# TOWARDS A FRAMEWORK FOR VISUALISING THE SEMANTIC WEB

A THESIS SUBMITTED TO THE UNIVERSITY OF MANCHESTER FOR THE DEGREE OF MASTER OF PHILOSOPHY IN THE FACULTY OF ENGINEERING AND PHYSICAL SCIENCES

2013

By
Petch Sajjacholapunt
School of Computer Science

Supervisor: Prof. Carole Goble
Co-supervisor: Dr. Steve Pettifer

# Contents

# List of Tables

# List of Figures

# Abstract

Additions to the Semantic Web data in general domains such as news, movies, music, and books give non-expert users a chance to consume these kinds of data. Although the general user can understand the content of the semantic web, they are still faced with another problem in the form of the poor visualisation of current semantic web tools and a technical problem in terms of the use of these tools. We have therefore designed an evaluation framework based on the literature reviews with regard to information visualisation components and the semantic web background. This allows us to explore the state of the art in terms of visualisation tools for the Semantic Web and allows us to discover potential gaps which can extend the framework for the interactive visualisation of the Semantic Web.

The sixteen-visualisation tools that are active with regard to testing against the evaluation framework were selected. Based on characteristics of these tools, this thesis classified them into four kinds of tools. These are Generic Browser, Search Engine, Exploration tool, and Mashup tool. The evaluation framework thereby addresses the current trends with regard to these types of visualisation tools in terms of data accessing, visualisation techniques and interaction techniques. A basic assessment involving tool testing in a simulated environment and using a test bed of semantic web data sources such as DBpedia and Musicbrainz, shows that there are limitations in the visualisation and interaction techniques of each visualisation tool. The current visualisation tools also lack data transformation and visual mapping processes in order to allow general users to be able to present semantic web data effectively and easily without custom coding.

The evaluation results also illustrate that there are some visualisation tools that provide a dynamic visual mapping process through a semi-automatic approach. We therefore believe that their architecture could be extended in such a way as to design a

framework for an interactive visualisation tool for the Semantic Web. In order to design a framework for a generic visualisation tool, we propose that future research should extend the architecture of the Exploration tool and the Mashup tool because these kinds of tools are somewhat similar to what we need as an ideal tool for general users. This thesis also presents some direction towards this proposal.

# Declaration

No portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

# Copyright

# Acknowledgements

This thesis would not have been possible without the support of many people. I therefore would like to express my gratitude to all those who gave me the opportunity to complete this thesis. Firstly, I am deeply indebted to my project advisor Professor Carole Goble of the University of Manchester, whose support, stimulating suggestions and encouragement helped me throughout my research, and especially while writing the thesis. I also deeply appreciate the generous support provided by Dr. Steve Pettifer of the University of Manchester. Special thanks also to all my friends, especially Dr. Panom Gunawong, who always provided me with useful guidance, cheering me up and offering invaluable assistance. I would also like to thank the School of Computer Science for giving me permission to use its computer resources to do the necessary research and implementation. Finally, I wish to thank my parents for all their moral support.

# Chapter 1

# Introduction

In recent times, the semantic web has become an important concept for expressing the meaning of data on the Internet. The phrase 'Semantic Web' refers to a web of data in which the semantics of data are established by describing and connecting data through formal statements. One standard that has been widely used as a data model for modeling and expressing formal statements on the semantic web is the RDF (Resource Description Framework). This chapter will describe research motivation including aims and objectives. The research problems and research question are also presented to provide an overview of the work to be answered at the end.

## 1.1 Research Motivation

Research on the semantic web concept is not only focused on specialist scientific domains, such as biological and chemical data, but also on more general domains [10, 11, 23]. By 'general domain', we mean the information that people consume in their daily lives on the internet, such as movie, music, news, books, and so forth. Furthermore, the information that users consume is not limited just to static visualisation, as presented by content providers, but also involves information that is dynamically presented as users need it [2]. The general web user, who is a non-technical expert user, is not familiar with the technical concepts of the Semantic Web technologies and generally expects to find a simple and comprehensible tool that can support them when searching for and navigating semantic web data [1]. Moreover, we believe that this kind of user is always full of curiosity about exploring semantic web data. When they find a result through searching, they want to investigate further, sometimes using more than one tool to get different view of the information. An obvious difficulty is that to explore semantic web data in general, users require knowledge about RDF data and its meaning. The users occasionally also need to use more than one visualisation technique or different tools for revealing their desired information. There is, therefore, a need to reduce the difficulties and increase flexibility so that users can discover and visualise semantic web data dynamically within a single application.

We focus our research on an investigation of the visualisation tools that can present RDF graphs and have identified a gap concerning the process of information visualisation. The purpose is to explore and analyse the utility of existing visualisation tools to help improve the visualisation process. Due to the research problems (section 1.2) and taking inspiration from semantic web technologies, such as RDF data, data interchange format, and data annotation, [4, 9, 10], we place our work in the context of the semantic web, in which a visualisation tool is supposed to be able to automatically process the meaning of input data and visualisation methods. We believe that if users provide more information that is machine interpretable regarding the visualisation, tasks and the RDF graph, the visualisation tool will perform better in terms of functionality, extensibility and flexibility. For example, the visualisation tool might make it easier to integrate new visualisation techniques, which would allow more functionality for presenting and discovering RDF data.

## 1.2 Research Problems

From the literature review and through experimentation with mashup tools, we found the following research problems.

- The visualisation tools for representing RDF data are normally designed either to be specialist, which are visualisation tools tailored for a specific domain, or general, which are tools for browsing the general domain of RDF data. The general visualisation tool is still faced with the problem of poor visualisation because it visualises RDF data purely based on its syntax, which comes in the form of a graph consisting of nodes and link [23]. The users therefore require basic knowledge of the semantic web in order to understand the data. Moreover, the complexity of a large graph is also difficult to use, particularly regarding extracting new knowledge. Therefore, we need solutions that allow a general visualisation tool to visualise RDF data in other different visualisation techniques.

- As there is no visualisation technique that can reveal all information, what users need is a dynamic visualisation and direct manipulation approach that allows users to interactively select and present data through the different visualisation techniques contained in the visualisation library. However, the current tools we have explored do not allow users to perform dynamic visualisation through the RDF data. We therefore need solutions that allow interactive and dynamic visualisation.

- In general, the mechanism for choosing visualisations and tasks is based on the target users and the structure of data. In addition, there have been some studies regarding the purpose of the architecture for mapping between structure of data, visualisation, and tasks, such as [40, 54, 58]. However, from our experiments [67, 79–81], semantic web visualisation tools and research that supports a practical mapping process between data, visualisation, and task are rare. We therefore propose to identify an intermediate language or method that would allow machines to communicate between the structure of the RDF data model and the structure of visualisation techniques, which can fill the gaps in the mapping process and reduce the requirement for a manual mapping process.

## 1.3 Key Research Questions

To provide the research with direction regarding the interactive visualisation framework, we have defined a number of key research questions:

- What are the common visualisations and tasks that are common in the visualisation and manipulation of semantic web data?

- What are the prominent relationships between visualisation structures, tasks and data?

- What are the key factors that prevent current visualisation tools performing dynamic visualisations and direct manipulation?

- What are potential techniques that can support the mapping process between data and visualisations?

## 1.4 Thesis Structure

This thesis consists of six chapters. The remainder of the thesis is structured as follows:

**Chapter 2**: This provides the background and related study of the semantic web and linked data, including the language of the Semantic Web, such as RDF, RDFS, and OWL, which will be used for designing an evaluation framework in Chapters 4 and 5.

**Chapter 3**: This provides the background and related studies on a process of information and visualisation components. These components are used as an evaluation checklist in the analysis of the visualisation tool in Chapters 4 and 5.

**Chapter 4**: This will discuss fundamental knowledge about general desktop-based visualisation and web-based visualisation, especially for semantic web data. We present the difference between web-based and desktop applications. The details and current status of both aspects are discussed. The components from Chapter 3 are used for designing an evaluation framework in order to review the existing systems of information visualisation in the next Chapter.

**Chapter 5**: This provides the evaluation result and discussion of 16-visualisation tools. This thesis has carried out further detailed analysis regarding the scope of experiment in exploration and mash-up tools. This thesis also provides the results of the analysis of the processes and architecture of these types of tools, which can be used to identify the potential gaps within these tools in the next chapter.

**Chapter 6**: This provides a conclusion to the results we presented in Chapter 5. This thesis presents the potential gaps and gives answers to the research questions mentioned in Chapter 1. Future areas of research are also discussed.

# Chapter 2

# Semantic Web Background

The term 'Semantic Web' has been frequently mentioned in web technology discussions. The concept relates to machines that can automatically process data on the web, rather than the reliance on humans, which is likely to become a new trend of web technology. This concept does not aim to replace, but to extend the current web technologies (web 1.0 and web 2.0), by giving meaning to data on the web, which will enable machines and people to work in cooperation. In order to design an evaluation framework and identify problems of visualisation, we require an understanding of the underlying data that will be visualised.

This chapter therefore provides an understanding of semantic web data regarding origin, concept, components, and related principles. This is useful for understanding the complexity of semantic web data, which is fundamental knowledge for designing an evaluation framework.

## 2.1 Semantic Web Origins

The World Wide Web (WWW) was created based on the concept of an open community, in which people could contribute their own ideas to others in different parts of the world. People could gain access to information through hyperlinks, which contained the location of a web page document. At present, there are many intelligent web applications that have been built to make the web smarter. By using the term 'smarter', we mean a web application can perform some tasks automatically without extra input and interaction from a user; for instance, a search engine enables people to find desired information by using relevant keywords and the users do not need to know the location of the documents.

To build a smart web however, it is not enough to focus only on building an intelligent application; the development of a web infrastructure is also essential as well [4]. The web infrastructure performs its function by providing a basic infrastructure to support a smart application on the web. The data and information are underlying factors of the web infrastructure, because the intelligent applications can only be as smart as the information available to them.

As a result, in order to make a smarter web, the web infrastructure must be improved, especially in terms of how to serve the required data for the needs of a smart application. The problems at the moment are that data on the web is inconsistent and it is difficult for a machine to extract knowledge. Therefore, many approaches to organise the data have been proposed. One common approach is to apply a relational database for integrating data and solving the problem of data consistency. The relational database, however, is not sufficiently flexible for connecting data over the web. It needs to perform data agreement when users or agents need to share data between two or more data sources. The other common approach is to perform custom programming; for example, writing source code manually for connecting data between a relational database and XML data. This approach is more flexible for connecting data, because a programmer can connect data from one source to another according to its structure. However, this approach is still not appropriate to use for sharing data on the web due to a lack of standards for custom programming, which means it is almost impossible to allow a machine to perform custom programming.

These two common approaches are not flexible and powerful enough to connect data on the web due to its scale. The semantic web concept, therefore, has been created to construct a web infrastructure as a step towards a solution. A standard data model was created to describe data over the web, which is called the Resource Description Framework (RDF). It is assumed that all resources on the web have their own uniform resource identifier (URI), which allows other data to connect through this address. The definitions of these components are presented in the next section.

## 2.2 The Semantic Web Concept

Semantic web data is a term that refers to a web of data in which data are described and connected in order to establish a meaning (semantic) through a defined format and language [29]. From the definition, the phrase web of data means an infrastructure of the web that is mainly focused on the connectivity of data rather than documents (web pages) in order to provide information on web content. The reasons that a semantic web focuses on the connectivity of data are that it aims to enable flexibility in sharing data, an increase in dynamic data, and an improvement in the expression of data on the web. This is why we need a standard, defined format and a language that can describe semantic data on the web.

Many formats and languages have been built to provide semantic data on the web in the past, such as Microformat, and JSON [11]. The data presented by these formats are limited and rely on a structure defined by a generator in which it is difficult to integrate data. Linked data has recently been proposed as a solution. The term Linked data refers to a set of best practices that aims to publish and connect data on the web [11]. One well-known best practice was defined by Tim Berners-Lee, and is called The 5 Stars of Open Linked Data [100].

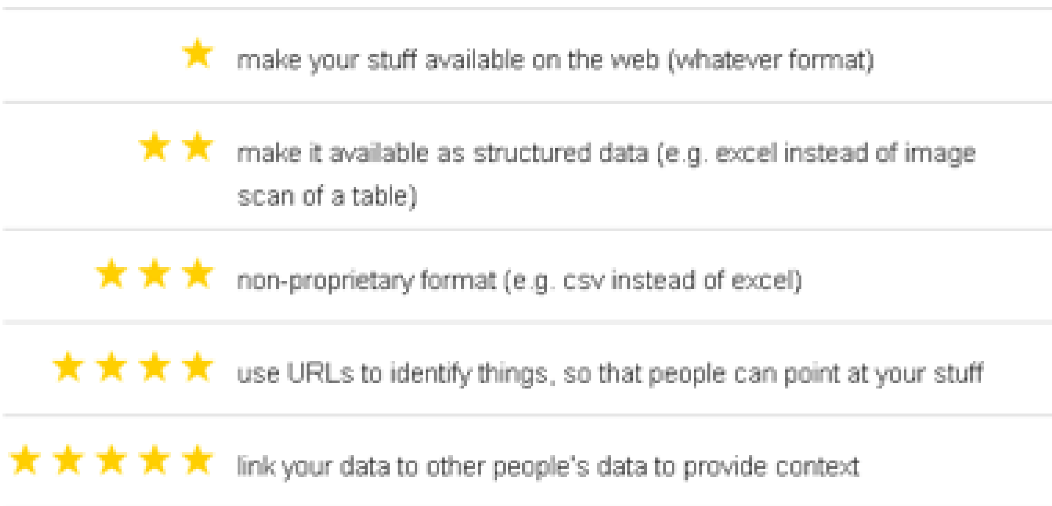


Figure 2.1: The Five Stars of Open Linked Data [100]

Figure 2.1 describes the five steps required to publish linked data. First, begin with publishing data on the web. Second, provide data in a structured format, such as publishing it in Excel rather than as an image of a table or random text. Third, use an open or non-proprietary format such as XML or CSV rather than a commercial one such as Excel. The fourth step is about all data on the web being identified using URLs that can be referenced or linked to by other people. The final criterion for making this approach complete is to link the published data to other people's data on the web in order to provide more contexts. Imagine that everyone follows the five star principles of open linked data: most of the data sources on the Internet will be connected together through semantic links as a big graph, which is presented in Figure 2.2.

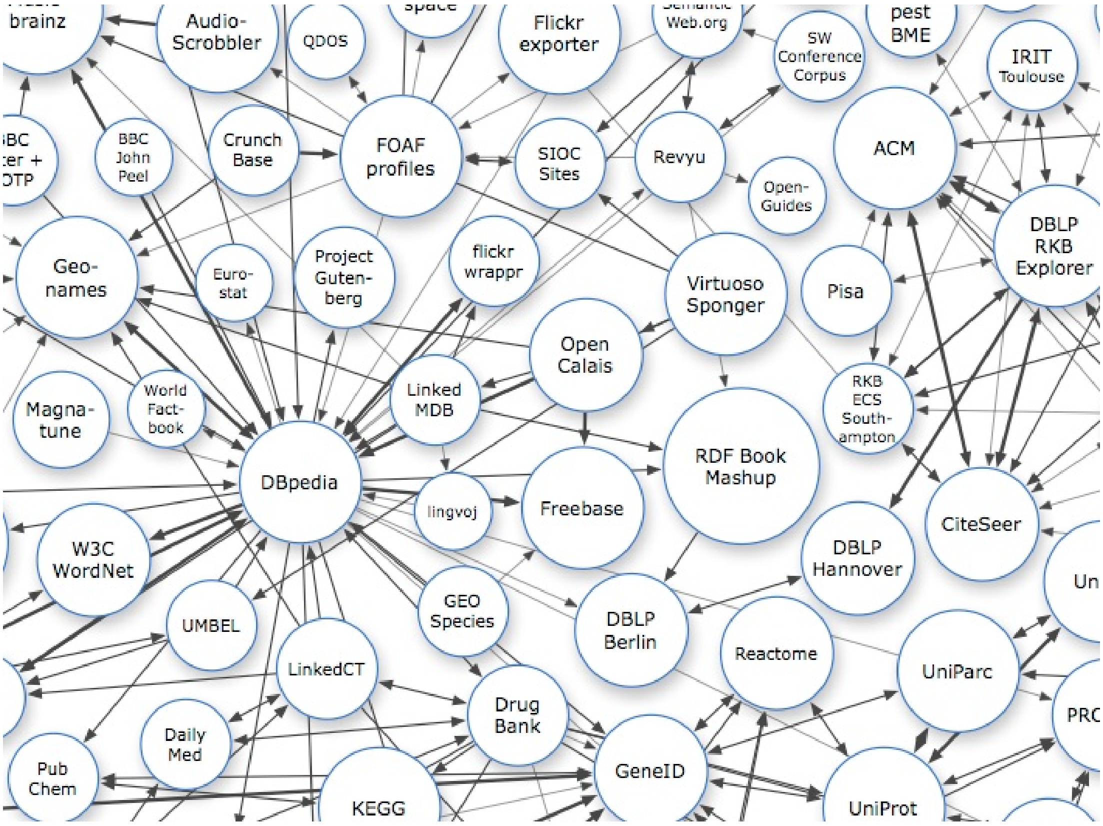


Figure 2.2: An Example of High-level in Linked Data between Data Sources [109]

Following the five stars of open linked data requires a formal statement for describing data on the web. A formal statement is a fundamental building block of the semantic web due to the fact that it provides information regarding the concepts, logic, and restrictions. The information can be extracted from statements that are linked together. A connection of statements therefore can create a network of semantic meaning in which these statements can be analysed and translated into new information or even knowledge. The formal statement supposes to be built up based on a standard data model to certify that the statements are allowed for connecting, sharing, and integrating data on the web scale. The RDF data model is a common standard proposed by Tim Berners-Lee [29] for modelling formal statements to establish semantic data. The details of RDF will be described in the next section.

## 2.3 Semantic Web Layers

The semantic web concept has been designed based on the web technologies and standards presented in Figure 2.3. This figure illustrates a key dependency of the technologies between each layer, such as the Unicode and URI layer, which are the foundation of other technologies in the semantic web layer [4]. A description of each layer is provided below [102]:

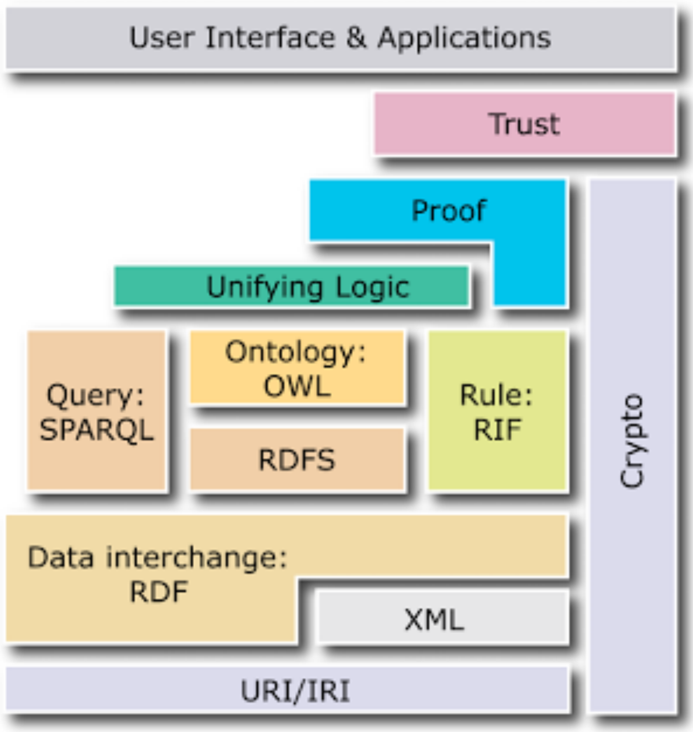


Figure 2.3: The Semantic Web Layers [102]

- **The Unicode and URI layer:** This layer deals with a set of standard characters and provides a pattern for identifying objects in the semantic web.
- **XML layer:** This layer is concerned with the namespace and schema definitions. This is to ensure that the semantic web definitions can be integrated with the other XML-based standards.
- **Data interchange (RDF) layer:** This layer relates to the data format of the semantic web. The RDF is used to make statements about the objects in the semantic web. Each statement is created with vocabularies that can be referred to by the URIs.
- **RDFSchema (RDFS) layer:** This layer is used to control the structure of vocabularies used in the RDF data model.
- **Query (SPARQL) layer:** This layer is used to contain the language and syntax for processing semantic web data format such as RDF, RDFS, and OWL.
- **Rule (RIF) layer:** This layer is used to control the rules languages for exchanging rule purpose due to the variety of rules languages.
- **Ontology (OWL) layer:** This layer is also used to control the structure of vocabularies used in the RDF data model like RDFS but more expressiveness. It more concerns the relationship between the concepts described in the particular domain.
- **Crypto or Digital Signature layer:** This layer is about the standard for detecting change in documents. It is a standard in the W3C working group [101].
- **Unifying Logic layer:** This layer supports the writing of rules.
- **Proof layer:** This layer is about executing rules. It works with the Trust layer in order to evaluate the trust of documents.
- **Trust layer:** This layer concerns a mechanism for evaluating whether documents can be trusted or not.

W3C states that the top three layers, Logic, Proof, and Trust are one of the challenging topics currently being researched. This thesis, however, focuses on the visualisation of sematic web information. We are therefore interested in studying the lower layers, especially the Ontology layer and the RDF and RDFS layers, as they relate to the data that have to be presented. The next section will discuss these layers in detail.

## 2.4 Resource Description Framework (RDF)

RDF is a framework for describing information [4]. Its structure aims to provide a simple data model for modelling resources and their relationships through the simple expression, which is known as triples. The triple, in the Figure 2.4, is a statement that represents relationships between things through the simple components: subject, predicate, and object. The URI reference is an identifier that identifies the address of a resource, such as *<http://www.profile.com/rdf/name>*. A blank node is a special node in RDF graph representing resources that it does not have a URI or literal (sometime is called anonymous resource), it uses as a reference to the same identifier. The literal is a value that contains one or two components. The details of each component are presented in the following list:

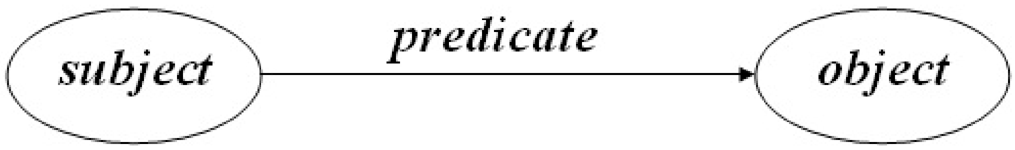


Figure 2.4: the RDF Triple Statement

- Subject: can be the URI reference, or a blank node
- Predicate (sometime referred to as property): can be the URI reference
- Object: can be the URI reference, a blank node, or literal

The abstract structure of the RDF data model can be represented as a directed-graph (see Figure 2.4), in which a node represents the subject and object while an edge represents a predicate. To connect data together over the web, each individual portion of data on the web has to provide its address and property information in a standard format. Moreover, the web resource can be anything that has a URI, such as a title, name, address, date, and all contents of a web page, property, and even property value as well. In addition, people or machine can navigate through web resources by following individual (predicate) resources, unlike the traditional web, which use hyperlinks to only tie up webpages. This allows each object in the webpage to be described in a meaningful way and able to be understood by a web browser. The RDF data model is commonly presented by the XML language (RDF/XML); however, it can also serialise into many format such as N-Triple, N3, and RDFa. The Figure 2.5 below illustrates the RDF data model in RDF/XML syntax to describe relationship of people.

```
<? xml version= "1.0"? >
<rdf:RDF xmlns:rdf= http://www.w3.org/1999/02/22-rdf-syntax-ns#
       xmlns:foaf= http://xmln.com/foaf/0.1/" >
         <rdf:Description rdf:About= "http://www.profile.com/peter" >
                <foaf:knows>
          <rdf:Description rdf:About= "http://www.profile.com/petch" >
                 <foaf:name>Petch Sajjacholapunt</foaf:name>
          </rdf:Description>
                 <foaf:knows>
          </rdf:Description>
</rdf:RDF>
```

Figure 2.5: An Example of RDF Data in XML Format

From the RDF/XML document in the Figure 2.5, we can interpret the information in a triple format as this simple statement.

Peter knows Petch

- The subject of this statement is *<http://www.profile.com/peter>* .
- The predicate is *foaf:knows*.
- The object is *<http://www.profile.com/petch>* .

In this statement, all resources are referred by URIs. Although, *<http://www.profile.com/petch>* represents as an object in this statement, it can be a subject in another statement. For example;

Petch has a name Petch Sajjacholapunt

- The subject of this new statement is *<http://www.profile.com/petch>* .
- The predicate is *foaf:name*.
- The object is a literal value, which is Petch Sajjacholapunt.

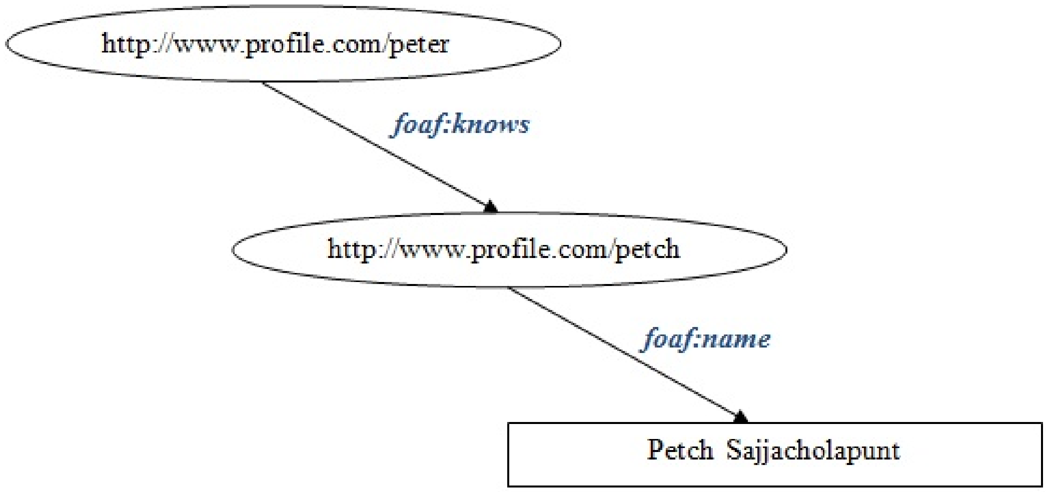


Figure 2.6: The RDF in Graph Format

We can see from the example in Figure 2.5 that RDF makes use of namespaces in which a container provides context for identifier; for example, *rdf* and *foaf* are used to identify namespaces and serve as abbreviations for the full URL. The property '*knows*' is a user-defined terminology and its definition can be found where the namespace *xmlns:foaf* points to. These RDF/XML statements can be represented in the RDF graphs form, which contain nodes and links as show in the Figure 2.6.

Although the RDF language is more powerful than other previous Markup Languages as it is a structured model, it suffers from high complexity when applied to an existing system. Moreover, the fundamental idea of semantic data is to give an address to all resources, which is identified by the identity uniform resource identifiers (URIs). This will enable connectivity among resources on the Internet. Not all resources in fact have a URI; there is a special case, which is called a *'blank node'*, and some resources theoretically have no name at all. It is used to describe ideas that are not for specific resources. An example of the idea is *"all music was composed by person"*. The person in this statement refers to a general term and does not mean any specific person. It is just used to express the idea that *'an artist'*, who is *'a person'*, composes all *'music'*. Although the RDF data model can model all web resources, it does not describe resources well in terms of a more complex structure. The other standard models defined by W3C that are used to extend the ability of the RDF data model, are RDFS and OWL. These data types are more expressive than the RDF data model.

To access RDF data, a special query language, which is called SPARQL (SPARQL protocol and RDF query language) [4] [110], is a common preference. SPARQL performs a query on the RDF graph rather than on the relational database. The result of a query can be serialised in many forms as mentioned earlier, for example: N-triple, N3, or RDF/XML. This is important because in order to use the open source visualisation methods, we need to know what kinds of input data types are accepted. The RDF data models are somewhat different from the data of the traditional web infrastructure. In addition, they were more likely designed to support the inference of a machine rather than humans. Consequently, human information seeking and visualisation techniques might have to be concerned with supporting these data characteristics.

## 2.5 RDF Schema (RDFS)

The RDFS is a framework that provides mechanisms for describing resources, properties and their relationships which are being used in an RDF data model [29]. The RDFS is built upon an RDF data model. It intends to develop a common vocabulary and collection of resources for building schema that can be consistently used to describe other resources and to define simple constraints.

The concept of schema construction in the RDFS is somewhat similar to Object Oriented Programming, in that the RDFS constructs schema based on using simple object elements; classes and properties [4]. The schemata can be constructed through a set of limited vocabulary of RDF Schema defined by RDF specifications [104]. The RDFS vocabulary can be classified into three basic categories that are Classes, Properties and Constraints. An example of each category is presented in the next section.

