Using Information Visualization Techniques to Support Web Service Discovery

By

Simone Beets

Submitted in fulfilment of the requirements for the degree of Magister Scientiae in the Faculty of Science at the Nelson Mandela Metropolitan University

December 2011

Supervisor: Prof. J. L. Wesson

Declaration

I, Simone Beets, hereby declare that the dissertation for the degree Magister Scientiae is my own work and that it has not previously been submitted for assessment or completion of any postgraduate qualification to another University or for another qualification.

Simone Beets

Acknowledgements

I would like to express my gratitude to my supervisor, Professor Janet Wesson for her continued support throughout my work on this degree. Her insight into this research, invaluable advice, feedback and direction helped to keep me motivated and interested in the work. Working with Janet also helped me grow as a student and a person.

I would also like to thank:

- Melisa Koorsse for her assistance with the usability evaluations;
- Bradley van Tonder for his support with statistical analysis; and
- Clayton Burger for the technical checking of this dissertation.

I would like to thank the Telkom / NMMU Centre of Excellence for funding this research. Finally, I would to thank the Department of Computing Sciences for the support I received throughout my degree.

Summary

The increasing number of web services published over the Web highlights the need for an effective method for users to find appropriate web services. Existing web service discovery methods do not effectively aid a user in finding suitable web services. The current methods provide textual lists of web services that the user is required to explore and manually evaluate. Thus, these methods lead to time-consuming and ineffective web service discovery.

The aim of this research was to investigate using information visualization (IV) techniques to effectively support web service discovery. The node-and-link network IV technique was selected as the most appropriate IV technique to visualize web service collections. A prototype, called SerViz, was developed as a tool for interactive visualization of web service collections incorporating the node-and-link IV technique and an alphabetical list-based technique. SerViz used the ProgrammableWeb web service collection as the sample web service collection. A usability evaluation was conducted to compare these techniques. Ninety percent of participants preferred the network IV technique for visualizing web service collections. The network IV technique was also faster for browsing.

Several usability problems were identified with the network IV technique. This motivated a need for implementing an alternative IV technique in SerViz. The node-and-link tree IV technique was selected as it was more structured than the network IV technique. A usability evaluation was conducted to compare the network and tree IV techniques. Participants slightly preferred the tree IV technique as the technique to visualize web service collections. The tree IV technique was faster for browsing the web service collection while the network IV technique was faster for searching and filtering.

This research has determined that IV techniques can be used to effectively support web service discovery. Future work will involve using IV techniques to support collaborative web service discovery.

Keywords: Web Service Discovery, Information Visualization, Web Service Collections, Information Visualization Techniques

Table of Contents

List of Figures

List of Tables

List of Abbreviations

API	Application Programming Interface
APP	Atom Publishing Protocol
BoT	Bag-of-Tricks
CSUQ	Computer System Usability Questionnaire
DoD	Details-on-Demand
DoI	Degree-of-Interest
ebXML	electronic business XML
FDL	Force Directed Layout
GML	Graph Mark-up Language
GUI	Graphical User Interface
HCI	Human-Computer Interaction
HTML	Hypertext Mark-up Language
HTTP	Hypertext Transfer Protocol
ICE	Information Connections Engine
IV	Information Visualization
NMMU	Nelson Mandela Metropolitan University
QoS	Quality-of-Service
QWS	Quality-of-Web-Service
REST	Representational State Transfer Protocol
RPC	Remote Procedure Call
SMTP	Simple Mail Transfer Protocol
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SOC	Service Oriented Computing
SOIA	Service Oriented Integration Architecture
SSDL	SOAP Service Description Language
TML	Tree Mark-up Language
UBR	UDDI Business Registry

UDDI	Universal Description, Discovery and Integration
UI	User Interface
UML	Unified Modelling Language
URL	Uniform Resource Locator
WS-I	Web Service Interoperability
WSDL	Web Service Description Language
WSIL	Web Service Inspection Language
XML	Extensible Mark-up Language
XMTP	XML Mail Transport Protocol
XSLT	Extensible Style-sheet Language Transformations

Chapter 1: Introduction

1.1. Background

Web services are software systems to aid machine communication over a network (W3C Web Services Architecture Working Group 2004b). Web service discovery provides a method of searching for web services that are specific to the requirements of a user (W3C Web Services Architecture Working Group 2004a). The increasing number of web services published over the Web highlights the need for an effective search technique for users to find appropriate web services (Al-Masri and Mahmoud 2007b; Sabou and Pan 2007). Existing web service discovery methods do not effectively support the user in finding suitable web services. The current techniques provide textual lists that the user is required to explore and manually evaluate to select appropriate web services to use. Minimal information is provided to distinguish between web services of similar functionality. Thus, these methods can result in time-consuming and ineffective web service discovery.

Various methods exist for discovering web services. The most well-known method is web service discovery based on the UDDI (Universal Description, Discovery and Integration) standard. There are several limitations with this standard. The two most notable issues are that UDDI was not designed for web service discovery and the standard's inability to provide quality-of-service parameters (QoS) for web services contained in the UDDI repository (Al-Masri and Mahmoud 2008a). Publishing sites such as XMethods and RemoteMethods provide another method of web service discovery (Bachlechner *et al.* 2006; Hagemann, Letz and Vossen 2007). Publication sites are dedicated public registry web sites that provide web services for service discovery and utilization. A third web service discovery method includes the use of general-purpose search engines to search for web services based on WSDL (Web Service Description Language) files (Bachlechner *et al.* 2006). Search engines are typically used to search for web pages. Web pages and web services differ in that web pages contain more detailed textual information for the search and information retrieval process. A limitation of using search engines as a web service discovery method is that the web service search results are displayed in textual lists similar to a web page search (Al-Masri and

Mahmoud 2008a). The problems identified with these techniques limit the ability to accurately and effectively discover web services.

Visualization of the results of web service discovery is a novel approach in service oriented computing (SOC) (Chua, Yuan and Kim 2007). Information overload is a key concern of the World Wide Web (Keim 2002). Visualization tools and techniques have been used to improve the visualization of web-based search results (Keim 2002; Rivadeneira and Bederson 2003; Carpineto *et al.* 2009; Tilsner, Hoeber and Fiech 2009), but have not been widely applied to visualizing web service collections. It needs to be determined whether information visualization (IV) techniques can be applied to support effective web service discovery.

1.2. Relevance of Research

As the number of available web services rapidly increases, the need becomes greater for an effective web service discovery method to support users in finding suitable web services to use. Current web service discovery methods provide lengthy textual lists that the user is required to explore to find relevant web services to use.

Little evidence has been found regarding the application of IV to web service discovery. This research will investigate how IV can be applied to web service discovery and whether this application provides effective support for web service discovery. The aim of this research is to improve the presentation of web service collections to support a user in effectively finding suitable web services.

1.3. Research Outline

The research outline of this research project is described by identifying both the problem and thesis statements. The research objectives and corresponding research questions are outlined in this section. The research methodology used to achieve the identified research objectives is described in detail. The scope and constraints as well as the envisaged contribution of the research are discussed and conclude this section.

1.3.1. Problem Statement

Current web service discovery methods do not support effective web service discovery. Existing registries, repositories and search engines provide lengthy textual lists which the user is required to manually browse to determine which web service is the most appropriate.

In addition, few parameters are provided for comparison of similar web services. The process of discovering the most suitable web service is currently time-consuming and ineffective.

1.3.2. Thesis Statement

The thesis statement for this research is defined as follows:

Information visualization techniques can be used to effectively support web service discovery.

1.3.3. Research Objectives

The objectives of this research are:

1. To identify the existing problems with web service discovery (Chapter 2);
2. To identify the IV techniques that can be used to support web service discovery (Chapter 3);
3. To develop a web service discovery tool using suitable IV techniques (Chapter 4); and
4. To evaluate the effectiveness of the proposed IV techniques (Chapters 5 and 6).

1.3.4. Research Questions

The following main research question will be addressed by this research:

> *How can information visualization techniques be used to effectively support web service discovery?*

The above research question will be answered by addressing the following sub-questions:

1. What are the existing problems with web service discovery?
2. What IV techniques can be used to support web service discovery?
3. How can IV techniques be applied to support web service discovery?
4. How effective are these IV techniques in supporting web service discovery?

1.3.5. Research Methodology

The research for this project is quantitative and measurable, and thus a positivism philosophy will be used for the project with a deductive approach. The project will make use of descriptive and experimental research. The research methods outlined in the following sub-sections will be used to address the identified research objectives and answer the corresponding research questions. This research will make use of literature study, prototyping and experimental research methods.

1.3.5.1. Literature Study

Literature studies are used to provide a summary of a topic in a specific research area (Hofstee 2009, 121). A literature study will be used to investigate the problems of existing web service discovery methods. Web services and web service discovery will be defined and described. The current methods of web service discovery will be discussed in detail. Shortcomings will be identified for each web service discovery method discussed. Criteria will be proposed for web service discovery and the existing web service discovery methods will be compared using these criteria. The problems of existing web service discovery methods will then be determined.

The literature study will also be used to identify which IV techniques will be suitable to visualize web service collections. IV will be discussed to provide a general background of this field. A brief discussion will be given of suitable IV techniques for different data types. The data structure of web service collections will then be investigated to identify which IV techniques would be suitable to visualize these web service collections. IV requirements will then be proposed to determine how IV can support web service discovery. Existing applications of IV to web service discovery will then be identified and reviewed to determine the shortcomings of existing systems.

1.3.5.2. Prototyping

A prototype can be used to express research in a useful way (Olivier 2009, 9). A prototyping or proof-of-concept method will be used to determine whether IV techniques can be used to visualize web service collections. An IV tool will be developed based on the problems identified with current web service discovery methods and IV requirements and will incorporate one or more IV techniques. A sample web service collection will be selected for the IV tool. Appropriate implementation toolkits will be compared to determine which toolkit will be suitable to develop the tool. The selected IV technique(s) will then be compared with an existing technique of visualizing web service collections using an experimental research method.

1.3.5.3. Experiment

An experiment is conducted to assess a theory or to examine the outcome of a given intervention (Hofstee 2009, 128). Usability evaluation(s) will be conducted to evaluate whether the IV technique(s) implemented in the IV tool provide effective support for the discovery of web services. The prototype will be compared with an existing technique used for web service discovery that does not incorporate visualization. If more than one IV technique is selected, the IV techniques will be compared with one another to determine which IV technique is more suited for web service discovery. Performance metrics and questionnaires will be used to determine the effectiveness of the prototype.

A limitation of this research will be that not all the problems identified with existing web service discovery methods can be addressed by this research. Only those problems relating to the visualization of web service collections will be addressed.

1.3.6. Scope and Constraints

The focus of this research will be on improving the presentation of web service collections. IV techniques will be selected to be applied to web service collections. These IV techniques will be evaluated to determine which IV technique(s) will be the most appropriate techniques to adopt. A prototype will then be developed using the selected IV technique(s).

As mentioned in the previous section (Section 1.3.5.), not all the problems identified with existing web service discovery methods will be addressed by this research. Only those problems pertaining to the presentation of web service collections will be addressed. The research will not include evaluating the search facility used for web service discovery. Various algorithms have been developed to improve this web service discovery component. It will be determined whether browse and/or search facilities should be provided to aid in the web service discovery process by analyzing existing web service discovery methods, but the search facility will not be evaluated as a separate element.

1.3.7. Envisaged Contribution

Limited research has been found regarding the application of IV techniques to web service discovery. The contribution of this research will be the IV technique(s) that are selected to be incorporated for web service discovery. These IV technique(s) will be identified in order to improve the presentation of web service collections.

Criteria will be identified to compare current web service discovery methods. IV requirements for web service discovery will also be proposed. Existing web service collections will be analyzed to determine which IV technique(s) will be the most appropriate. Usability evaluation(s) will be used to evaluate whether IV can be used for effective web service discovery. These evaluation(s) will assist in determining any design recommendations or suggestions for improvement.

1.4. Chapter Outline

The research questions outlined in the dissertation chapters and addressed using the corresponding research methods are summarized in Table 1-1.

Table 1-1 Summary of Research Questions, Methods and Chapters

Research Question	Research Method	Chapter
1. What are the existing problems with web service discovery?	Literature Study	Chapter 2
2. What IV techniques can be used to support web service discovery?	Literature Study	Chapter 3
3. How can IV techniques be applied to support web service discovery?	Prototyping	Chapter 4
4. How effective are these IV techniques in supporting web service discovery?	Experiment	Chapter 5 Chapter 6

This chapter introduced the topic and motivated the purpose of this research. This chapter has also introduced concepts related to this research. The problem statement, research questions, research objectives and thesis statement of the research were presented. The scope and constraints were identified. The research methodology to be utilized for the research was also discussed. Envisaged contributions of this research were then discussed.

The literature study will be covered in Chapter 2 and Chapter 3. Chapter 2 will introduce and discuss web services in detail. The concept of web service discovery will be described and existing web service discovery methods will be discussed. Advantages and shortcomings of these methods will be identified. Web service discovery criteria will be identified and problems with current web service discovery methods will be determined by comparing each web service discovery method with these criteria.

IV will be discussed in Chapter 3. IV concepts will be introduced and IV techniques will be discussed in terms of visualizing different data types. Advantages and shortcomings of the different IV techniques will be identified. Existing web service collections will be analyzed to determine the web service collection data structure and suitable IV techniques. The most appropriate IV technique(s) to visualize web service collections will then be identified. IV requirements for web service discovery will be proposed. Existing IV applications for web service discovery will be discussed to determine the features and shortcomings of these systems.

Chapter 4 will discuss the design and implementation of the IV tool for web service discovery. The sample web service collection that will be used for the IV tool will be identified. Tasks to be supported by each technique implemented in the IV tool will be discussed. The visualization techniques used to visualize the web service collection will also be specified. The selection of an appropriate implementation tool will be discussed. This chapter will discuss the design and implementation of the techniques provided by the IV tool and any design issues and improvements to be made to the IV tool.

The most appropriate IV technique selected to visualize web service collections will be compared with an existing web service discovery technique in Chapter 5. An alternative IV technique will be identified and compared with the first IV technique in Chapter 6. Usability evaluation(s) will be used to compare the different techniques used to visualize web service collections in the IV tool implemented in Chapter 4. Based on the results of a first usability evaluation, an additional IV technique may be required. The evaluation(s) will be used to determine whether the IV tool provides effective support for web service discovery.

Chapter 7 will conclude the dissertation. The research objectives will be reviewed and research achievements will be discussed. The theoretical and practical contributions of this research will then be identified. Limitations of the research will be documented and problems encountered will be discussed. Future work to be conducted on this topic will then conclude the chapter. Conclusions will be made on whether or not IV techniques can effectively support web service discovery.

A summary of the chapter outline is shown in Figure 1-1.

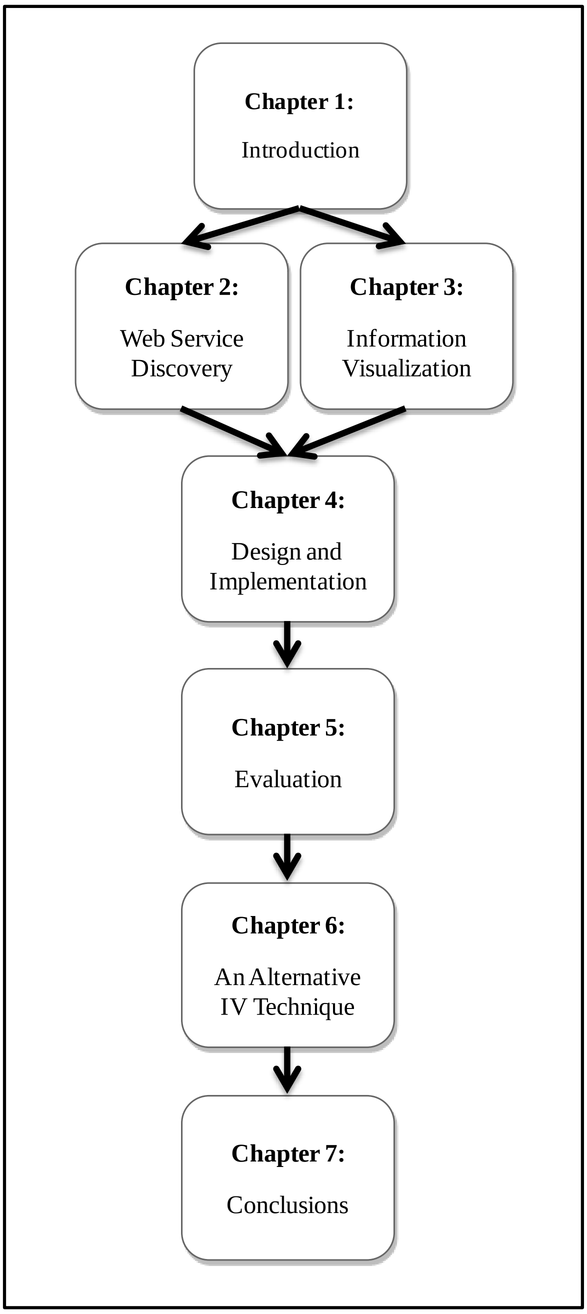


Figure 1-1 Chapter Outline

1.5. Conclusion

Web service discovery is an important research area in SOC. As web services are becoming increasingly popular, there is an increased need for an effective web service discovery method to support the user in finding suitable web services to use.

This research will determine how IV techniques can be applied to web service discovery. The aim of the research is to determine whether IV can improve the presentation of web service collections.

The research will involve a research methodology comprising a literature review, prototyping and experimental research. These research methods will be used to address the identified research objectives. The following chapters will address the identified research questions. Chapter 2 will address the first research question by identifying the problems with existing web service discovery methods.

Chapter 2: Web Service Discovery

2.1. Introduction

Web service discovery is an important research area in the service oriented computing (SOC) field due to the increasing need for web service utilization (Makris *et al.* 2005). Web service discovery needs to be investigated to identify which information visualization (IV) techniques can be applied to web service discovery. The chapter addresses the first research question identified in Chapter 1, namely w*hat are the existing problems with web service discovery?*

This research question is answered by addressing the following sub-questions: what are web services; what methods exist for web service discovery; and what are the shortcomings of existing web service discovery methods.

The above-mentioned sub-questions are addressed by the corresponding sections in the chapter.

2.2. Web Services

The first sub-question is addressed by identifying the components of web services, providing a definition of a web service utilizing these components and outlining the general web service structure. The benefits of using web services are then identified, the web service users are specified and the web service properties used to distinguish one web service from another are explained.

2.2.1. Components of a Web Service

Web services are software systems to facilitate machine communication over a network (W3C Web Services Architecture Working Group 2004b). The web service components are defined below.

The Web Service Description Language (WSDL) describes a web service in an XML (Extensible Mark-up Language) format (W3C Web Services Architecture Working Group 2004a). The WSDL describes the web service interface, which includes the service's location on the Web as well as the functionality the web service provides and interaction specifications (Cavanaugh 2006; Bachlechner and Fink 2008). The user makes use of the web service interface to find and utilize a web service (Bashir *et al.* 2010). An alternative to the default WSDL is the SSDL (SOAP Service Description Language) (Parastatidis and Webber 2005). SSDL is focused on web services which make use of SOAP (Simple Object Access Protocol) where WSDL provides a general structure for a description of a web service.

The Universal Description, Discovery and Integration (UDDI) standard is a method used to publish web services in a public or private UDDI registry to allow web service providers to make their web services available and allows users to find and utilize these web services (Cavanaugh 2006). Public UDDI registries are no longer available (Wu and Chang 2007; Platzer, Rosenberg and Dustdar 2009). Other registries that provide an alternative to UDDI but that are not as well-known include ebXML (electronic business XML) and WSIL (Web Service Inspection Language). The ebXML standard is a centralized web service registry which allows businesses to communicate and engage (Dustdar and Treiber 2005). The registry is broader than the UDDI standard as it is able to store more data than UDDI. The WSIL "provides a distributed metadata model for web service information" (Dustdar and Treiber 2005, 160). That is, there are no limitations to the type of web service information published for a web service.

SOAP is a protocol that provides arrangement and communication rules of a SOAP message. The SOAP messages provide a means of communication with the web service (W3C Web Services Architecture Working Group 2004a). A developer uses the WSDL to assemble SOAP messages to exchange information with the web service over HTTP (Hypertext Transfer Protocol) in the form of service requests and responses in a specified format documented in the WSDL definition (Cavanaugh 2006; Bachlechner and Fink 2008). SOAP is supported by major software platforms and limits the integration challenges of web services (Landre and Wesenberg 2007). The alternative to SOAP is REST (Representational State Transfer Protocol). REST is "a software architectural style for distributed systems such as the Internet..." (Landre and Wesenberg 2007, 931). The main difference between SOAP and REST is that REST makes use of "simple course [sic] grained operations ... to transport a resource across the network and then change the resource state locally by invocation of local accessible operations". SOAP uses refined methods that are accessed over the network corresponding to the RPC (Remote Procedure Call) programming model. The benefits provided by REST include the clear of separation between local and networked operations and that a single protocol is used for accessing different resources. SOAP has been the major communication protocol for web services, but REST is emerging as a competitor to SOAP. Landre and Wesenberg (2007) conclude that SOAP should be used for local web services and REST should be used for networked web services.

HTTP is the leading communication means among web service mediators (W3C Web Services Architecture Working Group 2004a). This protocol is used for the exchange of SOAP messages between the web service provider and web service user (Bashir *et al.* 2010). Other alternatives to HTTP are SMTP (Simple Mail Transfer Protocol) and XMTP (XML Mail Transport Protocol), but HTTP is the most widely used protocol (Landre and Wesenberg 2007).

Various definitions have been identified relating to web services. The web service definition used by this research is defined in the following section and includes the above-mentioned web service components.

2.2.2. Defining Web Services

A SOA (Service Oriented Architecture) is defined as a modern software development approach that is used to assemble distributed systems which present functionality of applications as services which have the properties of being language and platform independent (Makris *et al.* 2005; Al Hunaity 2008). This architecture is realized by web service technology (Al Hunaity 2008). A web service may also be referred to as an Internet service.

The W3C Web Services Architecture Working Group (2004b) defines a web service as follows:

> *"A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP-messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards."*

2.2.3. Structure of a Web Service

The web service structure is illustrated in Figure 2-1. A search component is required in order for a user, referred to as a client in Figure 2-1, to find and utilize a web service. A UDDI registry is a method for discovering web services over the Web. The UDDI registry contains at least a WSDL document for each web service to allow users to search for the web services. The WSDL document describes the web service. The user will search this registry provided on the Web to find the most suitable web service according to his requirements. The user and the HTTP server communicate via SOAP messages when the user requests to interact and make use of the web service.

