice.com: Weaving the open linked data. pages 552–565. 2008.
- [Teeo0] David J Teece. Strategies for managing knowledge assets: the role of firm structure and industrial context. *Long range planning*, 33(1):35–54, 2000.
- [TKIoo] N. Tractinsky, A. S. Katz, and D. Ikar. What is beautiful is usable. *Interacting with Computers*, 13(2):127–145, December 2000.
- [Trao4] Noam Tractinsky. Toward the study of aesthetics in information technology. In *in Proceedings of the 25th International Conference on Information Systems*, pages 771–780, 2004.
- [Tuf86] Edward R. Tufte. *The visual display of quantitative information*. Graphics Press, Cheshire, CT, USA, 1986.
- [Tuf90] Edward Tufte. *Envisioning information*. Graphics Press, Cheshire, CT, USA, 1990.
- [Tuk77] John W Tukey. *Exploratory Data Analysis*. Pearson, 1977.
- [TVo1] Haridimos Tsoukas and Efi Vladimirou. What is organizational knowledge? *Journal of Management Studies*, 38(7):973–993, 2001.
- [TvdB10] Orhan Tuncer and Jan van den Berg. Implementing bi concepts with pentaho, an evaluation. *Delft University of Technology. Netherlands*, 2010.
- [vdS96] R. Spek van der and A. Spijkervet. *Knowledge Management - Dealing Intellegently with Knowledge*. The Hague, CIBIT., 1996.
- [VMSCo2] Karel Vredenburg, Ji-Ye Mao, Paul W. Smith, and Tom Carey. A survey of user-centered design practice. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '02, pages 471–478, New York, NY, USA, 2002. ACM.
- [Waro4] Colin Ware. *Information Visualization: Perception for Design*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004.
- [WAso8] Max L. Wilson, Paul André, and mc schraefel. Backward highlighting: Enhancing faceted search. In *Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology*, UIST '08, pages 235–238, New York, NY, USA, 2008. ACM.
- [WDM⁺07] Ryen W. White, Steven M. Drucker, Gary Marchionini, Marti Hearst, and m. c. schraefel. Exploratory search and hci: designing and evaluating interfaces to support exploratory search interaction. In *CHI '07 extended abstracts on Human factors in computing systems*, CHI EA '07, pages 2877–2880, New York, NY, USA, 2007. ACM.

- [WHP⁺04] Pak Chung Wong, Beth Hetzler, Christian Posse, Mark Whiting, Susan Havre, Nick Cramer, Anuj Shah, Mudita Singhal, Alan Turner, and Jim Thomas. In-spire infovis 2004 contest entry. In *INFOVIS '04: Proceedings of the IEEE Symposium on Information Visualization*, page 216.2. IEEE Computer Society, 2004.
- [Wii94] Karl M. Wiig. *Knowledge Management Foundations: Thinking about Thinking - how People and Organizations Represent, Create, and Use Knowledge*. Schema Press, Limited, 1994.
- [Wil81] T. D. Wilson. On user studies and information needs. *Journal of Documentation*, 37(1):3–15, 1981.
- [Wil99] T. D. Wilson. Models in information behaviour research. *Journal of Documentation*, 55(3):249–270, 1999.
- [Wil00] Thomas D. Wilson. Human information behavior. *Informing Science*, 3(2):49–56, 2000.
- [WJF⁺06] Pak Chung Wong, George Chin Jr., Harlan Foote, Patrick Mackey, and Jim Thomas. Have green - a visual analytics framework for large semantic graphs. In *IEEE VAST*, pages 67–74, 2006.
- [WJLW85] John Whiteside, Sandra Jones, Paula S. Levy, and Dennis Wixon. User performance with command, menu, and iconic interfaces. *SIGCHI Bull.*, 1985.
- [WR09] Ryen W. White and Resa A. Roth. Exploratory Search: Beyond the Query-Response Paradigm. *Synthesis Lectures on Information Concepts, Retrieval, and Services*, 1(1):1–98, January 2009.
- [WRE⁺10] Stuart N. Wrigley, Dorothee Reinhard, Khadija Elbedweihy, Abraham Bernstein, and Fabio Ciravegna. Methodology and campaign design for the evaluation of semantic search tools. In *Proceedings of the 3rd International Semantic Search Workshop, SEMSEARCH '10*, pages 10:1–10:10, New York, NY, USA, 2010. ACM.
- [WRLP94] Cathleen Wharton, John Rieman, Clayton Lewis, and Peter Polson. *The cognitive walkthrough method: a practitioner's guide*. John Wiley and Sons, Inc., 1994.
- [WS92] Christopher Williamson and Ben Shneiderman. The dynamic homefinder: evaluating dynamic queries in a real-estate information exploration system. In *Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval, SIGIR '92*, pages 338–346. ACM, 1992.
- [WvHdV⁺10] Niels Willems, Willem Robert van Hage, Gerben de Vries, Jeroen H. M. Janssens, and Veronique Malaise. An integrated approach for visual analysis of a multisource moving objects knowledge base. *Int. J. Geogr. Inf. Sci.*, 24(10):1543–1558, October 2010.

- [WWo7] Hugh J. Watson and Barbara H. Wixom. The current state of business intelligence. *Computer*, 40(9):96–99, September 2007.
- [WWo8] Hai Wang and Shouhong Wang. A knowledge management approach to data mining process for business intelligence. *Industrial Management and Data Systems*, 108(5):622–634, 2008.
- [WWMo8] Peter Wright, Jayne Wallace, and John McCarthy. Aesthetics and experience-centered design. *ACM Trans. Comput.-Hum. Interact.*, 15(4):18:1–18:21, December 2008.
- [Yam88] F. Yamaguchi. *Curves and surfaces in computer aided geometric design*. Springer-Verlag Berlin, 1988.
- [YAPMo8] Xintian Yang, Sitaram Asur, Srinivasan Parthasarathy, and Sameep Mehta. A visual-analytic toolkit for dynamic interaction graphs. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '08*, pages 1016–1024, New York, NY, USA, 2008. ACM.
- [YWZ⁺08] Ying Yan, Chen Wang, Aoying Zhou, Weining Qian, Li Ma, and Yue Pan. Efficiently querying rdf data in triple stores. In Jinpeng Huai, Robin Chen, Hsiao-Wuen Hon, Yunhao Liu, Wei-Ying Ma, Andrew Tomkins, and Xiaodong Zhang 0001, editors, *WWW*, pages 1053–1054. ACM, 2008.
- [ZTG11a] Jasni Mohamad Zain, Mengkar Tey, and Yingsoon Goh. Does aesthetics of web page interface matters to mandarin learning? *CoRR*, abs/1101.1608, 2011.
- [ZTG11b] Jasni Mohamad Zain, Mengkar Tey, and Yingsoon Goh. Probing a self-developed aesthetics measurement application (sda) in measuring aesthetics of mandarin learning web page interfaces. *CoRR*, abs/1101.1606, 2011.