### 2.5.1 Core Classes

The RDF data model that is created through the RDF Schema namespace is most likely to include the following core vocabularies:

| RDFS Vocabulary | Description |
|---|---|
| *rdfs:Resource* | This class represents all resources that are described in the RDF data model. In other words, all these resources are considered as an instance of this class. |
| *rdfs:Class* | This is a class which declares a resource as a class of other resources. |
| *rdf:Property* | This class represents the class of RDF properties which are use to describe a relationship between subject and object. |

Table 2.1: Example of Core Classes Vocabulary

### 2.5.2 Core Properties

The core properties vocabulary is an instance of the class *rdf:Property*, which provide information regarding relationships between classes and their instances:

| RDFS Vocabulary | Description |
|---|---|
| *rdf:type* | This is used to indicate that resources are an instance of a class. |
| *rdfs:subClassOf* | This property is used to declare hierarchies of classes. In other words, it is used to state that all instances of one class are instances of another. |
| *rdfs:subPropertyOf* | This property is used to state that all resources related by one property are also related by another. |
| *rdfs:seeAlso* | This property is used to indicate an additional resource that provides extra information about the subject resource. |

Table 2.2: Example of Core Properties Vocabulary

### 2.5.3 Core Constraints

The core constraints provide a mechanism for describing the constraints and limitations of using classes and properties in the RDF data model:

| RDFS Vocabulary | Description |
|---|---|
| *rdfs:range* | This is an instance of *rdf:Property* that is used to state that the value (object resource) of a property is an instance of one or more classes. |
| *rdfs:domain* | This is an instance of *rdf:Property* that is used to state that any resource (subject resource) that has a given property, is an instance of one or more classes. |

Table 2.3: Example of Core Properties Vocabulary

A Schemata which is defined through the RDFS vocabulary will be used depending on the purpose of an application. For example, a reasoning application will use it for inferring the class, and to reveal an inconsistency, while a validator application might look for an error through constraints. The full version of the RDFS vocabulary is provided in the RDFS guide [104]. The following example statement illustrates a basic use of the RDF schema vocabulary for describing an RDF data model:

- Boat is a vehicle
- Car A is a car
- Cars are vehicles
- Toyota is a brand of vehicles
- Toyota is a brands of car B

These statements can be interpreted to an RDF/Turtle format which is a simple alternative serialization format of RDF, and as an RDF graph as presented in Figure 2.7 and Figure 2.8 respectively.