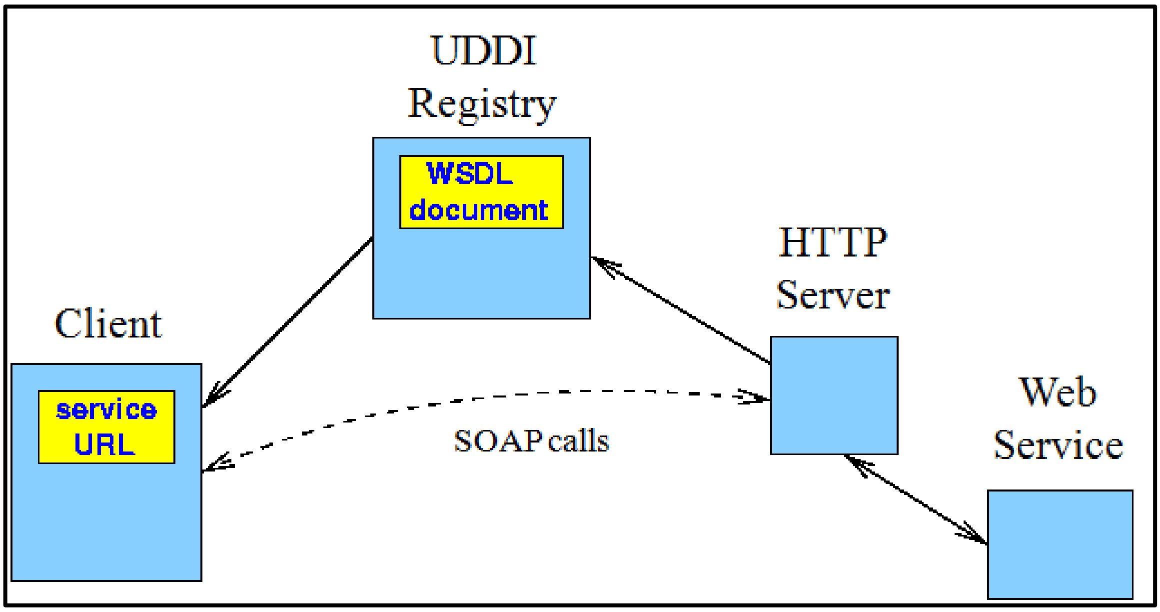


Figure 2-1 Web Service Structure (Newmarch 2003)

2.2.4. Web Service Benefits

Web service technology is widely recognized as an important aspect of service oriented integration architectures (SOIA) as this technology is moving in the direction of seamless integration (Bachlechner and Fink 2008). The benefits of using web services include the following (Cavanaugh 2006):

- *Application and Data Integration*: The use of XML and HTTP enable applications to communicate easily via web services due to the vendor, platform and language independent characteristics of XML and the ubiquitous properties of HTTP. Data integration is encouraged due to the fact that only the WSDL definition is required to communicate with a web service. It is not required to have knowledge of the implementation and format details.

- *Versatility*: Web services can be accessed in various ways. Multiple web services may be combined to form a composite web service to perform more complex functionality. Changes to the databases used by a system will not affect the web service as information is exchanged between the system and the web service.
- *Code Re-use*: Code re-use is an advantage that is produced by the interoperability and agility of web services. Services may be used by more than one developer and custom services do not need to be developed each time a new development requirement arises.
- *Cost Savings*: The above advantages of web services encourage cost savings. The ease of interoperability, as well as using and combining existing infrastructure within a business for development, improves cost savings and adds value in a business.

Any application can make use of a web service regardless of the programming language (w3schools 2010). W3schools (2010) provides a simple example of a web service that can convert temperatures from Fahrenheit to Degrees Celsius. This web service is considered as re-usable code and eliminates the need for a programmer to develop this functionality again.

2.2.5. Web Service Users

Web services can be used in the form of development components by the general population to assemble applications and other services (Zhuge and Liu 2004). Programmers and software agents make use of web services to aid in program development by re-using code, i.e. by invoking web services (Lu and Yu 2007). Web services are made available on desktop computers and mobile phones. Desktop developers and mobile phone developers can thus make use of web services.

The main stakeholders that would typically search for web services are identified as follows (Lu and Yu 2007):

- *Web Service End User:* These stakeholders are programmers that require web services to invoke directly in a written program.

- *Web Service Assembler:* These stakeholders search for web services in order to build composite web services to accomplish functionality not available from a single web service.
- *Web Service Broker:* These stakeholders are programs assisting web service assemblers to recommend suitable web services for the duration of the build process.
- *Web Service Agent:* These stakeholders are intelligent programs used to automatically search for suitable web services for use at system run time, for example in the case that the selected web service is not functioning as it should be and another web service is required to replace the selected web service.

Web services are thus used by various users and are used for different purposes by the various stakeholders.

2.2.6. Web Service Properties

Web service properties can be defined by means of functional and non-functional properties. The functional and non-functional properties of web services are discussed in this section.

2.2.6.1. Functional Properties

The functional properties of web services describe the capabilities of a web service, i.e. what the web service can accomplish (Badr *et al.* 2008). The following properties are considered to be the functional properties of a web service:

- *Web Service Name:* The web service name is the name used to describe in one or more words what the web service is capable of.
- *WSDL:* The WSDL describes the capability of a web service as an interface definition language (Bashir *et al.* 2010).
- *Textual Description:* A brief textual description is usually provided to describe the capabilities of a web service. This description enables the user to easily understand the functionality of the web service over the WSDL document.
- *URL:* A URL (Uniform Resource Locator) is provided as a link to the web service.
- *Category:* Web services are occasionally classified into different categories as this web service property is also an indication of the functionality of the web service.

Keyword searches provided by web service discovery methods to find web services use the web service name, WSDL and textual description to search for the web services.

2.2.6.2. Non-Functional Properties

Non-functional properties describe the characteristics of a web service to meet a user's goal, i.e. how the web service can accomplish what it is capable of (Badr *et al.* 2008). The non-functional properties of web services are described in terms of Quality-of-Service (QoS) and pricing (Al Hunaity 2008). QoS support is a new research area in web service technology (Yeom, Yun and Min 2006). QoS, also referred to as Quality-of-Web-Service (QWS), represents the ability of a web service to achieve the requirements of a user (Sha *et al.* 2009). QoS properties aid in distinguishing between providers as well as competing services of similar functionality (Yeom *et al.* 2006; Al-Masri and Mahmoud 2009b). QoS properties also assist in finding suitable web services by satisfying the goal of the user and the user's perception of quality (Yeom *et al.* 2006). QoS properties have the potential to assist the user in avoiding utilization of low quality services (Al Hunaity 2008). Currently, finding suitable web services is mainly based on functional web service properties and non-functional properties are rarely considered (Badr *et al.* 2008).

Various methods are used to provide QoS information and are identified as follows (Al Hunaity 2008):

- *Self-Advertising:* Expected QoS information is provided by the service providers.
- *User-experience:* Feedback is provided by users of the web services or by web service monitoring.
- *Third party evaluation of a web service owner:* This method is the most expensive to implement where the web service will be tested and the QoS information published.

QoS properties can be classified into two main categories (Sha *et al.* 2009):

- *Generic Attributes:* These attributes are properties possessed by all web services. Examples of these attributes include response time, availability, price, accessibility, reliability and security.
- *Domain-Relevant Attributes:* These attributes provide the information about the business that provides the web service.

The QoS attributes can also be categorized as subjective or objective attributes:

- *Subjective Attributes:* These attributes are provided by user evaluation of a web service. Examples of these attributes include reputation, user rating, user feedback and/or comments.

- *Objective Attributes:* Objective attributes are attributes computed during running time of the web service. Examples of these attributes include response time and availability.

Various measurements exist for QoS including those that describe implementation details, deployment details and user experience issues. The following properties are considered to be the non-functional properties of a web service:

- *Web Service Reputation:* This parameter is a measurement of users' opinions regarding the QoS information of a web service.

- *Service Level Agreement:* This agreement defines the level of performance agreed upon between the web service user and web service provider.

- *User Feedback:* This parameter involves past users of web services adding comments of their experience of using the web services (Xu *et al.* 2007).

- *Status:* Only 63% of web services available on the Web are considered to be active (Al-Masri and Mahmoud 2008b). This attribute displays whether the web service is active or inactive.

- *Version Updates:* An attribute is required to display which web service version is applicable to the web service if more than one version of the web service is available.

- *Price:* This attribute concerns the payment amount required to access the web service.

- *Rating:* This attribute could be considered to be a user rating or the web service discovery method's rating of a web service and is a rating assigned after using the web service.

Other non-functional QoS properties and their corresponding definitions were identified by Al-Masri and Mahmoud (2008c) and are displayed in Table 2-1. The units of measurement for each attribute are also displayed in the table.

Table 2-1 QoS Properties Defined by Al-Masri and Mahmoud (2008c)

	Property	**Description**	**Units of Measurement**
1.	Response Time	Time taken to send a request and receive a response	Millisecond
2.	Availability	Number of successful invocations / total invocations	Percentage
3.	Throughput	Total number of invocations for a given period of time	Invocations per second
4.	Likelihood of Success	Number of response / number of request messages	Percentage
5.	Reliability	Ratio of the number of error messages to total messages	Percentage
6.	Compliance	The extent to which a Web Service Description Language document follows the WSDL specification	Percentage
7.	Best Practices	The extent to which a web service follows the Web Service Interoperability (WS-I) Basic Profile	Percentage
8.	Latency	Time the server takes to process a given request	Millisecond
9.	Documentation	Measure of documentation (that is, description tags) in WSDL	Percentage

There are more QoS attributes available, but a sufficient subset has been discussed to provide an idea of what is considered to be a QoS attribute. When evaluating which web services are the most suitable web services to select amongst web services with similar functionality, it is important to consider the functional and non-functional properties of web services (Badr *et al.* 2008). Non-functional properties have become essential for web service discovery and selection. QoS attributes were identified as important attributes for web services to aid the user in decision making regarding which web service to select and utilize. It thus becomes important to identify which QoS attributes should be displayed for each web service to aid in finding the most suitable web services. D'Mello, Ananthanarayana and Narasimhan (2010) discussed current web service research issues. Some of these issues included providing an effective method of distinguishing between similar web services using non-functional web service properties and the need for an improved web service discovery method using web service descriptions. Web service discovery is discussed in the following section.

2.3. Web Service Discovery

This section discusses different definitions of web service discovery and describes the web service discovery process. Several methods of discovering web services are discussed and criteria for web service discovery are identified.

2.3.1. Defining Web Service Discovery

As the number of available web services on the Internet increases, a similar problem as web page discovery becomes relevant to web services as the process of finding suitable web services becomes a challenging and time-consuming task (Al-Masri and Mahmoud 2007b). It thus becomes necessary to develop an effective method of finding these services (Sabou and Pan 2007).

Web service discovery provides a method of searching for web services that are suitable and specific to requirements of a user's development (W3C Web Services Architecture Working Group 2004a). Web service discovery is considered a research challenge in the SOA domain (Guinard *et al.* 2010; Ismaili *et al.* 2010).

Web service discovery can be manual or autonomous (Al Hunaity 2008). Manual web service discovery takes place at design time and a human service user is required to make use of a discovery service to find suitable web services. Autonomous web service discovery takes place at design or run time and makes use of a discovery service or agent to search for suitable web services.

For the purpose of this research it is important to distinguish between web service discovery and web service selection (Badr *et al.* 2008). Web Service Discovery involves finding web services that match the requirements of a user. Web Service Selection involves evaluating and ranking the web services found to identify which web service(s) most appropriately meet the user's specified non-functional requirements.

2.3.2. The Web Service Discovery Process

The web service discovery components include a service user or requestor, a service provider and a discovery agent or engine, as illustrated in Figure 2-2.

The roles of the components of the web service discovery process are as follows:

- *Service Requestor / User:* Requests a web service.
- *Service Provider:* Deals with the web service request.
- *Discovery Agency / Engine:* The location where the web service is published to allow a service requestor to discover the web service.

The operations of the web service discovery process are as follows:

- *Publish:* Publication of the Service Description - It is necessary for a web service to be published in order to be discovered.
- *Find:* Finding and Retrieving Service Descriptions – The service requestor searches and retrieves a service description of the web service to invoke.
- *Interact:* Binding or Invoking of Services – The requestor or user utilizes the web service by invoking it.

The service requestor or user and service provider interact with the discovery engine. A web service will typically have at least a service name and service description attached to it. The service provider will then publish the service to the discovery engine. The service requestor then searches for the web service via the discovery engine. Once the service requestor has found the appropriate web service, the requestor then uses the service description to bind to the provider. The requestor will then interact with the web service and invoke the web service.

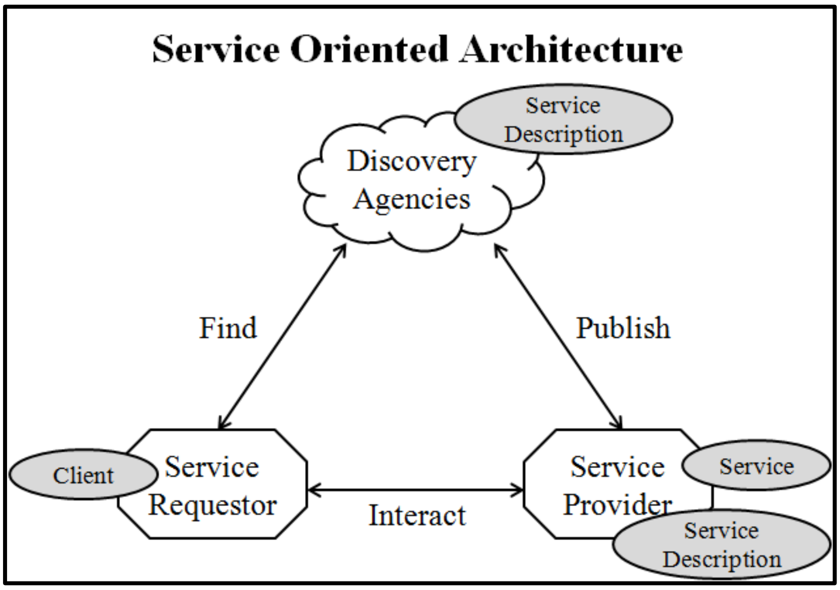


Figure 2-2 The Web Service Discovery Triangle (W3C Web Services Architecture Working Group 2004a)

Figure 2-3 illustrates the web service discovery and invocation process using a UDDI registry (Bashir *et al.* 2010). According to the figure, web services are registered in a UDDI registry. A user searches for suitable web service providers in this registry. Each web service in the registry has a WSDL document which describes the web service. The user makes use of the web service interface described in WSDL to discover and invoke the web service. A SOAP message is sent to the server requesting the use of the selected web service. The server replies with a SOAP response message.

A discovery agent for web service discovery is considered to be the interface for data and service queries made by users and provides information about the web services resulting from these queries (Berners-Lee 2009). The relationship and interaction between the service requestor or user and service provider is an important component of web service discovery.

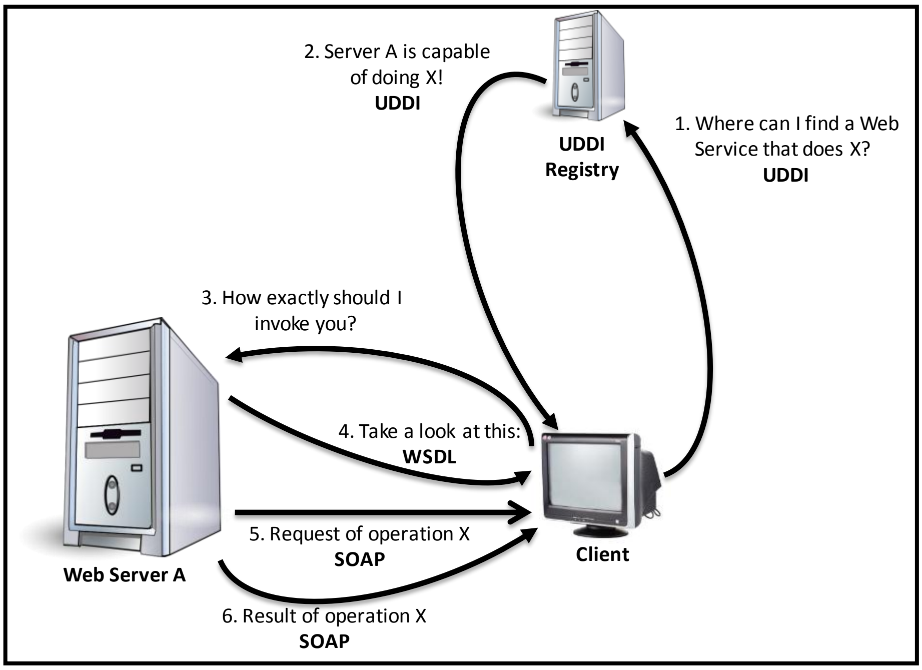


Figure 2-3 Process of Web Service Discovery and Invocation (Bashir *et al.* 2010)

2.3.3. Methods of Discovering Web Services

Various methods exist to discover web services (Al-Masri and Mahmoud 2008a). These methods include the following:

- Universal Description, Discovery and Integration (UDDI);
- Search Engines; and
- Publication Sites / Registries.

The above-mentioned web service discovery methods differ in terms of functionality and in the information that is displayed regarding each web service. Shortcomings and issues can be identified for each web service discovery method and are discussed below.

2.3.3.1. Universal Description, Discovery and Integration

UDDI is a standard technique for publishing, managing, discovering and utilizing web services (OASIS 1993-2010). The purpose of the UDDI registry is to represent information about web services. This web service information is displayed using a list-based technique (Ismaili *et al.* 2010). The UDDI registry is the most well-known method for web service discovery (Pedrinaci *et al.* 2010). Information required and provided about each web service in a UDDI registry includes a web service description, publisher information, the technical details of the service and additional metadata (OASIS 1993-2010). The main purpose of the UDDI is to publish web service information to a registry and to allow searching the registry for web services. This is a direct example of the discovery process illustrated in Figure 2-2. A UDDI registry allows the user to search for web services based on keyword searching and categorization (Rajasekaran *et al.* 2004). A UDDI registry allows a requestor or user to search for web services based on business category, business name or web service (Ladner *et al.* 2006). UDDI has a standard structure for its registry in terms of white, yellow and green pages. White pages display business information, yellow pages describes a web service in terms of categories and green pages provide the technical information about a web service (Atkinson *et al.* 2007). Many companies have supported the UDDI specification (Al-Masri and Mahmoud 2007b).

The UDDI specification has been discussed and researched in detail. Keyword based searching represents a limitation of UDDI (Rajasekaran *et al.* 2004). Web service information is limited in the amount of detail provided. If a user were to search for a web service based on keywords, the web service name and description would be the most likely information used to find the web service. If these keywords are not satisfactory to find the relevant web service a user would be required to modify the keywords in order to find the service (Rajasekaran *et al.* 2004). Keyword searching may provide unnecessary or inappropriate search results (Ram, Hwang and Zhao 2006).

Browsing categories to find a web service presents limitations as it requires a user to have prior knowledge or information about the web service they are searching for (Ram *et al.* 2006). The more categories that are added to the registry, the more generic the categories become and thus, the more difficult it becomes to browse for relevant web services (Atkinson *et al.* 2007). This limitation is applicable to any web service discovery method using categorization.

Another limitation is that the largest UDDI registries were closed and so it would be difficult to collect web service descriptions from the UDDI (Wu and Chang 2007; Platzer *et al.* 2009). Since public UDDI registries are no longer available it becomes difficult to illustrate the web service collection presentation using a UDDI registry and identify any issues that are associated with the registry's presentation. A distinguishing difference between the UDDI and a publication site is the UDDI registry's inability to represent QoS properties (Al-Masri and Mahmoud 2008c). QoS properties are necessary to distinguish between web services of similar functionality as mentioned in Section 2.2.6. Self-evaluating web service capabilities by the user can become an incredibly time-consuming task (Atkinson *et al.* 2007). Other limitations listed by Al-Masri and Mahmoud (2008a) with regards to UDDI are that the UDDI was not designed as a web service discovery search engine, it does not provide version management of web services and it does not provide guarantees of the quality of information of web services or whether the information provided is valid. Lastly, Al-Masri and Mahmoud (2008a) state that the UDDI and the Web are disconnected. The UDDI registry also does not include a form of ranking or matching and does not include the ability for feedback from users using the registry for web service discovery (Al Hunaity 2008). The UDDI registry does not provide a suitable web service discovery method that is able to satisfy the user's goal (Al Hunaity 2008). Al-Masri and Mahmoud's conclusion was that search engines would thus become the preferred web service discovery method in the future.

2.3.3.2. Search Engines

Search engines have become the latest resource for discovering web services (Al-Masri and Mahmoud 2008b; Ismaili *et al.* 2010). Examples of search engines used for web service discovery include Google, Alexa, Baidu and AlltheWeb. Web services are discovered by search engines based on the web service's WSDL document and file type of the web service. For example web services can be found on Google by specifying a keyword describing the web service with the addition of a filter in the search to restrict the search on WSDL file types, such as "FILETYPE:WSDL" (Wu and Chang 2007). The file type filter on .ASMX file extensions may also be used. Figure 2-4 illustrates the search results when searching for a web service with email functionality.