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix ex: <http://example.org/> .
@prefix car: <http://example.org/car/> .
  ex:boatA rdf:type ex:vehicle .
  ex:carA rdf:type ex:car .
  ex:car rdfs:subClassOf ex:vehicle .
  car:brand rdfs:range ex:vehicle .
  ex:toyota car:brand ex:carB .
```

Figure 2.7: Example of an RDF Model in the RDF/Turtle Format

The property *'rdfs:subClassOf'* provides extra information to the machine and allows it can interpret that 'car is a vehicle'. The property *'rdfs:range'* also creates a constraint that the object resource of the property *'car:brand'*, which is car B, is an instance of the class vehicle. Hence, if the user builds a query to present all instances that is a vehicle, the result will present *'ex:car A'* , *'ex:car B'*, and *'ex:boat A'*. The simple RDF graph of this RDF document is manifested in Figure 2.8. However, the use of RDFS is not enough for processing in a complex RDF data model. Other formal languages such as OWL [105] and SKOS [107] that have more expressiveness are needed to fill these gaps.

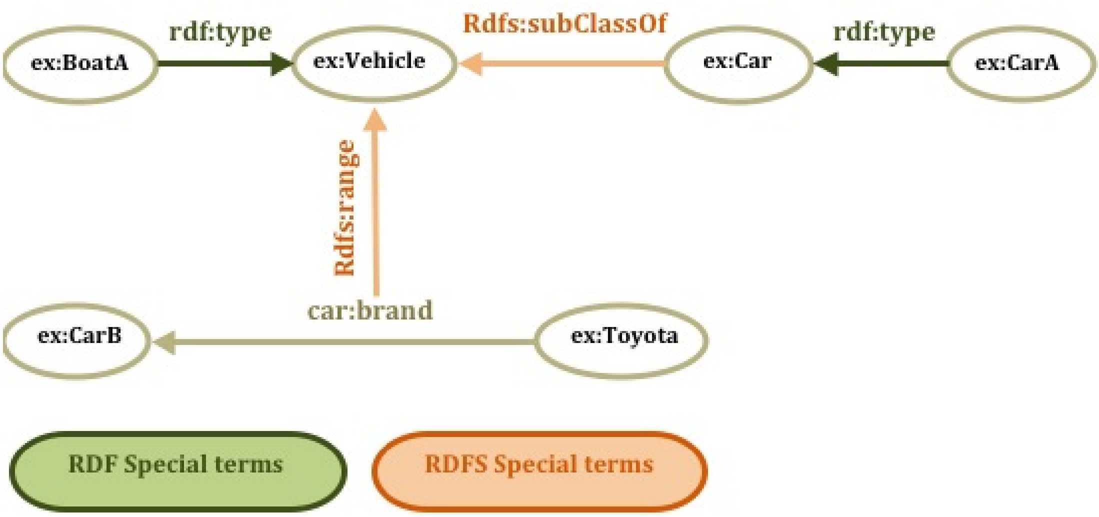


Figure 2.8: An Example of an RDF Graph Representing a Simple RDFS Schema Case Study

## 2.6 Web Ontology Language (OWL)

The OWL is a knowledge representation language that has been builds on top of RDF like the RDFS for authoring ontologies. These ontologies are used for describing data as a set of "individual" and as a set of "property", which relate these individuals to each other [105]. However, unlike RDFS, the OWL provides additional vocabulary and semantics by which a machine can perform a greater computation and interpretability of web contents and so overcome RDFS limitations. For example, a vocabulary in RDFS cannot specify that Person and Vehicle classes are disjointed.

An ontology designed by OWL relies on the "open world assumption" in which everything we do not know is undefined [4]. This is opposite to the close world assumption that implies everything we do not know is false. By following the open world assumption, statements that cannot be inferred to be true with current knowledge, cannot be implied as being false. The ontology consists of a set of axioms, statements that being accepted to be true, which place constraints on a sets of individual (sometime called "classes") and the types of relationships between them. The semantics of web content are provided through these sets of axioms, which allow a machine to interpret extra information from the explicit data that has been provided.

The OWL has different levels of expressiveness from simple to complex, which are OWL Lite, OWL DL, and OWL Full respectively. Each of them is designed to support a different specific web developer and user. They are also a syntactic extension of its simpler predecessor. For example, every legal and valid OWL Lite ontology is a legal and valid OWL DL ontology as well. The following list presents details in each sub-language ordered by expressiveness respectively.

- **OWL Lite**: This is a lite version of OWL that is intended for a user who needs to create a simple constraint, and is concerned more with a classification hierarchy.
- **OWL DL**: This is an OWL sub-language represented by description logic. It is intended for users who need a maximum amount of expressiveness and a computation of completeness and decidability. This sub-language is more appropriate for reasoning in computer application.
- **OWL Full**: This is a full version of OWL that is intended for maximum expressiveness as is the OWL DL, but is less restricted in terms of constraints. This

is designed for users who need syntactic freedom of RDF, but with no need for computational guarantees. It is unlikely that any reasoning software will be able to support the complete reasoning of OWL Full.

Furthermore, there is an extension to, and revision of, OWL in the form of OWL 2 [113]. OWL2 introduces an alternative way of expressing OWL endeavours to eliminate some of the limitations that have appeared in the previous version of OWL such as increased language expressivity, qualified number restrictions, relational expressivity, etc. Further information can be obtained in [114]. Moreover, to overcome the issue of OWL Lite and OWL DL, OWL2 has its own trimmed-down version, which is called OWL 2 profile. There are three OWL2 profiles which are useful in different contexts. These are:

- **OWL2 EL**: is apparently useful in application ontologies that have a large number of classes and properties.
- **OWL2 QL**: is appropriate for application ontologies that have large instances of individual data.
- **OWL2 RL**: is designed for applications that require scalable reasoning without much need for expressiveness. OWL2 RL can be implemented using rules-based engines.

This thesis focuses on the study of ontological language such as RDFS and OWL because the RDF data model requires a standard for expressing meta-data and constraints. Furthermore, the visualization of RDF data also requires a consideration in terms of presenting these meta-data, and constraints in terms of RDF data. The OWL vocabulary for building an ontology can be seen in the OWL guide [106].

## 2.7 Summary

This chapter has explained the origins and concepts of the semantic web. It intends to provide the reasons for developing the semantic web concept, and presents the revolution behind it, and related principles. This chapter has also given some brief examples of what RDF looks like in terms of both simple language expression and graphs. However, the RDF data model requires a method for organising information. Using schema and ontology to build up constraints is a conventional approach that has been used as a relational database technology. Three commons languages and vocabulary for building schemata have been introduced in this thesis: RDFS, OWL, and OWL2. A machine gain benefits from these schemata information in the form of a meta-data. This allows the web content to be processed by computer application.

The schema and ontology of RDF, however, is difficult to create, implement, and maintain due to their scope. From the linked data principles point of view, this can present the idea that semantic web data has an issue with visualisation when there is a lot of data to be presented. This leads to a big-fat-graph [23] which is difficult for the general user or even a user with some expertise to extract patterns and knowledge. To obtain desired information through RDF data, a user is required to perform searching, navigating, or querying processes. This thesis therefore intends to study and evaluate visualisation tools for finding common visualisations and tasks. If successful, this will help users to visualise these huge amounts of data, and also identify the factors that prevent the current visualisation tools from performing rich visualisation in the generic browsing and navigating context.

In the next chapter, we will discuss the background to visualisation, with an emphasis on the process of information visualisation and its components. The relationship between each component will also be discussed in order to establish a framework for evaluating semantic web data.

# Chapter 3

# Information Visualisation

Information visualisation is the process of displaying data in a more meaningful way than just plain text [3, 20] and it aims to enhance the performance of the information-seeking and searching processes. Historically, there have been two common ways to represent data: either in text or in a graphical format. Although text format representation is easier to implement, representing data through it requires a high cognitive effort. Moreover, it is difficult to spot and analyse trends through a large amount of text. On the other hand, representing data in a graphical format simplifies the result and also allows humans to use fewer cognitive aspects of their memory to understand patterns in a large amount of data. Semantic web data, however, are mostly presented in text or graph structure. We would therefore like to study the possibility of presenting RDF data in other graphical formats. We are also interested in investigating the current limitation that prevents the semantic web tool from visualising such a presentation.

To study the visualisation of semantic web data, it is important to understand the underlying theories of information visualisation, including related components for visualisation. This chapter therefore provides an understanding of the process of information visualisation and its components. This is useful for gaining an insight into each stage of the information visualisation pipeline and for addressing the problems of the current semantic web visualisation. In addition, studying related visualisation components that have been used in an information visualisation pipeline will lead to ideas for designing our research framework.

## 3.1 The Process of Information Visualisation

The process of information visualisation is a set of mechanisms or steps to convert visualisation structure from one to another, such as converting raw data in a data table structure to an expansion tree visualisation in a tree structure. Two mechanisms, transformation and mapping process, are core elements in the process of information visualisation. Card's diagram (Figure 3.1) illustrates the underlying stages in the process of information visualisation [7]. The first step is to transform the raw data from whichever format it is into a standard data structure such as data tables containing relationally structured data including metadata. This step can be performed either by manual coding or by using conversion tools such as DERI Pipes [81], Parse-O-Matric [98], and Explorator [78]. The next step is to create a mapping between data tables and visualisation structures (e.g., chart, graph). While an expert user can perform this step manually, some visualisation tools are available to a non-expert user, such as Google's Public Data Explorer [67] and IBM's Many Eyes [77]. The final step is to transform this from visual structures into views of its structure. Examples of tools that can support this task are Exhibit [80], mSpace [90], and the Google Chart Tool [71]. These views can then be transformed interactively through users interaction processes, such as searching, browsing or zooming.

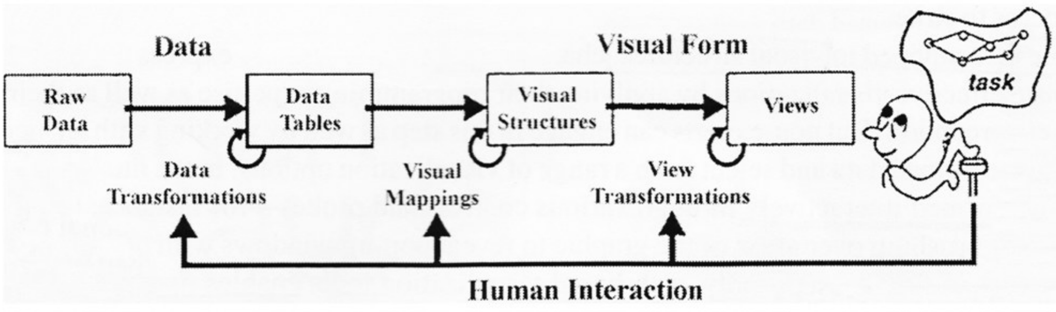


Figure 3.1: A Process of Information Visualisation [7]

In fact, human-system interaction can occur at any stage of the information visualisation process. An expert user might feel comfortable with an exploration of data by querying, while a non-expert user might prefer visualisation tools to help them explore their data in a pictorial form. This can imply that different users require different views and approaches to explore data. An ideal visualisation tool is supposed to take care of all stages in the process of information visualisation and allow the user to interact dynamically with the visualisation results. In desktop-based visualisation tools, there are some tools that support the user to carry out all these stages, such as the Advanced Visualisation System (AVS) [8]; however, there are not many visualisation tools focused on the Semantic Web data. Most of the current tools that support visualising Semantic Web data were developed on the web-based application. This thesis, therefore, focuses on examining web-based visualisation tools. An analysis of the applications and tools is given again in detail in Chapters 5 and 6. The next section, therefore, gives an explanation regarding information visualisation components, which are extracted from the process of information visualisation.

## 3.2 Information Visualisation Components

In this section, the relevant components of an information visualisation pipeline are discussed. Designing an evaluation framework for answering research questions, mentioned in Chapter 1, requires consideration of the fundamentals of user activities (tasks), appropriate visualisations (visualisation), and the characteristics of the data (data) [1]. Each concern interrelates with all the others. The relationship between data, tasks, and visualisation can be pulled out of the Card's diagram and be presented in a simplified form as represented in Figure 3.2 . The diagram illustrates the data flow of information visualisation components, in which the data component is related to the visualisations and tasks component. To visualise raw data, a structure of data, such as data tables format in relational database, must be supported with a library of visualisation. For instance, a pie-chart visualisation supports the data tables structure. Moreover, other details of data components embedded in visualisation, such as size or type of data suppose to be related to the tasks. For example, a zooming task requires granularity of information in detail in order to magnify the results. In another point of view, a user activity is related to a visualisation technique and data embedded in that technique. Hence, the structure and data regarding this visualisation technique must be manipulated and organized by user tasks in order to achieve its purpose.

When visualising data, it is necessary to understand the types of data that are compatible with the types of visualisation, as well as the tasks that can feasibly be performed. For example, if the user activity is to allow people to sort information, then the type of data with some characteristic that allows ordering (e.g., being numeric) is most appropriate. In addition, a visualisation that can present the results in a sequence (e.g., a table or a bar chart) would be most useful. These three core components, therefore, will have an effect on designing the evaluation framework for semantic web tools which are discussed in Chapter 4.

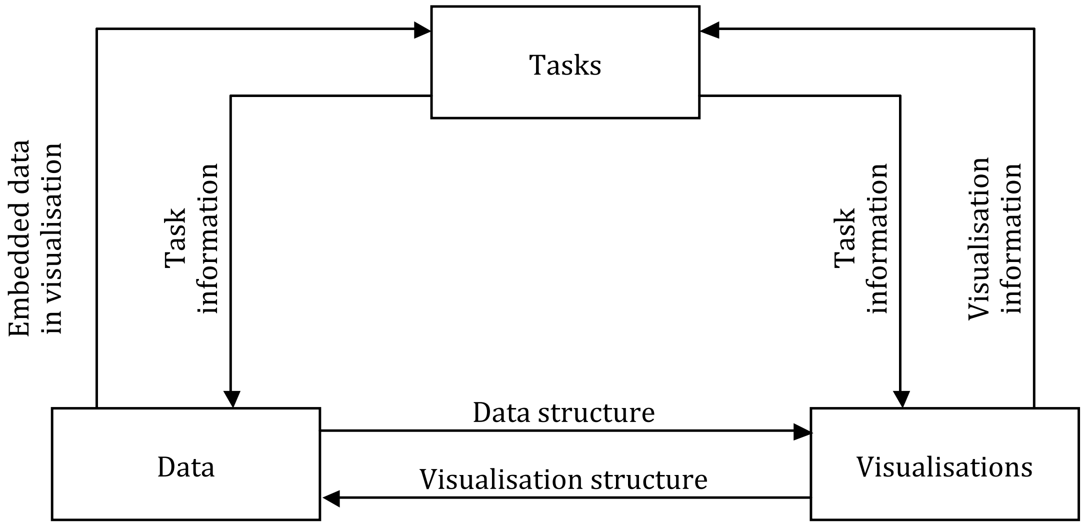


Figure 3.2: A Data Flow of Information Visualisation Components

### 3.2.1 Types of Data

Data is a collection of facts, such as values or measurements, which are collected for analysis and for drawing conclusions. It is helpful to understand the characteristics of the information that is being looked at before it can be presented. In Chapter 2, the characteristics and model of semantic web data are described. This section, therefore, is concerned with understanding other general dimensions of data. In addition, due to the fact that information visualisation is involved with different forms of data, it is necessary to be informed of what type of data needs to be interacted with and how best to present it. As a result, this section provided a short explanation of the data type dimension.

Bertin states that the type of data for display is related to the form of representation [5]. He also suggests that there are two forms of data, a data value and a data structure, which are commonly used to represent a problem. The data value is used to represent numerical or categorical attributes relevant to a problem while the data structure is used to present a whole characteristic of datasets through links, mathematical equations or constraints. It can be inferred that Bertin classified data to be displayed into low-level and conceptual-level views of visualisation [6]. Furthermore, the different visualisation techniques will represent different level of information result. For example, bar charts represent attribute-values of data, whereas tree diagrams represent the relationship between the data in the dataset. From Bertin's argument, a process of information visualisation is concerned with, not only a representation of data value, but also a boundary of presentation. Common dimensions concerning a presentation of data can be illustrated in Figure 3.3.

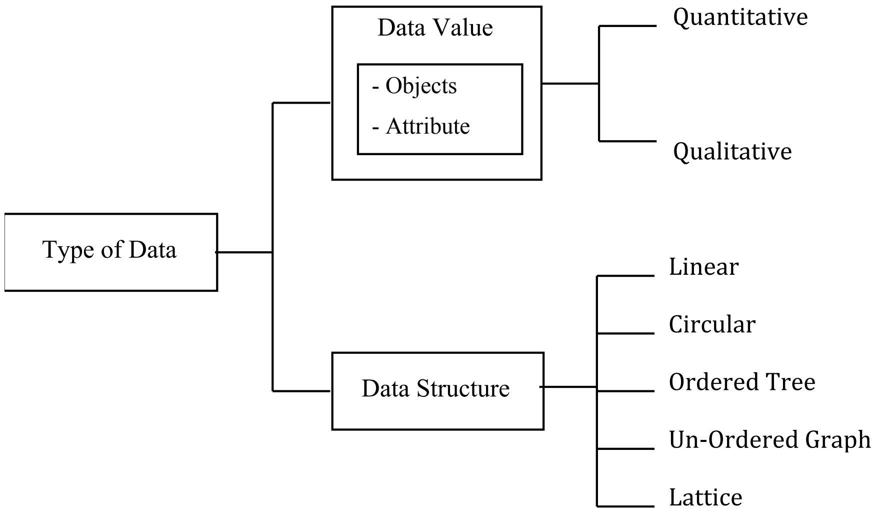


Figure 3.3: Data Types Classification

Tweedie was interested in classifying a boundary of information visualisation in two levels of data value: attributes and objects [25]. The concept of Tweedie's classification concerns all main data in which users are interested, as the objects to be presented and the attributes are meta-data (data about data) of the objects. In relational databases, for instance, the table entity can be seen as an object; each field in the table represents an attribute. From the semantic web point of view, Tweedie's concept can be applied such that user-focused data is an object, while the information linked around the data is the attributes of that data. To present these data, however, it is still necessary to understand the type of data value in greater detail.

As Bertin mentions, one form of data to be presented is a data value. Senay categorises two further data type values, namely quantitative and qualitative. Quantitative data (e.g., size, distance, time) consists of nominal and ordinal values, while qualitative data (e.g., discrete categories) comprises scalar, vector and tensor (mathematical objects that can present both scalar and vector) data. See [13]for more detail and explanation of these types.

When visualising an overview of data, it is beneficial to understand the different structure of data. Basically, the structure of data refers to a specific form of data that is stored and organised in order to be used or presented effectively in the future. Bertin categorises the structure of data into five levels: Linear, Circular, Ordered Tree, Un-Ordered Graph, and Lattice. In this dissertation, the un-ordered structure has been the focus, because the nature of RDF data is a graph as mentioned in the previous chapter.

### 3.2.2 Types of Visualisation

Visualisation is one significant component that is related to the user's perceptions. Having gathered the data, the data structure and data value must be mapped with related visualisation techniques. Understanding the classification of primitive visualisation elements, therefore, allows users to identify which visualisation techniques support which data structure. This is helpful as a criterion to study in the current state of semantic web visualisation tool presented in Chapter 4. This section will discuss the proposed ideas for primitive visualisation elements gathering from Senay theory [13], including how to categorise them. The primitive visualisation elements are fundamental components used in the visualisation of data. It is classified into three dimensions, namely positional, temporal and retinal. Each dimension is described in detail in the following sections.

#### 3.2.2.1 Positional Visualisation

Positional visualisation is the most significant element of the three. It is concerned with presenting data based on position through a set of primitive visualisation structures. It can be used to present data in both quantitative and qualitative forms. Data types and data structures must be considered when using this element. The thesis however focuses on Semantic Web data in which the raw data is represented in RDF graph format. It is therefore important to study the primitive visualisation technique that current visualisation tools use for presenting RDF graphs. The following primitive positional visualisation structure, which derives from the open APIs are listed below:

- Single Axis (1D) (Google API [71]) : presents a set of data in row and column format.
- Multiple Axis (2D, 3D) (Google API, Flare) : displays a set of data based on two values, one of which determines the horizontal axis and the other determines the vertical axis. The data is displayed as a collection of points.
- Area plot (Google API) : presents a set of data with the two data values connected by a line and the area below the line shaded a different colour to represent the other data. This type of structure is similar to that of the line plot type.
- Bar plot (Google API, Flare) : presents a set of rectangular bars based on the values that they represent. This type of structure can be plotted both horizontally and vertically.
- Arrow plot (JGraph [73]): displays a set of data from the base value. Each has a direction point to the other value.
- Network plot (HyperGraph [74], Graphopt [75]): displays a set of each data which is called a node, that connects together through the link.
- Line plot (Google API): presents a set of data with the two data values connected by a line.
- Candlestick plot (Google API): displays a set of data in n-dimensional space through a backdrop that is drawn as equally spaced parallel lines.
- Parallel Coordinate plot (Google API): displays a hierarchical structure normally used to represent probability spaces.

#### 3.2.2.2 Retinal Visualisation

Retinal visualisation is a fundamental technique concerned with how humans perceive information through their eyes [3]. Brodbeck argues that a human has the abilities to scan, recognise and remember patterns through visual elements such as length, shape, orientation, texture, and colour. These visual elements can generate meaningful visualisations and makes it easier to differentiate and extract patterns. Seney classifies visual elements into four components, which are Orientation, Size, Shape, and Colour. The detail of each component is explained in the following list:

- **Orientation**: Orientation is a visualisation revealed by the positioning of data. This enables the complex structures which are hidden inside abstract graph data to emerge.

- **Size**: Size is commonly used to differentiate between two concepts, and users can quickly distinguish concepts in terms of size.

- **Shape**: Shape is another common visualisation for differentiation among concepts of size. The factor of size is sometimes not enough to allow differentiation between data in more detail; hence, it is necessary to consider other factors as well.

- **Colour**: Colour is a rich visualisation used to represent a large amount of data. It is apparent that colour is used to differentiate concepts more effectively than size. In addition, it can present granularity or shade of concepts, such as human temperature information.

These four components are also embedded in the eight dimensions of Pfitzners classification [22]. He believes that any display screen only has eight dimensions with which to convey meaning: plan, colour, value, size, texture, orientation, shape and relationship. Each visual element in the retinal visualisation, however, also needs to be considered as part of the whole, because each element has different function and purpose for helping in visualisation. For example, a task about differentiating will involve size, colour and shape of data in order to make it easier for users.

#### 3.2.2.3 Temporal Visualisations

Temporal visualisation focuses on time series and represents the changing of information over time. This component works effectively when it is presented in an animated format, which is an action of objects that are changing from one state to the other, such as blinking of text, changing in colour or size. This is because users can insightfully perceive changing concepts over time.

These three dimensions, as defined by Pfitzner [22] and Senay [13], can be used as a guideline for both systems and users to determine visualisation elements in each direction. For example, the positional technique is most relevant in representing quantitative data, especially business reports that require rich visualisations such as bar charts, line charts, etc. Moreover, the retinal techniques might also require inclusion since they can help in distinguishing or emphasising the information. In addition, sometimes presenting dynamic data over time (temporal visualisation), such as by animation, is required to allow easier understanding for users. To present useful data, it might be necessary to aggregate many elements and dimensions together; for example, integrating bubble charts with Google maps, plus concerning make use of a different of colours in each country to present the size of populations around the world. Therefore, the study of semantic web visualisation tools requires these elements as a schema for an evaluation performance.

### 3.2.3 Types of Task

A task is an activity performed by the user at the view stage. The purpose of a task for the user is to find and seek information. In the context of a visualisation tool, a task can be compared as a function or module of the tool, which allows the user to interact with embedded data in the visualisation interface in order to obtain specific results. Using the same task with different visualisation techniques might result in a different information result; for instance, someone browsing data through a timeline visualisation might get different information to someone using a graph visualisation. The study of tasks is related to user's behaviours for the discovery of desired information, known as information-seeking behaviour. The emphasis is on studying a user's purposive seeking for information in order to achieve goals. Wilson demonstrates that information-searching behaviour is a micro-level behaviour nested to information-seeking behaviour [19]. This section, therefore, is concerned with understanding the fundamental behaviours of users and types of task. Studies in related fields, such as information-seeking behaviour, also help to identify the common tasks of users, which is mentioned in the research questions.

Information-seeking behaviour helps to clarify the human information perception. In 2002, Bates [15] proposed an integrated model that incorporates both information seeking and searching (Figure 3.4). This model demonstrates how humans perceive information. It classifies seeking and searching modes using two factors, namely:

- **Direct and Undirected**: whether a seeking task can be specified to some degree or is about seeking information more randomly.

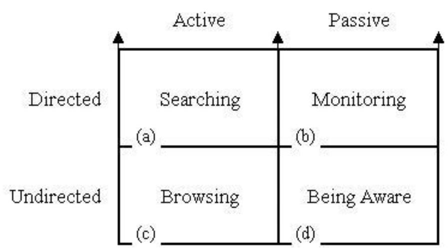


Figure 3.4: Modes of Information Seeking [15]

- **Active and Passive**: whether a user performs any action purposely to get information or just passively absorbs information without any effort being made to seek it.

The model includes four subsections of information-seeking and -searching tasks: awareness, monitoring, browsing, and searching. The research mentions that the two passive modes, awareness and monitoring, are the ways that the majority of users approach learning during their life; for instance, children learn to speak from their environment. The only difference between monitoring and being aware is that in monitoring the user has to have a question in mind, but not act to find its answer, while in awareness, the user absorbs knowledge without having any specific intention at all. The other two methods, namely browsing and directed searching, present active information-seeking behaviour. Directed search is performed when users know what information they want; on the other hand, browsing is used when they do not know what they want and therefore need to make discoveries through serendipity. The results of Bate's model show that most human perception processes come from awareness. However, Bate's model is not sufficient to use on its own. It does not contribute enough of information seeking and searching tasks. For this reason, the task classification theories are required to further study.

Ellis, Cox and Hall proposed eight different tasks or features associated with information-seeking behaviour (Figure 3.5) [21]. The features comprise starting, chaining, browsing, differentiating, monitoring, extracting, verifying, and ending. Ellis indicates that these tasks which users perform do not necessary occur in a linear sequence; however, some features are dependent on one another and have to be processed linearly. The eight different tasks of Ellis's work seem to cover the majority of the activities of

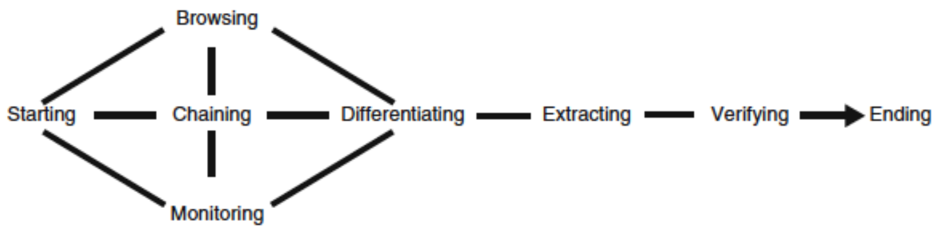


Figure 3.5: Ellis 1993 Behaviour Model for Information System Design [21]

end users on the web; however, this framework still lacks some detail on information-seeking tasks, such as the degree of interaction. Although there are many theories on information seeking, most of them have a common task, such as searching, browsing, navigation, monitoring, and so forth. These common tasks concern the manipulation of an RDF data model at the view stages due to the need for a dynamic visualisation result for the user.

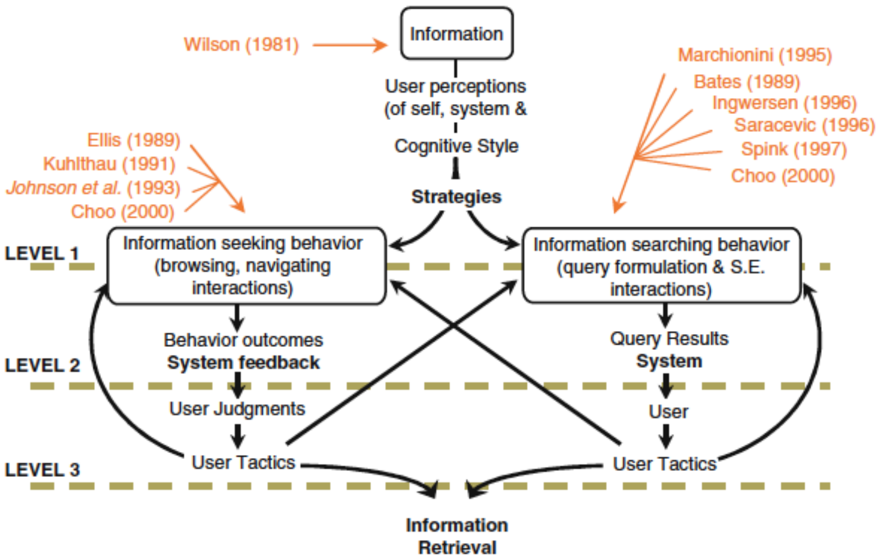


Figure 3.6: A Macro Model of Human IR Behaviour on the Web [21]

A macro model of human IR behavior on the web in Figure 3.6 can illustrate a combination of information-seeking and -searching behavior theory. This model argues that information begin with information need, which is influenced from user's cognitive style. The need of user will appear in making use of specific information seeking and searching. Users normally periodically change information behavior styles between these two strategies. In case that user knows the where is a target, however, it seem like they can directly perform search style strategy without the need of seeking strategy.

## 3.3 The Visualisation Framework

The general visualisation framework in Figure 3.7 has been inspired by the literature review regarding the relationship of visualisation components in a process of information visualisation and its components in the previous sections. This framework aims to provide the basic elements required by a developer who plans to design a generic visualisation tool. It is also used for designing an evaluation framework in Chapter 5. The framework is comprised of five major components: data, user's characteristics, user's task, level of interaction, and visual representation. A detail of each component is described below.

### 3.3.1 Data component

The data component in the visualisation framework concerns both data type values and the structure of data. The theories of Tweedy, Bertin, and Seney are used to categorise these two elements. The data type value comprises both raw data and meta-data. Raw data is data that is desired by the user while meta-data is data that describes or relates to the raw data. These two data type values can be expressed in both quantitative and qualitative approaches. The structure of data is an element that is considered as a method of storing and organising data and it relationships. The data component has a direct effect on the visualisation component in which the visualisation elements must support type of data and data structure.

### 3.3.2 Visualisation Component

The visualisation component contains a set of visualisation elements that are used for representing data value and structured data. This component is comprised of three visual elements, which are positional visualisation, retinal visualisation, and temporal visualisation. Each element is aimed at answering the question of how to represent different types or structures of data, or how a system should present specific content. The visualisation components have a direct relationship to the data component and tasks component. This means the visualisation components must be assured that their elements are compatible with the data component and task component.

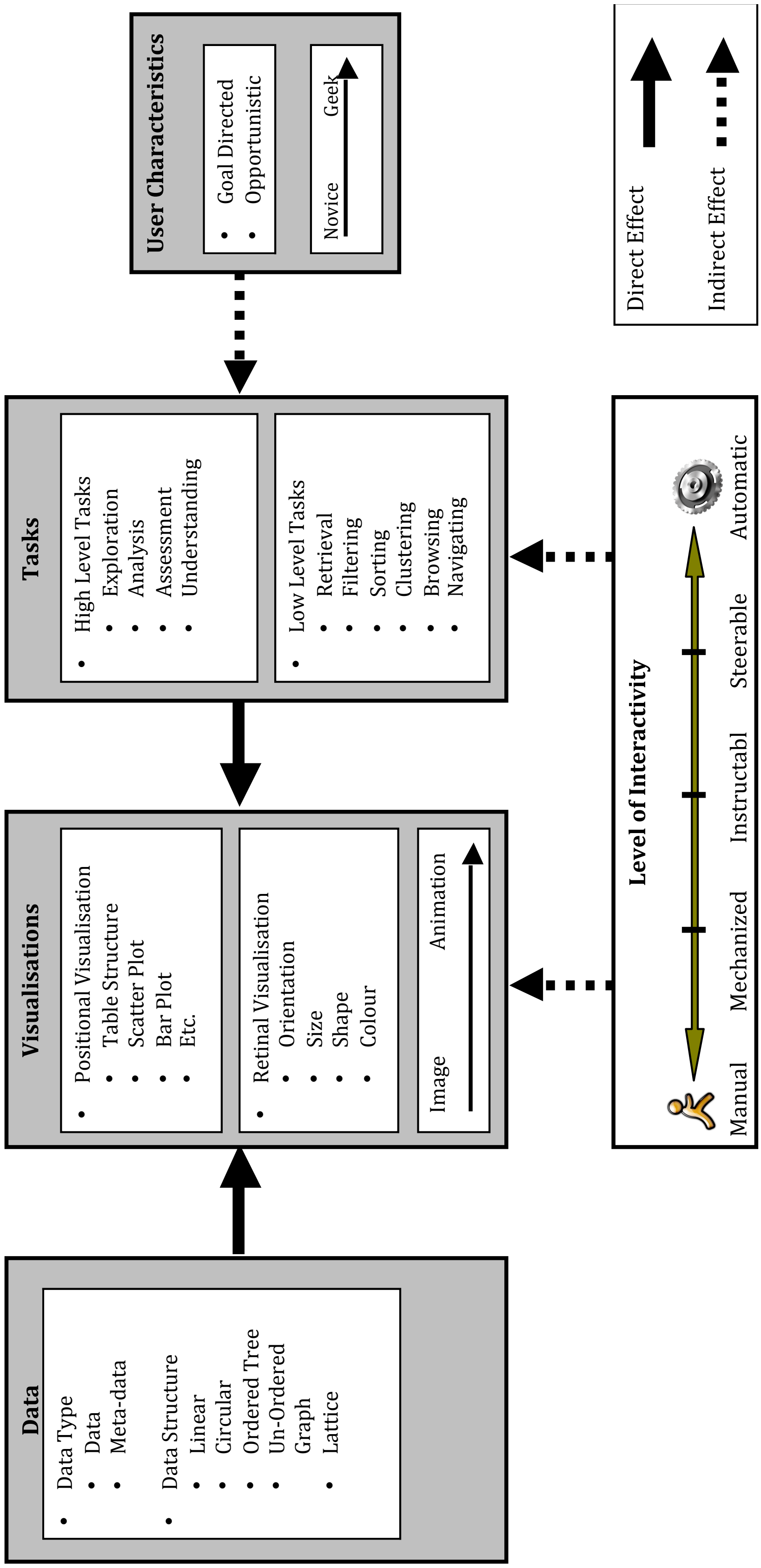


Figure 3.7: The General Visualisation Framework

### 3.3.3 Task Component

The task component is a user's activity, which concern a function or module that interacts with the visualisation components. For example, when browsing information on the web, a positional visualisation such as a bar chart is required to support the mouse-click function, in order to allow the user to perform browsing through the detail of a particular region. This thesis classifies the user's tasks from the literature review into two basic categories, which are low-level and high-level tasks. The low-level task is an activity that aims to manipulate data directly in order to uncover the pattern of the data, for instance, retrieving value, filtering, sorting, clustering, etc. The high-level task is a multiple combination of low-level tasks in order to achieve an ultimate goal. Besides that, the high-level task is intended to provide understanding to the user, by means of, for example, exploration, analysis, assessment, understanding, etc.

### 3.3.4 User's Characteristics Component

This thesis classifies the level of users into two types, novice user, and expert user. The novice user is defined as a user who does not have much knowledge about the content, while the expert is one who has experience in some skill or content, for example an ontologist, content-curator, programmer, or specialist. Users take one or two approaches to discovering answers: a goal-directed approach and opportunistic-approach [15]. The goal-directed approach is a characteristic in which a user knows what they want from the system. This can be seen as a direct and active mode as mentioned in the discussion of information-seeking behaviour. Conversely, the opportunistic approach is a characteristic where users have a rough idea but do not know exactly what they want. The user's characteristic component has an indirect effect on the task component because the user's characteristics are not relevant to the task component and are not necessary for performing any action.

### 3.3.5 Level of Interactivity

Interactivity is defined as the level of interaction when a user is performing tasks during a discovery process. In fact, without interactivity from a user, a system cannot generate any result. However, there are many levels of interaction, which is aimed at measuring a level of automaticity in a system. The level of interactivity in this thesis is applied from Pfitzner's work [22] , which defines a level of interactivity into five steps: Manual, Mechanized, Instructable, Steerable, and Automatic.

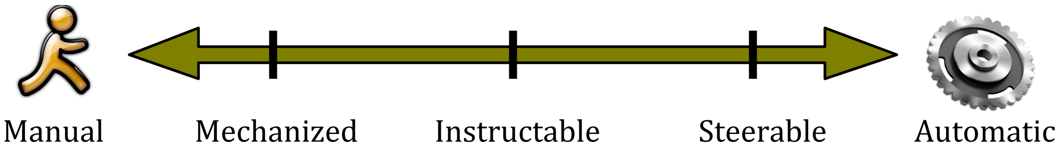


Figure 3.8: The Level of Interactivity

- **Manual**: the activity of a user using physical actions to interact with a system, such as dragging something with a mouse.
- **Mechanized**: the activity in which the user interacts with an object or controls a system, such as increasing focus by a selection of a value from a drop-down lists menu.
- **Instructable**: the activity in which user interact with the system through a complex query interface such as creating a formulas in a spreadsheet, or query.
- **Steerable**: the activity in which the user interacts with a system by providing an algorithm which can be applied to some specific purpose such as visual programming.
- **Automatic**: the activity in which the user interacts with the system through control, which is embedded a complex algorithm to achieve a result such as an embedded calculation algorithm in a button of the calculation application.

In this framework, the level of the interaction component has been set as an indirect relationship with visualisation and task because this component is not a major component that affects the process of visualisation.

## 3.4 Summary

This chapter has presented the process of information visualisation and its components. It briefly gave some examples of visualisation tools, which support the user through the process. The relationship between each component was also discussed. The general visualisation framework (in Figure 3.7) was synthesized from the literature review in this chapter. This framework presents all the visual component and elements that need to be considered for developing a visualisation tool. To generate interactive dynamic visualisation, compatibility between the major components; visualisation, task, and data, in visualisation framework needs to be concerned. For example, the system should be able to answer the question of how many visualisation techniques can present a particular group of data, how much interaction can be supported in any particular library, and what data type is required for future visualisation. These three components, therefore, are the core components when proposing a framework. Moreover, the elements in each components that can be combined to generate more complex visualisations.

From the literature review, it is obviously seen that data, visualisation, and task component have a relationship between each other. Hence, developing a tool for visualisation must be considered these factors. Moreover, this thesis aims for studying a component that is related to a visualisation of semantic web data. This thesis believe that, it is possible to identify a common relationship and pattern between each components in semantic web visualisation tool by survey method. The background of RDF data in Chapter 2 and the general visualisation framework in this chapter are used as a guideline to design an evaluation framework in Chapter 5. The expected result from the evaluation is to answer the research question mentioned in Chapter 1. It also could be used to analyse a gaps, which leads to a development of interactive visualisation framework for semantic web in the future.

In the next chapter, a background of semantic web visualisation will be discussed. The example of semantic web visualisation applications and tools will also be provided. In next chapter, the objective is to manifest and understand the overviews of current semantic web visualisation theory. The majority of this chapter provides the fundamentals of the components of the information-visualisation process, since we will be analysing the semantic web visualisation tools in Chapter 6.

# Chapter 4

# Visualisation Application for a Semantic Web Framework

We gained an understanding of the process of information visualisation and its components in Chapter 3. In this chapter, we are going to apply these components to a review of existing semantic web visualisation tools. Before discussing a framework for reviewing visualisation tools, this thesis discusses the fundamental knowledge about general desktop-based visualisation and web-based visualisation, especially for semantic web data. The main objective is to study the success and limitations of these two architectures in order to understand the challenges of developing a semantic web visualisation tool, before performing an analysis.

## 4.1 General Desktop-Based Visualisation Tools

A desktop-based visualisation tool can be seen as a standalone application that allows users to extract, transform and present information in a local machine. Basically, desktop-based applications have to be installed before being run by the user. The general advantage of this approach over a web-based visualisation tool is the speed of computation due to the fact that no time need be spent transferring data between the client and server. A rich interface, therefore, can be more easily built. Deploying an application, however, is one of the biggest issues associated with desktop applications. Imagine if a user moves from one workplace to another, s/he has to be concerned as to whether or not the application can be installed.

To understand the current issues regarding the data visualisation of desktop-based tools, this thesis studies four desktop-based visualisation tools. The reasons we selected three of these four systems is because Tableau [61, 62], Quickview7 [63, 64] and Spotfire (Tibco) [60], are commonly seen tools on the internet, and people are always comparing them with one another [99]. In addition, Advance Visualisation System (AVS) [8] is a classic desktop-based visualisation tool that has been rebuilt over the last ten years. The description and further details of each tool are described in the following subsection.

### 4.1.1 Tableau

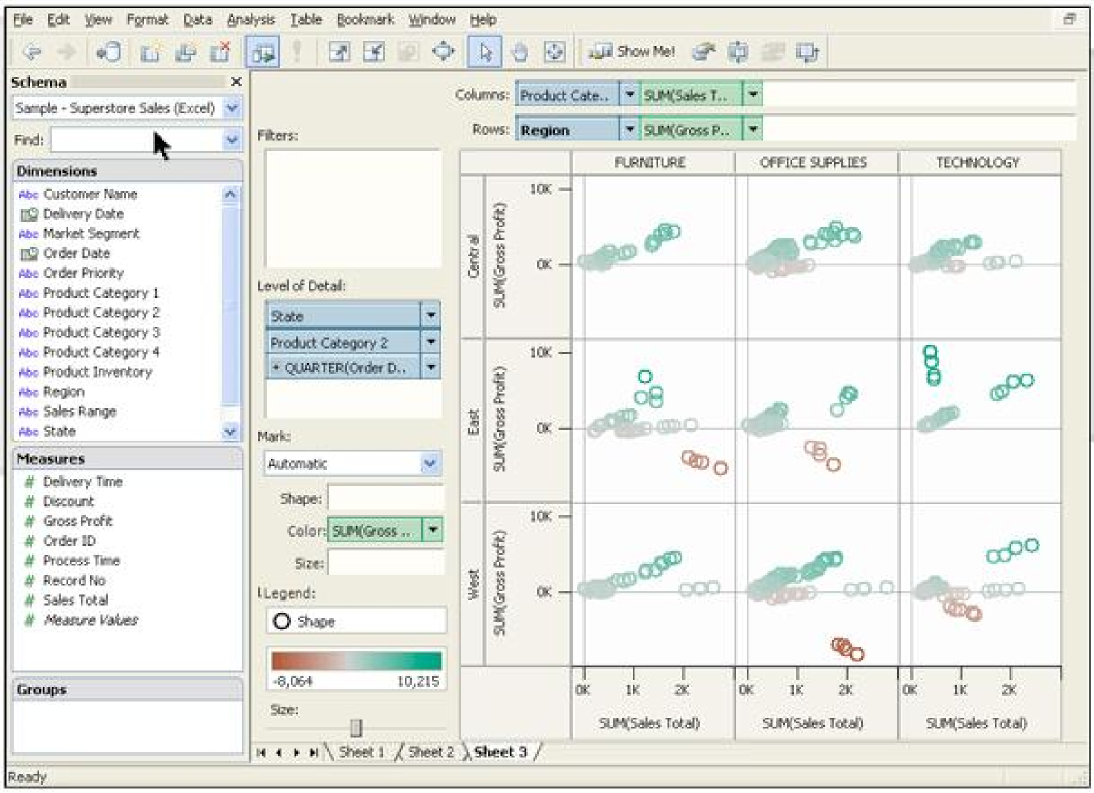


Figure 4.1: Example of the Tableau Desktop Showing Product Categories by Region Plots which are Created by Dragging Dimensions and Measures from the Schema Area, on the Left, over to the Plot Area on the Right [61].

Tableau Desktop is a visual analysis tool that allows users to simply explore and analyse data using drag and drop operations [61]. It allows users to create an interactive dashboard in a few steps. Tableau applies VizQL technology that translates the user's actions into a database query, matches the visualisation techniques and then presents the data result graphically, usually in Data Table format. The approach taken by Tableau provides effective support for the non tech-savvy user who requires a simple dynamic visualisation of the data to a high-level. A simple visual analysis interface is provided which, in addition, combines visualisation and data analysis. Visual analysis means a presentation of information that is intended to support human visual thinking [61]. It also supports interactive exploration and a users' ability to explore a different visualisations of the data.

### 4.1.2 Quickview

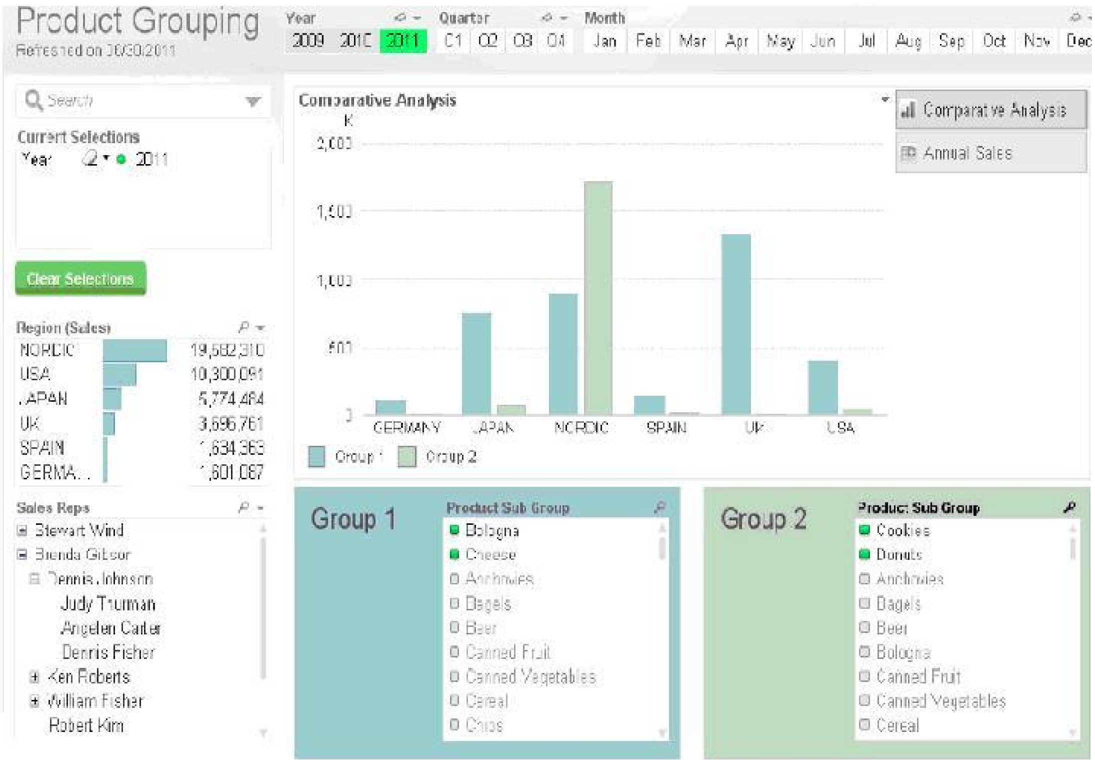


Figure 4.2: Example of Quickview Showing Comparative Analysis between two Product Types Regarding Annual Sales by Country [63]

QuickView has been developed as a kind of business intelligence software, used for conducting searching and analysis of data from any device [63]. It supports a comparative analysis using multiple different views or selections of data, which allow users to easily spot trends, patterns or outliers with regard to the data, as shown in

Figure 4.2. Comparative research [99] between Tableau, Quickview and Spotfire states that Quickview is the fastest tool when performing a drill-down visualisation operation. It is also the best in terms of simplicity and useful dashboard support.

### 4.1.3 Spotfire

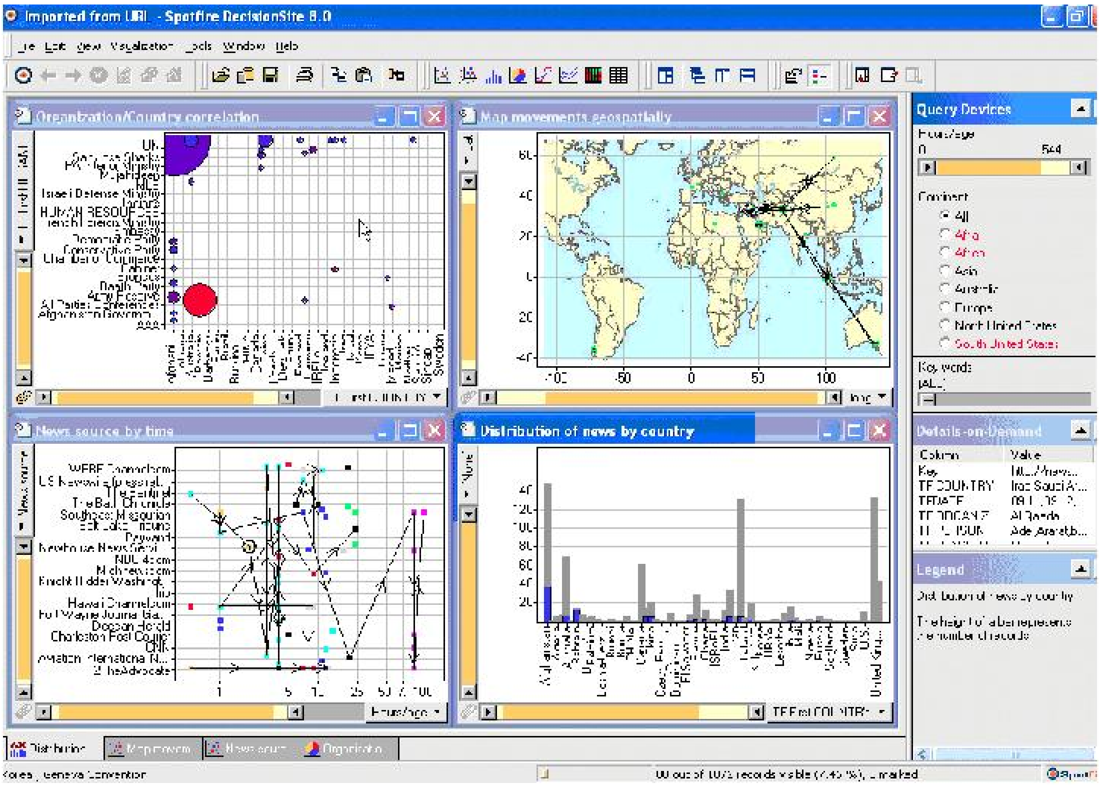


Figure 4.3: Example of Spotfire Showing Four Different Aspects of Data About News [60]

Spotfire's DecisionSite is also a kind of business intelligence software which provides a simple user interface for users to analyse data based on an interactive visualisation approach, as shown in Figure 4.3. Its primary purpose is for use in the business management market. The major advantage over Tableau and Quickview is that Spotfire provides a rich API for developers, which allows them to build a customised application [99]. It has therefore been applied to integrating and visualising data for use as a data-mining tool and as a statistical tool. Spotfire's DecisionSite, like the other two visualisation tools, can generate visualisation based on the imported data [60]. This allows the non-expert user to conduct his/her data analysis without the need for any programming skills.

### 4.1.4 AVS/Express

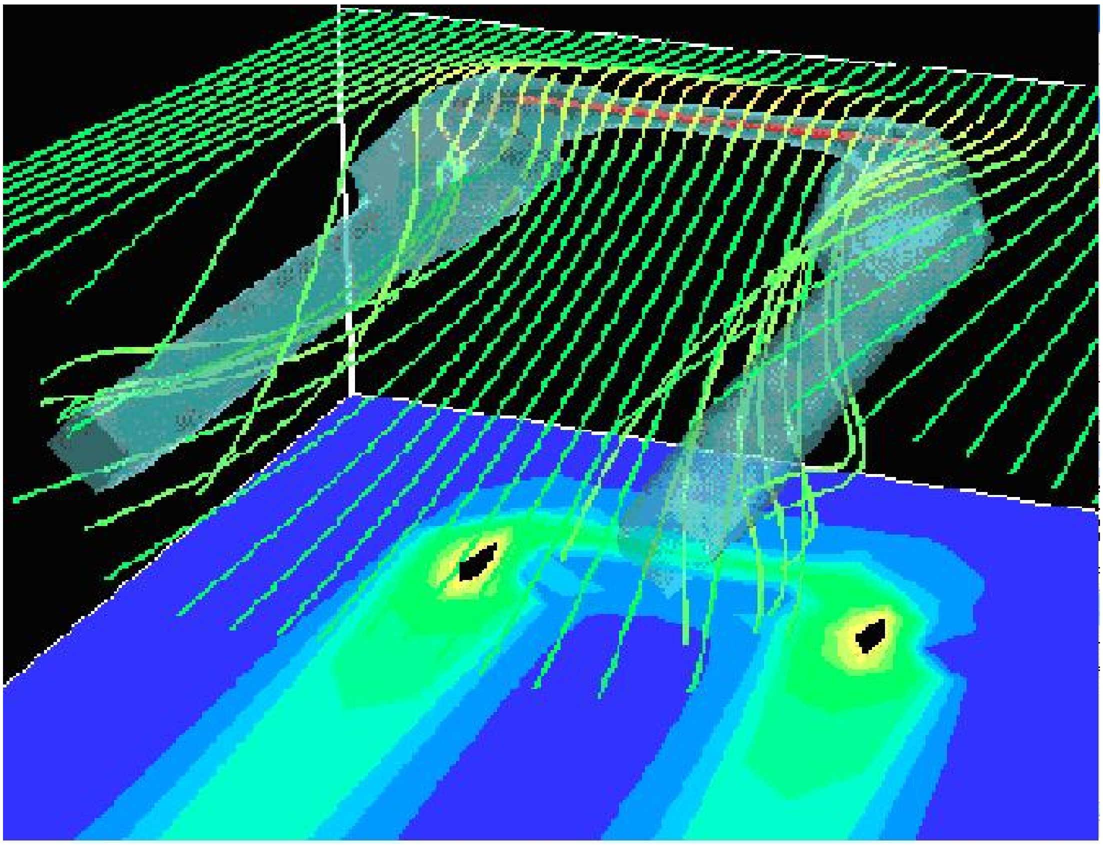

Figure 4.4: Example of AVS/Express Application Showing a 3D Visualisation of the Flow Around A Finite Wing [8].

AVS/Express is a graphical tool that is designed to support powerful visualisation methods [8]. It is normally used in the fields of science, engineering, medicine and telecommunications, and also in business research. Its aim is to analyse and visualise complex datasets. The purpose of visualisation is usually to support simulation and modelling data, as shown in Figure 4.4, which illustrates an example of using AVS/Express to visualise in engineering research. The AVS/Express however, requires a high performance stand-alone computer to support such a complex visualisation.

Based on the study of desktop-based visualisation tools, we observe that, at the current time, desktop-based visualisation tools tend to focus on the development of

Business Intelligence (BI) applications, involving business report information in charts such as bar charts and pie charts. Their purpose is mostly to provide a convenient way for a non-technical user to be able to analyse and visualise data without programming knowledge. Furthermore, these tools are designed to support a user in all the processes of information visualisation, starting with data transformation, visual mapping and view transformation, as discussed in Chapter 3. Although, some desktop-based visualisation tools such as Tableau, Quickview7 and Spotfire (Tibco), have provided a web-based tool version that supports visualisation over the Internet, their architecture still does not support a visualisation of an RDF graph structure. The purpose of allowing the user to visualise data over the Internet, is to enhance the flexibility of the user in terms of accessing results. The comparison details between these tools can be seen at [99].

## 4.2 Web-Based Visualisation Tools

A web-based visualisation tool transforms data into visual views based on client-server architecture. The process of information visualisation is mostly executed on the server-side machine. Users can visualise data concurrently through the web browser, any time and anywhere. As a result, this allows web-based visualisation to be easy to manipulate in order to process information. For example, a developer can add and update the features and functions of a visualisation tool only once on the server, and then any machine is able to run the most recently updated tool.

In this thesis, the web-based visualisation tools are focused on those exploring semantic web data. Sixteen semantic web visualisation tools were selected from those commonly seen in conference papers related to semantic web visualisation: Tabulator [84], DISCO [83], Zigist [85], Open Linked Data Explorer [86], Sindice (SIG.MA) [87], Falcons [88], Information Workbench [92], mSpace [90], Visual Data Web Tool [91], Exhibit [80], Swoogle [89], Explorator [78], RKBExplorer [79], Google Public Data Explorer [67], IBM Many Eyes [77] and DERI Pipe [81]. The further details of each tool are provided in Appendix A.

## 4.2.1 Types of Tool

This thesis presents the tools by grouping them according to their similar functions. This will allow us to narrow our focus with regard to visualisation tools, with the aim of identifying any problems. For the visualisation and semantic web tools that were studied in this chapter, we based the grouping on various tasks, visualisation components, characteristics and the architecture of the data processing that would represent each type of tool, such as a generic browsers or mash-up tools. However, the classification of type is not intended to be exhaustive, because the tool might fall into more than one category. In this thesis, we present four common types related to the semantic web visualisation tools that we studied: generic browser, search engine, mash-up tool and exploration tool. All definitions are provided in the semantic web aspect.

### 4.2.1.1 Generic Browser

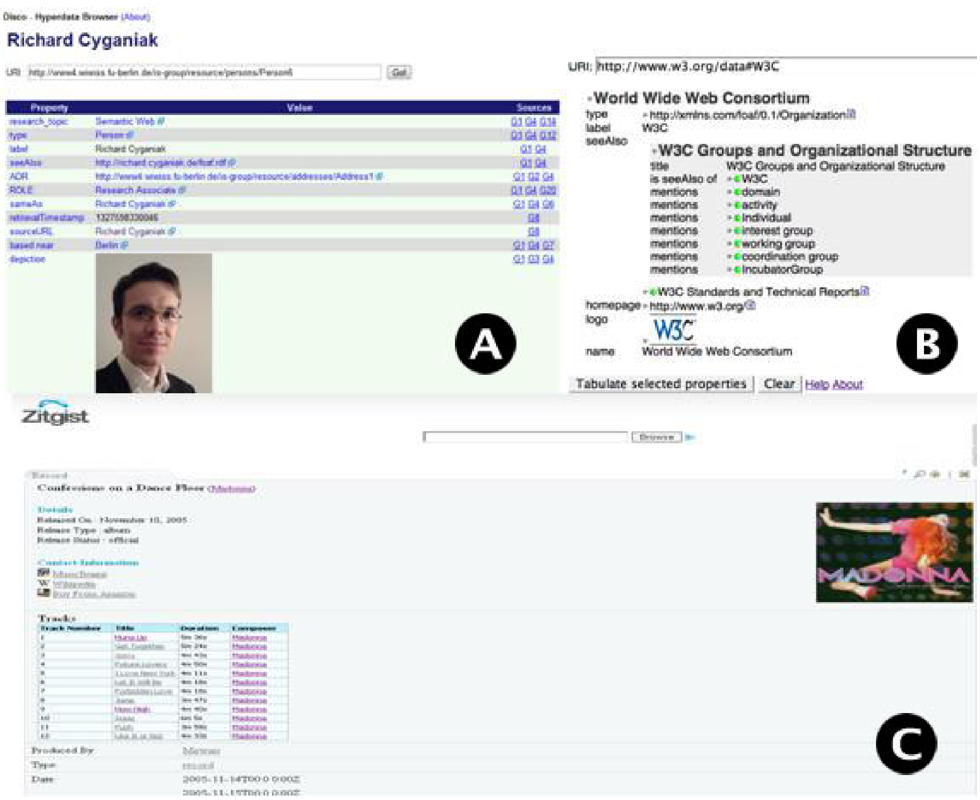


Figure 4.5: Example of Generic Semantic Web Browser; DISCO (A), Tabulator (B), and Zigist (C)

The Generic Browser is a tool that focuses mainly on surveying semantic web content, through the URI links, represented in Linked data, commonly in an RDF format. Users normally scan through a link superficially to gain a basic idea of the contents. It is definitely a kind of web-based tool, which enables users to access information over the internet. The generic browser however does limit to the content in the sense that it limits formats; it allows the presentation of any data that has URI and in a format that it supports, normally a standard format such as HTML, XML or RDF. This kind of tool requires the ability to retrieve, traverse and dereference the URI of the information resources. Mostly, visualisation is text-based, and a rich presentation is rarely supported. Examples of applications that we consider to be generic browsers are provided in the Table 4.1.

#### 4.2.1.2 Search Engine


Figure 4.6: Example of Semantic Web Search Engine; Sigma (A), Swoogle (B), and Falcon (C)

The Search Engine is a tool that allows users to discover information, using a URI, keywords or query-commands. It takes the users' query in the form of keywords. Then matches up with these data that is indexed, before returning the result. In a semantic web context, the search engines is supposed to consider a contextual meaning of data, in order to provide an extra relevant result. Search engines generally require the ability to search, browse and dereference data in almost real time.In semantic web context search engine can be categorized into two type which is one that use semantic content to improve performance (e.g. the use of rich snippets or micro-data) such as Sig.ma and one that provide search over the semantic content, e.g. Swoogle or Falcon. Examples of applications that we considered to be search engines are provided in Table 4.2.

#### 4.2.1.3 Mash-up Tool

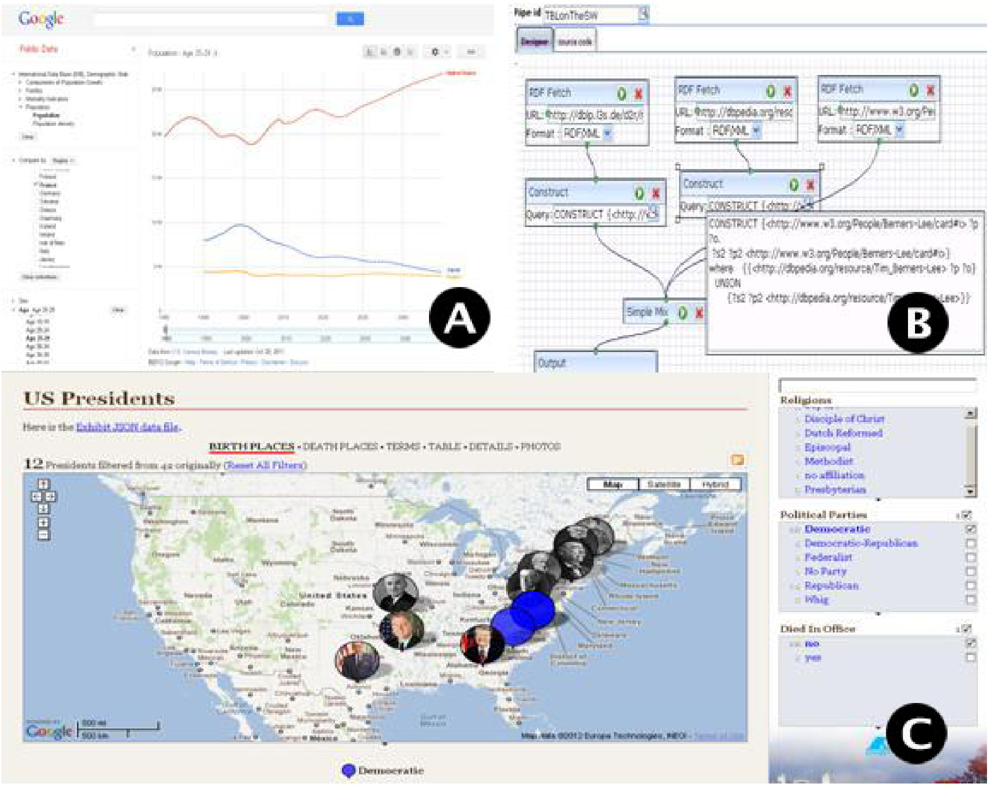


Figure 4.7: Example of a Mash-up Tool; Google Public Data Explorer (A), DERI-Pipes (B), and Exhibit (C)

The mash-up tool is a tool that mainly enhances the visualisation of data by a combination of data, presentation, and functionality in order to create a special tool for a specific purpose. It refers to a tool that is simple and fast when it comes to producing an enriched result. It does this by integrating data sources and using a set of presentations. This type of application allows users to customise a user interface more flexibly than using the other tools. The characteristics of a mash-up tool requires the ability to aggregate data input, combine visualisation and task. Examples of applications that we considered to be mash-up tools are provided in Table 4.3.

#### 4.2.1.4 Exploration Tool

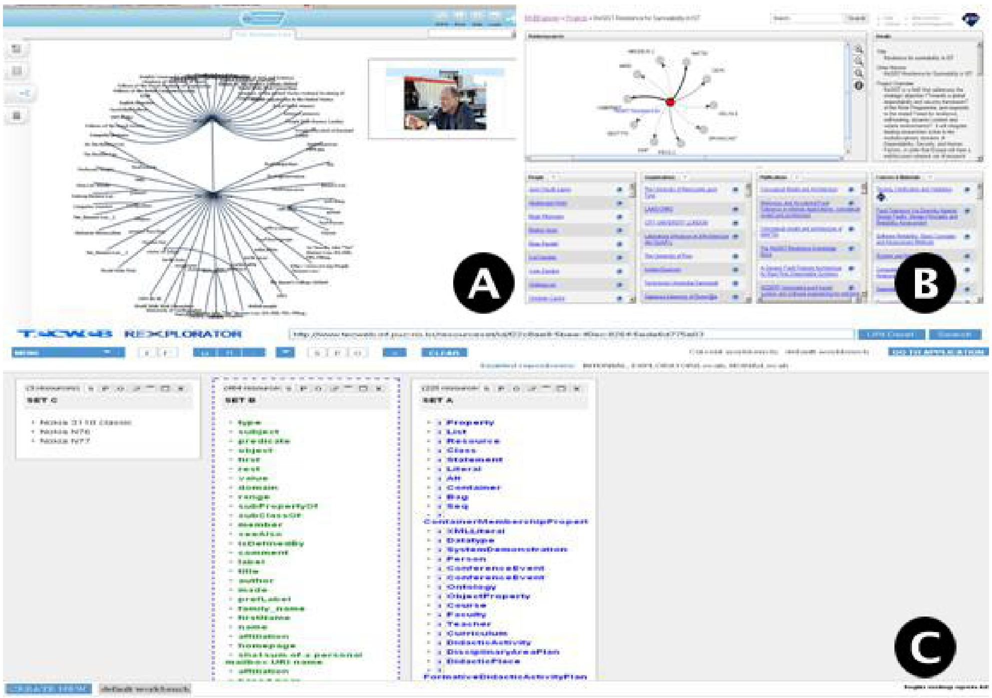


Figure 4.8: Example of Exploration Tool; Information Workbench (A), RKBExplorer (B), and Explorator (C)

The exploration tool is a tool for searching and navigating data in order to discover resources or knowledge. It operates by allowing users to perform an operation on the semantic data to reveal a hidden pattern. The ability of this kind of tool is like a search engine and browser combined, but it needs a set of functionalities that helps the user to scope down data, such as filtering, zooming, etc. Examples of applications that we considered to be exploration tools are provided in Table 4.4 below.

| Generic Browser | URL | Description | Citation |
|---|---|---|---|
| Tabulator | http://www.w3.org/2005/ajar/tab | Tabulator visualises RDF data using outline (tree structure) and table modes that allows users for dereferencing and browsing | [30, 31, 84] |
| DISCO | http://www4.wiwiss.fu-berlin.de/bizer/ng4j/disco/ | DISCO is a simple browser for visualising RDF data using table modes and enabled users to navigate semantic web data. | [83] |
| Zigist RDF Browser | http://zitgist.com/ | Zigist is a semantic web browser that visualises RDF data using template system. It allows users to browse data from many different sources. | [85] |
| Open Linked Data | http://ode.openlinksw.com/ | Open Linked Data is a browser extension, only supporting firefox, safari, and google chrome for browing RDF data | [86] |
| Sindice (SIG.MA) | http://sig.ma/ | Sigma is an application that builds on top of semantic data. It was built by the Sindice plateform. | [32, 41, 87] |
| Falcon | http://ws.nju.edu.cn/falcons/objectsearch/index.jsp | Falcon is a semantic web search engine that allows users to browse RDF data. | [49, 88] |
| Swoogle | http://swoogle.umbc.edu/ | Swoogle is a semantic web search engine that allows users to search semantic web ontologies, documents, terms and data published on the web. | [47, 48, 55] |
| Explorator | http://www.tecweb.inf.puc-rio.br/rexplorator | Explorator is an Exploration tool for discovery semantic web data based on direct manipulation. | [33, 78] |

Table 4.1: Generic Browser Tools

| Search Engine | URL | Description | Citation |
| --- | --- | --- | --- |
| Open Linked Data | http://ode.openlinksw.com/ | Open Linked Data is a browser extension, only supporting firefox, safari, and google chrome for browing RDF data | [86] |
| Sindice (SIG.MA) | http://sig.ma/ | Sigma is an application that builds on top of semantic data. It was built by the Sindice plateform. | [32, 41, 87] |
| Falcon | http://ws.nju.edu.cn/falcons/objectsearch/index.jsp | Falcon is a semantic web search engine that allows users to browse RDF data. | [49, 88] |
| Swoogle | http://swoogle.umbc.edu/ | Swoogle is a semantic web search engine that allows users to search semantic web ontologies, documents, terms and data published on the web. | [47, 48, 55] |
| Explorator | http://www.tecweb.inf.puc-rio.br/rexplorator | Explorator is an Exploration tool for discovery semantic web data based on direct manipulation. | [33, 78] |

Table 4.2: Search Engine Tools

| Mashup Tools | URL | Description | Citation |
|---|---|---|---|
| Exhibit | http://simile.mit.edu/exhibit/examples/presidents/presidents.html | Exhibit is a tool for website author for creating interactive visualisation that can be searched and browsed through facet browsing. | [46, 80] |
| Google Public data Explorer | http://www.google.com/publicdata/home | Google Public Data Explorer is a tool for visualising and exploring data. It provides many visualisation APIs to support users. | [67, 68] |
| IBM Many Eyes | http://www-958.ibm.com/software/data/cognos/manyeyes/ | IBM Many Eyes is a tool for visualising and exploring data. It provides many visualisation Library to support users. | [52, 77] |
| Open Linked Data | http://ode.openlinksw.com/ | Open Linked Data is a browser extension, only supporting firefox, safari, and google chrome for browing RDF data | [86] |
| DERI Pipe (Semantic pipes) | http://sig.ma/ | DERI Pipes is a data mashup tool that provides features for users to integrate between heterogeneous of semantic data sources. | [53, 81, 82] |

Table 4.3: Mashup Tools

| Exploration Tools | URL | Description | Citation |
|---|---|---|---|
| Information Workbench | http://www.fluidops.com/information-workbench/ | Information Workbench is a tool for exploring semantic web data. | [43, 92] |
| mSpace | http://mspace.fm/ | mSpace is a tool for website author for creating interactive visualisation that can be searched and browsed through facet browsing. | [44, 45, 90] |
| Visual Data Web Tool | http://www.visualdataweb.org/ | Visual Data Web Tool is a portal of tool for exploring semantic web data. it composed of gFacet, RelFinder, and tFacet tool. | [91] |
| Exhibit | http://simile.mit.edu/exhibit/examples/presidents/presidents.html | Exhibit is a tool for website author for creating interactive visualisation that can be searched and browsed through facet browsing. | [46, 80] |
| Explorator | http://www.tecweb.inf.puc-rio.br/rexplorator | Explorator is an Exploration tool for discovery semantic web data based on direct manipulation. | [33, 78] |
| RKBExplorer | http://www.rkbexplorer.com/ | RKBExplorer is an exploration tool that explore a specific domain of RDF data. The prototype is subject about the researcher and their publication data. | [50, 50, 79] |
| Google Public data Explorer | http://www.google.com/publicdata/home | Google Public Data Explorer is a tool for visualising and exploring data. It provides many visualisation APIs to support users. | [67, 68] |
| IBM Many Eyes | http://www-958.ibm.com/software/data/cognos/manyeyes/ | IBM Many Eyes is a tool for visualising and exploring data. It provides many visualisation Library to support users. | [52, 77] |

Table 4.4: Exploration Tools

## 4.3 Comparison of Desktop-based and Web-based Visualisation Tools

Some characteristics of both desktop-based visualisation tools and web-based visualisation tools for semantic web data are briefly sketched out in Sections 4.1 and 4.2. One major difference regarding visualisation between these two kinds of tools is the boundary of content. With the desktop-based visualisation tools considered in this study, most of the data for visualisation is imported from a database or spreadsheet such as Excel; the visualisation is more likely to be presenting numerical data that is well defined by the user. From our survey, there is no desktop-based tool that emphasises the visualisation of semantic web data. However, they are tools for publishing ontologies and linked data such as Protege [112]. The web-based visualisation tools are more concerned with presenting a larger body of data. There is also the challenge of how to present this large scale data in a way that makes sense to a user. Furthermore, it appears that due to a limitation in the performance of computation by the server, which is slower than desktop-based visualisation tools [24], most web-based visualisation tools would prefer not to support complex visualisation that requires a high level of computing, such as 3D simulation modelling.