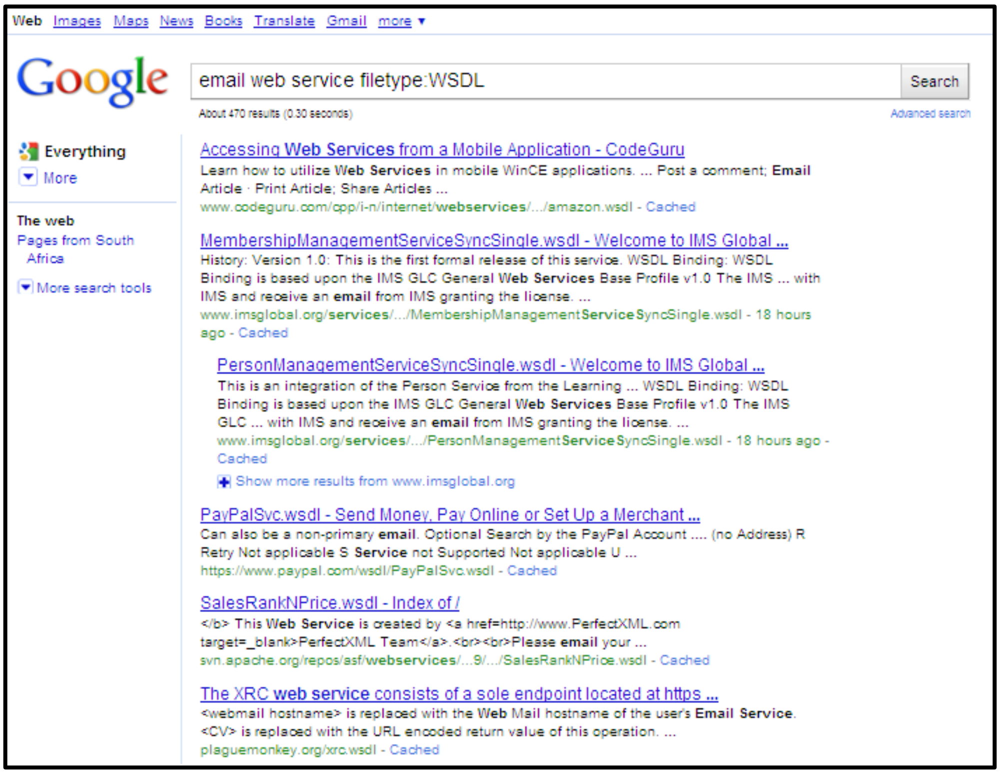


Figure 2-4 Google Search Results for an Email Web Service

Several limitations exist regarding search engines. A significant limitation of using search engines for web service discovery is that search engines are not concerned with the web service properties (Al-Masri and Mahmoud 2008b). Search engines do not acknowledge the importance of displaying web service information such as binding information, operations, ports, service endpoints or QoS attributes. Web service pages are handled in the same way as a web page regarding indexing and ranking the search results (Gomadam *et al.* 2008). Relevant web service items are thus spread throughout the search result list, which usually contains many items in the resulting list, making it difficult to find suitable web services. In addition to this, web pages and web services differ in their structure. More textual information is typically provided on a web page than for a web service and a search engine is not specifically designed to search for web services. This could lead to irrelevant web service search results. Despite this, Al-Masri and Mahmoud (2008b) concluded that search engines may become useful for web service discovery in the future. Google does not support the WSDL file type and so the validity of the discovered WSDL files cannot be guaranteed when searching for a web service by WSDL file types (Wu and Chang 2007). Search engines also lack the ability to provide or display QoS properties (Al-Masri and Mahmoud 2009a). A UBR (UDDI Business Registry) is a registry for publishing web service information. A comparison was conducted between UBRs and search engines (Al-Masri and Mahmoud 2008b). The comparison is shown in Table 2-2.

Table 2-2 UBR and Search Engine Comparison (Al-Masri and Mahmoud 2008b)

Features	UBRs	Search Engines
Contains business information?	Yes	No
Uses tModels?	Yes	No
Is publishing (listing) voluntary?	Yes	Yes
Any service-like structure?	Yes	No
Stores WSDL Documents	No	Yes
Any update interval?	No	Yes
Any support for range-based searching?	No	No
Any support for caching?	No	Possibly
Search Capabilities	Limited	Keyword matching
Any Web service subscription/business model?	Yes	No
Can handle versioning?	No	No
Validates, governs, or secures Web services?	No	No
Any support for Web service specific measurements?	No	No

The table highlights the lack of support for range-based searching and the limited search capability of UBRs and search engines. Both UBRs and search engines do not provide support for web service specific information. Search engines also do not provide any business information. The conclusion of the comparison was that although UDDI and search engines provide different techniques for discovering web services, it is still unsure whether the techniques could be combined or consolidated.

2.3.3.3. Publication Sites

An alternative web service discovery method is the use of publication sites often referred to as service registries, portals or directories. Publication sites are web sites that contain registries for web service discovery. WebServiceList, RemoteMethods, and XMethods are examples of web service publication sites (Al-Masri and Mahmoud 2008b). Searching is provided through HTML (Hypertext Mark-up Language) (Bashir *et al.* 2010). An important advantage of this web service discovery method is that it is able to provide and display QoS properties for each web service. Publication sites are becoming a popular method of web service discovery as public web service registries were recently developed to provide centralized domain-specific web service registries in the Life Sciences domain (Bhagat *et al.* 2010; Pettifer *et al.* 2010).

An example of the presentation of the RemoteMethods web service publication site is illustrated in Figure 2-5. RemoteMethods provides an initial display of categories and sub-categories from which the user can select web services. A user may also search for web services. The figure depicts the web services listed in the Business & Commerce category. The name, price, description, hits, the last update date, the number of reviews and a rating of each web service are displayed in the list. The list may be sorted by name, price, rating, version, hits and date. Links to the WSDL document, reviews and comments for each web service are also provided.

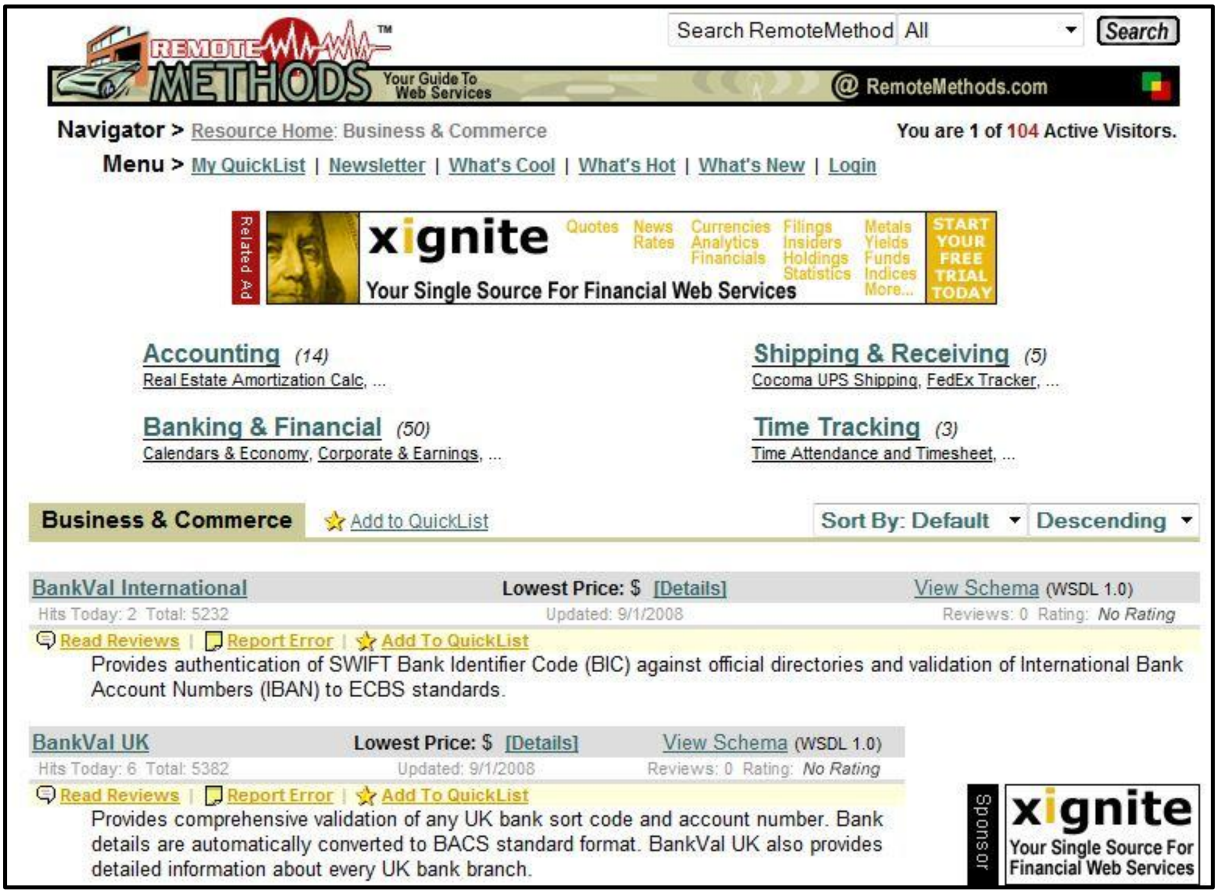


Figure 2-5 Web Services Provided by the RemoteMethods Publication Site

Figure 2-6 illustrates the presentation of web services available on the XMethods publication site. The publication site provides a lengthy list of web services sorted by the time the web service was submitted. The publisher, style, web service name, description and implementation of each web service are displayed in the list. A user is limited to manually browsing the list as no search or filter facilities are provided to find the web services that the user is searching for. Evaluating web services of similar functionality may become a difficult task with this publication site.

Publisher	Style		Service Name	Description	Implementation
SoaMoa	RPC	Try It	SoaMoa Sample Services	A sample artist registry webservice	JAX-RPC
XWebServices	DOC	Try It	XWebACHDirectory - FedACH Participant RDFIs Lookup Web Service	XML/SOAP based Web Service which provides a standards-based approach to lookup FedACH Participant RDFIs.	MS .NET
Xignite	DOC	Try It	XigniteGlobalQuotes	Provides global delayed stock quotes and for U.S. and international equities.	MS .NET
Xignite	DOC	Try It	XigniteLogos	Provides company logos for around 70% of companies listed on NASDAQ, NYSE, and AMEX.	MS .NET
Xignite	DOC	Try It	XigniteIndexComponents	This web service provides stock index component information.	MS .NET
Xignite	DOC	Try It	XigniteBATSLastSale	This web service provides real-time BATS Exchange Last Sale stock prices and trade data for US-listed stocks traded on NASDAQ, NYSE, AMEX and OTC exchanges.	MS .NET
Xignite	DOC	Try It	XigniteNASDAQLastSale	This web service provides real-time NASDAQ Last Sale stock prices and trade data for US-listed stocks.	MS .NET
tibco-sms	RPC	Try It	SMS Services	You Can Send Sms for free	
konakart	RPC	Try It	KonaKart eCommerce	Enterprise Java based eCommerce System with Web Services	AXIS
khthana	DOC	Try It	RequestStatusByCitizenID	Request Status of Road Permission By Citizen ID	MS .NET
ecubicle	DOC	Try It	Google Search Results in RSS	Provides Google Search Results in RSS Format	

Figure 2-6 Presentation of the Available Web Services Provided by XMethods

Another example of web service presentation is illustrated in Figure 2-7. This list of web services is provided by the ProgrammableWeb publication site. ProgrammableWeb is considered to be the most popular publication site (Pedrinaci *et al.* 2010). Although a search facility is provided, the search results are not entirely accurate and so the figure displays the initial web service list.

ProgrammableWeb provides a list of web services displaying for each web service the web service name, description, category and date the web service was last updated. A search and filter facility is provided to aid the user in finding suitable web services. The list can also be sorted by web service name, date, popularity and category. Although this publication site provides different methods to assist the user in finding suitable web services to select, the presentation of the web service collection may not assist the user with this task.

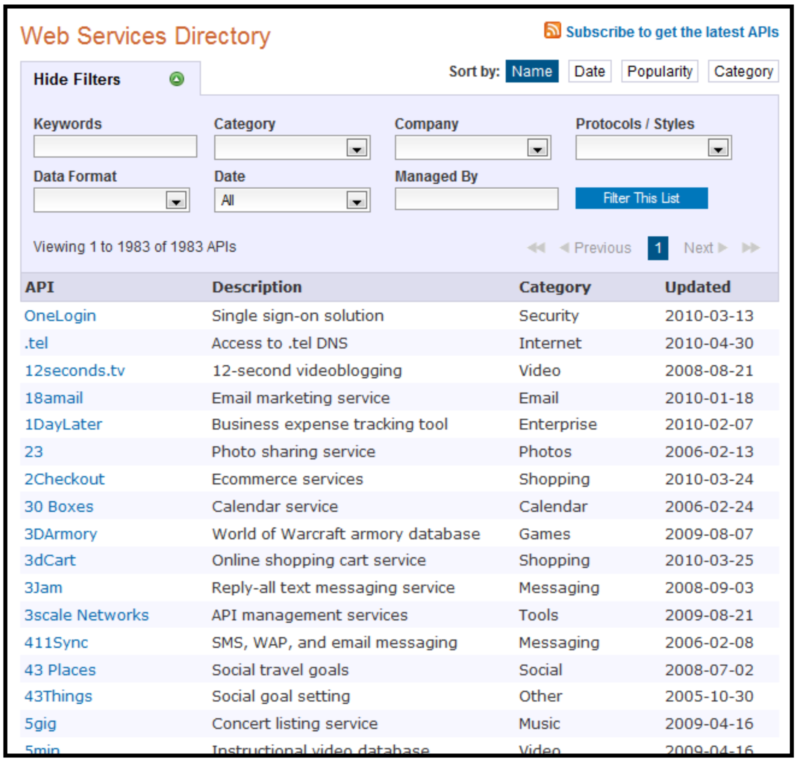


Figure 2-7 Presentation of the Available Web Services Provided by ProgrammableWeb

Web service search results of a third publication site are shown in Figure 2-8. These results are web service search results displayed by the Service-Finder publication site. The results are displayed when searching for a web service with email validation functionality.

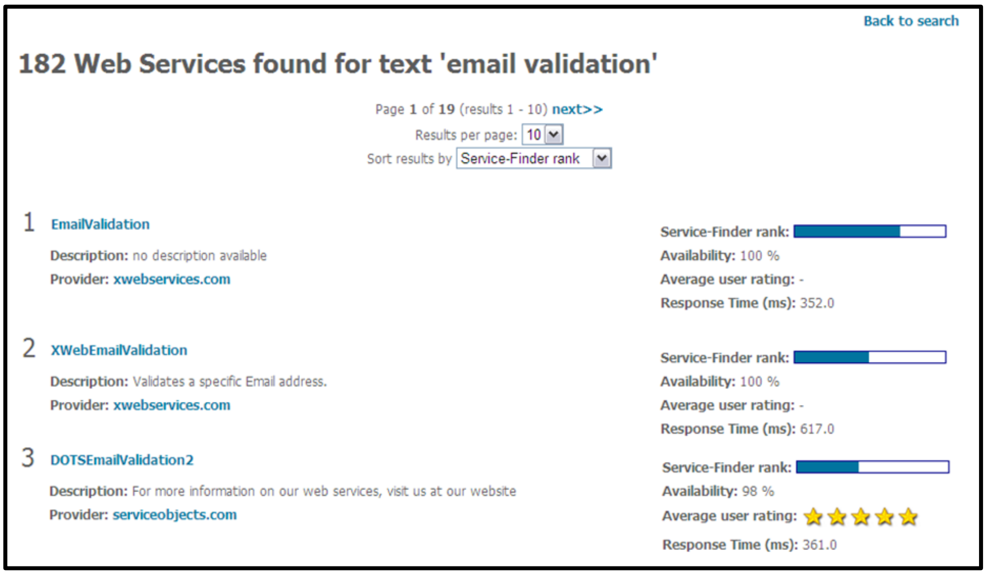


Figure 2-8 Presentation of the Available Web Services Provided by Service-Finder

The web service name, description and provider are displayed for each web service. Web services are ranked according to relevance with regards to the search keywords. Some QoS properties are also displayed for each web service including the website’s ranking of the web service, availability, an average user rating and a response time property. Displaying these web service properties assists the user in evaluating which web service is the most appropriate to select amongst web services of similar functionality.

The use of publication sites as a web service discovery method is also not without limitations. Most of these web-based registries do not follow web service standards such as the UDDI and so become susceptible to untrustworthy sources for web service discovery (Al-Masri and Mahmoud 2008b). Due to this limitation, publication sites such as Woogle and SalCentral have closed their portals. Only 60% of web services registered on these web sites are valid and so the remaining web services are inaccessible (Wu and Chang 2007). Many of these publication sites also provide only long textual lists that a user is required to browse in order to find the most suitable web service to use. Other sites arrange links to web services for manual browsing (Atkinson *et al.* 2007). Bashir *et al.* (2010) identify that, while browsers provide the required web service discovery results, a user is required to spend time filtering to find suitable web services and thus dedicated service portals or publication sites provide the most desired results. While this is encouraging, the identified limitations imply that finding appropriate web services using web service publication sites can also become a time-consuming task.

2.3.4. Criteria for Web Service Discovery

An effective method is needed for users to find suitable web services to fulfil their requirements (W3C Web Services Architecture Working Group 2004a). Web service discovery provides a method for finding these web services. Hagemann *et al.* (2007) compared different web service discovery methods according to certain characteristics. The conclusions that were made from this comparison were that a small number of public registries are expanding and increasing their web service collection size, including the ProgrammableWeb publication site.

To discover suitable web services effectively, the web service discovery methods should adhere to certain criteria. Little evidence has been found regarding web service discovery criteria. The following list includes proposed criteria for web service discovery based on the previous discussion.

- *Functional Web Service Properties:* Functional web service properties can be considered as the primary properties provided regarding web services. It is essential to provide functional properties of each web service explaining the capabilities of web services (Al-Masri and Mahmoud 2007a). This includes a textual description explaining the functionality to provide additional textual information and thus aid in the search process.

- *Non-Functional Web Service Properties:* Literature has supported the need to provide non-functional properties for each web service. Non-functional web service properties aid the user in distinguishing between web services with similar functionality. The non-functional properties support the user in finding web services that meet the user's goal and finding those web services that are able to achieve the necessary functionality with a reasonable level of QoS (Al-Masri and Mahmoud 2009c).
- *An Effective Graphical User Interface:* Providing an effective graphical user interface (GUI) so that the user can easily find and understand available and suitable web services is necessary as the user interacts with the web service discovery interface (Steinmetz *et al.* 2008)
- *Classification Facility:* Providing an effective classification method to categorize web services to aid in the discovery of web services is essential. This provides an additional method of finding web services. This should not be the only additional method of searching for web services as classification is based on the assumption that a user has prior knowledge of the web service functionality that they require and also limits informational queries (Ram *et al.* 2006).
- *Search and/or Browsing Facility:* If the number of web services contained in a dataset used for web service discovery is large, providing a search and browse facility to aid the user in finding web services should be considered. Browsing capabilities allow the user to perform informational queries as the user does not need to have prior knowledge of the required web service functionality. Search capabilities aid a user who has a specific web service query. Various studies were completed regarding the search facility used for web service discovery and so the optimal search mechanism to be used should be provided. According to Al-Masri and Mahmoud (2008c), a keyword search facility combined with an additional method to aid in web service discovery will become the accepted solution. Using a keyword search method and no other alternative method to aid in the discovery process can result in ineffective web service discovery. Filters may provide an additional component to the search facility as a user can easily refine his search.

- *Ranking Facility:* If a search mechanism is used for web service discovery then providing a ranking facility to identify the relevancy of web services in the search result list (ranking using functional as well as non-functional web service properties), provides a useful method to aid the user in web service selection (Al-Masri and Mahmoud 2007a; Gomadam *et al.* 2008).
- *Result List Sort Facility:* It may be useful to allow the user to sort web service search result lists in the order he would prefer to rank the resulting web services.

These above-mentioned aspects are considered to be important criteria to support the user in web service discovery. It still needs to be identified on what to base each criterion in order to be effectively used in web service discovery. For example, to provide effective web service discovery based on non-functional properties it would be necessary to determine the desired non-functional property combination to aid in the web service discovery process.

2.4. Shortcomings of Web Service Discovery

Methods for web service discovery were explained and limitations regarding these methods were identified in the above section. Several web service discovery criteria have been proposed. In order to identify existing problems with web service discovery, the methods for web service discovery are compared against the criteria to identify the extent of support each method provides regarding the proposed web service discovery criteria. The problems that have been identified from literature are discussed in this section.

This section identifies problems with the web service discovery methods identified previously. Each web service method is compared against the proposed web service discovery criteria. This section compares the shortcomings of the UDDI, search engines and publication sites. A summary of web service discovery shortcomings is then presented.

2.4.1. Existing Methods

It is difficult to compare the proposed criteria with the UDDI web service discovery method as public UDDI registries are no longer available. UDDI is only available via private registries with access provided to authorised users. The UDDI web service discovery method was thus compared according to previously identified limitations.

Search engines have become a popular web service discovery method. Google was used as a search engine example in the comparison.

Dedicated web service publication sites or web service portals are considered to provide the user with the desired results for web service discovery. Many publication sites are available on the Web providing their own web service collections. These web service publication sites do not follow any standard and so each publication site provides its own version of web service discovery. It is difficult to compare the web service publication site method for web service discovery with the proposed criteria due to the diversity of these sites. Publication sites considered for the comparison include RemoteMethods, ProgrammableWeb, XMethods, StrikeIron, seekDa, WebServiceX, theWebService, Service-Finder, WebServiceList and WebServiceCentral. Publication sites not mentioned were either closed or offline at the time of comparison.

2.4.2. Summary of Web Service Discovery Shortcomings

The support provided by the web service discovery methods for the proposed criteria is summarized in Table 2-3.

Table 2-3 Summary of Web Service Discovery Shortcomings

	Support		
Criterion	**UDDI**	**Search Engine**	**Publication Site**
Functional Web Service Properties	Yes	Limited	Yes
Non-Functional Web Service Properties	No	No	Yes
An Effective Graphical User Interface	Textual List, Focus on Business Information	May not be suited for web services	Textual Lists
Classification Facility	Yes	No	Yes
Search and/or Browsing Facility	Limited	Keyword Search with file type restriction	Yes
Ranking Facility	No	May not be suited for web services	Yes
Result List Sort Facility	No	No	Yes

From Table 2-3, it can be seen that publication sites provide the most support for the proposed web service discovery criteria. All three web service discovery methods provide support for functional properties of web services, although the support provided by search engines is limited. Only publication sites provide support for non-functional web service properties and a result list sort facility. The majority of publication sites also provide support for non-functional properties. Each publication site provides different QoS properties for each available web service.

All of the existing web service discovery methods have a potential problem regarding the GUI criterion. Current research regarding web service discovery components such as search or ranking do not emphasize the importance of the GUI used for web service discovery as well as the presentation of web service collections (Uddam 2008; Sellami *et al.* 2009). This property is difficult to compare as public UDDI registries are no longer available. Keyword searching is mainly provided to search on business information and so the search results mainly focus on the provider of the web service and not the web service itself. The search results are also displayed in textual lists with limited information displayed for each web service. Regarding the GUI for search engines, the user interface is effective for a web page search but may not be suitable for web service discovery as textual lists are provided similar to a web page search. User interfaces for each publication site seem to be usable but all publication sites provide lengthy textual lists of available web services.

A classification facility is provided by the UDDI and publication sites. Classification in the UDDI is supported, but it is not clear how web services are classified and how categories are identified. Most publication sites considered provide a web service classification facility and some sites even provide more specific web service sub-categories within a category.

Only publication sites provide support for browsing and search facilities. UDDI and search engines provide search facilities. Regarding UDDI, a search facility is provided to search for web services based on keyword and category. The keywords that can be searched for include business name, business category and web service. Limited search capability is supported by means of keyword searching. As far as literature is concerned it seems that browsing without a search as well as a filtering facility is not supported in the registry. Initial browsing is not supported using a search engine but keyword search capability is provided with file type restriction. Due to the fact that web service search is dealt with in the same way as web page search, relevant web services are displayed among other unnecessary results. Filter capability applicable to web services is also not supported. The majority of publication sites provide search and browse capabilities with one publication site, XMethods, only providing a browse facility. ProgrammableWeb provides a search facility with filtering based on category, company, protocols / styles, data format, date and who is responsible for the management of the web service.

Only publication sites provide ranking suited for web service discovery. No ranking support is provided by the UDDI registry according to literature. Ranking is provided by most search engines but this ranking is suited to web page searching. Only one publication site, Service-Finder, provides a ranking facility.

Finally, only publication sites provide support for a result list sort facility. Limited information is provided in search engine search results and so most search engines do not provide a sort facility. A sort facility is provided by most publication sites.

Other problems of web service discovery not highlighted by this comparison include lack of security, transaction support, service quality guarantees, execution control and lack of support for web service discovery (Al Hunaity 2008).

Public UDDI registries are no longer available. Search engines have been identified as the web service discovery method to be used more widely in the future. Publication sites provide the most support for the proposed web service discovery criteria. Publication sites have thus the potential to provide support for all the proposed criteria except for the effective graphical user interface as lengthy textual lists are displayed.

2.5. Conclusion

This chapter has identified that web services are software systems used to facilitate machine communication over a network. The benefits of web services were described including application and data integration, versatility, code re-use and cost savings. This chapter also identified the importance of providing non-functional properties for each web service.

As the number of web service available increases rapidly, an effective method of finding suitable web services becomes necessary. Web service discovery is a technique for searching for web services with specific and required capabilities. Methods that exist for web service discovery include UDDI, search engines and dedicated web service publication sites. Each web service discovery method has a set of advantages and limitations. Several criteria were proposed for web service discovery. These criteria include the provision of functional and non-functional web service properties, an effective GUI, a classification facility, a search and/or browse facility, a ranking facility and a result list sorting facility.

The existing web service discovery methods were compared using the proposed web service discovery criteria. The result of this comparison identified that problems exist with the presentation of web service collections due to the use of lengthy textual lists and the search and browse facilities. Current web service discovery methods do not focus directly on the presentation of web service collections. Only web service publication sites provide non-functional web service properties and a result list sort facility.

The next chapter will identify appropriate IV techniques that can potentially address these web service discovery problems and assist the user in effective web service discovery. The importance of using IV for visualizing large amounts of data will be discussed. The structure of web service collections will be analyzed to assist in identifying which IV technique(s) would be suitable to adapt for web service discovery. IV requirements will then be identified for web service discovery. Existing IV systems will also be discussed and compared according to the proposed requirements.

Chapter 3: Information Visualization

3.1. Introduction

Information visualization (IV) is a useful way of dealing with large amounts of data so that a user can understand and use the data easily (Fekete *et al.* 2008). The key problems with web service discovery identified in Chapter 2 involve shortcomings of the search facility and the poor presentation of web service collection information. This chapter identifies appropriate IV techniques that can be used to aid the user in finding web services effectively. The chapter addresses the second research question identified in Chapter 1, namely, *what IV techniques can be used to support web service discovery?*

This research question is addressed by reviewing existing IV techniques. IV techniques that are suitable for web service discovery are then identified and discussed. IV requirements for web service discovery are then outlined. Existing IV systems for web service discovery are discussed in terms of functionality and are compared to the proposed requirements.

3.2. Background

Information overload is a key problem addressed by IV. Many definitions exist relating to IV and these definitions are discussed in this section. The purpose of IV is then explained.

3.2.1. Information Overload

Information overload is a key concern of the World Wide Web (Keim 2002). Data is generated much faster and in greater volumes than what users can understand and examine but there is a significant need for this ability (Keim, Mansmann, Schneidewind, *et al.* 2008). Users and decision makers are required to make effective use of the vast amounts of data that are presented to them (Keim *et al.* 2009). A crucial requirement for decision makers is to mine data that is important and relevant from the large amount of data provided (Keim, Mansmann, Oelke, *et al.* 2008). If data is irrelevant, inappropriately processed or unsuitably presented then users may become lost in the data that they are required to use and understand.

Information overload may lead to loss in time, money and missed scientific or business opportunities as users are unable to effectively utilize the data they are presented. Once this problem is overcome, users can make intelligent use of the volumes of data and can comprehend advancement with regards to technology and the ability to succeed in business (Keim *et al.* 2009).

This problem relates to web service discovery as web service collections are large repositories of information. To find a suitable web service, a user would have to explore the web service collection thus possibly becoming overwhelmed. Web service collections are presented in lengthy textual lists and the methods for web service discovery do not present the web service collections in a consistent manner. As a result, this can lead to loss in time that users or developers could have used to focus on other work activities and a consequent loss in work productivity. Ineffective web service discovery can result in loss of revenue for service providers if the web services that these businesses provide cannot be found by the users or developers that need to discover the services.

3.2.2. Definition of IV

IV is a combination of representation and interaction (Yi *et al.* 2007). Representation stems from computer graphics and deals with transforming data to a suitable illustration and displaying this illustration. Representation is not time dependent. Interaction stems from HCI (Human-Computer Interaction) and deals with communication between the user and the system when the user interacts with the representation. Interaction is time dependent as it can result in a change in representation.

Many definitions have been derived for IV (Kosara 2007). Heer, Card and Landay (2005) define the purpose of IV as making use of the visual capacity of users to supplement their cognition for sense-making of the data provided. Chen (2010, 387) describes IV as a "computer generated interactive graphical representation of information". Risch *et al.* (1997, 42) define IV as a method of converting "abstract, typically non-numerical information (such as text) into more easily understandable, graphical forms". Card, Mackinlay and Shneiderman's definition (1998, 7) is considered to be the mostly accepted definition of IV and describes the concept as follows:

> *"the use of computer-supported, interactive, visual representations of abstract data to amplify cognition."*

The idea underlying these definitions is that IV increases user cognition by presenting large volumes of data visually to enable a user to analyze and query the data easily and thus extract relevant and useful information effectively. Visualization is widely used for different purposes and as a result the concept of IV can be confused with scientific visualization, knowledge visualization or visual analytics. Statistics and data mining are separate concepts. These concepts need to be defined in order to be differentiated from IV.

The main difference between scientific visualization and IV is that scientific visualization deals with scientific data provided by tests, simulations and sensors and the data to be visualized involves 3-dimensional geometries with allusion to time and space (Keim, Mansmann, Schneidewind, *et al.* 2008; Keim *et al.* 2009). IV deals with more general data abstraction with no allusion to time and space. Knowledge visualization makes use of visual representations to transfer useful information between two users (Cañas *et al.* 2005). Concept maps are used for the visualization of knowledge and other media such as images and videos are used for IV.

IV and visual analytics are correlated but remain different concepts (Keim, Andrienko, *et al.* 2008). Visual analytics combines visualization, human factors and automated data analysis for decision making. IV focuses on providing displays and useful interaction techniques for specific data. IV does not focus on data analysis processes. Data mining relies on statistical algorithms and machine learning to extract useful information (Shneiderman 2001). IV relies on providing an overview and insight to the provided data. Statistics is a composite concept encapsulating descriptive statistics, classical statistics, Bayesian statistics and exploratory data analysis (Fekete *et al.* 2008). IV stems from exploratory data analysis. The next section discusses the main purpose of IV.

3.2.3. The Purpose of IV

A number of researchers have discussed the purpose of IV. Yi *et al.* (2007) stated that a fundamental purpose of IV is to expose unclear and unidentified properties of data and the relationships between these properties. Yi *et al.* (2008) indicated that an additional core purpose of IV is to provide insight about data. Amar and Stasko (2004, 144) noted that "the status quo of information visualization is one concerned primarily with what is being visualized" in order to use the data to provide knowledge to the user. Li and Takatsuka (2004) stated that an objective for IV is to present a vast amount of information in a restricted display space.

A challenge of IV is to develop techniques and metaphors to present information that is easy to understand and observe (Rabelo *et al.* 2008; Vande Moere 2008). IV systems are useful when users are not certain what data should be extracted or when users require complex data extraction (Fekete *et al.* 2008). These systems aid users in finding specific information within a large data space. These systems should also support the user in finding data features and aid in data interpretation (Cook, Earnshaw and Stasko 2007). IV has been widely applied as a means for military and national security programs (Viégas and Wattenberg 2007).

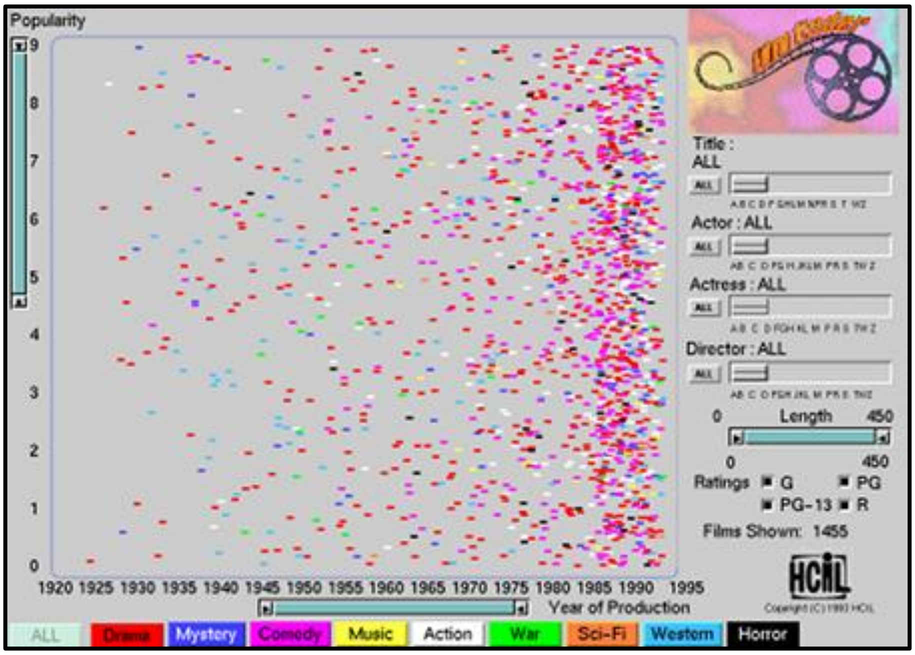


Figure 3-1 The FilmFinder System (Card 2008, 510)

Another well-known illustration of the use of IV is the FilmFinder system shown in Figure 3-1. FilmFinder is used to find films by browsing a film database (Card 2008, 510). Figure 3-1 displays a scatter-graph of films providing an overview according to popularity and the year the film was released. Different types of films are encoded using colour. Sliders are provided on the right side of the interface for the user to drill down to zoom in to more specific information. Films may be selected by double-clicking on the marker. Detail is then provided of the selected film including the name of the actors in the film. This system allows the user to find a film easier and much quicker among the vast collection of films displayed than when viewing a lengthy textual list of available films.

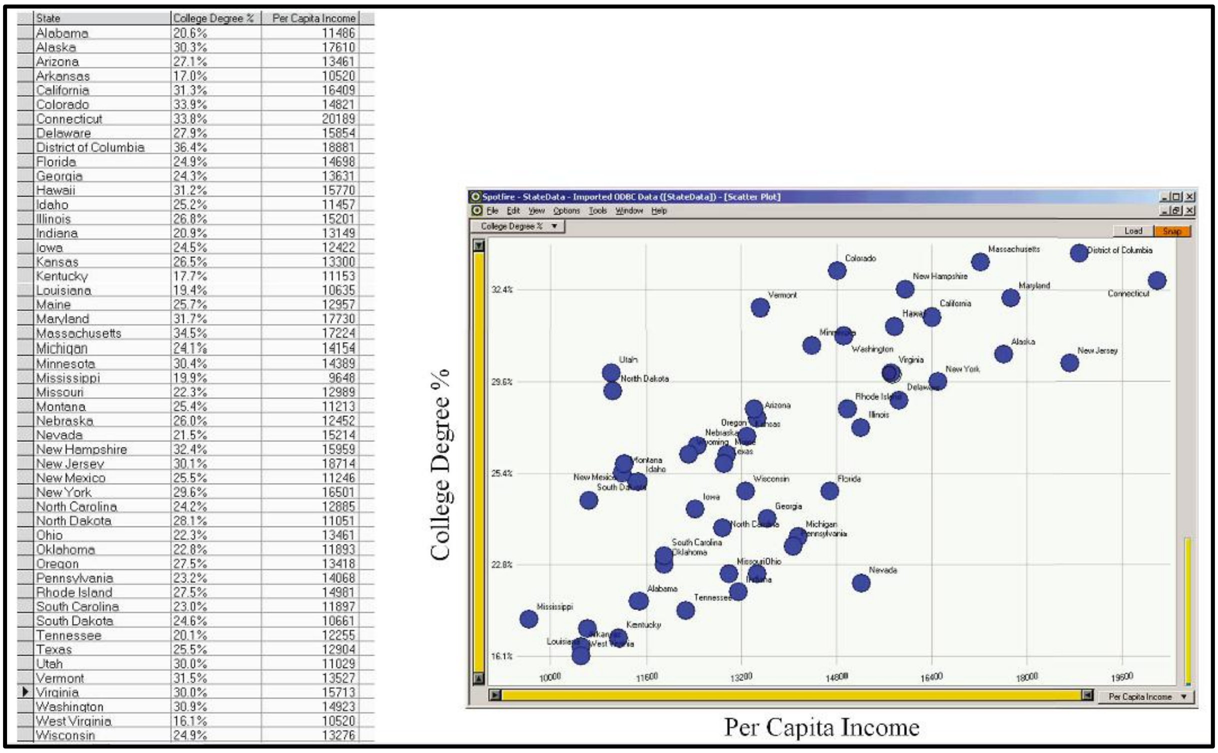
3.3. Benefits of IV

This section assists to identify the key benefits of IV. Challenges exist for IV but these are outweighed by the many benefits that IV can provide. There is a great need for sense-making of complex and ever-changing information which has led to the continuing development of IV techniques.

It is essential for all stakeholders or users such as decision makers, analysts and emergency response teams to be able to find useful information from the vast amount of data provided (Keim, Mansmann, Schneidewind, *et al.* 2008). Effective data presentation requires no explanation to gain insight and is directly useful (Amar and Stasko 2004). As the IV concept becomes established in research, the techniques and tools that are developed are becoming available to users (Plaisant 2004). IV is also being included in everyday applications such as Microsoft Outlook. Taxonomies have been identified and derived to avoid the adoption of unsuitable IV techniques (Amar and Stasko 2004). Data types and existing IV techniques can now be mapped to allow the use of the appropriate IV techniques with specific types of data. IV is a concept which is continually being researched (Viegas *et al.* 2007). This provides an opportunity of furthering and maturing the IV concept and a greater opportunity for collaboration in IV systems. As discussed previously, users can understand and analyze information faster and more effectively when using IV systems compared to textual representations (Deligiannidis, Kochut and Sheth 2006). IV assists decision makers in understanding the data to perform the decision making process quicker (Borzo 2004). IV overcomes the difficulty of visualizing the results provided by data mining (Rabelo *et al.* 2008). IV reduces the need for the user to do work to gain insight into the data he is analyzing (Keim *et al.* 2006).

A simple example to illustrate the benefits of IV is depicted in Figure 3-2. The left part of the figure displays records in a spreadsheet for the fifty states and the District of Columbia in the United States of America with the percentage of citizens in each state that have a college degree and the income of the citizens in each state (Fekete *et al.* 2008). The spreadsheet may be difficult to analyze when the user has a specific complex query for the data. The right section of the figure displays the same information visually in a scatter-plot chart where complex queries may be easy to address and much faster to answer.

IV has been used to aid in minimizing information overload (Keim 2002). IV is especially useful for large data sets. The purpose of studying information visually is to illustrate the data in a visual manner in order to enable users to gain knowledge about this data and allow users to interact with the information.



State	College Degree %	Per Capita Income
Alabama	20.6%	11486
Alaska	30.3%	17610
Arizona	27.1%	13461
Arkansas	17.0%	10520
California	31.3%	16409
Colorado	33.9%	14821
Connecticut	33.8%	20189
Delaware	27.9%	15854
District of Columbia	36.4%	18881
Florida	24.9%	14698
Georgia	24.3%	13631
Hawaii	31.2%	15770
Idaho	25.2%	11457
Illinois	26.8%	15201
Indiana	20.9%	13149
Iowa	24.5%	12422
Kansas	26.5%	13300
Kentucky	17.7%	11153
Louisiana	19.4%	10635
Maine	25.7%	12957
Maryland	31.7%	17730
Massachusetts	34.5%	17224
Michigan	24.1%	14154
Minnesota	30.4%	14389
Mississippi	19.9%	9648
Missouri	22.3%	12989
Montana	25.4%	11213
Nebraska	26.0%	12452
Nevada	21.5%	15214
New Hampshire	32.4%	15959
New Jersey	30.1%	18714
New Mexico	25.5%	11246
New York	29.6%	16501
North Carolina	24.2%	12885
North Dakota	28.1%	11051
Ohio	22.3%	13461
Oklahoma	22.8%	11893
Oregon	27.5%	13418
Pennsylvania	23.2%	14068
Rhode Island	27.5%	14981
South Carolina	23.0%	11897
South Dakota	24.6%	10661
Tennessee	20.1%	12255
Texas	25.5%	12904
Utah	30.0%	11029
Vermont	31.5%	13527
Virginia	30.0%	15713
Washington	30.9%	14923
West Virginia	16.1%	10520
Wisconsin	24.9%	13276

Figure 3-2 "A Picture is Worth a Thousand Words" (Fekete *et al.* 2008)

IV has also been used to improve web based search results. Web search results contain large amounts of data which needs to be managed and displayed in an adequate method to provide effective exploration of these search results. Keim (2002) reviewed IV techniques in relation to the type of data to be managed. Rivadeneira and Bederson (2003) made use of search result clustering to compare textual and zoomable user interfaces. The outcome of this comparison was that although users preferred the textual based clustering interface, the zoomable interface showed promise. Carpineto *et al.* (2009) provided a survey of current web clustering techniques. Tilsner *et al.* (2009) presented CubanSea, a method of using clustering to visualize search results.

To take full advantage of the benefits provided by IV, the disadvantages or challenges of IV need to be discussed. The issues are as follows:

- *Usability:* IV researchers need to provide convincing evidence of the usefulness of IV systems (Plaisant 2004; Chen 2005). Evaluation methods include controlled experiments comparing design elements, usability evaluations of IV tools, controlled experiments comparing tools and case studies. These evaluation methods may not convince a larger audience of the usability of the IV system as these methods are restricted and are not necessarily how a user would use the system in a real world setting.

- *Uncertainty:* Uncertainty of the capability of current IV systems may hinder decision making. This uncertainty is due to the following reasons: limited affordance, simple operations, predetermined representations, static representations and the decline of determinism in decision-making. The uncertainty factor is generally not dealt with in current IV systems (Amar and Stasko 2004).
- *Scalability:* Information overload was identified previously as an important aspect that IV attempts to overcome. As datasets become larger, displays become more cluttered, systems become slower and data becomes disorganized (Deligiannidis *et al.* 2006). IV systems need to be able to deal with this problem.
- *Unsuitable adoption of IV Techniques:* IV systems may only be useful when they use the appropriate IV techniques depending on what type of data is provided. If unsuitable IV techniques are adopted, the information retrieved may not be useful or understandable. IV techniques can also be developed for customized applications (Masui 1998).

The above-mentioned challenges are important issues to consider when designing an IV system.

3.4. IV Techniques

IV techniques can be used for a range of data types. Certain IV techniques are suitable for different data types. An analysis of web service collections assisted in determining which IV techniques would be suitable based on the data type of existing web service collections. The IV technique(s) that support the different types of web service collections were then analyzed further to determine which specific technique(s) would be the most appropriate to visualize these web service collections.

3.4.1. Data Types

IV techniques have been structured using a taxonomy according to data type (Shneiderman 1996; Card 2008, 523-525). The IV techniques can be classified according to the following data types (Card 2008, 525-531):

- *1-Variable:* 1-dimensional or 1-variable data consists of mapping a variable or attribute against a set of objects. Examples of this mapping include a simple textual list, a 1-dimensional scatter-graph mapping distances from home to a petrol station, a single-variable visualization mapping an object with a variable attribute and a pie chart (Figure 3-3).

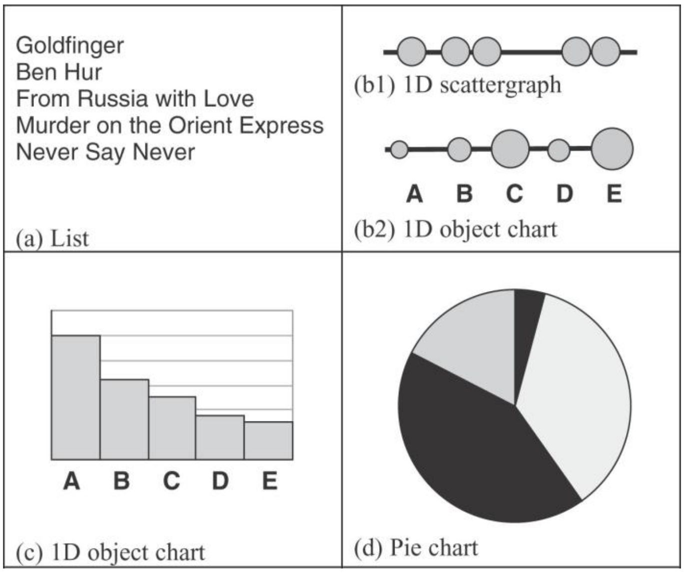


Figure 3-3 Examples of 1-Dimensional Visualizations (Card 2008, 526)

- *2-Variables:* 2-Variable or 2-dimensional data can be visualized by plotting a variable or attribute against the second variable or attribute along two basic axes and making a mark where the two variables' values coincide. An example of this visualization includes a 2-dimensional scatter-graph (Figure 3-4).