From the purposes of this study, we derived three challenges for the web-based visualisation of semantic web data that are related to the research question mentioned in Chapter 1. These challenges need to be overcome in order to devise a tool for generic visualisation which is effective for a non-expert user. The challenges are as follows:

- The need for dynamic visualisation for presenting different views of the content to a user [39]. This challenge aims to help a non-expert user to gain insight into data by allowing users to change data to a different aspect that is appropriate for them;
- The need for reducing information overload by representing semantic web data in a way that optimise the limited visual space through a visualisation library or APIs [65]. This is to help the user to eliminate unfocussed data, in order for it to be easier to detect and in order to identify data errors such as unexpected data or a missing link;

- The need for an interactive function, to allow users to easily manipulate and organize the information [38]. This is about helping the user to discover knowledge literately through a set of functionalities.

The benefits and limitations of visualisation in both desktop-based and web-based aspects, including the challenges, are considered when designing the evaluation framework for answering the research question. The aim is to study the factors that will support a dynamic and interactive visualisation for semantic web data. We set out to study existing visualisation tools in current use for presenting and exploring semantic web data. The next section will explain the methodology and requirements that are used in order to design an evaluation framework which is used to perform the necessary analysis.

## 4.4 An Evaluation Framework

The evaluation framework was created using the structure of the information visualisation pipeline (Figure 3.1) and its components as mentioned in Chapter 3. We selected sixteen examples of visualisation tools based on the literature available and their accessibility on the internet. The products chosen for evaluation are those commonly seen, as cited by top reviews. We aimed to select only those tools which are actively available and whose prototypes we can obtain. Commercial tools that provide only literature but not a prototype to evaluate have not been included in this empirical study.

The evaluation framework comprises three table schemas shown in Tables 4.5, 4.8 and 4.12 for the assessment of web tools. These table schemas are derived from the general visualisation framework proposed in Chapter 3, and are selected based on the considered criteria for the characteristics of a dynamic visualisation tool [56]. Specifically, these are: a method to extract dynamic information, a method for analysing the extracted data, and a method by which the result of the visualisation are presented. Each schema is designed to measure the functionalities of the semantic visualisation tools, grouping similar tools in order to answer the research question. The attribute in each schema comes from the background study in Sections 3.2.1, 3.2.2, 3.2.3 and the literature related to the existing systems. Although each of the three components has its own attributes and criteria for evaluation, there is a common attribute used in every table schema, and that is a semantic web tool. This refers to the name of each semantic web tool that is used in the evaluation. The rest of this section will explain each component and its attributes in detail.

### 4.4.1 Data Schema

Data is at the heart of the semantic web concept. In a semantic web world, every data object and entity is unique. Moreover, it ties this data together to create a large network of data available over the internet. To visualise and access the semantic web data, it is necessary to study the type of data and the way that applications are used to process such data. The table schema of data, therefore, is designed to answer the question regarding what type of data model is used in a semantic web tool, and how to import a new dataset. The purpose of doing this is to find limitations and areas for improvement in the data representation of the semantic web application.

| | Type of Data support | | | | Back-End Data Processing | | | |
|---|---|---|---|---|---|---|---|---|
| SW Application | Data Format | Input Unit | Category Support | Data Privacy | Access Methods | Importing Techniques | Intermediate Language | is Pre-Processing Required |

Table 4.5: Data Schema

The schema in Table 4.5 divides the framework architecture of data into two parts: data characteristics and back-end data processing. Data characteristics, from Chapter 3, are focused on the characteristics of the data model that users interact with, while back-end data processing is concerned with pre-processing and the internal process of data as an input to the system. The checklist of attributes is explained below:

| Data characteristics | Definition |
|---|---|
| Data Format | This is a type of data model that is used by a semantic web application. |
| Input Unit | This is the volume of input data that the system can process at any time. For example, a single URI, multiple dataset support (Single or Set unit) |
| Category Support | This is a type of data, whether qualitative such as text content, or quantitative such as numerical. |
| Scope of Data | This covers whether an application is designed for the general domain or a specific one. |

Table 4.6: Data Characteristic Terms and Definition

| Data characteristics | Definition |
|---|---|
| Access Methods | This is about what method is used for connecting with semantic web data sources; for example, SPARQL Endpoint, Dump file, APIs. |
| Importing Techniques | This is a technique that is used to import data to an application. |
| Intermediate Language | This is the tool needed for any specific format of input in order to visualise data. |
| Is Pre-Processing Required | This is to understand whether an application requires any pre-processing of data. |

Table 4.7: Back-end Data Processing Terms and Definition

### 4.4.2 Users Activity (Task) Schema

A user activities in the semantic web context are a set of tasks that aim to help them to obtain knowledge. This table schema has been influenced by many information searching and seeking framework theories such as the Marcia Bates theory about information searching and seeking related to human behaviour [15], the eight task features proposed by Ellis, Cox and Hall [18] and the classification scheme by Cool and Berlin [5, 17]. Furthermore, this schema concerns a measuring of functionalities in terms of low-level tasks.

| | Information Gathering Process | | | | Refinement process | | | | | | Support Interaction Process | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Web-based visualisation tool | Searching | Dereferencing | Browsing | Navigating | Zooming | Filtering | Highlighting | Grouping | Ordering | Comparing | Data Transformation | Visual Mapping | Visual Transformation |

Table 4.8: Tasks Schema

The task framework can be categorised into three main groups as shown in Table 4.8. The first group is about the information gathering process in which a user can get details of data related to their demands through basic tasks such as search, browse and navigate. The next group is a refinement process which is about focusing, integrating, organizing and generating information. This will allow the user to gain an overview of the data, or to scope down and focus only on the specific data they need. The last group is a process of information visualisation which is designed for measuring the availability of each process in application. The end task is the process of extracting and verifying a result. This framework is performed in order to discover opportunities for improving user activities as well as techniques which are related to the performance of the discovery process. The details of the sub tasks and their abbreviations are provided in Tables 4.9, 4.10, and 4.11.

| Gathering Techniques | Process | Additional detail |
|---|---|---|
| Searching | Looking for particular data | This is a process of using keywords or commands to retrieve information directly from data sources. |
| Dereferencing | Referring information from an address | URI is an address |
| Browsing | Probing or traversing through data on the web | This task aims in the hope for getting information by travelling through the data without pre-concern about data path. Serendipitous discovery can happen from this task. |
| Navigating | Traversing from one point to another | This process different from browsing by systematical traversing through the connectivity and data path, such as how to get there and which way to go next. |

Table 4.9: Information Gathering Terms and Definition

| Refinement Techniques | Process | Additional detail |
|---|---|---|
| Zooming | Scoping down the data | This process aims to understand detail or expand a view to see an overview of data. |
| Filtering | Eliminating irrelevant information | This process aims to make it easier to extract patterns of data, e.g. facet browsing. |
| Highlighting | Emphasising an object or information | This process intends to make data more outstanding by emphasize the data using size, colour, etc. to make it more visible. |
| Grouping | Putting objects together | This process groups data together based on some shared vocabularies of a property. |
| Ordering | Arranging data in order. | This process arranges data based on the attributes, e.g. name, ages. |
| Comparing | Describing the similarities and differences between multiple objects | This process requires at least two objects. |

Table 4.10: Refinement Process Terms and Definition

| Interaction Process | Process | Additional detail |
|---|---|---|
| Data transformation | Transferring a raw data format into a standard data model,such as a relational data table as RDF. | This process is located between the data and data table stage. |
| Visual Mapping | Mapping between a standard data model and a visual structure | This process is located between the data table and visual structure stage. |
| Visual Transformation | Transferring a visual structure into a visualisation view | This process is located between the visual structure and views stage. |

Table 4.11: Support Interaction Terms and Definition

### 4.4.3 Visualisation Schema

Information visualisation can be seen as the final output of a semantic web application. This component affects human perception directly. An appropriate visualisation therefore, allows users to understand the content more easily. The design of the table schema for the visualisation component has been influenced by Ptzner's [22] and Ben Schneider's work [2], which proposes the framework and concerns for developing information visualisation. Defining the table schema for the visualisation component aims to identify common visualising techniques among the semantic tools. Moreover, this thesis also focuses on the number of graphical representations that are used in each tool. This is to understand how the current visualisation supports a dynamic visualisation.

| | Basic Visualisation | | Positional visualisation | | | | | | | Temporal Visualisation | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| SW Application | Table Structure | Tree Structure | Multiple Axis (2D, 3D) | Area Plot | Bar Plot | Network Plot | Line Plot | Candle Strick plot | Parallel Coordinate Plot | Animation | TimeLine |

Table 4.12: Visualisation Schema

The visualisation component in Table 4.12 is divided into three major attributes - basic visualisation structure, positional visualisation structure and temporal visualisation structure. The attributes in each component derive both from the theory considered in Chapter 3 and the empirical study of existing tools. The details of the sub sections in each category are provided in Sections 4.13, 4.14 and 4.15.

| Basic Visualisation | Definition |
|---|---|
| Table Structure | It represents data as a list of value in a table which is consisted of row and column. |
| Tree Structure | It represents the hierarchical nature of a structure in a tree hierarchical structure. |

Table 4.13: Basic Visualisation Terms and Definition

| Positional Visualisation | Data representation |
|---|---|
| Multiple Axis (2D, 3D) | It represents data more than one dimension in both horizontal and vertical axis, such as chart, 2D graph, etc. |
| Area Plot | It represents quantitative information based on line plotting the area between the axis and line commonly emphasized by colour. |
| Network Plot | It represents the interconnection between a set of objects by node and link. |
| Line Plot | It displays data as a series of data points connected by straight-line segments. |
| Candlestick Plot | It shows an opening and closing value overlaid on top of a total variance. |
| Parallel Coordinate Plot | It visualises multivariate data. An n-dimensional space is represented as n parallel lines. |

Table 4.14: Positional Visualisation Terms and Definition

| Temporal Visualisation | Definition |
|---|---|
| Animation | This is about the representation of moving objects to make the content easier to understand. |
| Timeline | This is data representation in the form of a temporal data structure as well, appropriate for presenting data that occurs over a long period. |

Table 4.15: Temporal Visualisation Terms and Definition

## 4.5 Summary

In this chapter, we give the reasons for the evaluation of the web-based visualisation tools for semantic web data as well as a criteria which will be used as a schema in designing matrix, for classifying category of visualization tool for semantic web data in the next chapter. This thesis also discussed about the designing of an evaluation framework for semantic web visualisation tools.

Having performed the literature review on both desktop and web-based visualisation tools, this research focuses on the web-based visualisation tool rather than the desktop one. The problems regarding the web-based visualisation tool, however, are still too large to be picked up. We therefore group the visualisation tools into four categories: generic browser, search engine, exploration, and mash up tool.

This thesis believes that assessment the visualisation tool through our framework, we can reveal the potential gaps of current visualisation tool as regards with accessing RDF data, visualisation techniques and the user's interaction techniques. From the evaluation framework, this thesis creates assumptions that the visualisation tools for semantic web should support at least basic visualisation structure and basic information gathering techniques. The visualisation tool should be classified into two group which are; the tool that have rich visualisation but complex and difficult for general user and tool that have poor visualisation but simple to use. The next chapter will provide the evaluation result and discussions of these tools in which these result can answer the thesis assumption and objectives.

# Chapter 5

# Evaluation Results and Discussion

This chapter provides an evaluation of the results of the survey of web-based visualisation tools. The evaluation of the application features and capabilities was made manually. Due to this being a capability assessment, the utilization of only one user is enough for the scope of the evaluation. Moreover, in this thesis we have experimented with semantic web tools to evaluate the features of the attributes described in the previous Chapter. The explanation of the assessment contains three major aspects: accessing semantic web data, visualisation techniques, and the user interaction techniques.

## 5.1 The Evaluation Result

The evaluation was done based on three table schemas regarding data, tasks and visualisations, the design of which were explained in Chapter 4. 16 tools are provided for analysis in the left hand side of each table. A cross ('x') indicates a requirement that is available in the tool. The following subsections provide a discussion of the results we obtained from the evaluation of the three tables (Tables 5.1 , 5.2 and 5.3). The results are discussed with regards to the supports that are provided for each type of semantic web visualisation tool, including the detailed analysis of some particular types of tools. This is considered from the point of view of both technical users and non-technical users.

| | Type of Data support | | | | Back-End Data Processing | | | |
|---|---|---|---|---|---|---|---|---|
| SW Application | Data Format | Input Unit | Data Category | Scope of Data | Access Methods | Importing Techniques | Intermediate Language | is Pre-Processing Required |
| Tabulator | RDF | Single | Qualitative | Open | APIs, SPARQL Endpoint | | | N |
| DISCO | RDF | Single | Qualitative | Open | APIs | | | N |
| Zigist | RDF | Single | Qualitative | Open | APIs | | | N |
| Open Linked Data Explorer | RDF | SET | Qualitative | Spec | APIs, SPARQL Endpoint | | | N |
| Sindice (SIG.MA) | RDF | SET | Qualitative | Open | APIs | | | N |
| Falcons | RDF | SET | Qualitative | Spec | Manual | | | N |
| Information Workbench | RDF | SET | Qualitative | Open/Spec | APIs | Upload | | Y |
| mSpace | ANY | SET | Qualitative | Open/Spec | Manual | Hard Coding | | Y |
| Visual Data Web Tool | RDF | SET | Qualitative | Spec | Manual | Hard Coding | | Y |
| Exhibit (tool for SW) | ANY | SET | Qualitative/ Quantitative | Spec | Manual | Hard Coding | JSON Files | Y |
| Swoogle | RDF | SET | Qualitative | Open | APIs | | | N |
| Explorator | RDF | SET | Qualitative | Spec | SPARQL Endpoint | Upload | RDF (N3) Files | Y |
| RKBExplorer | RDF | SET | Qualitative | Spec | Manual | Hard Coding | | Y |
| Google (Public Data Explorer) | ANY | SET | Quantitative | Spec | Manual | Upload | DSPL Files | Y |
| Many Eyes | ANY | SET | Quantitative | Spec | Manual | Upload | Data table | Y |
| DERI Pipe | RDF | SET | Qualitative/ Quantitative | Spec | Manual,SPARQL Endpoint | | | N |

Table 5.1: Data Result: missing data is ‘unknown’

### 5.1.1 Accessing Semantic Web Data

The mainstream users of the semantic web are technical users who can read and understand semantic data such as RDF in its raw form, and they are usually expert domain users who are skilled in their field and can understand a set of semantic web or linked data in text-based visualisation [35]. One reason for this is because, in the past, most semantic web data has been published as a specific domain, especially scientific research such as in the case of bioinformatic. A user requires knowledge about the semantic web concept or at least the vocabulary in the field of the domain to use the tool for discovering information and gaining an insight into it. In the past few years however, general topics such as music, films, people and places are becoming available as semantic web data. For example, BBC Music [93], MusicBrainz [94], Jamendo [95] and DBpedia [37]. This is an opportunity for general users to be able to understand the semantic web content through human readable text-based visualisation. Although the content is more general for them, they still encounter problems accessing data and information through the use of current tools such as the complicate of user layout or structure that represent the RDF. Table 5.1 illustrates the evaluation results of semantic

| | Basic Visualisation | | | Positional visualisation | | | | | | Temporal Visualisation | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **SW Application** | Table Structure | Tree Structure | Graph Structure | Multiple Axis (2D, 3D) | Area Plot | Bar Plot | Line Plot | Candle Strick plot | Parallel Coordinate Plot | Animation | TimeLine |
| Tabulator | x | x | | x | x | | | | | | |
| DISCO | x | | | | | | | | | | |
| Zigist | x | | | x | | | | | | | |
| Open Linked Data Explorer | x | | | x | x | | | | | | |
| Sindice (SIG.MA) | x | | x | x | | | | | | | |
| Falcons | x | | x | | | | | | | | |
| Information Workbench | x | | x | x | | | | | | | |
| mSpace | x | | | x | | | | | | | |
| Visual Data Web Tool | x | | x | x | | | | | | | |
| Exhibit (tool for SW) | x | | | x | | | | | | | x |
| Swoogle | x | | | | | | | | | | |
| Explorator | x | | | | | | | | | | |
| RKBExplorer | x | | x | x | | | | | | | |
| Google (Public Data Explorer) | x | | | x | x | x | x | x | x | x | x |
| Many Eyes | x | x | x | x | | x | x | | | | |
| DERI Pipe | x | | | | | | | | | | |

Table 5.2: Visualisation Result

| | Information Gathering Process | | | | Refinement process | | | | | | Support Interaction Process | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Web-based visualisation tool | Searching | Dereferencing | Browsing | Navigating | Zooming | Filtering | Highlighting | Grouping | Ordering | Comparing | Data Transformation | Visual Mapping | Visual Transformation |
| Tabulator | | x | x | | | | | | | | | x | x |
| DISCO | | x | x | | | | | | | | | | x |
| Zigist | | x | x | x | | | | | | | | x | x |
| Open Linked Data Explorer | x | x | x | x | x | x | | x | | x | | x | x |
| Sindice (SIG.MA) | x | x | x | | x | x | | x | x | x | x | | x |
| Falcons | x | x | x | | | | | | | | | | x |
| Information Workbench | x | | x | | x | x | | x | x | | | x | x |
| mSpace | | | x | x | | x | | | | | | | x |
| Visual Data Web Tool | x | | x | | | x | | | x | | | | x |
| Exhibit (tool for SW) | x | | x | | | x | | x | x | | | | x |
| Swoogle | x | x | x | | | | | | | | | | x |
| Explorator | x | x | x | | | x | | x | | x | | x | x |
| RKBExplorer | x | x | x | x | x | x | | | | | | | x |
| Google (Public Data Explorer) | | | | | x | x | x | x | x | x | | x | x |
| Many Eyes | | | | | x | x | x | x | x | x | | x | x |
| DERI Pipe | | | | | | x | | x | | | x | | |

Table 5.3: Tasks Result

web tools in terms of the data accessing aspects. This thesis has analysed the underlying data in Table 5.1, and has organised that data into a simple statistical graph form presented in Figure 5.1 and 5.2. These two figures present an ability of accessing RDF data on four types of tools presented in Table 5.1 in the form of stacked columns. Numbers, which are located besides the types of tools, show total number of tools in each categorie.

From Figure 5.1 (A), three mashup tools and half of exploration tools are able to present any kind of data including RDF while all data from generic browsers and search engine are shown only in RDF. Although these kinds of tools are not designed specifically to present semantic web data, they can nonetheless present in an RDF data format by user are required to converting them to their standard format, or as described in this thesis, their intermediate language such as DSPL, JSON, or IBM data table. Figure 5.1 (B) illustrates that all selected tools are designed to support a set of input data apart from three of tools in generic browsers, which have been developed

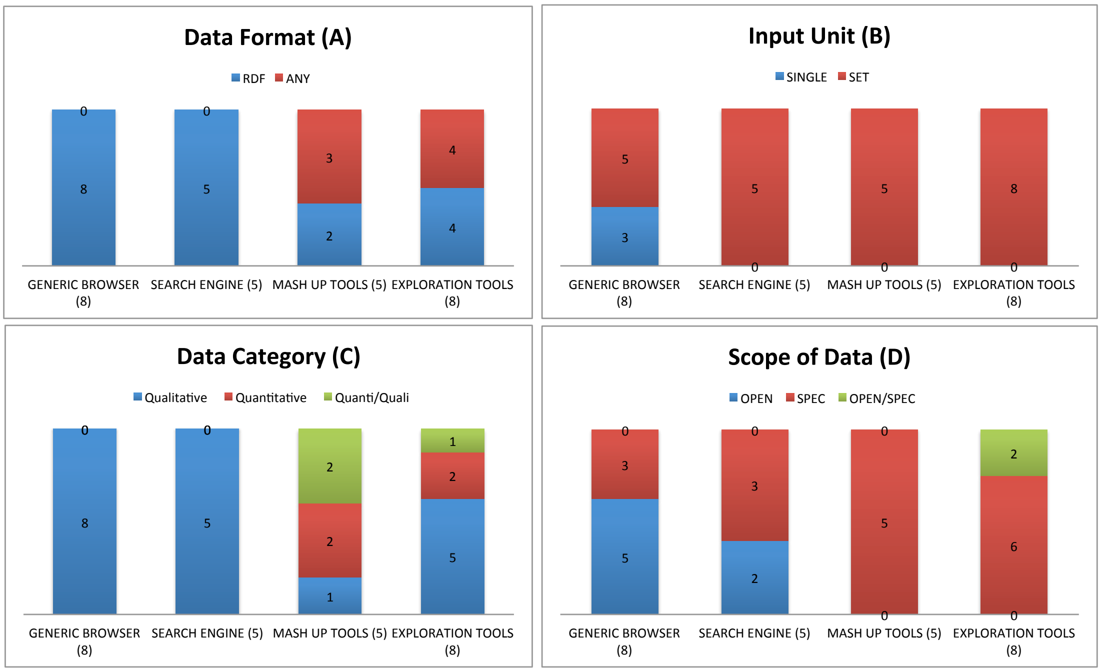


Figure 5.1: Type of Data that Each Category of Visualisation Tool is Supported

to explore only single URIs. The browser tools such as Tabulator, DISCO, and Zigist provide a URI dereferencing method, which is a process of looking up the URI of a semantic web resource and then retrieving information about the referenced resource [12].

These tools access semantic web data through the APIs that were built by the developer. Some of the tools were implemented as an extension of a web browser such as Tabulator in the Firefox-plugin, which is convenient for browsing semantic web data. A user is only required to provide a single dereferencable URI as an input. Although these tools allow simple access to semantic web data using URI dereferencing, it is difficult for a non-technical user to find the URI input that refers to a resource they want. The user needs to understand the basic concept of the semantic web and the content of data. The browser, therefore, is more supportive for technical users who wish to dereference and navigate the semantic web URI in pure RDF triples. Another simple approach to accessing semantic web data is by using semantic web search tools such as Falcon, Swoogle, and Sindice (SIG.MA). These tools remove a technical barrier for the non-technical user by providing a keyword search, which is similar to a web search using a search engine such as Google or Yahoo. Most of them also support a

URI dereferencing function. However, the results of using tools such as a search engine mostly present data as text-based visualizations, in which a user might need basic background knowledge of the semantic web in order to understand. The visualisation issues will be discussed in the next section.

Figure 5.1 (C) presents types of tool that have an ability to manipulate qualitative or quantitative data. The stacked bar shows that only three of the experimented mashup tools and exploration tools support both qualitative and quantitative data. For the presentation of qualitative data, it is obvious that the percentage of exploration tools that are able to manipulate qualitative data is more than twice that of those in mashup tools. Due to a large number of poor type of visualisation tools that are supported quanlitative data (e.g. Generic Browser, Search Engine, and Exploration tools), this can imply that the qualitative data might need a rich visualisation such as a chart or graph in order to be understood by the end user. The quantitative data, however, requires knowledge to organise information before presentation. Furthermore, the generic browser and search engine tools are visualisation tools that support only qualitative data. However, these kinds of tools are helpful for end users that need a simple and quick presentation of semantic web data in a text-based format.

The scope of data in Figure 5.1 (D) shows that most kinds of visualisation tools support an internal data visualisation rather than generic data. From the graph, we can see that two of selected exploration tools support the end user in visualising both specific sets of data and open data. However, there are some other tools such as Exhibit, mSpace and RKBExplorer that are designed for exploring only specific sets of data. Most of them are not concerned about linking data with general open linked data cloud. They are not appropriate for exploring a random set of semantic web data, because to apply it to a different data set, a user is required to have basic knowledge of programming, knowledge of SPARQL queries, data ontologies, (as presented in Figure 5.1 (A)), and also semantic web background knowledge. In addition, both generic data and search engine data are built to support general data such as the open linked data cloud that anyone can use. This can imply that linked data, which is used in the present, is more specific data than that used for specific purposes rather than general purposes.

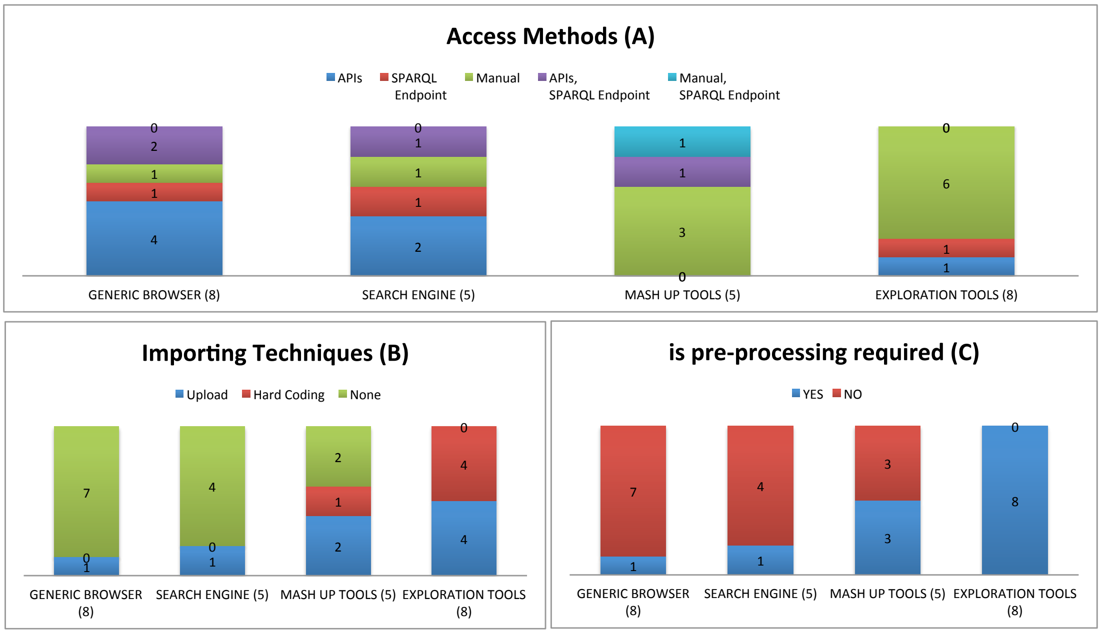


Figure 5.2: Back-End Data Processing that Each Visualisation Tool is Supported

From Figure 5.2 (A), we can see that the generic browser and search engine have an emphasis on providing methods to access RDF data such as APIs or SPARQL endpoints, while exploration tools and mashup tools provide less support in such methods. Moreover, Figure 5.2 (B) and (C) show that seven of generic browser tools and four of search engine tools do not need to import data and pre-process data to the system. This is because these tools are general tools and most of them have been designed to use open linked data and sometimes data has been integrated. six generic browsers and three of search engine tools provide APIs that help non-technical users to access RDF data without technical knowledge.

The exploration tool, however, requires a lot of technical knowledge for the visualisation of RDF data due to six out of eight exploration tools require manual processes for connecting with RDF datasets such as extract dump files, and half of exploration tools require custom programming in order to import data to the system. The upload technique for importing a set of RDF data is outstanding in both exploration and mashup tools in Figure 5.2 (B). Most of these techniques can be found in exploration tools that have mashup ability such as Google Public Data Explorer and IBM Many Eyes which will be discussed in Section 5.2.3. These tools require a process of transformation from semantic web data to their standard format that is used for visualisation.

This researcher believes that the simple way to access semantic web data for non-technical users is by using a keyword search, because the user requires fewer requirements to learn new knowledge in order to access data than with any other approach. Moreover, the simpler and more intuitive user interface can help reduce the cognitive load of users who wish to use such tools. The semantic web search engine is probably the simplest tool for accessing and finding semantic web data. This does not include the consumption of semantic web content from predefined interface websites, for instance BBC Music [93], which has converted the RDF data and layout to make it accessible for non-technical users. However, if we are not concerned about a specific data domain, the hybrid between exploration and mashup tools provide a rich and simple user interface for the user, not only in terms of finding information, but also exploring it.

### 5.1.2 Visualisation Techniques

Having discussed the problem of accessing semantic web data, this section considers the way users consume semantic web data using a visualisation aspect. Although there are approaches for simply accessing semantic web data for non-technical users, the presentation and visualisation of data is another important factor to be considered. The techniques for transforming and organising data before presenting it in an appropriate way for the user, is a key point that effect visualisation. From the review in Table 5.4, we have found that all kinds of visualisation tools support the representation in form of the Table Structure (text-based visualisation), by which the technical user can read and understand. The content in Table Structure has also sometimes been organised into different related sections based on the template of the tool, which allows the user to consume data easily. However, there are some other visualisation tools that provide rich visualisation techniques to support non-technical users such as Multiple Axis (image-based), Bar charts and Line Plots. These techniques can enrich semantic web data by transforming them into a simpler and more understandable form, for instance, images, graphs or charts.

Customised applications such as Exhibit and mSpace, are normally built for the non-technical user. This is because conventionally, the developer has customised these tools based on concerns for the data and the target user. The visualisation, therefore, is more fixed and only suited for certain groups of users. The term fixed here means that there are limited visualisation techniques available and they cannot be extended. The summary of the average number of common visualisations represented in each type of tool is presented in Table 5.4. These numbers are derived from Table 5.2 by counting a total number of tools in different type that support each visualisation techniques.

| | Basic Visualisation | | | Positional visualisation | | | | | | Temporal Visualisation | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Type | Table Structure | Tree Structure | Graph Structure | Multiple Axis (2D, 3D) | Area Plot | Bar Plot | Line Plot | Candle Stick plot | Parallel Coordinate Plot | Animation | TimeLine |
| Generic browser (8) | 8 | 1 | 3 | 5 | 2 | 0 | 0 | 0 | 0 | 0 | 1 |
| Search Engine (5) | 5 | 0 | 3 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| Mash up Tool (5) | 5 | 1 | 2 | 4 | 2 | 2 | 2 | 1 | 1 | 1 | 2 |
| Exploration Tool (8) | 8 | 1 | 4 | 7 | 1 | 2 | 2 | 1 | 1 | 1 | 2 |

Table 5.4: Type of Visualisation Summary

From Table 5.4, we realise that the visualisation techniques that are commonly used in most tools are Table Structure (text-based), Multiple Axis (images-based), and Graph Structure (graph-based). The table structure refers to a text-based visualisation presented in an organised way; it can be in separated sections as a table or just as a list of information. The text-based visualisation however, offers a poor presentation for an overview of semantic web data. It is better to reveal an overview of RDF structure through a Graph Structure or Chart Structure because they can reveal the related information surrounding focused semantic web data or a group of relevant data in the easier way, for non technical savvy to understand, compared to the big-fat graph citemcschraefel. The details of the information, however, still require the use of table structure for presentation.

It is obvious that from Figure 5.3, which is derived converting Table 5.4 into percentages, most kinds of tools have a lack of techniques to present quantitative data. Although there is some hybrid tool (between mashup and exploration, such as Google Data Explorer) that provides techniques to present such data, it appeared from the review that these techniques are not commonly used in visualisation tools for presenting RDF data. Moreover, there are only 20% and 40% of mashup tools provide animation and timeline functions to present data that has the highest support availability among

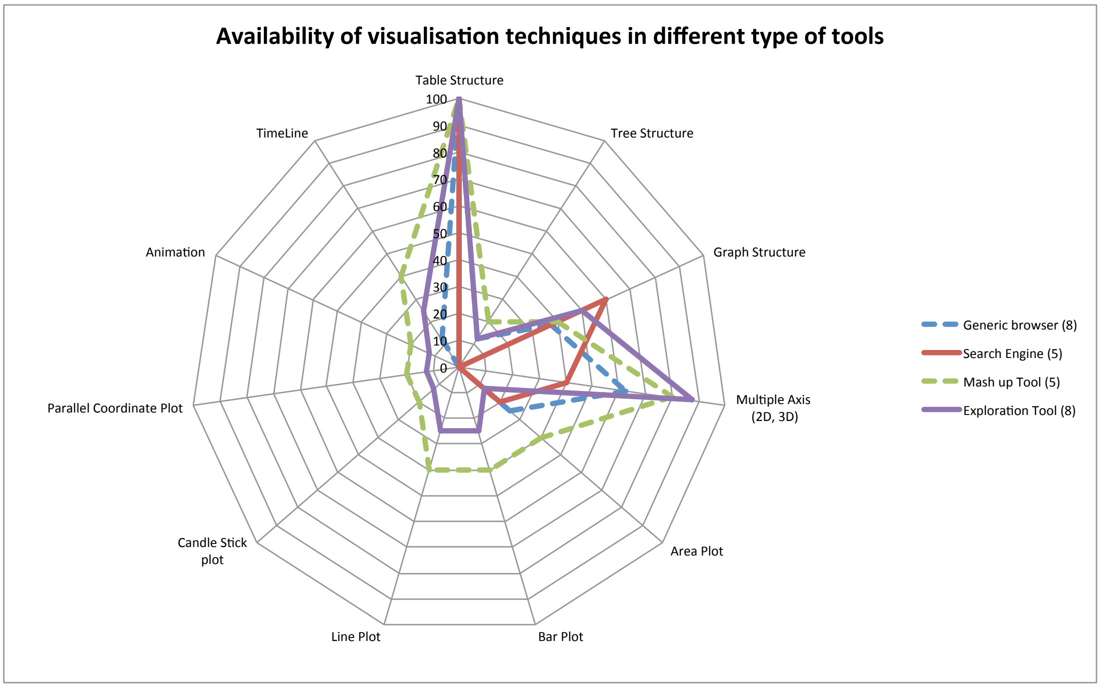


Figure 5.3: The Availability of User's Tasks in Different Type of Tools.

other kinds of tools. These two functions however really help users to understand data in a chronological way.

Technical users need visualisation techniques to discover specific information including checking for errors, and validating data. The visualisation tools that are particularly appropriate for such a user requires the ability to present simple overview and basic details of the data. As we can see from Figure 5.3, the generic browser and search engine offer the same techniques that meet the criteria of the technical user. The search engine moreover specialises in presenting RDF data in the form of a graph. These kinds of tools do not emphasise the visual representation of data, but they are considered as a way to simply access data.

Due to the fact that different users having different ways of understanding the data; designing visualisation for non-technical users is quite a challenging task. Non-technical users basically need a rich visualisation that is easy to understand. From Figure 5.3, we can see that the green line and purple line are structured in the same shape, mashup tools and exploration tools mostly provide the same visualisation techniques that support richer visualisations than the rest. However, a rich visualisation normally has a trade off which requires user effort to manually prepare data as mentioned in Section 5.1.1. This thesis believes that there is no single best visualisation technique to reveal all information. The ideal approach is to allow users to be able to change visualisation techniques based on their need. This approach is relevant to the user's interaction process, which will be discussed in the next section.

### 5.1.3 User Interaction Techniques

The users interaction techniques starts after the end-user obtains results from the system. The reasons for end-users to further discover data is because they might need to gain more knowledge or acquire support and evidence that relates to the theory. Figure 5.4 presents the ratio between the information gathering process and the refinement process in different groups of tools. These Figure are derived from the Table 5.5 by summing up number of tools,in each type, that presented in the sub-task of information gathering process and refinement process respectively before converting them into percentages. We therefore get a big picture of how the tools support these task process. We can see that the characteristic of generic browsers and search engine have a higher percentage in supporting the information gathering process, which is 66% and 57%, than the refinement process, which only accounts for 34% and 43%. The pie chart for generic browsers and search engines, as shown in Figure 5.4, is quite similar to each other. This similarity may come from the property they both have, by which most search engine tools have the same functionalities as the generic browser. On the other hand, the functionality of mashup tools and exploration tools are focused the opposite way round, which is concerned more supporting with the refinement process, which is 76% and 64%, than the information gathering process, which accounts for 24% and 36%. It can imply that these two types of tool aim to allow users to manipulate a set of data rather than just finding the information.

From the evaluation results in Table 5.5, we can see that all of search engines and browsing tools support browsing activity. However, only two of mashup tools and six of exploration tools support browsing activity. This is because there are two hybrids of visualisation tool, which are Google Public data explorer, and IBM Many Eyes that do not provide functionality for browsing RDF data. The visualisations of these tools are present through a pre-defined template that does not fully support browsing or dereferencing RDF data on its own. However, to allow users to search, browse or navigate through these tools, it requires much pre-coding work with regards to data

| | Information Gathering Process | | | | Refinement process | | | | | | Support Interaction Process | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Type | Searching | Dereferencing | Browsing | Navigating | Zooming | Filtering | Highlighting | Grouping | Ordering | Comparing | Data Transformation | Visual Mapping | Visual Transformation |
| Generic browser (8) | 5 | 8 | 8 | 2 | 2 | 3 | 0 | 3 | 1 | 3 | 1 | 4 | 8 |
| Search Engine (5) | 5 | 5 | 5 | 1 | 2 | 3 | 0 | 3 | 1 | 3 | 1 | 2 | 5 |
| Mash up Tool (5) | 2 | 1 | 2 | 1 | 3 | 5 | 2 | 5 | 3 | 3 | 1 | 3 | 4 |
| Exploration Tool (8) | 5 | 2 | 6 | 2 | 4 | 8 | 2 | 5 | 5 | 3 | 0 | 4 | 8 |

Table 5.5: Type of Task Summary

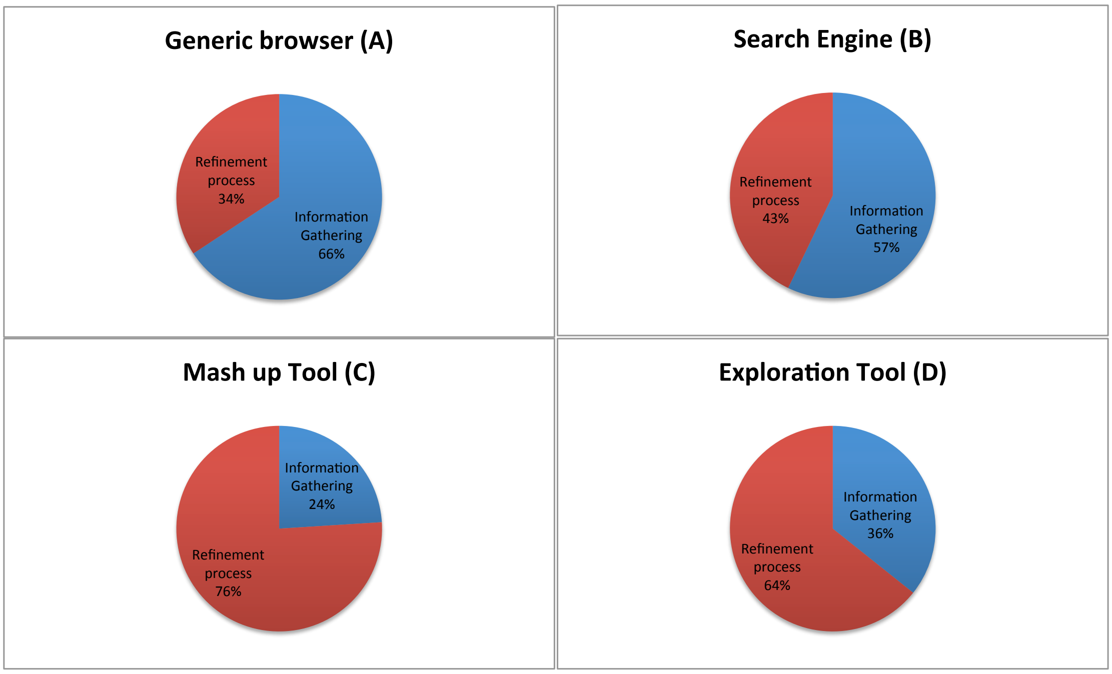


Figure 5.4: The Overview of Type of Interaction in Different Group of Tools.

integration and visualisation mapping. There is one kind of mashup tool that does not have the ability to browse data and that is the DERI Pipe. It has been designed mainly for integrating data from many sources. The functionalities between generic browsers and search engines are quite similar, which has more common functionality in dereferencing techniques and browsing techniques. However, the refinement process, such as grouping and filtering techniques, are more readily available in search engine tools.

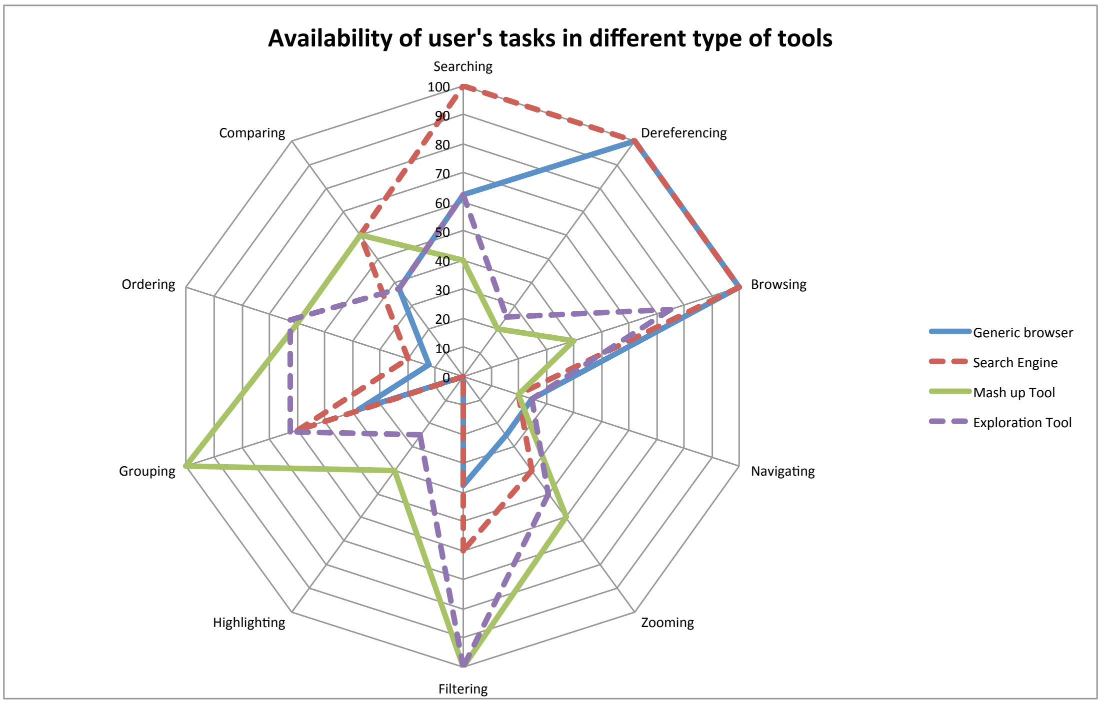


Figure 5.5: The Availability of User's Tasks in Different Type of Tools.

In term of functionalities, we believe that the Generic Browser and Search Engine tools are appropriate for both technical and non-technical users. This is because these kinds of tool provide a simple and quick functionality to gather RDF data such as searching, browsing and dereferencing as presented in the Figure 5.5, which is derived converting Table 5.5 into percentages,. The Mashup tool also provides basic information gathering function which are searching and browsing techniques for both technical and non-technical user. However, this kind of tool less support dereferencing techniques than the other. The end-users sometime require functionality to organise data to gain more understanding. From the Figure 5.5, the line of Exploration tool is similar to Search Engine tool, but it more emphasises on a refinement techniques such as filtering and ordering techniques. The functionality of these kinds of tool is more likely to support both technical and non-technical users.

## 5.2 The Detailed Analysis of the Visualisation Tools

The analysis result in Section 5.1 presents facts that support our assumption, as stated in Chapter 1 in this research investigation. The goal of this thesis is to find out the potential gaps and obstruction of current visualisation tools that can be filled to create an interactive framework for a generic semantic web visualisation tool. In order to identify the problem in detail, this thesis reduces the scope to discuss two types of visualisation tool: exploration and mashup tools. The potential and constraint in the study of these tools are provided at the end of this chapter. The reasons for narrowing the scope are because they fit to the problem domain that is interested in a rich and dynamic visualisation of generic RDF data. These tools are more to support end-user cognition and to gain insights and knowledge from exploring and visualising information compared with the other two categories of tool which are: search engine and generic browsers.

### 5.2.1 Scope of the Exploration Tool in a Process of Visualisation

The exploration tool is an interactive visualisation tool that allows users to explore and visualise data. By exploring, we mean the capability for searching and browsing around the information space for the purpose of discovering information and resources. This section will explain the scope of the exploration tool over a process of information visualisations, as mentioned in Chapter 3. The key characteristic of these kinds of tool is about supporting a direct manipulation at the view stage. The scope of the exploration tool is that it covers the process of visual mapping and view transformation, as presented in a process of information

From the evaluation results discussed in Section 5.1, we can see that exploration tools support the feature of importing data for exploration in both uploading and hard coding. Furthermore, the majority of tools that we have studied, such as mSpace [90], RKBExplorer [79], and visual data web tools (gFacet, relFinder, semLens), are simple exploration tools that can perform common tasks such as browsing and navigating.

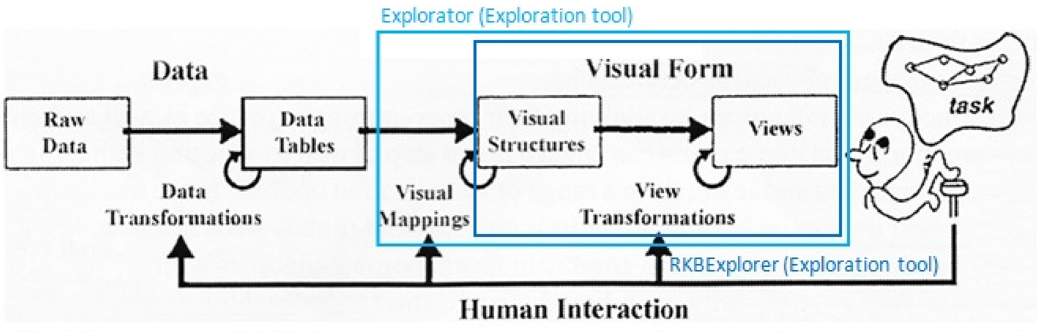


Figure 5.6: The Scope of Visualisation of the Exploration Tool

They also require a hard coding approach for mapping between data tables and the visual structures. For example, to explore data through mSpace, users must understand the type of data and visualisation structures of mSpace, for instance the facet browsing structure, before writing code for mapping. On the other hand, some tools such as Explorator [78] are built with some special tasks that support a process of exploration such as direct manipulation. This allows users to continuously manipulate data to discover the object of interest. Moreover, it allows users to import data to the tool without the need for understanding the back-end process.

We selected Explorator [78] and RKBExplorer [79] as examples of exploration tools to study further and to be able to compare their architecture with a process of information visualisations. The reason that we performed an analysis on these tools is due to their characteristics, which are also common in other tools and also because of their interesting features, such as direct manipulating and graph-based visualisation. In this section, however, we do not mention tools that are classified into both exploration and mashup categories, such as Google Public Explorer [67] and IBM Many Eyes [77]. These tools will be discussed later in section 5.2.3.

#### 5.2.1.1 Explorator

Explorator [78] is a probing tool, which is used to explore RDF triples. Explorator supports to have many datasets, which allows a user to import them from any RDF triples data sources. Furthermore, it represents data in a table format by using the Ordered Tree structure to illustrate between class, subclass, and property. Besides that, the user interface allows the user to move window panels everywhere. Explorator is designed for limited users who have knowledge about technical aspects and the subject domain. The limitation of this tool is that its visualisation results are not rich enough for some types of RDF data, such as a series of data. It also does not support the presentation of quantitative data.

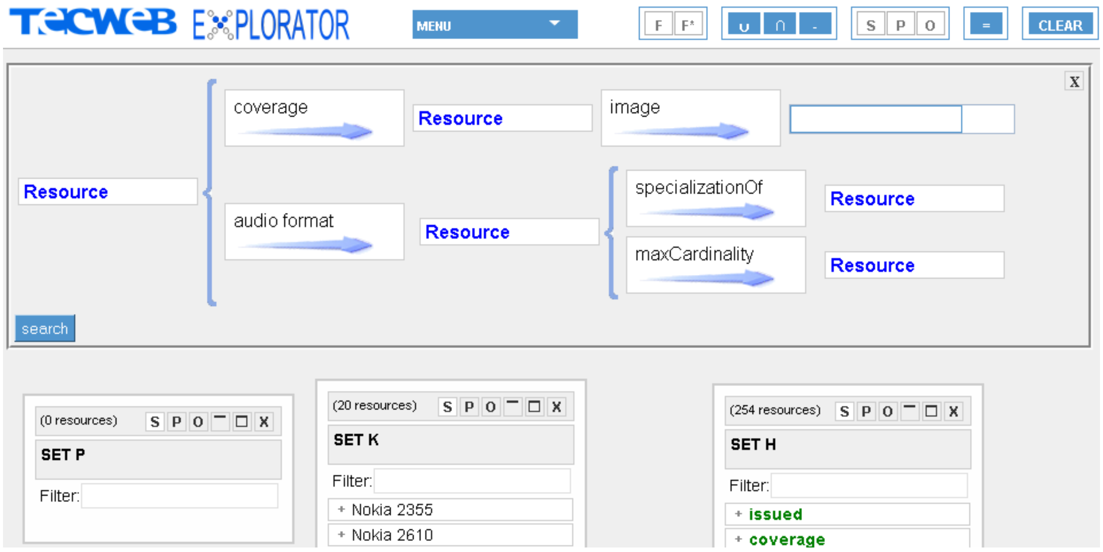


Figure 5.7: The User Interface of Explorator [78]

#### 5.2.1.2 RKBExplorer

The RKBExplorer [79] is a domain specific semantic application (in an area of academic research activity with an emphasis on software engineering). To represent the RDF data, RKBExplorer uses the simple Un-ordered Graph to represent connections between the related information. The graphical representation of RKBExplorer is a fixed image without an animation. It is simply designed and easy to understand the meaning of the picture. The system is designed for both goal directed and opportunistic users. It is more likely designed for novice users rather than '*geeks*' because the user interface is so simple and does not allow the user to interact much. The limitation of RKBExplorer is that it is a domain specific application, which does not allow users to explore other datasets. Users have to rewrite the entire code to do so.

#### 5.2.1.3 Potential and Constraints

The nature of an exploration tool emphasises rich user tasks for manipulating data, as presented in Figure 5.5. They use the refinement operation such as grouping, filtering,

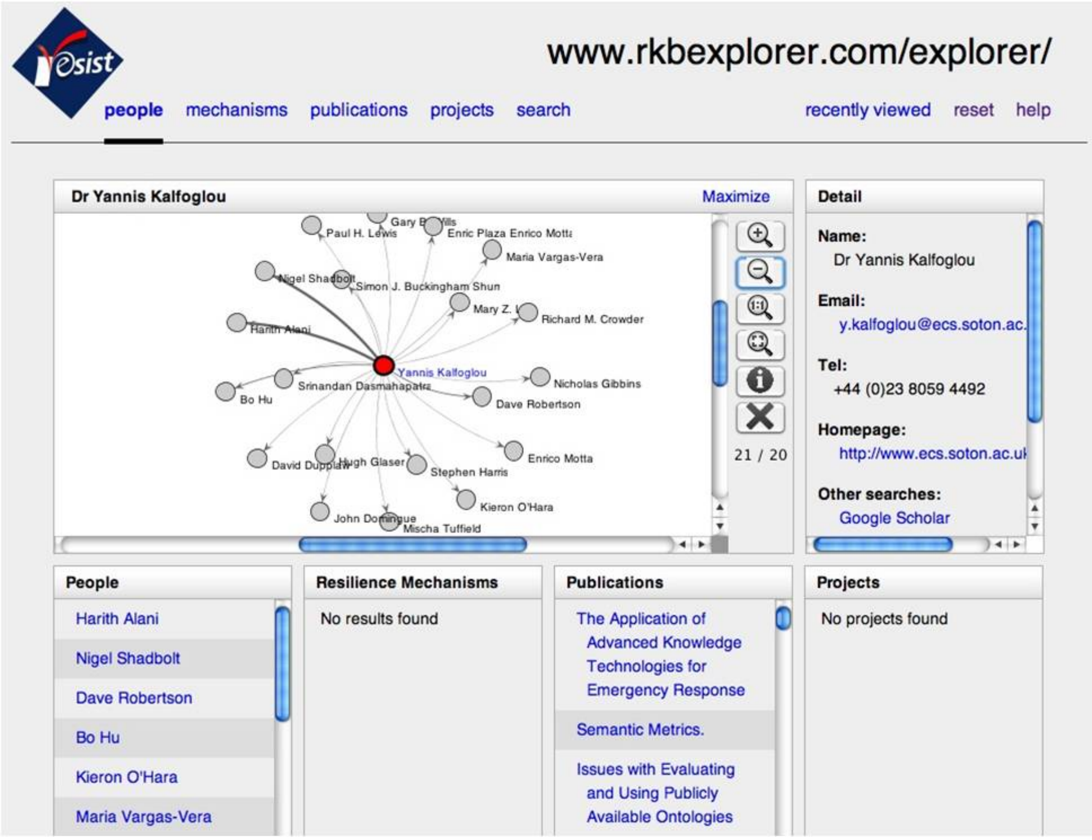


Figure 5.8: The User Interface of RKBExplorer [79]

or ordering to manipulate and discover information and resources. The current state of the tool provides a friendly user interface for users to manipulate data rather than a custom programming approach. This allows users to easily manipulate data at the view stage.

The visualisation, however, is a main constraint for these kinds of tools. Our survey (Table 5.2) illustrates that these tools have insufficient visualisation compared with the other types, such as a mashup tool. Although not many tools require rich visual representation, sometimes it might help users gain more understanding. For example, imagine that we need to find the number of researchers in each year who were involved in a semantic web visualisation project for ten years. Using an exploration tool, the result you get is mostly presented in text and table format, which will take time to read and understand. However, if you can present it in a bar chart, this will allow users to understand the overview of the result quickly.

The other constraint is that users are required to perform the transformation process by themselves. The exploration tools that we have studied only support users in the visual mapping and transformation process. Users however, have to extract, transform, and load data into the tool manually. Although there are some open source scripts that are designed to help users with that process, they have to find this out themselves. This will take non-technical users a while before they can explore the data.

### 5.2.2 Scope of the Mashup Tool in a Process of Visualisation

A mashup tool, as mentioned in the previous Chapter, is a tool that uses and combines data, visualisation, or tasks from two or more sources to build new specific services. The key characteristic of a mashup is combination and aggregation. In this research, we discuss two types of mashup tools, namely data and visualisation mashups. Task mashups, however, are counted as embedded components in both data and visualisation mashups. The scope of a mashup tool to cover all processes of information visualisation is presented in Figure 5.9.

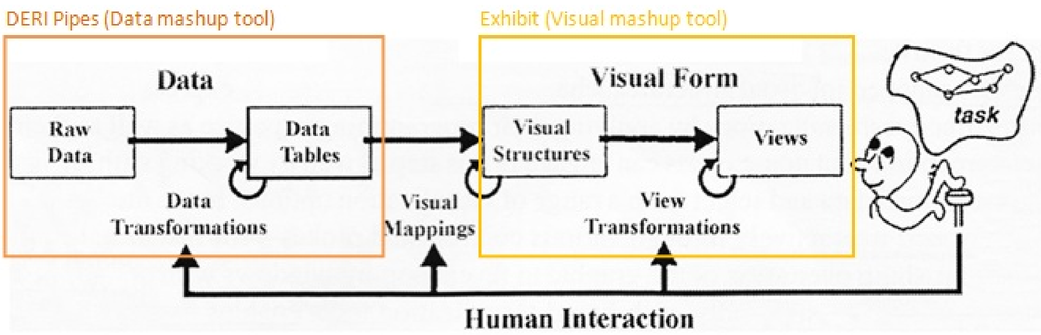


Figure 5.9: The Scope of Visualisation of the Mashup Tool

From the evaluation results discussed in Section 5.1, the common characteristics of a mashup tool is the combination of multiple visualisations, functions, or data. This can imply that a mashup tool can fit into more than one category of visualisation tool. For example, Open Linked Data Explorer is a search engine tool. However, it supports multiple visualisations for browsing results, which can be counted as a search engine tool that has the ability of visual mashup. Exhibit [80] and DERI Pipes [81] have been chosen as examples of mashup tools for analysing the scope of mashup tools in a process of information visualisation; this is due to the fact that both of these tools have an outstanding mashup characteristic such as multiple visualisation and operation. The scope of Exhibit and DERI Pipes is illustrated in Figure 5.9.

#### 5.2.2.1 Exhibit

Exhibit [80] is a lightweight data publishing framework that provides rich visualisations (e.g. timeline, map) and tasks (e.g. sorting, filtering). The reason Exhibit (Figure 5.10) does count as a mashup tool is because it provides a set of APIs and tools for supporting limited users, who only know basic HTML, to build their own visualisation through the rich visualisation and functions. The benefit of using Exhibit is that users do not need to install any applications. However, to use Exhibit to represent RDF data, users must perform data transformation and visual mapping processes themselves due to Exhibit only providing a certain amount of APIs and libraries that support users to visualise data. This set of APIs helps users a lot in constructing a rich user interface; however, the users need to have some basic knowledge, for instance of basic HTML.


Figure 5.10: The User Interface of Exhibit [80]

#### 5.2.2.2 DERI Pipe

DERI Pipe [81] is a tool for integrating heterogeneous data sources based on the semantic concept, which is inspired by Yahoo Web Pipe. Instead of focusing on the RSS feed of the web resources like Yahoo Pipe, this concept concerns integrating RDF data from various sources through the SPARQL query and a set of special operators (e.g. Fetch, Construct, and Mix). To build a semantic web pipe concept, there is an example of a tool, which is called the DERI Pipe (Figure 5.11).

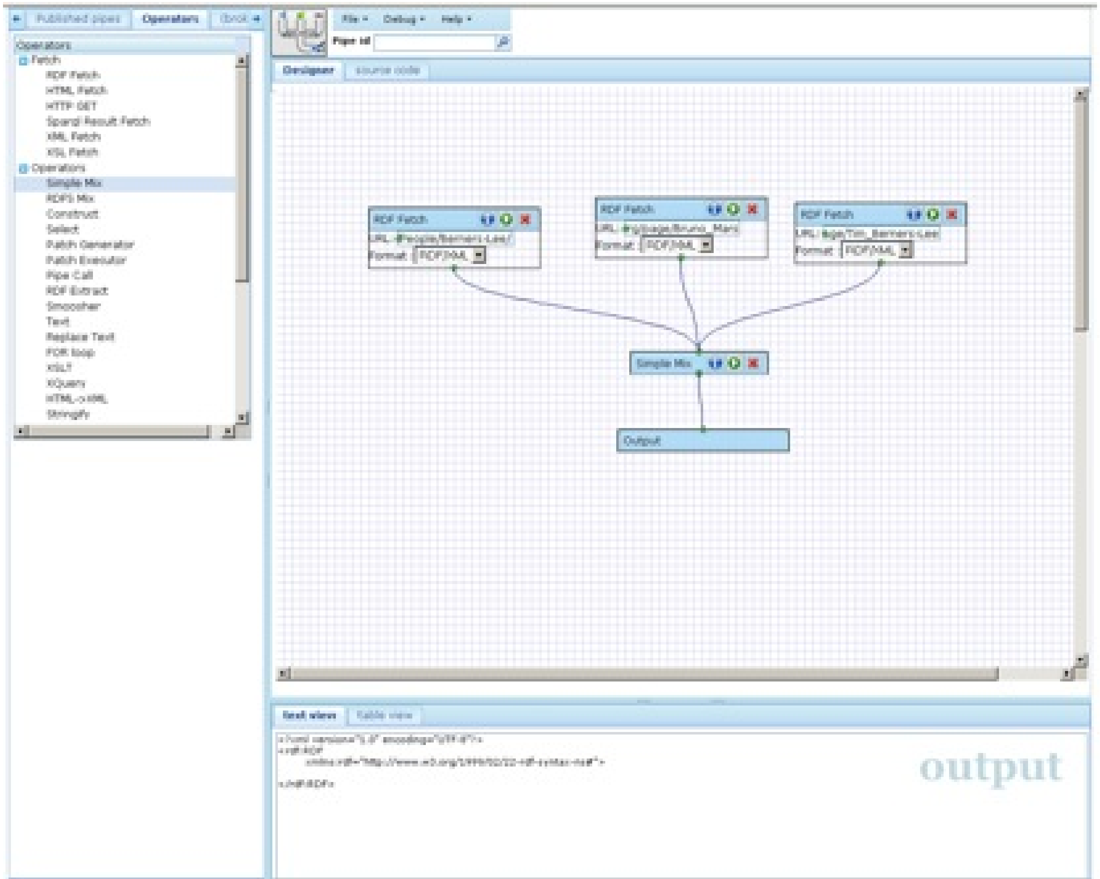


Figure 5.11: DERI Pipe User Interface [81]

The DERI Pipe aims to support a process of transformation and mashup semantic data such as RDF, XML, Microformat, and JSON. Moreover, it produces and outputs streams of data that can be used in other applications. In addition, DERI Pipes provide an end user GUI for manipulating data sources and browsing results, as shown in Figure 5.11. It allows users to select operators from the operator lists, such as RDF fetch, Simple Mix, and then users just put in the data source and link between the operators. The visualisation result of the DERI Pipe can present both text and table views, as shown in Figure 5.12.

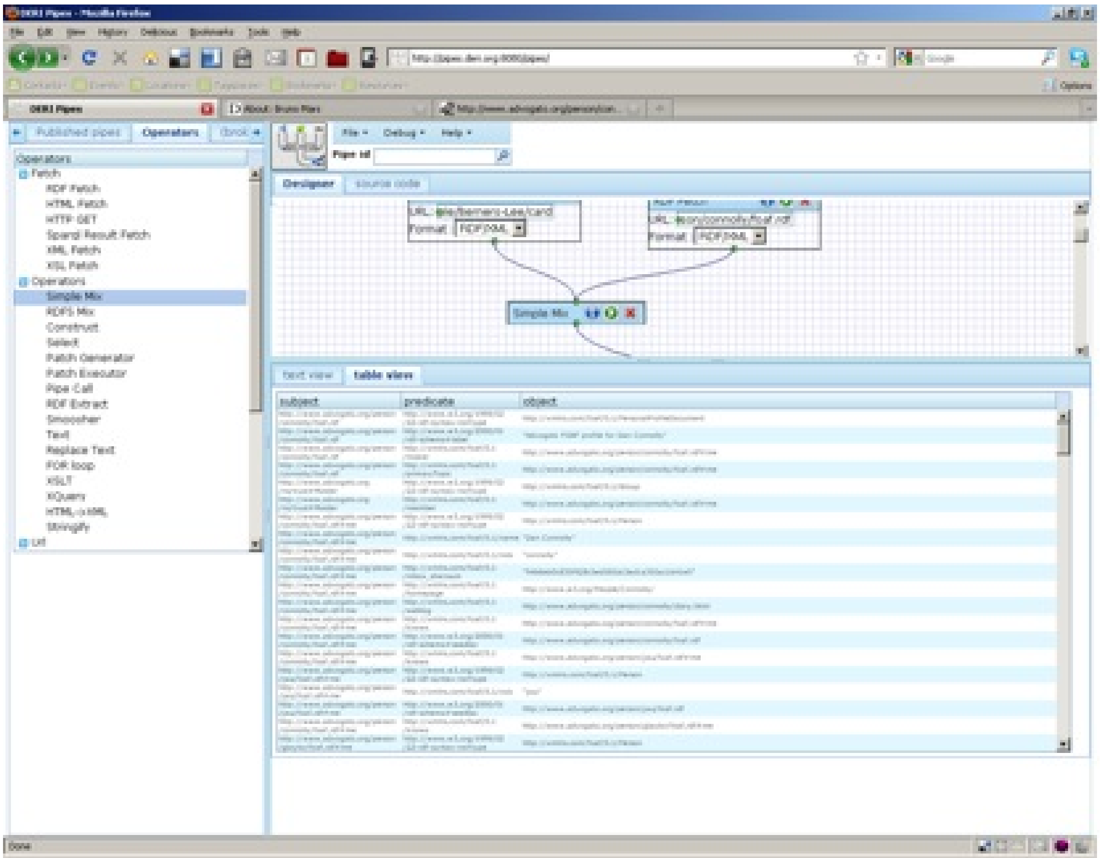


Figure 5.12: Visualisation Result in the DERI Pipe [81]

The DERI Pipe tool [81] focuses on a stage of data transformation rather than the visual mapping and view transformation stages, while Exhibit is interested in the visual transformation process. It aims to integrate data and allow other applications and users to use the results in the way they want. For example, having aggregated data through the DERI Pipe, we can visualise the results of data through the Exhibit APIs.

#### 5.2.2.3 Potential and Constraints

The nature of mashup tools highlights the aggregation of data, visualisation, and functions. This research concerns the mashup tool that provides the visual component and functions that support users to build their own visualisation. The current state of this kind of tool is a visual programming interface, such as DERI Pipe. This allows users with limited skills, who only know basic HTML, to mashup data easily. Regarding the scope of our study, however, there is a lack of web-based visual programming that supports the user to customise visualisation on the fly. We propose that by combining both tools, DERI Pipe and Exhibit, results in a reduced process of data and visual transformation; however, there is still a need to perform a visual transformation.

### 5.2.3 Scope of a Combination Between Exploration and Mashup Tools in a Process of Visualisation

From the study of two types of visualisation tool, exploration and mashup tools, we can see that some tools fit into both categories, such as Google Public Data Explorer [67] and IBM Many Eyes [77]. These tools are significant since they have common functionalities; for example, rich visualisation and direct manipulation. In this study, we studied the Google Public Data Explorer's process of visualisation and compared it with the individual processes of an exploration and mashup tool that we mentioned before. The results of the experiment on this tool are provided in the following subsection.

Although the Google Public Data Explorer tool fits into both exploration and mashup categories, this kind of tool still only covers the visual mapping process and the view transformation, as shown in Figure 5.13. It provides the intermediate language to support users to perform a visual mapping process. Users do not have to hard code to map the data with the library of the visual structure; they just need to transform data into the intermediate language, which is easier. The details of the application and the gap are presented in the following section.

#### 5.2.3.1 Google Public Data Exploration

The Google Public Data Explorer [67] is a kind of exploration tool that allows users to explore and visualize data through a specific intermediate language. It sits between data and visualisation. It provides a pre-defined set of visualisation techniques. To explore a dataset, a user must at least have knowledge of the domain that they work within and understand the basic XML structure. Users have to prepare their dataset in the Dataset Publishing Language (DSPL) as the intermediate language, which is composed of CSV and XML files, as an input for visualizing. Using the intermediate language reduces a lot of time and complexity for non-expert users, who are not

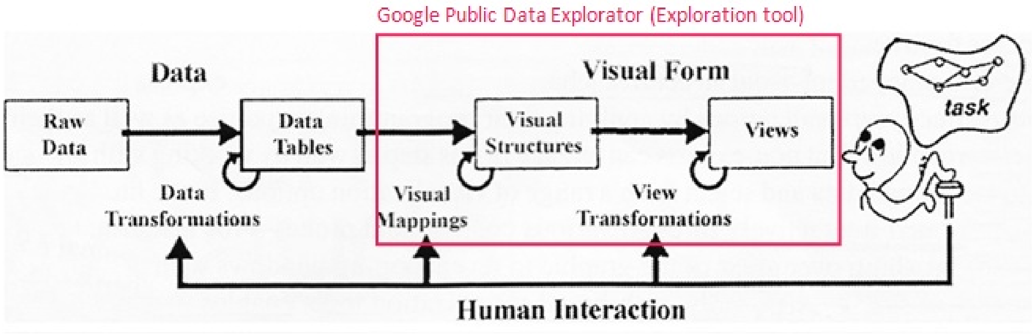


Figure 5.13: The Scope of Visualisation of both Exploration and Mashup tool

familiar with custom programming, to visualise data.

The difficulty of using this approach, however, is the process of transformation from raw data into the DSPL format (Figure 5.15). This tool does not support the transformation process. For example, the users have to perform a query themselves to extract and retrieve the information that they want. However, the Python script, desplgen.py, which Google provides, offers an automated data conversion for generating the XML schema from the CSV format. This, however, does not completely do the job. Users still have to put in effort to amend the results for visualisation. Visualising data through Google Public Data Explorer therefore requires some basic programming skills, such as SPARQL query and XML. In addition, due to the fact that the visualisation methods in the Google Public Data Explorer are predefined, the developer cannot add new visualisation techniques by themselves. This leads to a challenge in the research of a new approach for finding a flexible dynamic visualisation development framework.