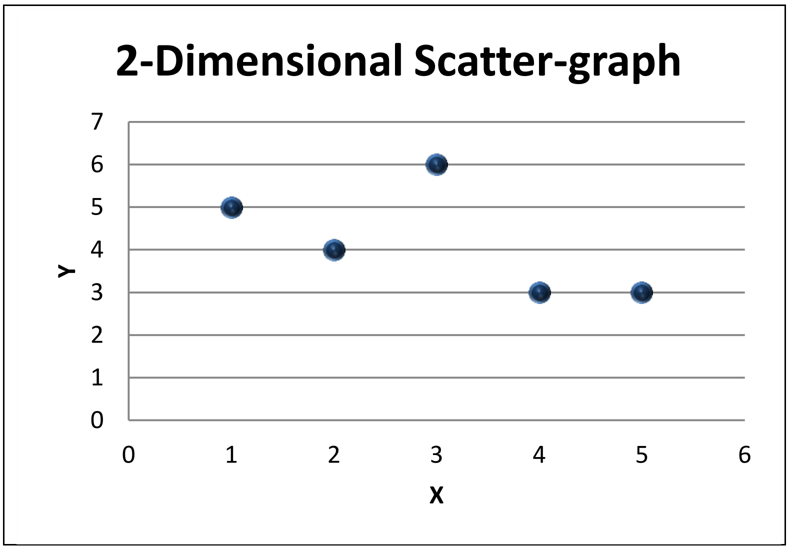


Figure 3-4 Example of a 2-Dimensional Scatter-Graph

- *3-Variables:* With 3-dimensional or 3-variable data three separate dimensions are used to encode the corresponding variables or attributes. An example of visualization of 3-dimensional data is illustrated in Figure 3-5.

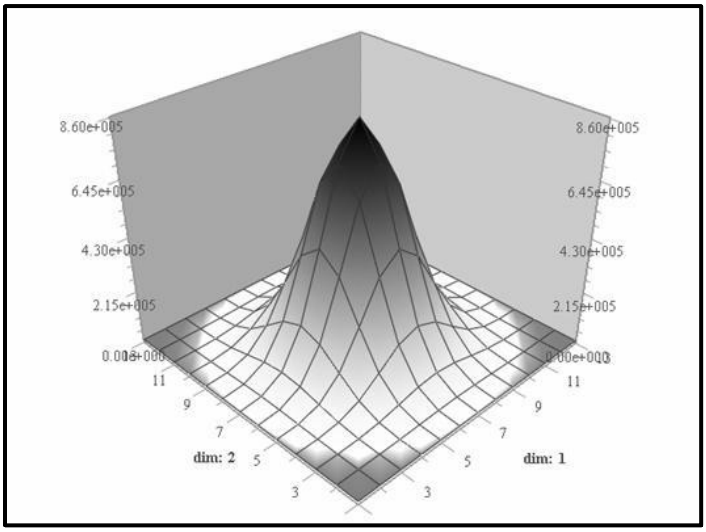


Figure 3-5 A 3-Dimensional Visualization of the Normal Distribution (Krus 2002)

- *Multi-dimensional:* N- or multi-dimensional data is an extension of 3-dimensional data and is more complex to visualize. Diagrams visualizing multi-dimensional data can be understood, but the effort required corresponds to the number of variables or attributes in the data.
- *Tree:* Trees can be used to visualize data by drawing linkages between variables and attributes that form a hierarchical data structure. Examples of tree visualizations include node-and-link trees (Figure 3-6), enclosure trees, tree maps and cone trees. The two central methods for tree visualization include a connection and an enclosure. A connection is used to join nodes in a tree using positioning to assist the readability of the visualization. An enclosure is used to encircle nested subsets in a hierarchy.

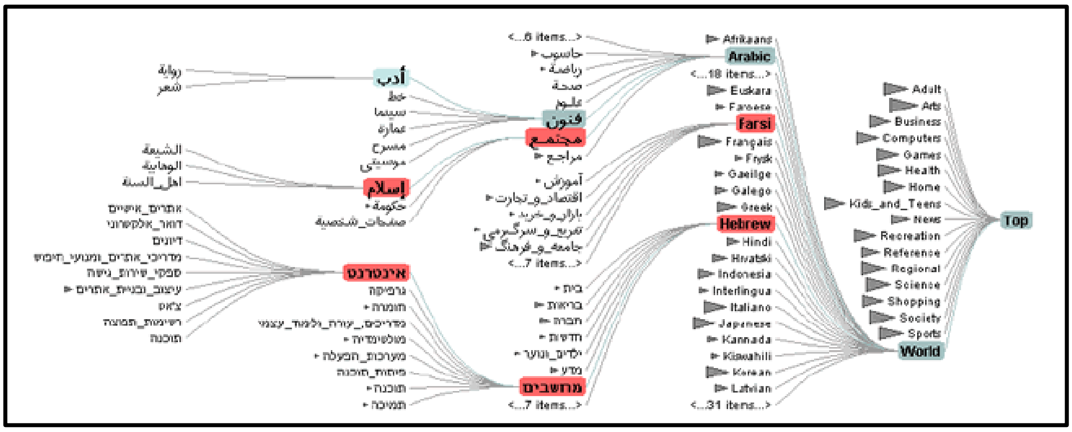


Figure 3-6 A Degree-of-Interest (DoI) Tree Visualization of the Open Directory Project (Heer and Card 2004)

- *Network:* Network data is similar to tree data but allow cycles and directional links in the data. Applications of network data visualization include visualizing the Internet and social networks. Examples of network visualizations include node-and-link diagrams (Figure 3-7) and square matrices.

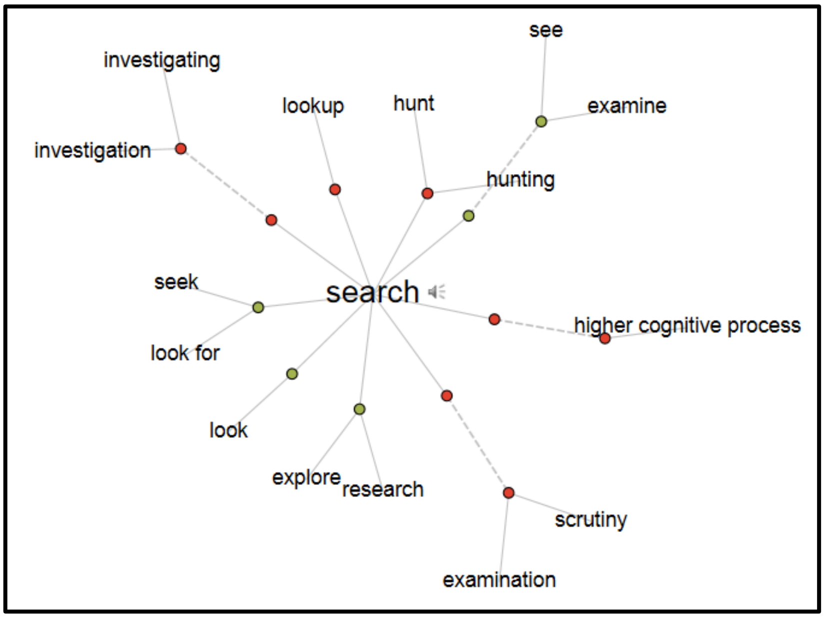


Figure 3-7 The Visual Thesaurus using a Network IV Technique (ThinkMap Inc. 1998)

3.4.2. Web Service Collections

The structure of web service collections can assist in determining which IV techniques will be suitable for this type of data. Analysis of publication sites can be used to determine the general structure of the web service collections.

Three publication sites, namely RemoteMethods (1999), ProgrammableWeb (2009) and Service-Finder (2008), were identified as the publication sites with the most support for the web service discovery criteria. These publication sites were analyzed to determine the data structure of the web service collections that are provided by each website.

The publication sites were individually analyzed to determine what information is displayed in the provided web service collection and also to determine whether and how this information is connected or linked.

3.4.2.1. RemoteMethods

RemoteMethods (1999) provides different categories of web services, the web service and some information about the web service as summarized in Figure 3-8. Categories consist of main, subsidiary and sub-subsidiary categories. No initial list is displayed of the available web services and so a user can browse by category or search for web services. At the time of analysis, RemoteMethods contained 396 web services. Web services are contained in one category only (i.e. hierarchical structure).

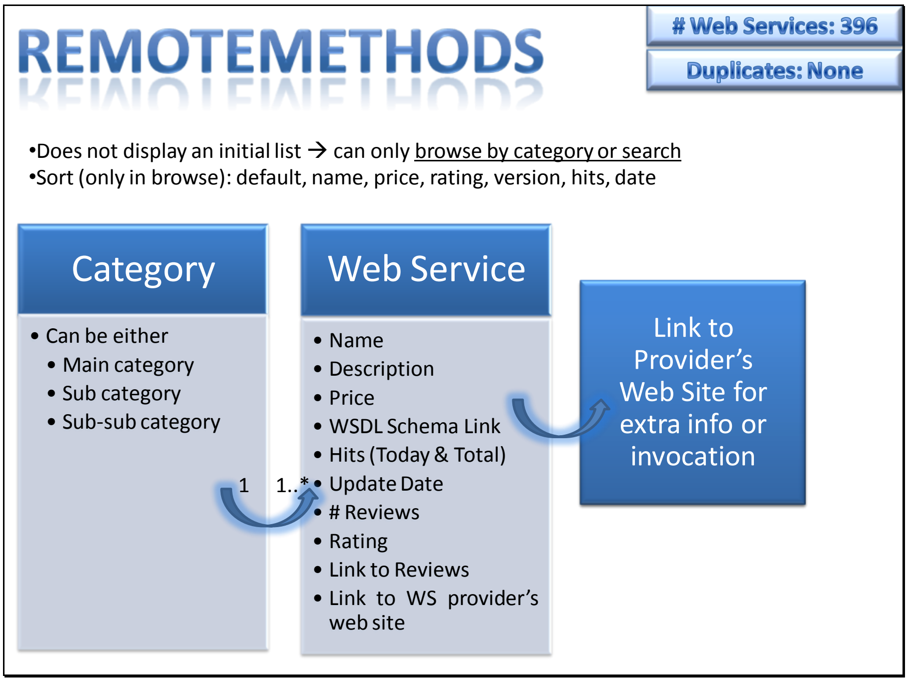


Figure 3-8 Data Analysis of RemoteMethods

3.4.2.2. ProgrammableWeb

Figure 3-9 illustrates a summary of web service data provided by the ProgrammableWeb (2009) publication site. ProgrammableWeb also provides different categories of web services, the web service and some information about each web service. Only a main category is provided. An initial list of web services provided by the site is displayed and a user may search for a web service or browse for a web service by category. Approximately 2,000 web services are provided by ProgrammableWeb. Again, a web service is contained in only one category.

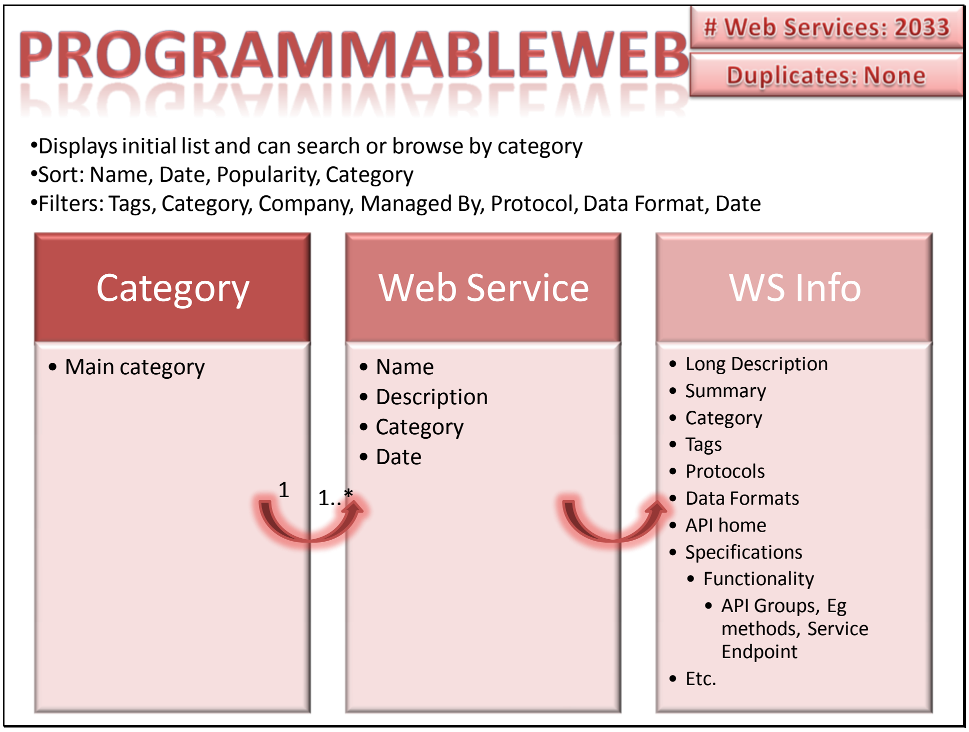


Figure 3-9 Data Analysis of ProgrammableWeb

3.4.2.3. Service-Finder

Service-Finder (2008) provides approximately 20,000 web services. Similarly to RemoteMethods, Service-Finder provides a category, the web service and some web service information with a user being able to browse for web services by category and search for web services. Main, subsidiary and sub-subsidiary categories are provided by this publication site. A user can tag web services in categories irrespective whether it is a main, subsidiary or sub-subsidiary category. Thus, web services can be contained in more than one category (i.e. network structure). Figure 3-10 illustrates the web service information provided by Service-Finder.

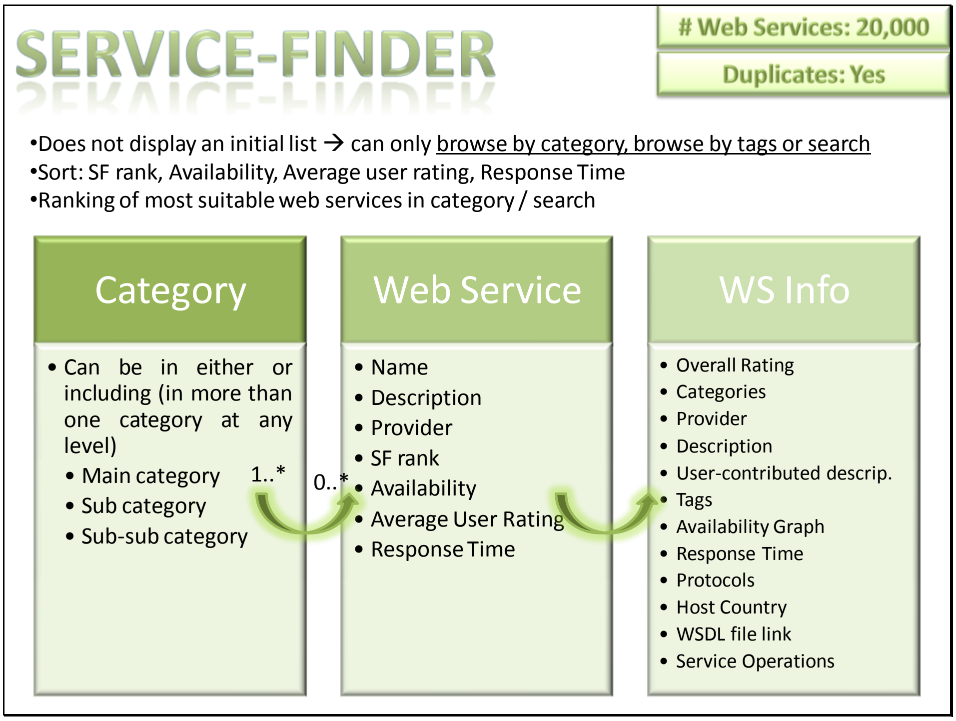


Figure 3-10 Data Analysis of Service-Finder

3.4.2.4. Summary of Data Analysis

Based on the analysis the general structure of web service collections is depicted by the UML (Unified Modelling Language) class diagram in Figure 3-11. A web service can be classified in a category according to its functionality. The web service can be contained in either a main, subsidiary or sub-subsidiary category. A variety of information is provided for each web service that describes the functional and non-functional properties of the web service. A provider makes the web service available. If a user selects a web service, this selection typically links to the provider's web site where more information can be found regarding the selected web service and where and how the web service can be invoked. Users can also provide a review or comment on a web service used. A link to the description file is generally provided for the user to view the file in XML (Extensible Mark-up Language) format.

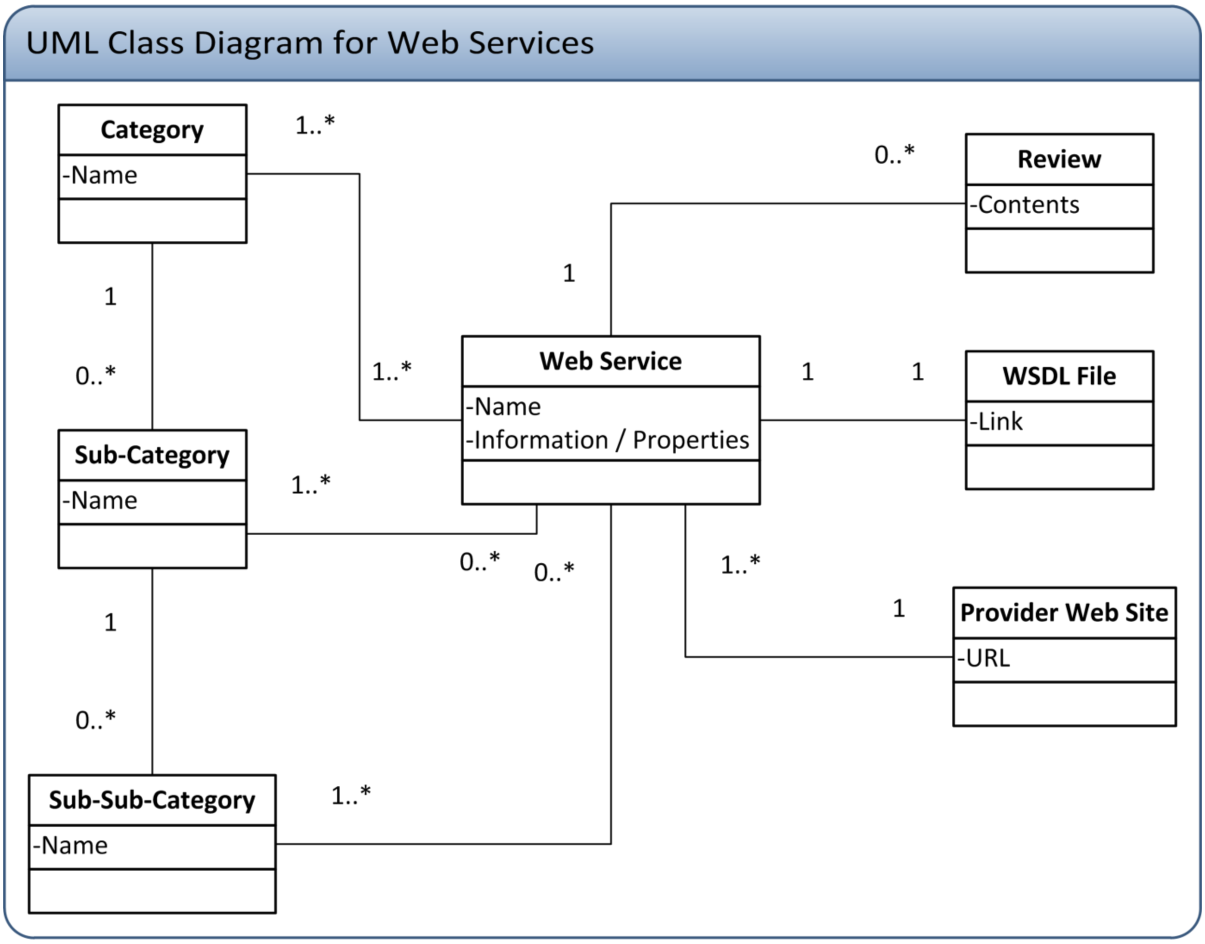


Figure 3-11 UML Class Diagram for Web Service Collections

From this analysis it can be concluded that the data structure of web service collections has either a hierarchical (Figure 3-12) or network data structure (Figure 3-13). A web service can have many attributes and is contained in one or more categories.

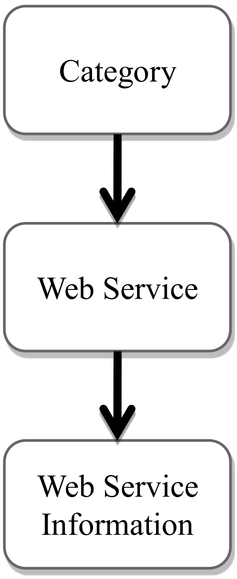


Figure 3-12 General Hierarchical Structure of Web Service Collections

The data structure of the RemoteMethods and ProgrammableWeb publication sites is hierarchical in that a web service may only be contained in one category. Service-Finder allows users to tag web services into different categories that describe the web service's functionality and so a web service can be contained in more than one category. Thus, the data structure of Service-Finder's web service information can be categorized as a network structure (Figure 3-13).

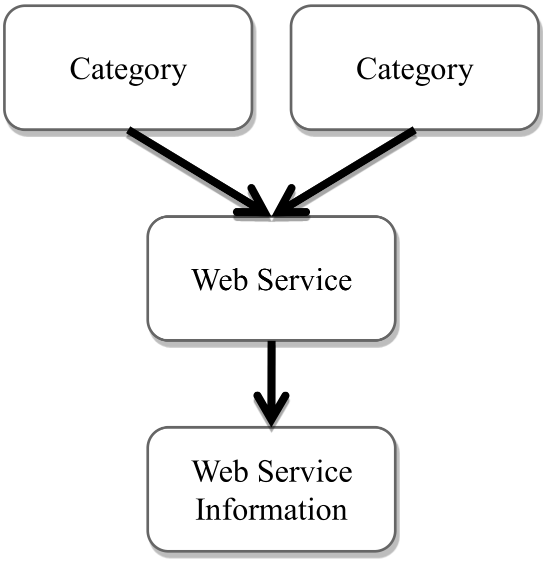


Figure 3-13 General Network Structure of Service-Finder's Web Service Collection

A decision needs to be made whether to use a hierarchical or network IV technique to visualize web service collections. The next section discusses these IV techniques in greater detail to determine which IV technique would be the most appropriate to visualize web service collections.