## 5.3 Summary

This chapter provides the results and discussions as regards with an assessment of 16 semantic web visualisation tool as well as presenting the common functionalities of the visualisation tools for semantic web data. This evaluation however is not the only measure of the availability of functions in each tool. We also aim to discuss how well these functionalities support the non-technical users. We analyse the tools that support such a user by following three aspects. These are: (1) accessing semantic web data, (2) visualisation techniques, and (3) user interaction techniques.

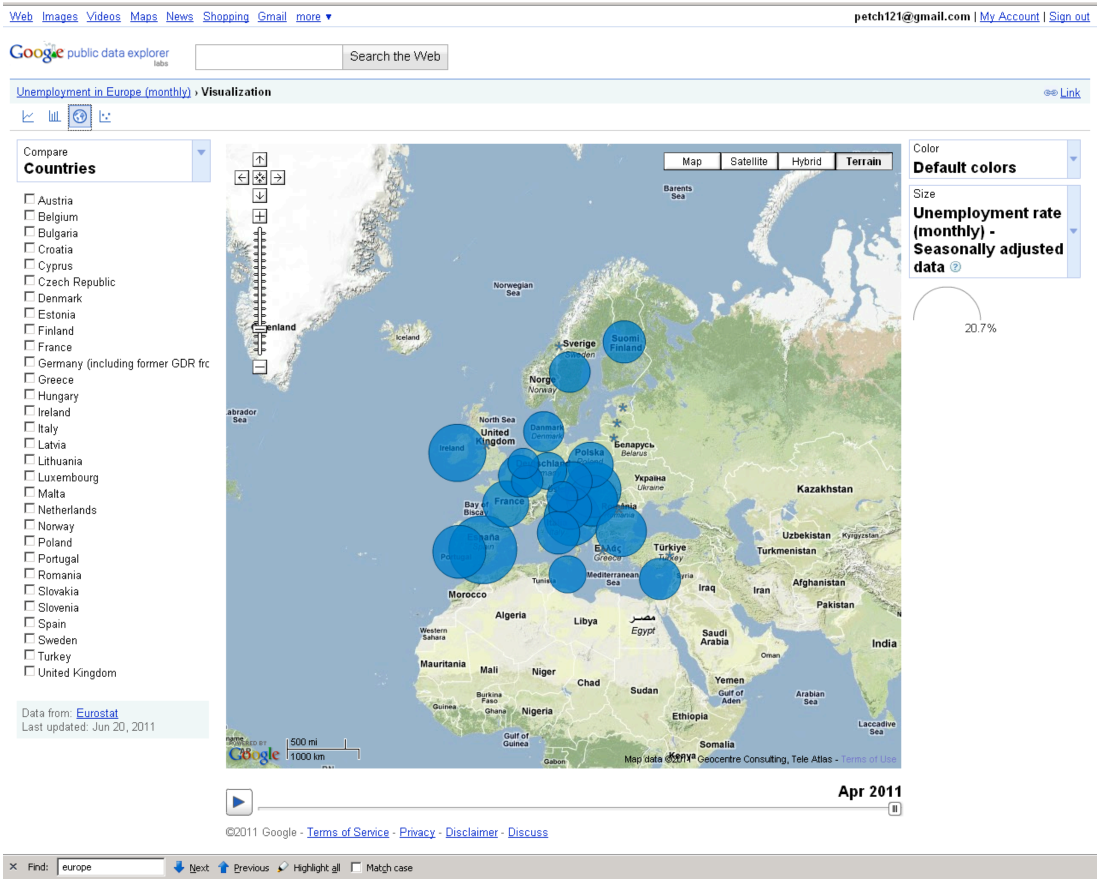


Figure 5.14: Google Public Data Explorer User Interface [67]

From the evaluation result, this thesis realised that four types of visualisation tool can be combined together and reduced down into two groups based on the similarity of accessing RDF data, common visualisation techniques and interaction techniques. The first group is the similarity between generic browsers and search engine tools. The second one is between exploration and mashup tools. The first group is more to support both technical and non-technical user due to the simplicity in accessing RDF data through a simple task such as searching and browsing. Although this kind of tool lacks rich visualisation techniques, the technical user does not require them. On the other hand, the mashup tool and exploration tools are more designed to support non-technical users due to the richness in visualisation techniques. Although, some of these tools still have difficulty in accessing RDF in which a user requires some technical knowledge, characteristics of these tools are more similar to what we need for a general user than the other types of tool.

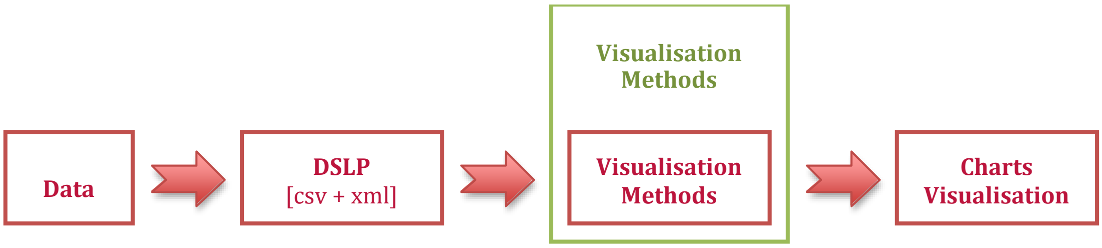


Figure 5.15: Google Public Data Explorer Process

This thesis therefore reduced the scope of the analysing tool into two types that are exploration and mashup tools. We aim to find the potential gaps of these groups of tools that can further research to build a framework for designing an interactive visualisation tool that supports a non-technical user. The summary results from the investigation into exploration and mashup tools for visualising RDF data are presented in Figure 5.16. This Chapter discussed the examples of exploration and mashup tools for understanding the scope of each tool over a process of information visualisation.

Explorator [78] and RKBExplorer [79] are kinds of exploration tool that support mostly in the view transformation process. These applications emphasise the interaction at the views stage. Explorator however provides some features that support users in the visual mapping process, a mapping between data table and visual structures, while RKBExplorer does not. Before using these tools, users have to perform data extraction and transformation themselves.

Exhibit [80] is a visual mashup tool that supports the views transformations process. It provides a small set of Application User Interfaces (APIs) for non-expert users to present data in various views. Exhibit does not, however, support data transformation and visual mapping processes. Users have to perform those processes by themselves.

DERI Pipe [81], a data mashup tool, focuses only on data transformation processes. It aims to support users to integrate data from various sources and different data formats easily through a graphic user interface based editor tool. The visual mappings and view transformation processes, however, have to be done manually or with the other visualisation tools.

Google Public Data Explorer [67], a combination between an exploration and mashup tool, further extends the scope of the Exhibit tool by covering a visual mappings process. It is a tool that allows users to describe data information through their intermediate language (DSPL format). Google Public Data Explorer, however, does not support the data transformation process. Users must perform a transformation process by coding manually or using conversion tools. Although there is a plethora of conversion tools available, the results do not exactly match with the requirements of the input in the visual mapping process. Users have to amend the results into the well-defined format following the structure of the data model. Moreover, these tools are pre-defined user interface functions. To use the other tools for representing the same data, different data transformation processes are required.

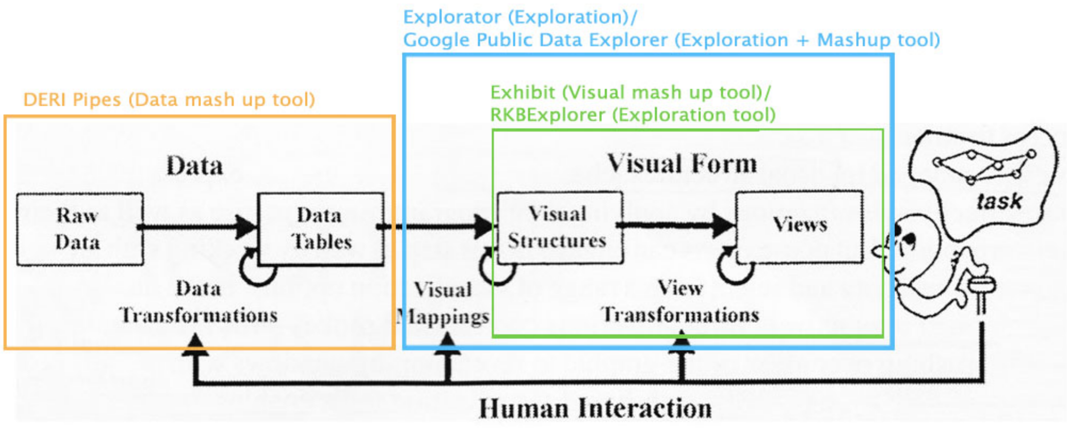


Figure 5.16: A Categorization of Selected Visualisation Tools through the Process of the Information Visualisation Framework

From the experimental results in this Chapter, we can see that most visualisation tools are focused on leverage in the views transformation process. The exploration tool provides features mostly to support user tasks while the mashup tool intends to support multiple kinds of visualisation. However, we can see that the trend of current tools for visualisation tends to support user customisation, allowing users to create their own views of data. This increases the development of a generic web-based tool for semantic web exploration and analysis. This chapter offered an overview of the domain analysis on the visualisation tools that we have studied. The evaluation result is a contribution of this work because it would be used as a resource for studying a potential gap in the next chapter. It is important to remember that the visualisation tools in this chapter were selected based on certain keywords, so there might be some other visualisation tool that we did not mention it here. Moreover, most tools fall into more than one category. In the future, we could therefore further improve the result by increasing the number of samples.

# Chapter 6

# Conclusion and Future Research Opportunities

In the previous chapter the evaluation results were provided. In this chapter, the conclusions arising from these results are discussed. This thesis examines the limitations and obstructions with regard to developing a general web-based visualisation tool for non-technical users. A contribution to the potential gaps in the development of such a tool and guidelines for the future research opportunities is provided. Researchers can extend these contributions towards achieving the goal of interactive visualisation for semantic web data mentioned in Chapter 1.

## 6.1 Discussion of the Result and Conclusions

The motivations that have led to the development of interactive visualisation tools for RDF data have been stated throughout this thesis. These are: (1) to help non-technical users to visualise RDF data; (2) to provide a simple but rich visualisation tool for discovery. Having evaluated all four categories of tools, this thesis has positioned them in Table 6.1. The evaluation results in Chapter 5 reveal some current issues with regards to visualisation tools in semantic web data. Most of the evaluated tools have been designed to conform to the thesis's assumptions that are two major groups of current visualisation tool: poor visualisation which is simple to use, and rich visualisation, which is complex. From the evaluation results, this thesis indicates that generic browser and search engine tools are similar in terms of data accessing, visualisation techniques, and interactive tasks. These tools fall into our first assumption which is positioned in category D in Table 6.1. They provide a simple means of access to RDF data through the use of URIs and keywords. Although generic browsers and search engines are easier for non-technical users to access through RDF data, the functionality and visualisation of these tools is still poor when it comes to supporting non-technical users.

Exploration tools and mash up tools however provide much richer visualisation and functionality which support non-expert users in terms of making it easy to understand and manipulate data. In order to present RDF data by using these tools, users require technical knowledge regarding how to access RDF data and how to visualise them. Most of the tools are designed to present a private data set or specific domain in which the developer has pre-integrated and designed visualisation techniques. These two tools therefore belong to the latter category (A) in Table 6.1.

| | Rich Visualisation | Poor Visualisation |
|---|---|---|
| Complex | Mash up tool, Exploration tool (A) | (B) |
| Simple | (IDEAL TOOL) (C) | Generic Browser, Search Engine (D) |

Table 6.1: A Categorisation of Visualisation Tool based on Richness of Visualisation and Complexity of Use.

During the evaluation however, we realised that there are some tools which have a rich visualisation and some features that help non-technical users to gain simple access to RDF data. It is felt that this tool can fall into both the exploration and mashup categories in terms of classification. This type of visualisation tool has the advantage in having characteristics of both types. For example, Google Public Data Explorer's process is designed to cover most of the processes of information visualisation, in that it allows users to import and visualise data on the fly by using an intermediate language. This tool can be classified as being closely related to the third category (C), which is an ideal tool for generic visualisation.

This thesis found that there were limitations and obstructions in terms of the development of a simple and rich visualisation tool category (C) in Table 6.1. Although there are many types of tools for supporting visualisation of semantic web, there are limited visualisation functions. Therefore we propose that future research should focus on improving mash-up and exploration tool architecture because they are more closely related to interactive visualisation than the other two types which are generic browsers and search engines. The following sub-sections identify and present the potential gaps that were revealed as a result of the assessment.

### 6.1.1 The Lack of Data Transformation in Visualisation Tools

As mentioned in Chapter 5, the trend of current visualisation tools is to support user customisation. This is to help the user to create their own view of data. However, the scope of the current exploration tools and mash up tools presented in Figure 5.16 shows that these kinds of tools, apart from data mash up tools, do not support the data transformation process. This means that it cannot be used easily by non-technical users when it comes to accessing RDF and visualising data. Most of these kinds of applications and the examples which were observed, have been pre-built by developers and are mostly built for a particular subject domain. The problem of visualising data using such tools is the difficulty of preparing data as part of the data transformation process, for instance, accessing and extracting RDF data. Non-technical users still require a basic knowledge of XML due to the fact that most tools used intermediate languages as an input and these languages such as DSPL or JSON are defined based on the XML format. They also need to know how to transform them into the intermediate format. This problem is supposed to be solved in order to allow non-technical users to easily use a mash up tool for visualising RDF data.

### 6.1.2 The Lack of Visual Mapping in Visualisation Tools

Figure 5.9 presents the supported scope of the mash up tool in terms of what it lacks with regard to the visual mapping process. Another gap that needs to be considered is a visual mapping process associated with the visualisation tool. In a mash up tool, users normally have to perform custom coding by transforming raw data into a format that is compatible with the visualisation techniques. Although there are many tools designed to convert data into a specific input format, the results are not yet satisfactory. Users are required to amend some parts of the code to allow the system to understand how to present data in a certain visualisation format. For example, there are some converter scripts, such as RDFOO [111], to convert RDF to JSON format as an input of Exhibit. However,the end-user has to revise the result to fit in the specific visualisation techniques.

During the evaluation, we also realised that there is a lack of flexibility in adding new APIs and libraries. Each visualisation tool has its own visualisation techniques which may not be enough for an end-user. Although the hybrid tool that mixes the Exhibit and mashup tool has some of the characteristics of a mashup tool which is easy and fast integration of the visualisation process, it is still difficult to insert new external APIs or libraries into the system. This is because of the differences between the structures and standards of APIs. By providing an intermediate language that can annotate/describe visualisation APIs, we believe this will enhance flexibility when integrating new visualisation techniques.

### 6.1.3 The Need for Quantitative Visualisation for RDF Data

Figure 5.3 indicates that the visualisation tools for presenting RDF data still rarely support quantitative data visualisation. The most common visualisation techniques are Table Structure (text-based), Multiple Axis (images-based), and Graph structure (graph-based) respectively. Although some mashup tools and Exploration tools provide some techniques for presenting quantitative data, these tool are not fully designed for presenting RDF data. It requires a lot of effort to convert and transform data before presenting.

### 6.1.4 The Need for a Navigating Function in Visualisation Tools

Figure 5.5 also presents the task availability in each visualisation tool test set. It can be seen that most kinds of tool lack the ability to navigate RDF data. Navigation is normally a common activity in web-based applications. This helps users to know where they are and allows them to trace back or define the path that they followed during the browsing process. In the visualisation of RDF data, users always probe using linked data in order to browse information node-by-node. We suggest that it is necessary to collect details of the data path that users travel along. This will show the current location of the user and the related data history. From the result, we can imply that most generic browsers are not browsers; they are more like a query result viewer.

## 6.2 Future Research Opportunities

Following on from the challenging discussion of visualisation RDF data, there are some alternative areas, which could be focused upon, namely data transformation, visual mapping, and view transformation. Both data and view transformation, however, have been investigated by many researchers. The trends of future research should focus more on improving the visual mapping process for the reason that this process is the most relevant to user's needs due to users requiring visualisation that is appropriate for their data and they do not want to be forced by a content provider [26]. There are also fewer studies aimed at improving this process. Furthermore, from our assessment in Chapter 5, this thesis has found some visualisation tools that provide a dynamic visual mapping process through the semi-automatic approach and we believe that their architecture can be extended towards filling the potential gaps we have identified. To bridge the gaps and build up the ideal tool mentioned in Section 6.1, we propose that research should extend the architecture of exploration and mashup tool from top left to the bottom left in Table 6.1 (C). The problems related to this kind of tool are about complication of use, which arises from two significant reasons: (1) lack of data transformation and visual mapping steps and (2) the difficulty in user interaction. It is somewhat difficult to deal with improving the process of user interaction and a lot of research into the behaviours in different groups of users is needed. We therefore propose to further study the common architecture of visualisation categorised in the type of exploration and mashup tool presented in the Figure 6.1 is needed. This would aim to reduce the complexity of the information visualisation process.

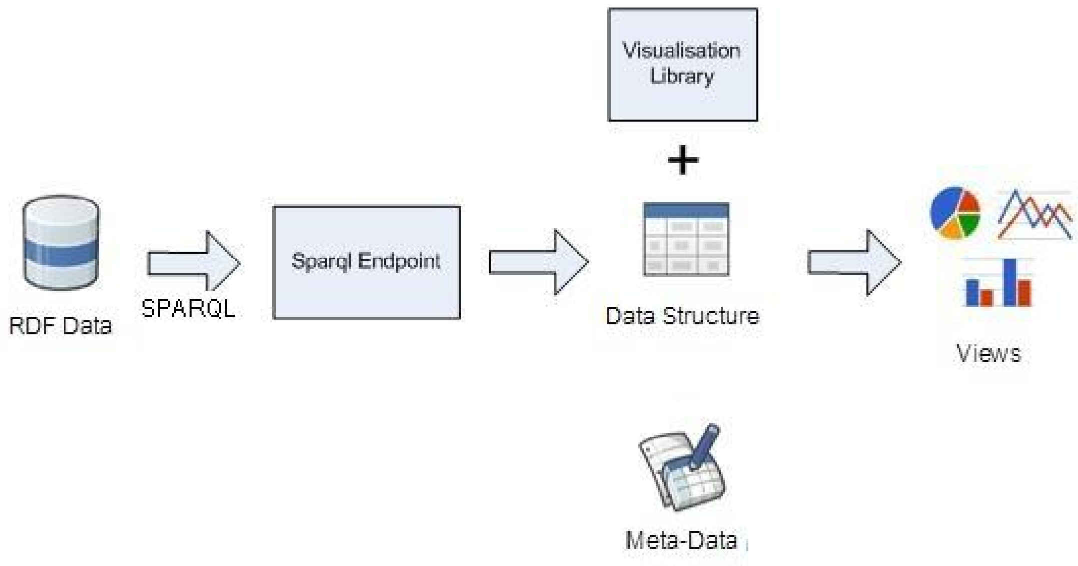


Figure 6.1: A Common Architecture of Visualisation [67] [77]

This architecture extracts from a common process of two visualisation tools and one programming script: Google Public Data Explorer [67], IBM Many Eyes [77], and LOGD script [96] respectively. The purpose of this approach is to provide extra information (meta-data) about the dataset such as datatype, description, and presentation structure to a system. The system, therefore, can then perform dynamic visualisation based on pre-defined information. Although these two visualisation tools and script are using the same concept in providing meta-data to the system, there are differences in the method and extra information that is used to present (Table 6.2).

| Visualisation tool/ Script | Methods | Meta-data information |
| --- | --- | --- |
| Google Public Data Explorer | Create DSPL file (CSV + XML) | • The concept of data<br>• The structure of concepts<br>• Abstract data table |
| IBM Many Eyes | Annotate the Data Table | • The unit of data<br>• The data type |
| LOGD Script | Function's argument | • Abstract data table<br>• Visual structure<br>• Configure information |

Table 6.2: Comparison of the Architecture Features

The current approach (Figure 6.1) so far is to predefine meta-data for the visual mapping process between the data structure and visualisations. Users are allowed to compose their own visualisation styles by providing meta-data about the primitive visualisation structure (see Table 6.2) to a system. Although providing meta-data to the system through this architecture can enable the system to understand a structure of the data, we noticed that this architecture does not support direct manipulation by users. To visualise further information, users need to rewrite the query and new meta-data to the system. Future work should focus on how to reduce these manual processes and increase the automaticity of visualisation processes, which will lead to enabling direct manipulation methods [28].

We propose that this architecture could embed more information regarding the task's structure and visual structure in the meta-data file. The information about the task's structure and visual structure will enable the automaticity of the system. For example, if the system has enough information such as the task and visualisation structure that users want to perform, it will be able to generate continuous visualisations from the user's actions. To perform such an automatic visualisation, the architecture also needs to include an abstract visualisation library, which provides information regarding visualisation libraries and what visual and task structures are supported and presented. This thesis proposes two possible approaches extended from the common one, to embed meta-data and an abstract visual library. These two approaches are represented in Figure 6.2 and Figure 6.3.

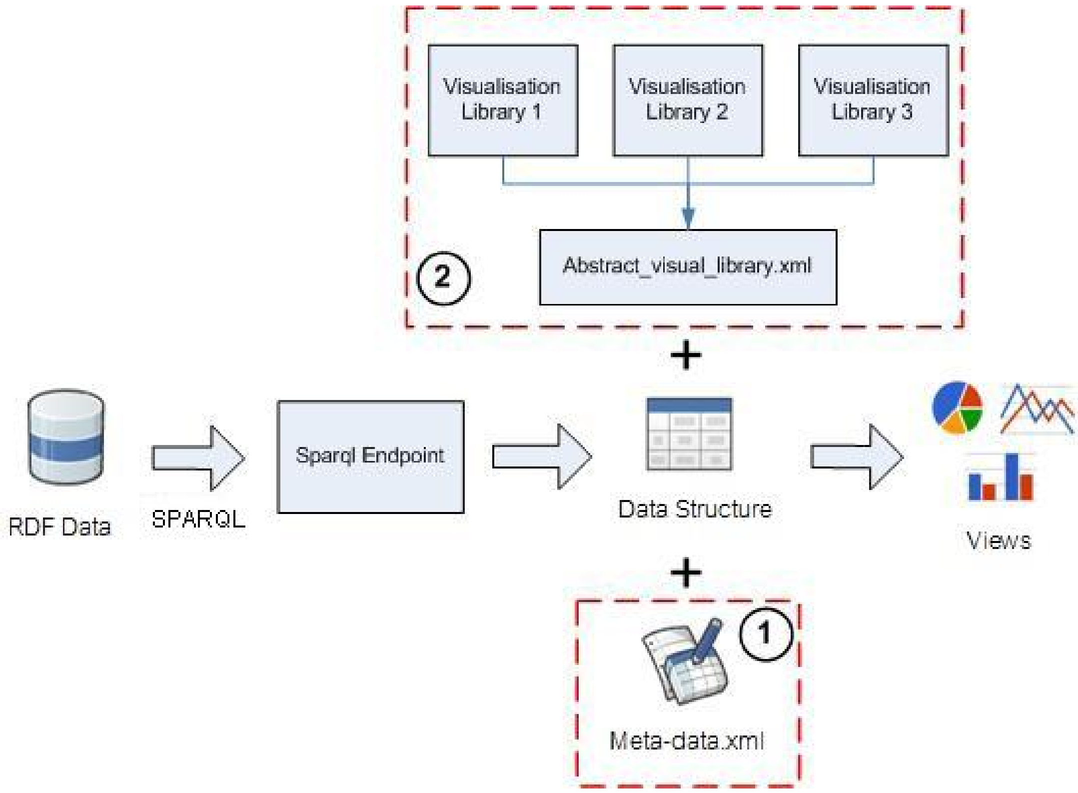


Figure 6.2: Rewrite the Structure of Meta-Data from Figure 6.1

Figure 6.2 illustrates an architecture extended from Figure 6.1. It shows that the meta-data file (1) can be extended by adding more descriptions concerning visual structure and task structure. The architecture also provides an abstract visual library (2) as an interface located between the concrete visualisation library components and mapping components. The mapping process between meta-data and the abstract visualisation library is carried out by mapping rules. The advantage of this approach is that it provides enough information to the system; the system therefore can use the information to automatically generate a SPARQL query when users perform an action, such as probing a certain part of the visualisation to reveal detail without rewriting the new SPARQL query and meta-data. The abstract visualisation library also enables the extensibility of the system; users can extend the visualisation library from other sources by defining the abstract visualisation library through the specific XML syntax. The major limitation of this approach is the users of the system need to have knowledge of the SPARQL query and basic XML. Therefore, to support general users who do not have such knowledge, a visual programming technique might be considered.

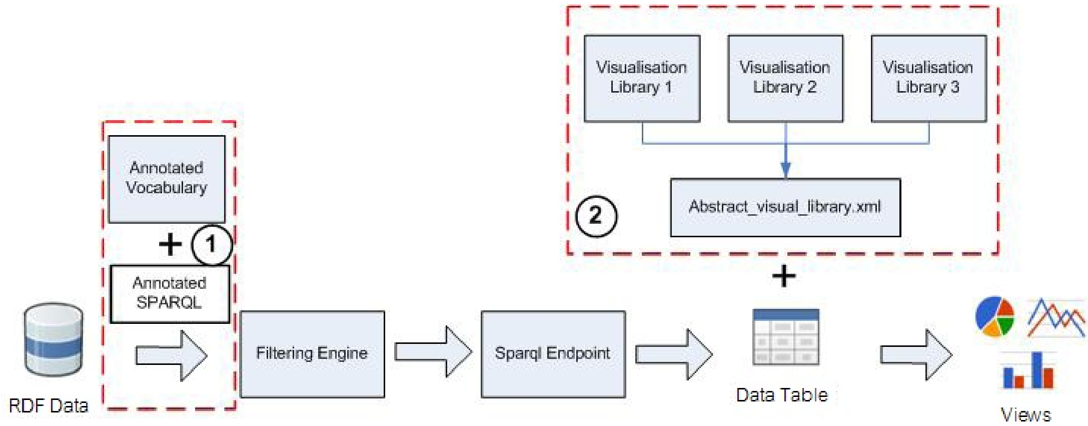


Figure 6.3: Annotate Meta-Data Directly through a SPARQL Query.

The second approach in Figure 6.3 is similar to the first approach in Figure 6.2 apart from that it provides meta-data information by annotating a SPARQL query (1) rather than separating it to the other file. The reason for doing this is because we believe that it might help users to reduce a step and the complexity in a process of data preparation. However, to annotate SPARQL query, we need to design a new vocabulary and engine to interpret the annotated query. An obstruction to this approach concerns the difficulty of designing an annotation language that has to serve the basic requirements of users and the system.

Although these two different approaches use different techniques for providing meta-data to the system, their main components in a framework are mostly the same. We believe that applying these two approaches may increase visualisation tool flexibility in extending visualisation techniques. Future research should study whether these two approaches can fill the gaps that have been identified and whether they conform to the assumption that we have mentioned. Moreover, it is important to experiment with the advantages and disadvantages of different intermediate languages such as DSPL, VizQL, and JSON in visualisation tools, and which intermediate language can most appropriately be extended as meta-data in these approaches.

## 6.3 Summary

Having analysed visualisation tools and types of tools, we believe that some functionalities, such as navigating RDF data, can be expected in future visualisation tools. All processes of information visualisation will be supported by the web-based visualisation tool. Moreover, some visualisation components are expected to support quantitative representation in the form of charts and graphs such as bar charts, line charts and area plots.

The problems with current visualisation tools have been identified in this study. The gaps in developing a generic web-based visualisation tool occur on data transformations and visual mapping processes. We specify those gaps in the following list:

- **Difficulty and time consuming:** manually performing a process of data transformation and visual mapping without any support tools is a difficult task for a non-expert user. Our experiment has shown that there is no visual programming tool that supports all processes of information visualisation.
- **Difficulty in extending the visualised structure:** there are no experimental tools that allow users to easily add extensible external visualisation and operation methods from other resources.

The ultimate goal of future research is to design a semantic web visualisation framework that can support users in exploring and analysing semantic data interactively in an easy way. This requires solving the problems of difficulty and time-consuming use of tools. From the experimental result in Chapter 5, this thesis has found a visualisation tool that falls into both exploration and mashup tool categories. This kind of tool offers a rich visualisation and is simpler to use compared with the other exploration tools or mashup tools. Moreover, we divided the complexity of using visualisation tool into two major parts that are (1) lack of data transformation and visual mapping steps and (2) difficulties in in user interaction. However, we recommend that an area of future research is improving data transformation and visual mapping steps. This is because the latter part is more complex and requires a lot of research in the user behaviour in different groups of users.

Currently, there are a few studies of tool architecture to solve the problem in data transformation and visual mapping steps. These helped us to study and propose the architecture extending from the common architecture presented in Figure 6.1 that aims to improve the simplicity of the visualisation process for non-technical users. Future research should perform an evaluation of the architecture we have mentioned in Figure 6.2 and 6.3 including whether they can reduce problems we have identified. Existing opensource visualisation libraries and APIs, such as Google API [69, 71] and Exhibit [80], are recommended to serve as a test bed. An empirical evaluation should be performed with a simulation of RDF data constructed to cover the evaluation scenarios in order to measure the accuracy, usability and flexibility of the architectures. It is also important to evaluate it using real world data extracted from semantic web data sources such as DBpedia [37] or Freebase [36].

# Bibliography

[1] Y. Rogers, H. Sharp, and J. Preece, *Interaction Design: Beyond Human-Computer Interaction*. John Wiley and Sons Ltd, 2002.

[2] B. Shneiderman and C. Plaisant, *Designing the User Interface: Strategies for Effective Human-Computer Interaction (5th Edition)*, 5th ed. Addison Wesley, 2010.

[3] J. Steele and N. P. N. Iliinsky, *Beautiful Visualization: Looking at Data through the Eyes of Experts (Theory in Practice)*. Sebastopol: O'Reilly Media, 2010. [Online]. Available: http://www.worldcat.org/isbn/1449379869

[4] J. Hebeler, M. Fisher, R. Blace, A. Perez-Lopez, and M. Dean, *Semantic Web Programming*. Indianapolis, IN: Wiley Publishing, Inc., 2009.

[5] J. Bertin, *Graphics and graphic information-processing / Jacques Bertin ; translated by William J. Berg and Paul Scott*. de Gruyter, Berlin ; New York :, 1981.

[6] D. R. Howe, *Data Analysis for Database Design*, 3rd ed. Newton, MA, USA: Butterworth-Heinemann, 2001.

[7] S. K. Card, J. D. Mackinlay, and B. Shneiderman, Eds., *Readings in information visualization: using vision to think*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1999.

[8] P. Dey, *Visualization Cookbook Using AVS/Express*. The University of Manchester: International AVS Centre Manchester Visualization Centre, November 2001.

[9] T. Heath, "How will we interact with the web of data?" *IEEE Internet Computing*, vol. 12, no. 5, pp. 88–91, Sep. 2008. [Online]. Available: http://dx.doi.org/10.1109/MIC.2008.101

[10] M. Hausenblas, "Exploiting linked data to build web applications," *IEEE Internet Computing*, vol. 13, pp. 68–73, 2009.

[11] C. Bizer, T. Heath, and T. Berners-Lee, "Linked Data - The Story So Far," *International Journal on Semantic Web and Information Systems (IJSWIS)*, vol. 5, no. 3, pp. 1–22, 2009.

[12] C. Bizer, R. Cyganiak, and T. Heath, "How to publish linked data on the web," Web page, 2007, revised 2008. Accessed 22/02/2010. [Online]. Available: http://www4.wiwiss.fu-berlin.de/bizer/pub/LinkedDataTutorial/

[13] H. Senay and E. Ignatius, "A knowledge-based system for visualization design," *IEEE Comput. Graph. Appl.*, vol. 14, pp. 36–47, November 1994. [Online]. Available: http://portal.acm.org/citation.cfm?id=616033.617929

[14] S. Knight and A. Spink, "Toward a web search information behavior model," *Springer Series in Information Science and Knowledge Management.*, vol. 14, pp. 209–234, 2008.

[15] M. Bates, "Toward An Integrated Model of Information Seeking and Searching," *Graduate School of Education and Information Studies*, 2003. [Online]. Available: http://www.gseis.ucla.edu/faculty/bates/articles/info_SeekSearch-i-030329.html

[16] "A classification of interactions with information." [Online]. Available: http://www.scils.rutgers.edu/~belkin/612-05/cool-belkin-colis-02.pdf

[17] C. Cool and N. Belkin, "A classification of interactions with information," vol. 03, 2002, pp. 1–15. [Online]. Available: http://www.scils.rutgers.edu/~belkin/612-05/cool-belkin-colis-02.pdf

[18] D. Ellis, D. Cox, and K. Hall, "A comparison of the information seeking patterns of researchers in the physical and social sciences," 1993.

[19] T. Wilson, "Human information behaviour," *Special Issue on Information Science Research*, vol. 03, pp. 49–55, 2000.

[20] H. Purchase, N. Andrienko, and M. Ward, "Theoretical foundations of information visualization," *Information Visualization,*, pp. 46–64, 2008.

[21] T. Wilson, "Models in information behaviour research," *Journal of Documentation*, vol. 55, pp. 249–270, 1999.

[22] D. Pfitzner, V. Hobbs, and D. Powers, "A unified taxonomic framework for information visualization," *2nd Australian Institute of Computer Ethics Conference (AICE2000)*, vol. 01, pp. 46–64, 2001.

[23] D. Karger and M. Schraefel, "The pathetic fallacy of rdf," 2006, position Paper for SWUI06.

[24] P. Pop, "Comparing web applications with desktop application an empirical study," 2002.

[25] L. Tweedie, "Characterizing interactive externalisations," *In Conference on Human Factors in Computing Systems*, pp. 375–382, 1997.

[26] D. Brodbeck, R. Mazza, , and D. Lalanne, "Interactive visualization - a survey," *Human Machine Interaction*, p. 2746, 2009.

[27] R. M. Rohrer and E. Swing, "Web-based information visualization," *IEEE Computer Graphics and Applications 53*, pp. 52–59, 1997.

[28] B. Shneiderman, "The eyes have it: A task by data type taxonomy for information visualizations," *Proceedings of the 1996 IEEE Symposium on Visual Languages (VL '96)*, vol. 96, pp. 1049–2615, 1996.

[29] T. Berners-lee, J. Hendler, and O. Lassila, "The semantic web," *Scientific American: Feature Article: The Semantic Web*, 2001.

[30] a. Y. C. Tim Berners-Lee, a. R. D. Lydia Chilton, and Dan Connolly, J. Hollenbach, A. Lerer, and D. Sheets, "Tabulator: Exploring and analyzing linked data on the semantic web," *Procedings of the The 3rd International Semantic Web User Interaction Workshop (SWUI06) workshop*, November 2006.

[31] a. J. H. T Berners-Lee, a. J. P. Kanghao Lu, and a. m. s. E. Prud'ommeaux, "Tabulator redux: Browsing and writing linked data," *CEUR-WS In WWW2008 Workshop on Linked Data on the Web*, November 2006.

[32] E. Oren, R. Delbru, M. Catasta, R. Cyganiak, and G. Tummarello, "Sindice.com: A document-oriented lookup index for open linked data," *International Journal of Metadata, Semantics and Ontologies*, vol. 3, p. 2008.

[33] S. F. C. de Arajo and D. Schwabe, "Explorator: a tool for exploring rdf data through direct manipulation," 2009.

[34] B. Hibbard, "Top ten visualization problems," *SIGGRAPH Comput. Graph.*, vol. 33, pp. 21–22, May 1999. [Online]. Available: http://doi.acm.org/10.1145/326460.326485

[35] A.-S. Dadzie and M. Rowe, "Approaches to visualising linked data: a survey," *Semantic Web*, vol. 1, no. 1-2, 2011. [Online]. Available: http://oro.open.ac.uk/28848/

[36] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, *Freebase: a collaboratively created graph database for structuring human knowledge*. ACM, 2008, p. 12471250. [Online]. Available: http://portal.acm.org/citation.cfm?id=1376616.1376746

[37] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, and Z. Ives, "Dbpedia: A nucleus for a web of open data," in *In 6th Intl Semantic Web Conference, Busan, Korea*. Springer, 2007, pp. 11–15.

[38] Q. Wang, W. Wang, R. Brown, K. Driesen, B. Dufour, L. Hendren, and C. Verbrugge, "Evolve: An open extensible software visualization framework." Press, 2002.

[39] V. R. Benjamins, J. Contreras, O. Corcho, and A. Gmez-prez, "Six challenges for the semantic web," in *In KR2002 Semantic Web Workshop*, 2002, p. 2004.

[40] M. Leida, A. Afzal, and B. Majeed, "Outlines for dynamic visualization of semantic web data," in *Proceedings of the 2010 international conference on On the move to meaningful internet systems*, ser. OTM'10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 170–179. [Online]. Available: http://portal.acm.org/citation.cfm?id=1948509.1948557

[41] G. Tummarello, R. Delbru, and E. Oren, "Sindice.com: Weaving the open linked data," in *In Proceedings of the International Semantic Web Conference (ISWC*, 2007.

[42] A. Westerski, A. Iqbal, G. Tummarello, and S. Decker, "Sindice widgets:

Lightweight embedding of semantic web capabilities into existing user applications." in *In Proceedings of the 4th International Workshop on Semantic Web Enabled Software Engineering*, Karlsruhe, Germany, 2008.

[43] P. Haase, A. Eberhart, S. Godelet, T. Math, T. Tran, G. Ladwig, and A. Wagner, "The information workbench. interacting with the web of data," in *3rd Future Internet Symposium (FIS2010)*, 2010. [Online]. Available: http://data.semanticweb.org/conference/fis/2010/paper/posters/11

[44] M. Schraefel, D. Alex, E. Smith, A. Russel, A. Owens, C. Harris, and M. Wilson, "The mspace classical music explorer: Improving access to classical music for real people," in *MusicNetwork Open Workshop, Integration of Music in Multimedia Applications*, 2005.

[45] M. L. Wilson and M. C. Schraefel, "mspace: What do numbers and totals mean in a flexible semantic browser," in *In Proceedings of The 3rd International Semantic Web User Interaction Workshop at ISWC2006*, 2006, pp. 401–408.

[46] D. F. Huynh, D. R. Karger, and R. C. Miller, "Exhibit: lightweight structured data publishing," in *Proceedings of the 16th international conference on World Wide Web*, ser. WWW '07. New York, NY, USA: ACM, 2007, pp. 737–746. [Online]. Available: http://doi.acm.org/10.1145/1242572.1242672

[47] L. Ding, T. Finin, A. Joshi, R. Pan, R. S. Cost, Y. Peng, P. Reddivari, V. Doshi, and J. Sachs, "Swoogle: a search and metadata engine for the semantic web," in *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, ser. CIKM '04. New York, NY, USA: ACM, 2004, pp. 652–659. [Online]. Available: http://doi.acm.org/10.1145/1031171.1031289

[48] T. Finin, L. Ding, R. Pan, A. Joshi, P. Kolari, A. Java, and Y. Peng, "Swoogle: Searching for knowledge on the semantic web," in *In AAAI 05 (intelligent systems demo*. The MIT Press, 2005, pp. 1682–1683.

[49] G. Cheng, W. Ge, and Y. Qu, "Falcons: searching and browsing entities on the semantic web," in *Proceeding of the 17th international conference on World Wide Web*, ser. WWW '08. New York, NY, USA: ACM, 2008, pp. 1101–1102. [Online]. Available: http://doi.acm.org/10.1145/1367497.1367676

[50] H. Glaser, I. Millard, and A. Jaffri, "Rkbexplorer.com:a knowledge driven infrastructure for linked data providers," in *European Semantic Web Conference*, vol. 5021?2. Springer, June 2008, pp. 797–801. [Online]. Available: http://eprints.ecs.soton.ac.uk/15152/

[51] H. Glaser and I. Millard, "Rkb explorer: Application and infrastructure ." in *Semantic Web Challenge*, ser. CEUR Workshop Proceedings, J. Golbeck and P. Mika, Eds., vol. 295. CEUR-WS.org, 2007. [Online]. Available: http://dblp.uni-trier.de/db/conf/semweb/challenge2007.html#GlaserM07

[52] A. B. Vigas, M. Wattenberg, F. V. Ham, J. Kriss, and M. Mckeon, "Many eyes: A site for visualization at internet scale," in *Proceedings of Infovis*, 2007.

[53] D. Le-Phuoc, A. Polleres, M. Hauswirth, G. Tummarello, and C. Morbidoni, "Rapid prototyping of semantic mash-ups through semantic web pipes," in *Proceedings of the 18th international conference on World wide web*, ser. WWW '09. New York, NY, USA: ACM, 2009, pp. 581–590. [Online]. Available: http://doi.acm.org/10.1145/1526709.1526788

[54] B. E. Rogowitz and L. A. Treinish, "An architecture for rule-based visualization," in *Proceedings of the 4th conference on Visualization '93*, ser. VIS '93. Washington, DC, USA: IEEE Computer Society, 1993, pp. 236–243. [Online]. Available: http://portal.acm.org/citation.cfm?id=949845.949889

[55] L. Han, L. Ding, and R. Pan, "Swoogle's Metadata about the Semantic Web," University of Maryland, Baltimore County, Tech. Rep., May 2006.

[56] M. J. Pacione, M. Roper, and M. Wood, "A review and evaluation of dynamic visualisation tools," Tech. Rep., 2003.

[57] G. H. Jacoby, J. Barnes, K. A. V. Test-bed, and R. Gooch, "Karma: a visualization test-bed," 1996.

[58] S. Kochhar, M. Friedell, and M. LaPolla, "Cooperative, computer-aided design of scientific visualizations," in *Proceedings of the 2nd conference on Visualization '91*, ser. VIS '91. Los Alamitos, CA, USA: IEEE Computer Society Press, 1991, pp. 306–313. [Online]. Available: http://portal.acm.org/citation.cfm?id=949607.949658

[59] M. Jern, "Information visualisation on the web."

[60] "Tibco spotfire analytics platform," 2010. [Online]. Available: http://spotfire.tibco.com

[61] C. Stolte, D. Jewett, and P. Hanrahan, "Tableau: A new approach to business intelligence rapid-fire bi."

[62] "Getting started with tableau desktop," May 2010. [Online]. Available: http://downloads.tableausoftware.com/quickstart/main-guides/desktop_getstarted6.0.pdf

[63] "Quickview architectural overview." [Online]. Available: http://qlikview.com

[64] "Quickview - simplifying analysis for everyone," 2006. [Online]. Available: http://www.intcomtech.net/QV7brochure.pdf

[65] C. Fluit, M. Sabou, and F. V. Harmelen, "3 ontology-based information visualization: Towards semantic web applications," 2002.

[66] D. Phuoc, "Semantic web pipes demo," White paper, Degital Enterprise Research Institute.

[67] L. Edwards, "Google introduces its public data explorer," PHYSORG.com, March 2010. [Online]. Available: http://www.google.com/publicdata/home

[68] L. Parfeni, "Upload your own data to the google public data explorer visualization tool," White paper, Softpedia.

[69] "Google chart web part," White paper, AMREIN ENGINEERING AG, January 2010.

[70] P. A. Bernstein, "Middleware an architecture for distributed system services," White paper, March 1993.

[71] "Google chart tools, http://code.google.com/apis/chart/," GOOGLE, June 2011.

[72] "Flare data visualisation for the web," UC Berkeley Visualization Lab., June 2011. [Online]. Available: http://flare.prefuse.org/

[73] "Javascript and java diagram library," JGraph Ltd, June 2011. [Online]. Available: http://www.jgraph.com/company.html

[74] "The hypergraph," sourceforge, June 2011. [Online]. Available: http://hypergraph.sourceforge.net/

[75] "Graphopt," Graphviz, June 2011. [Online]. Available: http://schmuhl.org/graphopt/

[76] "Yahoo pipes," Yahoo, June 2011. [Online]. Available: http://pipes.yahoo.com/pipes/

[77] "Many eyes," IBM, July 2011. [Online]. Available: http://www-958.ibm.com/software/data/cognos/manyeyes/

[78] "Explorator," TecWeb, July 2011. [Online]. Available: http://www.tecweb.inf.puc-rio.br/explorator

[79] "Rkbexplorer," ReSIST, July 2011. [Online]. Available: http://www.rkbexplorer.com/

[80] "Exhibit simile," Simile, July 2011. [Online]. Available: http://simile.mit.edu/wiki/Exhibit:_Semantic_Web_Tools

[81] "Deri pipes," DERI, July 2011. [Online]. Available: http://pipes.deri.org/

[82] C. Morbidoni, D. L. Phuoc, A. Polleres, M. Samwald, and G. Tummarello, "Previewing semantic web pipes," 2008.

[83] "Disco - hyperdata browser," the Freie Universitt Berlin, July 2011. [Online]. Available: http://www4.wiwiss.fu-berlin.de/bizer/ng4j/disco/

[84] "Tabulator," July 2011. [Online]. Available: http://www.w3.org/2005/ajar/tab

[85] "Zigist," July 2011. [Online]. Available: http://zitgist.com/

[86] "The openlink data explorer extension," July 2011. [Online]. Available: http://ode.openlinksw.com/

[87] "Sig.ma," July 2011. [Online]. Available: http://sig.ma/

[88] "Falcon," July 2011. [Online]. Available: http://ws.nju.edu.cn/falcons/objectsearch/index.jsp

[89] "Swoogle," July 2011. [Online]. Available: http://swoogle.umbc.edu/

[90] “mspace,” July 2011. [Online]. Available: http://mspace.fm/

[91] “Visual data explorer,” July 2011. [Online]. Available: http://www.visualdataweb.org/

[92] “Information workbench,” July 2011. [Online]. Available: http://www.fluidops.com/information-workbench/

[93] “Bbc music website relunch,” March 2009. [Online]. Available: http://www.bbc.co.uk/blogs/bbcinternet/2009/03/bbc_music_website_relaunch.html

[94] “Musicbrainz,” July 2011. [Online]. Available: http://musicbrainz.org/

[95] “Jamendo:open your eyes,” July 2011. [Online]. Available: http://www.jamendo.com/en/

[96] “Building logd visualizations,” TWC LOGD, August 2011. [Online]. Available: http://logd.tw.rpi.edu/tutorial/building_logd_visualizations

[97] “Rap - rdf api for php,” August 2011. [Online]. Available: http://www4.wiwiss.fu-berlin.de/bizer/rdfapi/

[98] “Parse-o-metic,” August 2011. [Online]. Available: http://www.parse-o-matic.com/

[99] “Data visualisation tool comparison,” August 2011. [Online]. Available: http://apandre.wordpress.com/tools/comparison/

[100] “the five star linked open data,” November 2011. [Online]. Available: http://inkdroid.org/journal/2010/06/04/the-5-stars-of-open-linked-data/

[101] “W3c standards,” March 2012. [Online]. Available: http://www.w3.org/standards/

[102] “W3c semantic web activity.”

[103] “Resource description framework (rdf),” December 2011. [Online]. Available: http://www.w3.org/RDF/

[104] “Rdf vocabulary description language 1.0: Rdf schema,” February 2004. [Online]. Available: http://www.w3.org/TR/rdf-schema/

[105] "Owl web ontology language overview," February 2004. [Online]. Available: http://www.w3.org/TR/owl-features/

[106] "Owl web ontology language guide," Online, February 2004. [Online]. Available: http://www.w3.org/TR/owl-guide/

[107] "Skos simple knowledge organization system," October 2009. [Online]. Available: http://www.w3.org/2004/02/skos/

[108] "Skos simple knowledge organization system reference," August 2009. [Online]. Available: http://www.w3.org/TR/skos-reference/

[109] "Linked data - connect distributed data on the web," December 2011. [Online]. Available: http://linkeddata.org/

[110] "Sparql query language for rdf," January 2008. [Online]. Available: http://www.w3.org/TR/rdf-sparql-query/

[111] "Rdfoo - convert rdf graphs into json objects," March 2012. [Online]. Available: http://conception.sourceforge.net/rdfoo/

[112] "Protg," March 2012. [Online]. Available: http://protege.stanford.edu/

[113] B. Motik, B. C. Grau, I. Horrocks, Z. Wu, A. Fokou, and C. Lutz, "Owl web ontology language profiles," W3C Recommendation, World Wide Web Consortium, 2009. [Online]. Available: http://www.w3.org/TR/owl2-profiles/

[114] B. C. Grau, I. Horrocks, B. Motik, B. Parsia, P. Patel-Schneider, and U. Sattler, "Owl 2: The next step for owl," *Web Semant.*, vol. 6, no. 4, pp. 309–322, Nov. 2008. [Online]. Available: http://dx.doi.org/10.1016/j.websem.2008.05.001

# Appendix A

# Additional Details of the visualisation tools

In this project, 16 visualisation tools are investigated in evaluation scheme presented in Chapter 5, so this section is going to provide basic description of these tools. We also intend to present our ideas concerning the visualisation based on our experiment.

## A.1 Tabulator

Tabulator is a generic semantic web browser [30,31,84].It is more likely to be designed for supporting a semantic web expertise. Due to its user interface it represents linked data in a tree-oriented structure based on the triple format; the user is required to understand the triple format. It is probably not appropriate for discovery knowledge. Even if a user has knowledge about the semantic web structure, it is even worse than normal HTML web content, which represents data for humans. To improve the generic web browser for linked data, it has to provide more features for organizing and managing information such as filtering, grouping, or zooming. These features allow a user to interact with data and gain different points of view from the publisher. To find some information, users needs to know what the URI of an object is that they are looking for. However, it provides features for a programmer to query semantic web data though a SPARQL command.

## A.2 DISCO

The DISCO is a simple generic semantic web browser like Tabulator [83]. However, it represents search results in table form which is simple and easy to read. To compare with the tree structure in Tabulator, the tree structure has an advantage in being able to expand the data which allows user to explore a related component. However, if users expand the tree structure too deep, it will result in infinity loop of information which is really confuse to a user. The main feature of the DISCO browser is to allow users to dereference a URI and verify its sources.

## A.3 Zigist RDF Browser

Zitgist RDF Browser is a semantic web browser in which users can browse the information available on the semantic web [85]. It provides a feature for dereferencing RDF data and presenting it through a template-based system in which data are arranged in the appropriate way but it does not allows users to customise the visualisation in their own way. Zigist RDF Browser supports data from many sources such as internal data sets, URIs, and on-the-fly conversion data sources (Microformat, RDFa).

## A.4 Open Linked Data Explorer

The Open Linked Data Explorer is a hybrid semantic web between searching and browsing tool for exploring RDF data [86]. Although it provides many features for users to gain an insight into the structure of linked data from different points of view, such as SVG graph, Image, Grid view, etc., it is still difficult for non-expert users who does not have knowledge in semantic web concept. On the other hand, this application seems to be designed for domain expert users to discover patterns of data and its structure. The natural language search produces extremely unsatisfactory results; the triples are shown in block form and seem not to relate to each other, even the view has been changed and it is still difficult to connect the content together. Dereferencing and searching from the URI provides much a better result; it is more readable than the natural language search result. The mash-up of technology such as Google maps or timeline is meant to enhance users vision by instead of showing location information as a numeric value, it represents it in the map which is easier to understand. Besides that, another feature is trying to change the view of displaying and navigating data. This feature is not always good to represent data; for instance, the SVG graph represents a relationship of thing by using node and link. When there are a lot of relationships, however, the graph of networks looks messy and users cant get any meaning from it. In this version of the application, the timeline, tag cloud, and custom function cannot be used at the moment. One feature I like from this application is that it allows the filtering of results based on the RDF predicate.

## A.5 Sindice (SIG.MA)

The Sindice project is an advance search engine for semantic web data [32,41,42,87]. It allows the end user to search by using a natural language and also to be able to browse the semantic data through its URI as well. The result of the SIG.MA search is displayed as a simple block structure, but is more effective than the other applications. The result seems to be summarized from various sources with references that allow users to be able to check, approve, and trace back the source of the information. One problem from experience is that when we have massive related information, it is difficult to focus on the data that we are looking for. The filtering feature after the result might be a good solution for the SIG.MA. Moreover, the Sindice also provides a feature for validating the URI concerning the syntax, common error, reasoning result. It also provides multiple points of view which allow the user to discover knowledge from a different point of view such as a graph, triple data, and ontology data. Besides that, when we search using Sindice, users can group the results that come from the same data set as well, and it is quite useful to classify the groups of information.

## A.6 Falcon

Falcon is a tool for searching and browsing RDF data [49, 88]. Falcon provides keywords search for semantic web entities. Falcon presents type and label of each semantic web entities in a result page. To use falcon users require experience in semantic web concept due to most of the visualisation are present in semantic form.

## A.7 Information Workbench

The Information Workbench is one kind of search engine that aims to improve state-of-the-art searching in the semantic web data environment [43, 92]. It is aimed at the general user to enable them to discover and search for information like wiki. The strong point of this application is that it provides a human readable format like wiki view and also allows a user to change the view into triple and graph formats as well. Moreover, it has function to filter the picture that is related to the topic of interest as well. I like the concept that we can navigate through the nodes and let it present the information for us. It has a feature to export information and go onto the website as well.

## A.8 mSpace

mSpace is also know as a facet browsing tool. It can be applied to both semantic web data and relational data [44, 45, 90]. The main feature of mSpace is to filter irrelevant data; therefore it will be easier for users to find what they want. Moreover, mSpace allows users to add and remove the filter column which is more useful for users to adjust their filtering criteria. mSpace allows users to use natural language search and display results in a summarized way, which is close to the normal web application. In this demo version the mSpace does not provide features for slicing, tagging, and grouping the data. The last feature that I noticed was the export data; once users have got data they can export it in a format that can be used by other programs.

## A.9 Visual Data Web Tool

Visual Data Web Tool is a set of visualisation tool for presenting RDF data [91]. it aims for making the Semantic Web Data visually more experienceable and also for general users who have less knowledge about the underlying technologies. There are four visualisation tools have already been develop in the project which are RelFinder, SemLens, Gfacet, and tFacet. The detail of each tool is explained by the following bullets;

- The RelFinder is a tool that aim to provide an overview of semantic Graph: It extracts and visualizes relationships between given objects in RDF data and makes these relationships interactively explorable. Highlighting and filtering features support visual analysis both on a global and detailed level.
- SemLens is a visual tool that allows to arrange objects in a scatter plot and to analyze them by user-defined semantic lenses. The lenses can be independently

defined for each of the objects' properties and can be combined by logical operators. The scatter plot provides a global overview and supports the discovery of dependencies and correlations also in large datasets. The semantic lenses facilitate the local analysis by providing an intuitive metaphor for the definition of semantic filters.

- Gfacet is adapt the concept of graph based representation and facet browsing together to show and allow user to discovery data. The application is still very slow for representing the data and not stable yet. The concept seems interesting in that each node represent thing and it related property or object. Users are able to filter results, show the relationship of the object which user can filter the result inside the node.

- tFacet applies known interaction concepts to allow hierarchical faceted exploration of RDF data. The aim is to facilitate ordinary users to formulate semantically unambiguous queries so as to support the fast and precise access to information. Used interaction concepts are e.g. a directory tree and interchangeable columns that are already well-known from other applications. The directory tree, for example, is used to enable the intuitive exploration and selection of hierarchical facets.

## A.10 Exhibit

Exhibit is one of the mash-up tools that is designed for enriching data on the web [46, 80]. When Exhibit is applied to the semantic web data, it enhances the ability of the general user to get and discover knowledge from the data in their own way. There are many example prototype projects that run by using Exhibit to visualise data; for example, data about the Noble Prize winner or US president. Exhibit applies a facet browsing technique for finding data as well. Moreover, it has the ability to sort information and display it in different views, such as a timeline, map, picture, or just a normal block of text.

## A.11 Swoogle

Swoogle is a semantic document search engine, which uses a crawler-based indexing and retrieval system for the semantic web documents [47, 48, 55]. Swoogle supports both RDF and OWL documents. It analyses and computes useful meta-data properties and their relationships concerning indexing. Moreover, Swoogle supports both character N-Gram and URIs as an input for matching between users query and semantic documents.

## A.12 Explorator

The Explorator is an exploration tool that allows users to import a dataset and explore it in the form of a nested block [33, 78].It can represent the relationships between class, property, and value as well. The Explorator has a strength in that it provides a set operation for users to perform an intersection and union between different data. They also used it with the feature of triple operation as well. There is no doubt that this application is appropriate for semantic web experts or ontologists, and even domain expert who want to explore information in deep detail in their own way.

## A.13 RKBExplorer

RKBExplorer applies the concept of graph based representation and facet browsing together as well [50, 50, 79]. It is, however, different from Gfacet because the data for filtering is not embedded in the node. RKBExplorer provides information on the subject we are looking for as well but it is more likely to emphasise the structure of the semantic data due to the small amount of information that is represented in the top right corner.

## A.14 Google Public Data Explorer

Google Public Data Explorer is an exploration tool in which rich visualizations are embedded. It can explore and visualize a large dataset through the rich visualization library [67, 68]. Google Public Data Explorer uses a DSPL data format as an intermediate language to describe the structure of data. It is designed to visualize all data that has been converted to a data table in the content separated value (.csv) format. The DSPL file contains data, which is presented in a CSV format, and the XML file contains the concept and composition structure of the data. Google claims that users dont have to be a data expert to use their system. In this thesis we raise issues that we had during importing our own dataset. Visualising data through Google Public Data

Explorer requires a lot of technical skill with SPARQL and knowledge of XML. The application was not as simple as we expected. However, it supports a technical user to easily explore useful information rather than discover it from the text-based user interface.

## A.15 IBM Many Eyes

IBM Many Eyes is an exploration tool that supports users to discover data and present it through the rich user interface [52, 77]. The aim of this project is not only to serve as a discovery tool for individuals but also as a medium for discussion among users. To present data through the IBM Many Eyes application, users have to transform data into a data table format. IBM Many Eyes also requires users to provide meta-data in order to help the system match the appropriate visualization library by annotating the schema of the data table.

## A.16 DERI Pipes

DERI Pipes is a visual programming tool for transforming and mashing-up data [53, 81, 82]. It extends the concept of Yahoo DERI Pipes but focuses on the mash-up semantic web data from data sources that are published online, such as DBPedia and Freebase. It provides functions to retrieve and integrate these data sources together through the graphic user interface. It also supports SPARQL and XQuery as query languages to retrieve semantic data.