3.4.3. IV Techniques for Web Service Discovery

From the previous data analysis it can be concluded that web service collections have a hierarchical or network data structure. The difference between the two data types is that a network allows for cycles in the data with directional links, if required. The IV techniques for each of these data types are discussed in detail in order to determine which IV technique will be suitable for web service discovery. The technique selected should provide support for large collections of web services, provide basic information as well as details on demand, allow for browsing and searching and support usability.

3.4.3.1. Hierarchical IV Techniques

Many hierarchical IV techniques exist and a review of these can be found on www.treevis.net (Schulz 2011). Hierarchical or tree data involves a dataset in which nodes are linked but an object in the hierarchy has a link to only one parent object (Shneiderman 1998, 531). Objects and the links between the objects may have many attributes attached to them. Hierarchical IV techniques are commonly used for computer directories and sales data such as stock markets (Shneiderman 1996; Plaisant 2004; 2005). The IV techniques that support hierarchical or tree data include the following:

- *Outline Style of Indented Labels:* This technique is used to visualize table of contents and a well-known application of this technique is Windows Explorer (Carr 1999) (Figure 3-14).

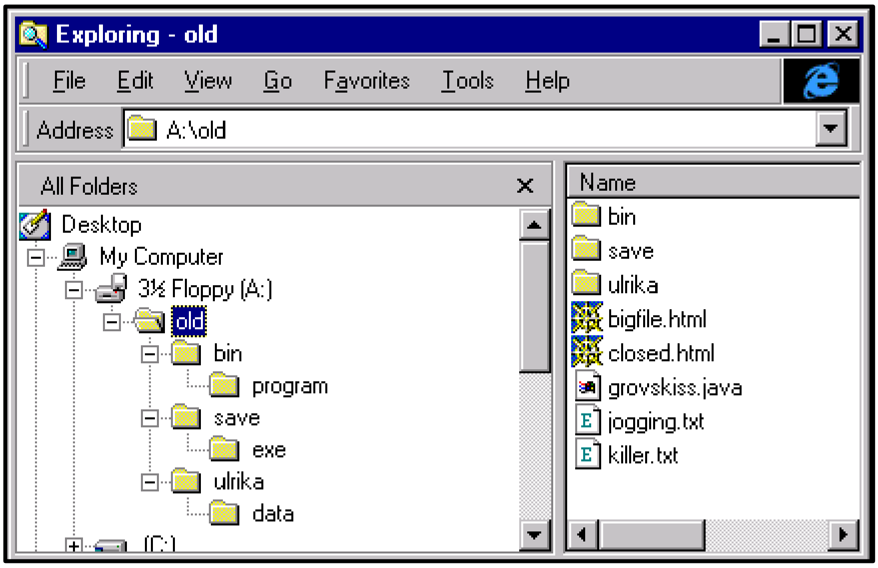


Figure 3-14 Windows Explorer using a Tree Visualization (Carr 1999)

- *Node-and-Link Diagram*: This technique is the most commonly used IV technique for hierarchical data. Examples of implementations of the node-and-link diagram include the hyperbolic tree, cam and cone trees. Some of the examples of these implementations are a space tree (Plaisant, Grosjean and Bederson 2002), a 3D cone (Shneiderman 1998, 535) (Figure 3-15), and a hyperbolic tree browser (Cockburn, Karlson and Bederson 2008).

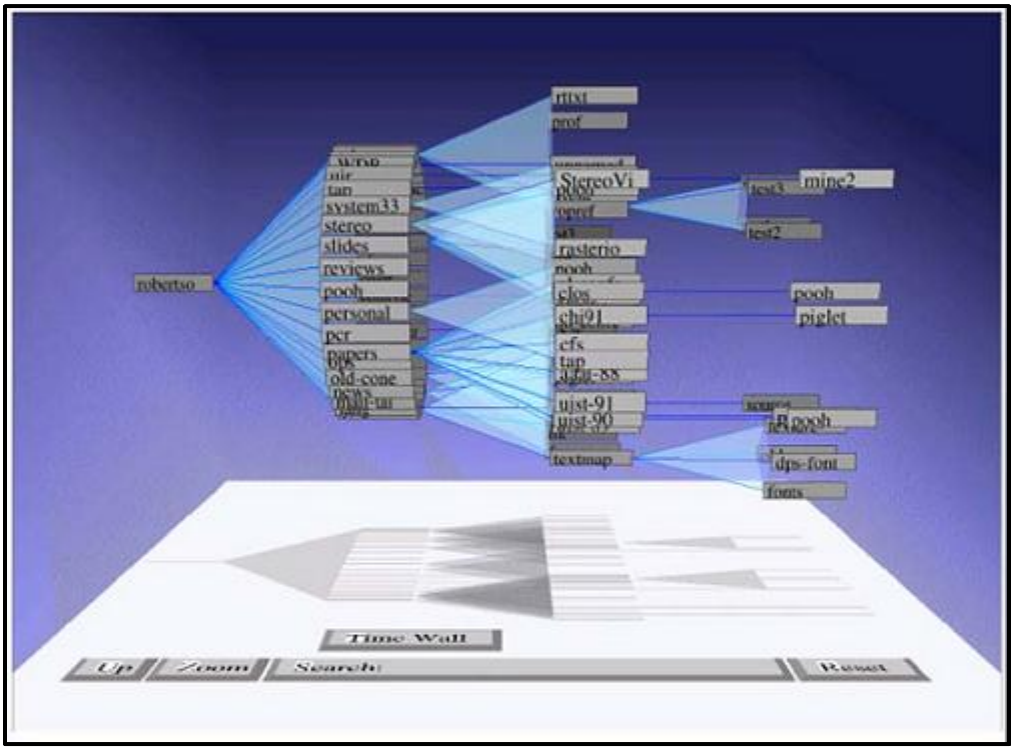


Figure 3-15 3D Cone using a Cam-Tree Design (Shneiderman 1998, 535)

- *Enclosure Tree:* As explained previously, enclosures are used to encircle nested subsets in a hierarchy. An application of this type of tree visualization is a tree map. Tree maps are typically used for business information such as visualization of stock markets and inventory. A well-known example of a tree map implementation is the SmartMoney Map of the Market which visualizes stock markets (Plaisant 2004) depicted in Figure 3-16.

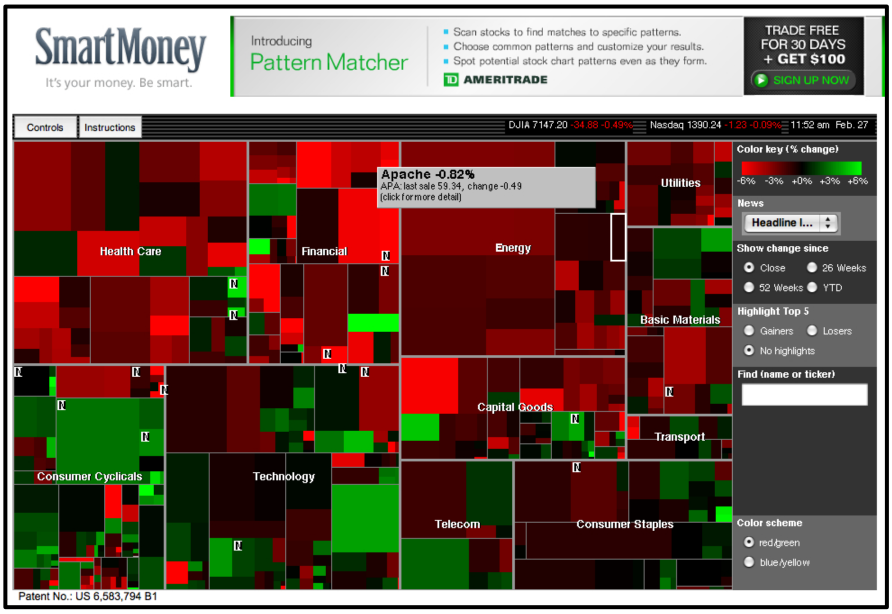


Figure 3-16 SmartMoney Map of the Market (Plaisant 2004)

The development and application of these hierarchical IV techniques support computer directories, stock markets and other hierarchical business information. These IV techniques only support hierarchical data and some of these techniques may provide poor usability when dealing with large hierarchies. For example, the indented labels list and tree map may be unsuitable to visualize large hierarchies. The data type of a web service collection needed to be visualized is not really comparable to the type of data that hierarchical IV techniques support with exception of computer directories. Additionally, enclosure trees which are commonly used to visualize stock markets may not suitably represent a web service collection as it does not require nested subsets.

3.4.3.2. Network IV Techniques

Network data is similar to hierarchical data except that objects can be linked to a number of other objects (Shneiderman 1998, 533). There are many types of networks including acyclic, lattice, rooted, un-rooted, directed and undirected networks, but all of these types fall under the network data type. Network IV techniques are used for visualization of network and traffic information (Keim, Andrienko, *et al.* 2008), digital libraries (Shneiderman 1996), the Web, geographical applications (Plaisant 2005) and social networks (Viégas and Wattenberg 2007). Although network IV techniques were developed some time ago the applications implementing these techniques seem to be more recent than hierarchical techniques. The IV techniques that support network data include the following:

- *Node-and-Link:* A node-and-link IV technique is the most common IV technique used for network data (North 2005). This technique has been used to visualize the Web and social networks. Some examples of implementations of a node-and-link visualization technique include ClusterMap (Deligiannidis *et al.* 2006), Vizster (Heer and Boyd 2005) illustrated in Figure 3-17, Jigsaw (Stasko, Görg and Liu 2008), Silo-Breaker (2009) and Touchgraph (2007).

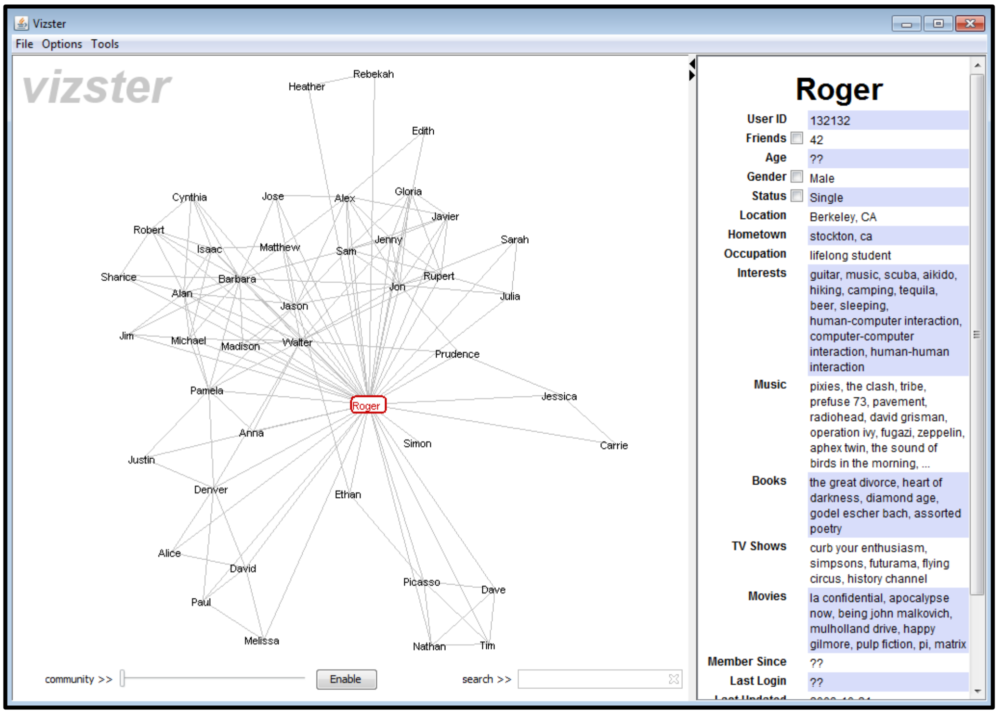


Figure 3-17 System Interface of Vizster (Heer and Boyd 2005)

- *Square Matrix:* A square-matrix concentrates on the connections of nodes instead of the nodes themselves and visualizes network data by placing a value of a link property in the row of the matrix and by placing the link in the column of the matrix (North 2005). Connex is an example of a square matrix implementation illustrated in Figure 3-18.

Data that is visualized by network IV techniques such as social networks and the Web is similar to web service collection data. Social network data typically consists of communities or groups to which people belong and people have certain information displayed about them. Additionally, network IV techniques are currently being used to visualize as well as improve on the visualization of large amounts of network data. The network IV techniques support visualization of network and hierarchical data as cycles may or may not occur in the data but the network IV techniques support the cycles if they occur. The node-and-link IV technique used to visualize network data may thus be the most suitable IV technique to visualize web service collections.

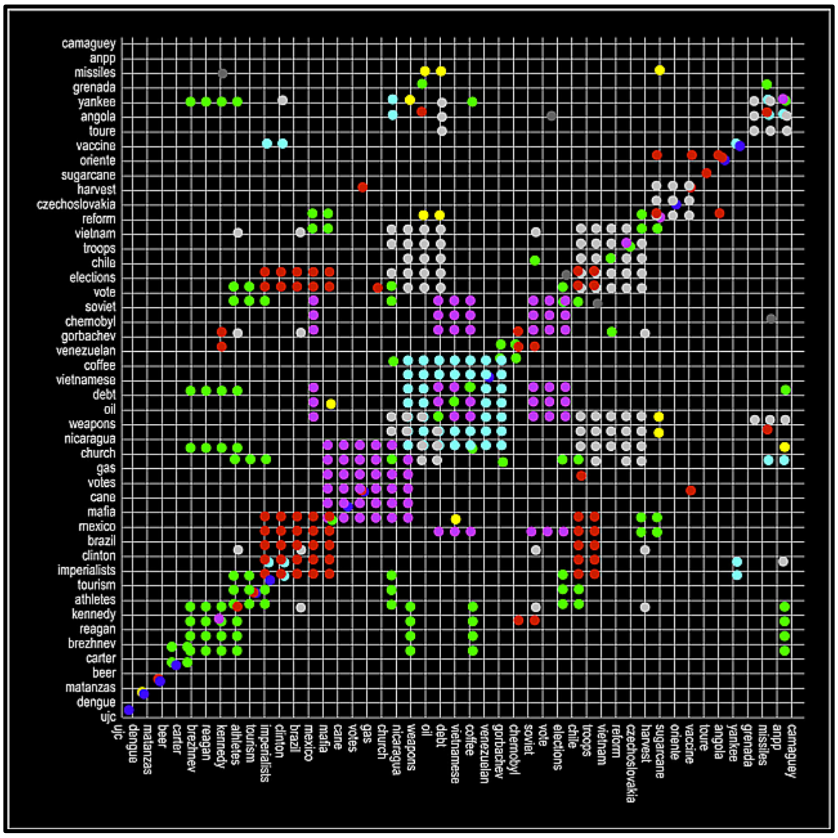


Figure 3-18 Connex implementation of a Square Matrix (Pacific Northwest National Laboratory 1998)

3.5. IV Requirements for Web Service Discovery

To present web service collections in a manner that supports effective web service discovery, the techniques used to visualize this specific information should adhere to certain requirements. Only a few applications using IV to represent web service collections have been found and thus there is a limited amount of information to guide this specific type of application. The requirements of IV for web service discovery are identified in terms of the visual information seeking mantra (Shneiderman 1996, 337):

“Overview first, zoom and filter, then details-on-demand...”

The mantra is discussed in terms of overview, zoom, filter, details-on-demand, relate, history and extract tasks. The visual information seeking mantra is considered to be an important contribution to IV research (Chen 2010). Any IV technique selected to represent data should support these tasks.

The following requirements are proposed in terms of the visual information seeking mantra used for IV (Shneiderman 1996), adapted for web service collection information and based on the web service discovery criteria identified in Chapter 2:

- *Overview:* The user should be able to view the entire collection of web services. A field-of-view window should be provided to allow the user to easily move through specific areas of a web service collection displayed in the detail window. The possible use of a fisheye strategy could assist the user in navigating to a position in the graph that is of interest (Shneiderman 1996). The user should also be able to easily navigate through the web service collection with pan or scroll controls. Making use of a combination of a field-of-view window as well as a detail window requires coordination between these displays (Card 2008, 534). The use of an overview is an important aspect in IV (Hornbæk and Hertzum 2011). The overview provides users with a general idea of the overall web service collection.
- *Zoom:* The user should be able to zoom in on web services, categories or sections of the collection that are of significance to him (Shneiderman 1996). A user would zoom in to focus on a section of the web service collection or to view more detail graphically (Carr 1999). A fisheye strategy may also be useful to assist in the zoom task.
- *Filter:* A user should be allowed to remove the items that are not of interest to him at any point in time allowing the user to concentrate on his interest (Shneiderman 1996). A user should be allowed to search for web services in the web service collection based on web service functionality. A search facility should be provided by the system to allow the user to find specific web services, especially if the collection is large and if the user has a specific query in mind. Web services that meet the search criteria should be highlighted or colour coded in some way to differentiate the relevant search results from the rest of the collection. It could also be possible to hide linkages of categories or web services that are not part of the search results. Filters should also be provided to refine the search based on certain criteria. Additional techniques to assist the filter task include sorting, grouping, highlighting, hiding or finding objects similar to the object of interest (Plaisant 2005).
- *Details-on-Demand:* A user should be able to select categories or web services in the web service collection and obtain details or be able to find more information about the categories or web services when required (Shneiderman 1996). A user should be able to expand or contract categories in the collection when needed. For example, a user should be able to expand the category objects until he finds a web service of interest. The user should then be able to view the web service information that describes the

functional and non-functional properties of the web service in a pop-up or additional window. The information that is displayed about the web service could also provide access to a list of related web services or web services with similar functionality.

- *Relate:* A user should be able to observe the associations between objects (Shneiderman 1996). This aspect can be used in combination with the filter aspect. For instance, QoS (Quality-of-Service) information can be filtered to identify web services that satisfy certain criteria. This can also be used in conjunction with the search facility. Users should also be able to view the WSDL (Web Service Description Language) file of a web service as well as user reviews related to the selected web service. A facility could also be provided to allow a user to compare web services of similar functionality. A search result could number or somehow encode the relevance of the web services or categories to the search criteria to identify which parts of the web service collection should be focused on.
- *History:* The user should be supported in reversing and re-doing actions performed to assist in the exploration process (Shneiderman 1996). Possible functionality that could be used to support this task includes the use of a login to keep a history of selected web services and searches based on previous actions that the user has performed with the system. The user should also be allowed to return to the original display at any time.
- *Extract:* This task involves allowing extraction of the collection or sections of the web service collection so that a user may export this to a file for printing, graphing or email purposes (Shneiderman 1996). This may not be required by a web service discovery system as the purpose of visualizing web service collections is to find a specific web service and not to simply visualize the web service collection itself. A user may not typically require functionality to save the web services selected to a file for printing, but an option may be available to save the selected web service's URL for future use or, as mentioned above, make use of a login so that selected web services or previous web service search results may be saved. This task could also be supported by allowing a user to select a web service to be invoked.

These tasks need to be considered when visualizing web service collections for the purpose of web service discovery. Colour-coding and highlighting could be used to support these tasks (Shneiderman 1996; Liang and Huang 2010). The minimum amount of information needs to be determined that should be displayed for a user to make web service selection decisions.

3.6. Existing IV Systems for Web Service Discovery

Visualization of the results of web service discovery is a new approach in service oriented computing (SOC) (Chua *et al.* 2007). Only two applications of IV to web service discovery were found in a literature review. Each application is discussed in terms of the proposed IV requirements and web service discovery criteria identified above.

3.6.1. Cluster Map IV

Sabou and Pan (2007) identified problems of web service repositories and explored the presentation of ontology-based metadata to overcome these problems. Their research discusses faceted browsing and visualization to present metadata. A cluster map IV technique was used to present the metadata. A cluster map is an example of a node-and-link IV technique for network data.

The purpose of a cluster map is to visualize “instances of a set of classes according to their classification into these classes” (Sabou and Pan 2007, 149). Figure 3-19 illustrates the implementation of the cluster map IV technique to aid web service discovery.

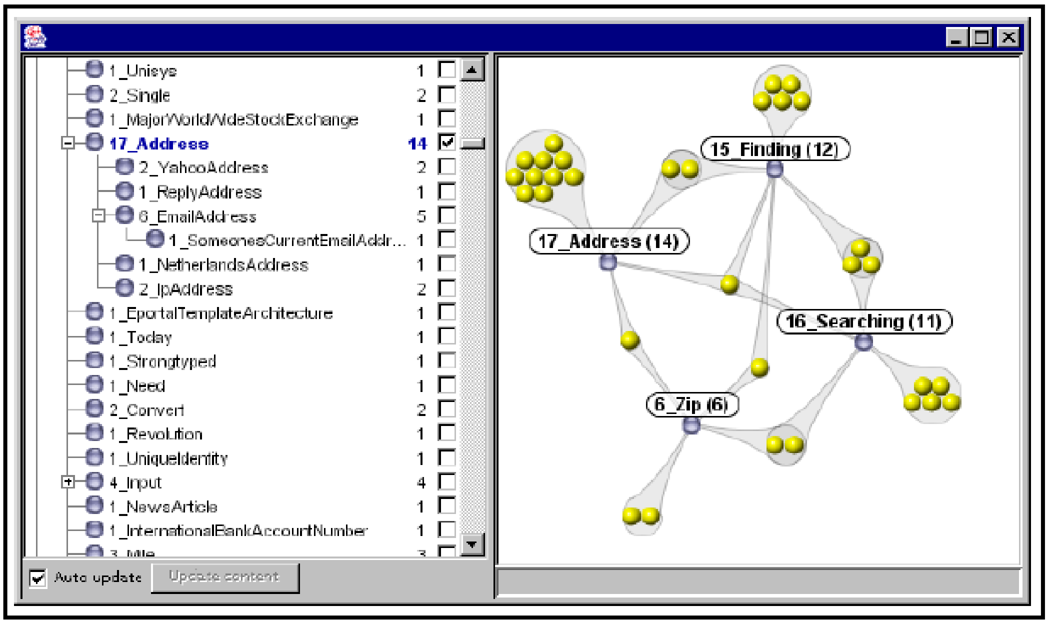


Figure 3-19 The Cluster Map Visualization Technique (Sabou and Pan 2007)

The left pane displays the ontology of terms which a user may browse to find terms that match the user’s query. The services that match the query are displayed in the right pane. Web services that have similar functionality and that are related to the specific terms are grouped into clusters. Services that match more than one term are linked. An advantage of this technique is that this visualization enables the user to easily explore or browse services in the repository that was used for this research.

Shortcomings of this application include the lack of a search facility and also that no QoS information is provided to assist a user in distinguishing between similar web services. The IV technique used for this application may be suitable to visualize web service collections, but emphasis is placed on the functionality of the web services in the collection rather than the web service details.

No clear indication is provided of what happens after finding the collection of web services that matches the selected functionality. The only indication of the procedure following the exploration process is that the instances, or web services, can be accessed with a single-mouse click (Sabou and Pan 2007). It is also not clear whether the Overview requirement is supported as a user is required to select functionality in order to have a sub-collection displayed in the right pane of the application. It also seems as though a user may only filter on functionality. As mentioned previously, there is no evidence of support for the provision of non-functional properties or additional functional properties of the web services that meet the selected criteria and so the details-on-demand task is not fully supported. There is also no evidence of support for the history task and it is not clear how a user would select and use an individual web service.

3.6.2. Graph Structure

Stollberg and Kerrigan (2007) present a goal-based visualization and browsing technique for web service discovery. The technique focuses on the user's goal and arranges the web services in a graph structure based on the web services' functional similarity. This technique is also an example of a node-and-link IV technique incorporating direction.

Knowledge is stored regarding the appropriateness of the web services with respect to the user's goal. The purpose of this application is to assist the user in forming his goal and to provide visual assistance in discovering web services that are most appropriate to this goal. Figure 3-20 demonstrates the graph structure implemented.

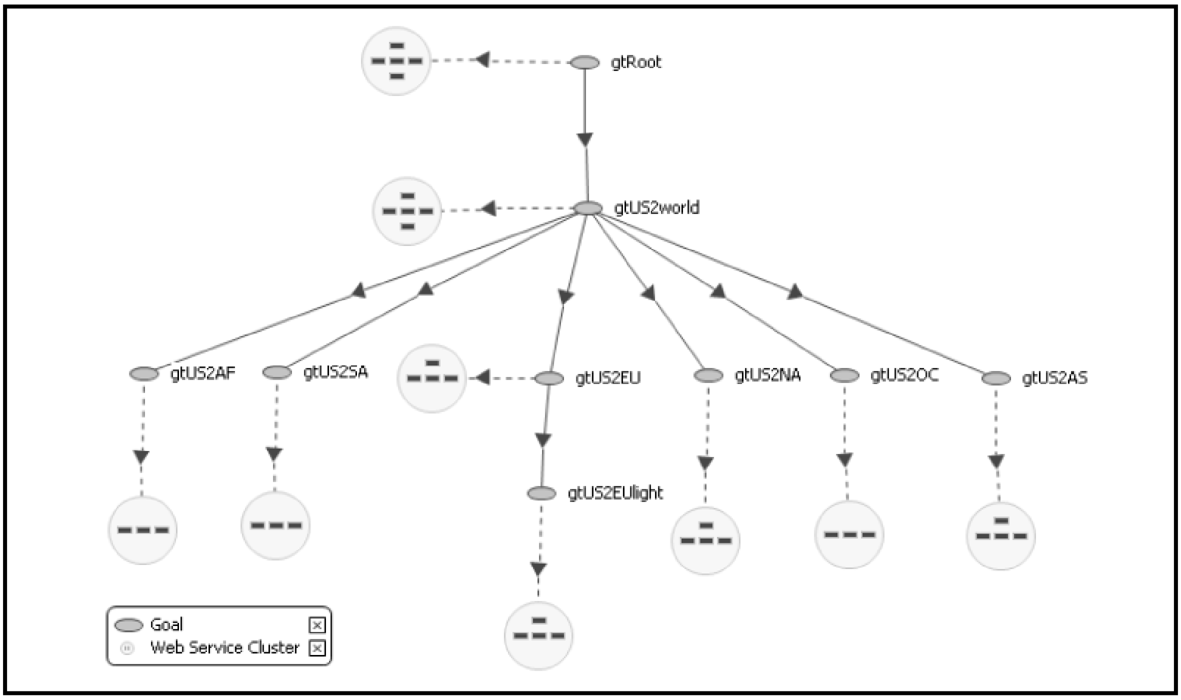


Figure 3-20 Graph Structure implemented in Stollberg's Technique (2007)

A clustering IV technique is used at the leaf nodes of the graph to display web services with functionality presented at the parent of each cluster. This technique provides a multi-level browsing ability. The graph structure is flexible in that the graph may be zoomed and rotated. The graph can also be modified by dragging and dropping nodes in the graph to other positions. Filtering of nodes is also provided. Technical details, such as the functional properties of each web service may be viewed.

This system has a similar limitation to the previous system as it only provides browsing facilities for the discovery of web services. No evidence was found regarding the provision of non-functional web service properties. The overview and details-on-demand tasks are supported in that a user may view the entire graph if required. Similar to the previous application, it seems as though a user may only expand the graph based on functionality; there is no evidence of support for the history task and it is not clear how a user would select and use an individual web service.

3.7. Conclusion

This chapter has highlighted the importance of IV as improving user cognition by presenting data visually. IV is used to present data clearly so that potentially hidden properties and relationships can be identified. Several benefits of IV were discussed including assisting in sense-making, ease of data interpretation, quicker decision making and reducing the effort required to gain insight into the data. Several challenges of IV were identified, including the usability, uncertainty and scalability of data.

Suitable IV techniques were identified by determining the type of data to be visualized. Data analysis was conducted on three publication sites considered to have the most support for the web service discovery criteria identified in Chapter 2. The three publication sites that were analyzed were ProgrammableWeb, RemoteMethods and Service-Finder. Two of the publication sites have a hierarchical data structure, while the third publication site has a network data structure.

The node-and-link IV technique that supports network data was selected as the most suitable technique to visualize web service collections. Data that is generally visualized by network visualization techniques is similar to web service collection data.

IV requirements for web service discovery were proposed according to the tasks identified by the visual information seeking mantra. The user tasks that should be supported by the selected IV technique included overview, zoom, details-on-demand, filter, relate, history and extract tasks. Colour and highlighting can be used to support these tasks.

Two existing applications that incorporate IV to visualize web service collections, namely the cluster map IV technique and the graph structure, were discussed in terms of their functionality and the support provided for the proposed IV requirements. No search facility is provided in either of these applications. No QoS information is presented for the web services contained in the collection. Thus, both these applications do not adequately support the proposed IV requirements for web service discovery.

The next chapter will discuss the design and implementation of an IV prototype to visualize web service collections. The system will be designed and implemented according to the IV requirements for web service discovery identified in this chapter. Low-fidelity prototypes will initially be used to determine how the system will support the proposed IV requirements. This prototype will then be implemented as a functional prototype to meet the requirements proposed in this chapter and will be evaluated in the subsequent chapter.

Chapter 4: Design and Implementation

4.1. Introduction

Chapter 3 proposed information visualization (IV) requirements for web service discovery. This chapter discusses the design and implementation of an IV prototype to visualize web service collections using the IV technique selected in the previous chapter (Beets and Wesson 2011a). This chapter addresses the third research question identified in Chapter 1, namely *how can IV techniques be applied to support web service discovery?*

The chapter discusses the design of the prototype in terms of the web service collection that is used for the visualization, the functions that the prototype supports and the techniques used to visualize the web service collection. A discussion regarding the implementation of the prototype follows the design section. The implementation section discusses the selection of an implementation tool to be used for the implementation. The chapter also includes a discussion of how the selected tool was used to implement the IV requirements.

4.2. Design

The prototype needs to visualize one or more web service collections. This section describes the web service collection used as the data source for the prototype and how this data was transformed for the prototype. The user tasks that need to be supported by the prototype are discussed in terms of the IV requirements identified in Chapter 3. The visualization techniques that are used for the prototype are then discussed and motivated.

4.2.1. Data

A live web service collection should be used to develop a prototype to determine to what extent IV can be used to support web service discovery. The ProgrammableWeb web service collection was selected as the sample web service collection for this prototype as it is a large collection that is increasing daily. The data needed to be extracted and converted into a usable and simpler format for the prototype, to be customized further as needed. This transformation process is discussed in detail in this section.

4.2.1.1. Data Source

ProgrammableWeb (2009) provides an API (Application Programming Interface) to allow developers to retrieve the web service collection. A comparison of public web service registries conducted by Hagemann *et al.* (2007) concluded that ProgrammableWeb was one of only two publication sites that are increasing in web service collection size.

The publication site currently (2011) provides 2400 web services contained in 54 categories. This web service collection is a dynamic collection which is increasing daily. ProgrammableWeb is also considered to be the most popular publication site (Pedrinaci *et al.* 2010). The data structure of this web service collection is hierarchical as a web service can be contained in only one category (Section 3.4.2.2.). Various information is provided regarding each web service such as a web service name, a summary, a detailed description of the provided functionality, the category to which the web service is assigned, the protocol used, a rating of the service and the provider of the service. The API allows developers to access the registry of web services provided by the publication site and allows users to access APIs, mash-ups, profiles of members and other information.

4.2.1.2. Format

The information retrieved using the API is structured using XML (Extensible Mark-up Language) and the Atom Publishing Protocol (APP). XML is used to describe or structure data in XML documents (W3C Web Services Architecture Working Group 2008). The APP uses HTTP (Hypertext Transfer Protocol) and XML to create, edit and publish Web resources (The Internet Society 2006).

Figure 4-1 illustrates the data conversion process from data retrieval to a suitable format containing the necessary web service information to be used by the prototype.

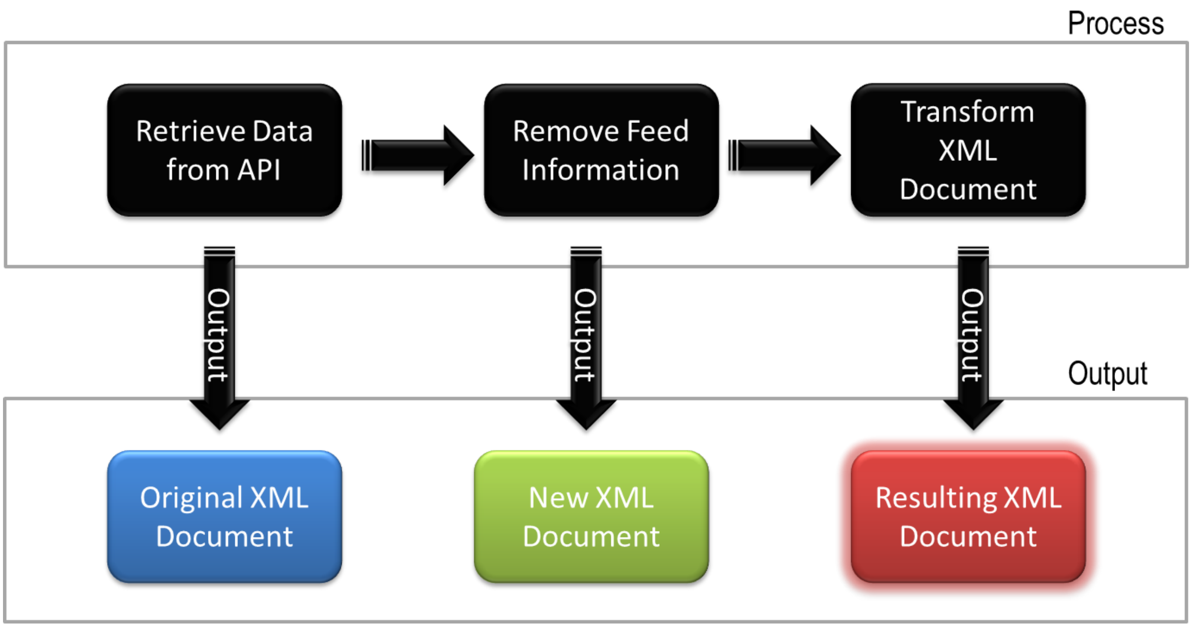


Figure 4-1 Data Conversion Process

Each call to the API retrieves 20 web services structured in XML contained in an Atom Feed. A loop was used to retrieve all the web services contained in the ProgrammableWeb web service collection.

The retrieved data was then converted into a usable format. The conversion process was completed using XPath and XSLT (Extensible Style-sheet Language Transformations). XPath is a language used to traverse an XML document (w3schools 2011a). XSLT is a template or language that makes use of XPath to transform XML documents into other XML documents or formats (w3schools 2011b). XSLT allows creation of customized XML documents. Using these tools the necessary web service attributes were extracted from the original XML document to minimize the data kept in the XML document to only data that was needed for the visualization. Additionally, the feed structure was removed so that the web services were listed in an XML document as if all the web services were retrieved at once. This was because the feed information interfered (or was incompatible) with the conversion process.

The web service attributes that were extracted for each web service included the following:

- Title: The web service name
- Category: The category the web service belongs to
- ID: The website link of the web service
- Summary: A short one-line description of the web service functionality

- Rating: An average user rating
- Description: A detailed description of the web service functionality
- Protocol: The protocol used for the service e.g. REST or SOAP
- WSDL: The link to the WSDL (Web Service Description Language) document if it is provided
- Provider: The provider of the web service
- Price: The cost of using the web service
- Date: The upload or update date of the web service
- Data Formats: The data formats supported by the web service

The resulting entity as contained in the converted XML document is shown in Figure 4-2.


```
- <WebService>
    <id>http://www.programmableweb.com/api/adgooroo</id>
    <title>AdGooroo</title>
    <summary>Keyword search marketing platform</summary>
    <rating>4.5</rating>
    <description>The AdGooroo API lets developers access competitive intelligence data t
      is based on the popular REST architecture and supports data transfer formats suc
      language of your choice, including .NET, Perl, PHP, or virtually any other common
      search engine marketers. Its proprietary technology tracks all search advertising
      agencies and advertisers with competitors&#039; keywords, ad copy, campaign s
      the highest possible return on advertising investment.</description>
    <category>Advertising</category>
    <protocol>REST</protocol>
    <version/>
    <wsdl/>
    <provider>http://www.adgooroo.com</provider>
    <price/>
    <date>2010-01-11T11:00:36Z</date>
    <dataformat>XML, JSON, CSV</dataformat>
  </WebService>
```


Figure 4-2 Converted XML Data Displaying the Resulting Web Service Entity

The web services were also sorted by category and then by web service name in the XML document, to make it simpler for future conversion.

4.2.2. Functions

A task list was drawn up according to the IV requirements for web service discovery discussed in Section 3.5. Functions provided by the prototype should support the visual information seeking mantra tasks that were used to identify these requirements (Section 3.5.). A low-fidelity prototype was created to identify initial ideas of how the prototype should support the proposed requirements as well as the web service discovery criteria identified in Chapter 2. A low-fidelity prototype (Figure 4-3) was drawn before the comparison of implementation tools and thus the final design was modified according to which tool was used. The prototype was called “SerViz”, as it represents a visualization tool for web service discovery.

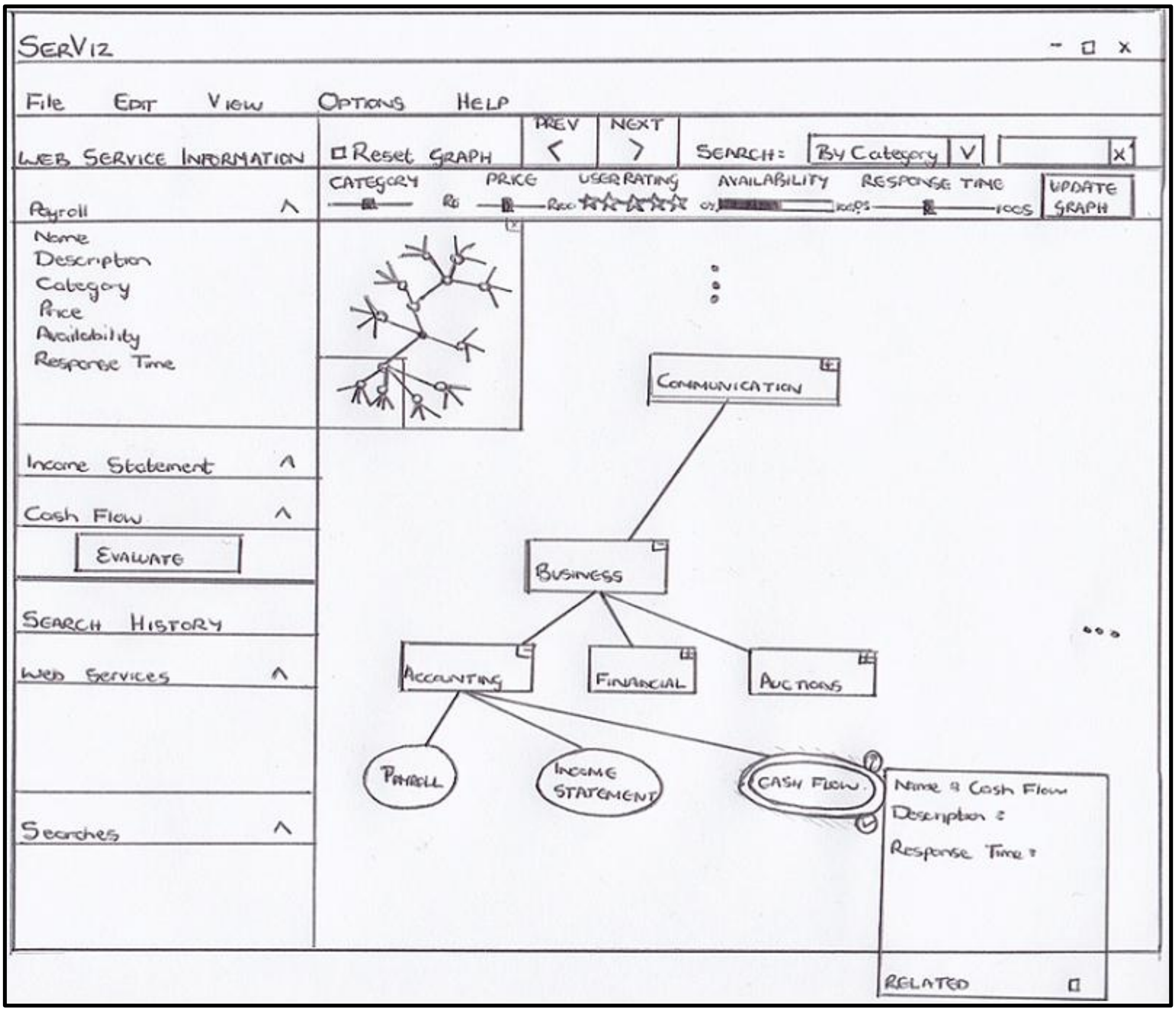


Figure 4-3 Low-Fidelity Prototype of SerViz

The prototype should generally aid searching or browsing for web services to support web service discovery. The tasks that were included in the low-fidelity prototype are listed below.

4.2.2.1. Overview

a) View the entire web service collection at any time.
b) Make use of the minimized interactive overview of the web service collection to navigate to sections in the web service collection.
c) Navigate through the web service collection with pan or scroll controls.

4.2.2.2. Zoom

a) Expand or contract the collection or categories to view more or less detail as required.
   a. View web services in a category.
   b. View categories in the collection.

4.2.2.3. Details-on-Demand (DoD)

a) View details of a category.
   a. Hover over the category to display the category name and the number of web services contained in this category.
b) View details of a web service.
   a. Hover over a web service to view a summary of the web service.
   b. Select a web service to view more detailed information.

4.2.2.4. Filter

a) Search for a web service based on the web service name (keyword search).
b) Filter the web service collection using a multiple filter selection based on certain criteria.

4.2.2.5. Relate

a) View the relationship between the collection and a category and the web services contained in a category.
b) View the WSDL file of a web service.
c) View search results in combination with the filtering results.

4.2.2.6. History

a) Undo / redo graph navigation actions.
b) View and manage a history of searches and selected or bookmarked web services.

4.2.2.7. Extract

a) Bookmark (or select) a web service to be used.
b) Save a search for a web service.
c) Save a selected web service's URL for future use.

4.2.3. Visualization Techniques

The network IV technique was identified to be the most suitable IV technique to visualize web service collections. Although hierarchical IV techniques can also be used, these techniques are too limited to visualize networked web service collections. To compare whether IV techniques can be used to visualize web service collections, a list view was also implemented.

4.2.3.1. List View

As discussed previously, existing web service discovery methods provide lengthy textual lists that the user is required to explore to select appropriate web services. Thus, this linear technique was implemented in order to compare the existing technique used for web service discovery with the proposed IV technique.

The list view was implemented similar to the current implementation used by publication sites. An initial list will be displayed with certain information shown for each service. A user will then search for a web service or browse the categories for a web service. If a certain web service in the list is selected, detailed information will then be displayed. The user will then return to the list to browse or search further.

4.2.3.2. Network View

The network IV technique was identified in Chapter 3 as the most appropriate IV technique to be used to visualize web service collections as it supports the web service collection data structure. By comparing this view to the list view, a usability evaluation will be used to determine to what extent this view supports web service discovery better than the list view currently used.

The network view will use nodes and edges to represent the web service collection. A collection node representing the web service collection will be the root node. This root node will connect the categories in the collection with edges. Each category connects to the web service nodes in the category with an edge.

4.3. Implementation

The design of SerViz was outlined in the above design section. This section discusses three tools considered to implement SerViz. An appropriate implementation tool is then identified. The process of converting the data to a format specific for graphing is discussed further. The implementation of the user tasks for the list and network views in SerViz is detailed. The section concludes with a discussion determining to what extent the selected implementation tool supports the users' tasks.

4.3.1. Implementation Tools

Silverlight was initially identified as an option to develop the IV prototype. As Silverlight provides limited support for graph visualization and interaction, Prefuse and Flare were identified as alternative toolkits that could be used for the prototype development. A comparison was conducted with Prefuse and Flare to determine which toolkit provided the most support for the IV requirements for web service discovery.

4.3.1.1. Silverlight

Silverlight (Microsoft 2011) is a plug-in for the .NET framework focused on developing highly interactive business and media applications. Web, desktop and mobile applications can be developed with Silverlight. Silverlight has progressed rapidly with the latest version, Silverlight 5, made available in a beta version in 2011.

Silverlight was initially considered as an option to use to develop the prototype. A feasibility study was conducted to determine whether the IV prototype could be developed with this toolkit. A number of Silverlight tools were found but only two of them, namely the Information Connections Engine (ICE) and the Silverlight Bag-of-Tricks (BoT) met the requirements of the prototype. These tools were limited in the support provided for graph visualization and interaction. The BoT only allows for a single node to have children nodes connecting to the parent node. ICE is a framework used to visualize connections between information and although ICE is described as being "heavily customizable" it is difficult to do so. As a result, alternative toolkits were identified that support graph visualization and interaction to develop the IV prototype.

4.3.1.2. Prefuse

Prefuse (Heer *et al.* 2005) is an open-source user interface toolkit for interactive data visualization and provides a framework for visualization developed in Java. Prefuse allows for different types of graph visualization including trees, graphs and network visualization. Support is provided for layout, searching, visual encoding, animation, dynamic queries and database connectivity.

An example of the network visualization in Prefuse is shown in Figure 4-4. The graph highlights the neighbours of a selected node and also displays a minimized overview of the graph so the user can always view the entire graph collection.

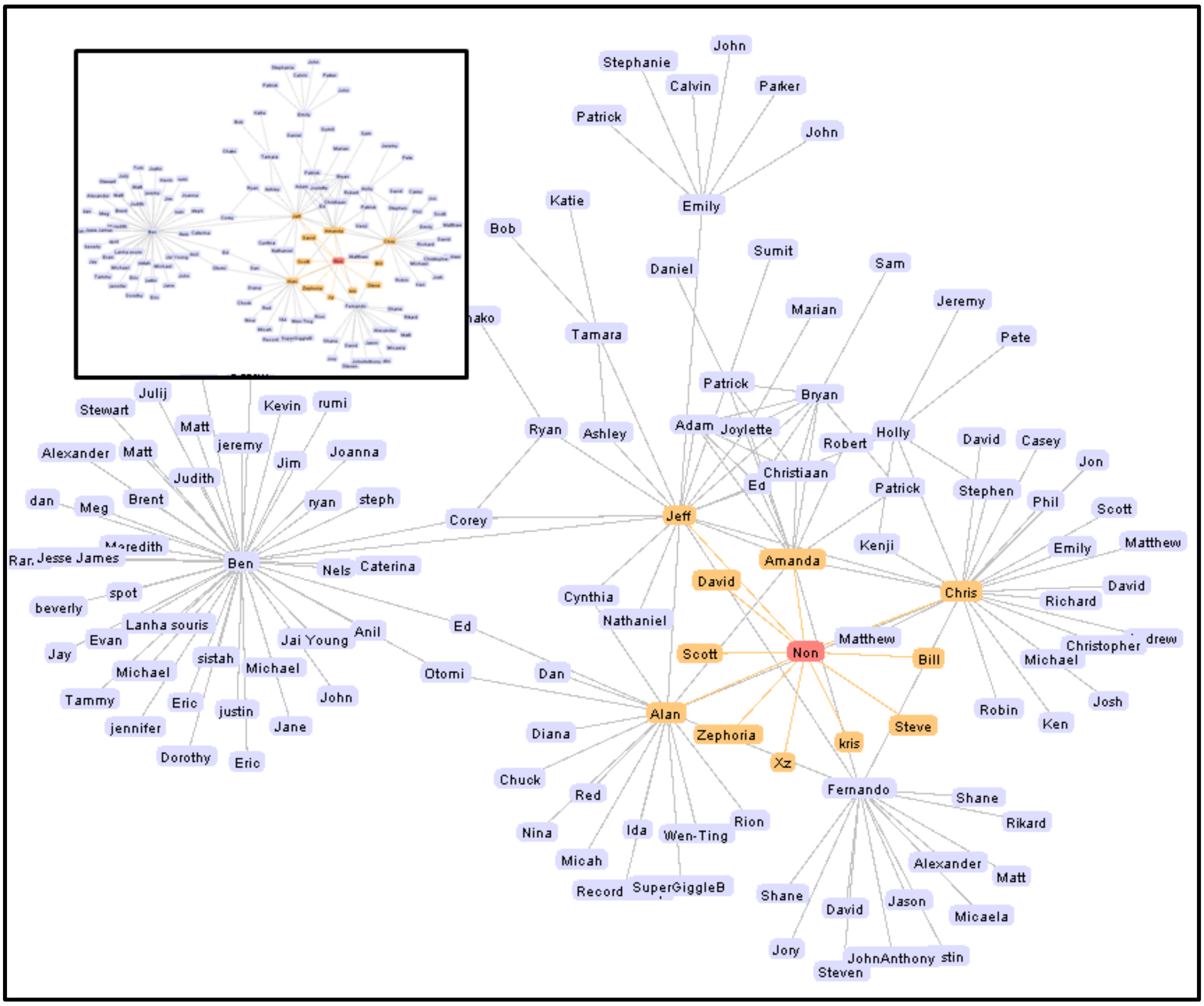


Figure 4-4 An Example of a Graph in Prefuse

The Vizster network graph (Heer and Boyd 2005) (Figure 3-17) and DoI tree (Heer and Card 2004) (Figure 3-6) were developed with Prefuse. Although Java provides limited support for a web-application, a desktop application was determined as sufficient for proof-of-concept purposes.

4.3.1.3. Flare

Flare (UC Berkeley Visualization Lab. 2009) is an open-source Flash-based implementation of Prefuse which consists of an ActionScript library for data visualization. Similar support is provided in Flare as in Prefuse, such as visual encoding and animation. Flare supports creation of customized IV techniques. An example of a simple graph visualization is displayed in Figure 4-5.

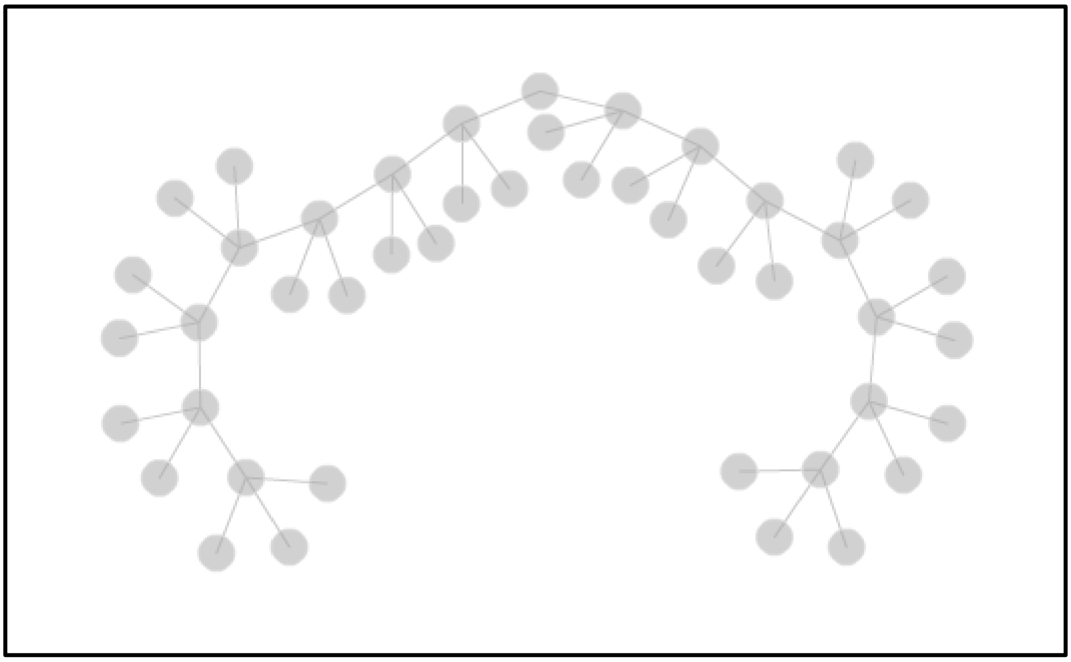


Figure 4-5 An Example of a Simple Graph in Flare

4.3.1.4. Comparison of Prefuse and Flare

Basic required functionality was implemented using Prefuse and Flare to compare the capabilities of each toolkit. The purpose of the comparison was to determine which of the implementation tools provided more support for the requirements of SerViz.

The tools supported the following functionality:

- Expand and contract collection or category nodes;
- Search and browse for web services;
- View a name and average rating of a web service; and
- Return to the original display i.e. how the graph was initially loaded when the application started only displaying the collection and contracted category nodes.

These basic tasks allowed for comparison of these two toolkits. The following tasks were used to compare the toolkits:

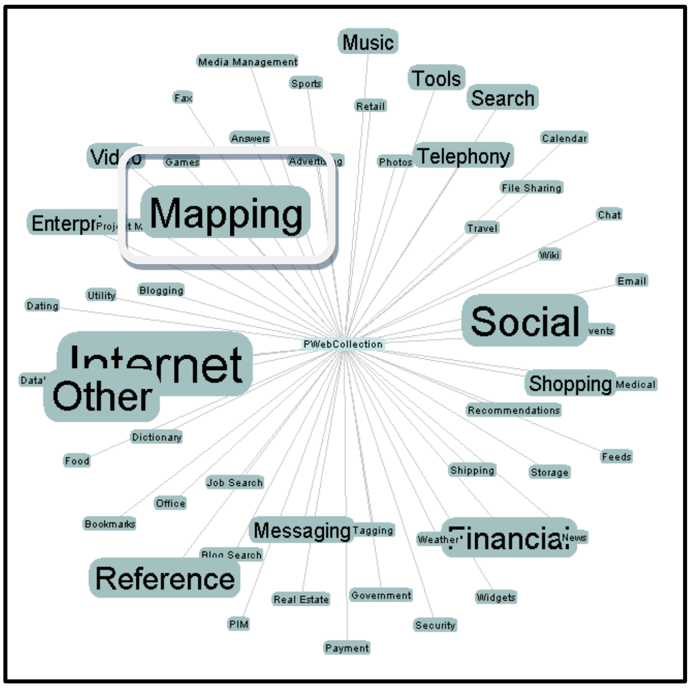
a) Browse the graph to find the *Mapping* category.
b) Use the search facility to find the *Shopping* category. Expand this category by clicking on the category node.
c) Return to the original display by using the checkbox (uncheck the checkbox once the graph has returned to its original layout with only the collection and category nodes displayed).

d) Browse the graph to find the *Venali Fax Web Service* web service.

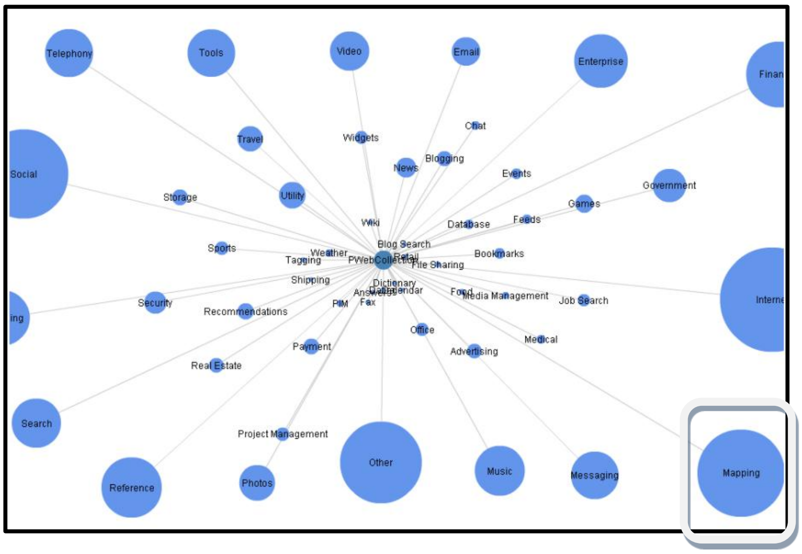
e) Select this web service to view the web service rating.

The functionality implemented in each toolkit was not necessarily the most appropriate functionality to be used for the prototype and was improved or extended upon once the more suitable toolkit was selected. The above-mentioned tasks were implemented as follows:

a) Browse the graph to find the *Mapping* category (Figure 4-6).



(a)

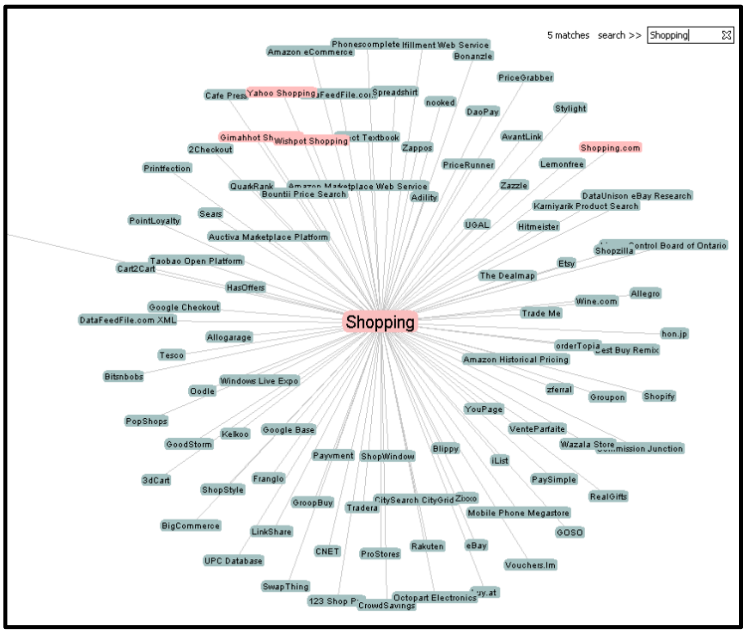


(b)

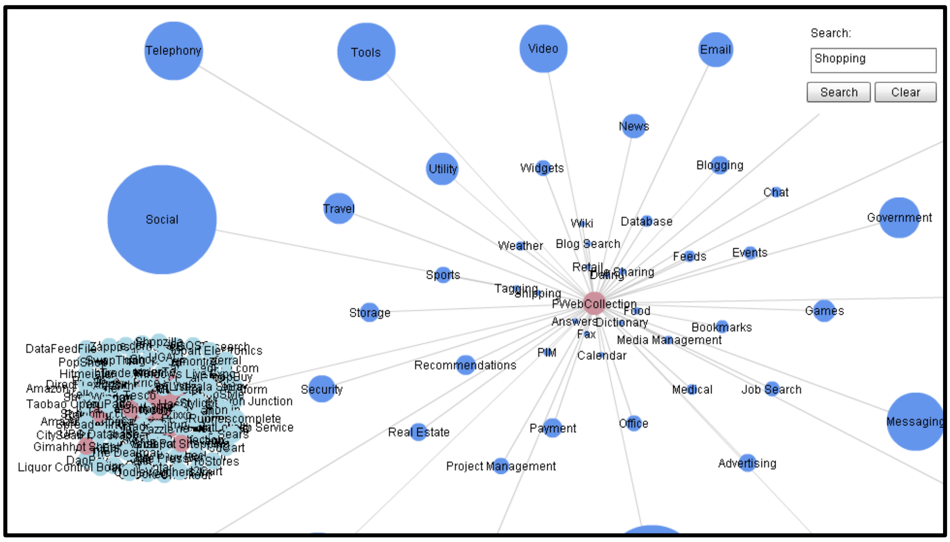
Figure 4-6 The Initial Graph: (a) Prefuse (b) Flare

When the application was run initially the layout of the graph in Prefuse was immediate. The graph in Flare took a while to lay out the nodes as the layout includes invisible nodes i.e. the web service nodes, when calculating where the nodes should be positioned. Each category node's size is representative of the number of web services contained in the category. It was a simple task to find a category node in either graph, although the category nodes in Prefuse have the text as well as the node resized based on the number of web services in the category.

b) Use the search box to find the *Shopping* category. Expand this category by clicking on the category node (Figure 4-7).



(a)



(b)

Figure 4-7 Searching and Expanding Categories (a) Prefuse and (b) Flare

When entering a search term in the search box on the interface, the Flare application seemed to lag. Prefuse responded immediately to any input. Prefuse handles any expanding or contracting layout calculations but Flare does not have support for this. Thus, Prefuse provided more support for expanding or contracting nodes and for searching within the collection.

c) Return to the original display by using the checkbox (uncheck the checkbox once the graph has returned to its original layout with only the collection and category nodes displayed).

This display is the same as the first task as the display should have looked the same as when the application was started. The collection node and category nodes are displayed and web service nodes are contracted. Both toolkits supported this task, but layout issues may arise if the expanding or contracting nodes' layout is considered in Flare. If the layout calculations are included and the user selects to return to the original display, the layout must then consider the space that was used to display any expanded nodes and place the category nodes in appropriate positions. Prefuse fully supported this task as the tool manages the layout.

d) Browse the graph to find the *Venali Fax Web Service* web service; and
e) Select this web service to view the web service rating.

Prefuse supported tasks (d) and (e) well as the user is able to expand the *Fax* category node to find available web services and find the specific web service. When the user selects the service, the details are immediately displayed in textboxes to the right of the display. The expanded category node's layout is handled well with Prefuse.

Tasks (d) and (e) were not well supported in Flare. A user can easily browse to find the *Fax* category node, but the layout becomes an issue when expanding the node. The nodes overlap resulting in occlusion and so it is difficult to identify the *Venali Fax Web Service*. If the user was able to select the correct node, the information was displayed similarly to Prefuse.

From this comparison, certain strengths and weaknesses were identified for each toolkit (Table 4-1 and Table 4-2).

Table 4-1 Strengths and Weaknesses of Prefuse

Prefuse	
Strengths	**Weaknesses**
• Layout is immediate (smoother) on load	• Desktop application
• Toolkit provides collision detection to avoid overlapping of nodes	• Limited UI (User Interface) controls in Java
• Layout of expanded nodes is provided	
• Built in overview control to display smaller view of the graph to view the entire collection	

A preliminary requirement of an IV tool was that it should support web development, thus the reason for the inclusion of Silverlight. Flare provides support for web applications and improved UI controls over Prefuse. Prefuse, however, provides better support for visualization and interaction with the web service collection. Prefuse also provides an overview control which is part of the Overview task identified in Chapter 3, whereas Flare does not support this. Therefore, a decision was taken to use Prefuse to develop the prototype.

Table 4-2 Strengths and Weaknesses of Flare

Flare	
Strengths	**Weaknesses**
• Web Application	• Performance of graph and application is slower
• Better UI controls	• No collision detection is provided in Flare thus leading to overlapping of nodes
• Can control animation	• Layout of expanded nodes needs to be implemented by the programmer
	• No built in overview control

4.3.2. Data

The simplest way to connect the XML document created using XSLT and the ProgrammableWeb API (Section 4.2.1.) is to convert the XML file to GML (Graph Mark-up

Language) or Graph-ML. Prefuse provides a GML reader that reads in the data from the GML file and converts the data into graph information in terms of nodes and connecting edges.

GML is structured according to nodes and edges. Nodes and edges can have data keys which act as node or edge attributes. A node does not necessarily need to have all data keys assigned. An edge requires a source and target node defined by an id. This id is assigned to each node for uniqueness.

The previous transformation discussed in Section 4.2.1. of this chapter resulted in an XML document where all the web services in the collection were listed and sorted by category and then by web service name. The following steps were involved in this transformation:

a) Declare general data keys for all nodes;
b) Include the collection node with a name and type;
c) Include the category nodes with a name and type;
d) For each web service retrieve the web service information for each data key and create the web service node;
e) Connect the collection node to the category nodes; and
f) Connect the category nodes to the web services contained in each category.

The resulting GML document includes the nodes with connecting edges. Figure 4-8 shows a snippet of the final GML document:

```xml
<graph id="WSGraph" edgedefault="undirected">
  <key id="name" for="node" attr.name="name" attr.type="string" />
  <key id="WSCategory" for="node" attr.name="WSCategory" attr.type="string" />
  <key id="type" for="node" attr.name="type" attr.type="string" />
  <key id="link" for="node" attr.name="link" attr.type="string" />
  <key id="summary" for="node" attr.name="summary" attr.type="string" />
  <key id="rating" for="node" attr.name="rating" attr.type="string" />
  <key id="description" for="node" attr.name="description" attr.type="string" />
  <key id="protocol" for="node" attr.name="protocol" attr.type="string" />
  <key id="wsdl" for="node" attr.name="wsdl" attr.type="string" />
  <key id="provider" for="node" attr.name="provider" attr.type="string" />
  <key id="price" for="node" attr.name="price" attr.type="string" />
<node id="1">
  <data key="name">PWebCollection</data>
  <data key="type">COLLECTION</data>
    </node>
<node id="2">
  <data key="name">Advertising</data>
  <data key="type">CATEGORY</data>
    </node>
<node id="3">
    ...
<node id="57">
  <data key="name">AdSpeed</data>
  <data key="WSCategory">Advertising</data>
  <data key="type">WEBSERVICE</data>
  <data key="link">http://www.programmableweb.com/api/adspeed</data>
  <data key="summary">Online advertising management services</data>
  <data key="rating">4.2</data>
  <data key="description">Use the AdSpeed API to access your zones, ads, campaigns,
     and reports programmatically from AdSpeed, if you are an existing client, or there
     is also an option for Web developers to access the AdSpeed platform and send the
     app to AdSpeed to showcase it. AdSpeed is an ad management solution where you
     enter your ads into the system and then AdSpeed serves the ads, tracks
     impressions, clicks, revenues, conversions, and shows real-time stats.</data>
  <data key="protocol">REST</data>
  <data key="wsdl" />
  <data key="provider">http://adspeed.com</data>
  <data key="price">None</data>
 </node>
    ...
  <edge source="1" target="2" />
  <edge source="1" target="3" />
  <edge source="1" target="4" />
  <edge source="1" target="5" />
  <edge source="1" target="6" />
  <edge source="1" target="7" />
  <edge source="1" target="8" />
  <edge source="1" target="9" />
  <edge source="1" target="10" />
```

Figure 4-8 The Final GML Document

4.3.3. Visualization Techniques

An alphabetical list technique and a network IV technique were implemented in SerViz for comparison purposes. Both techniques are referred to as views in SerViz and provide support for the tasks listed in Section 4.2.2. A user can toggle between the two views by means of tabs and the default display is the network view.

4.3.3.1. List View

The list view was implemented using the ProgrammableWeb publication site and is shown in Figure 4-9.

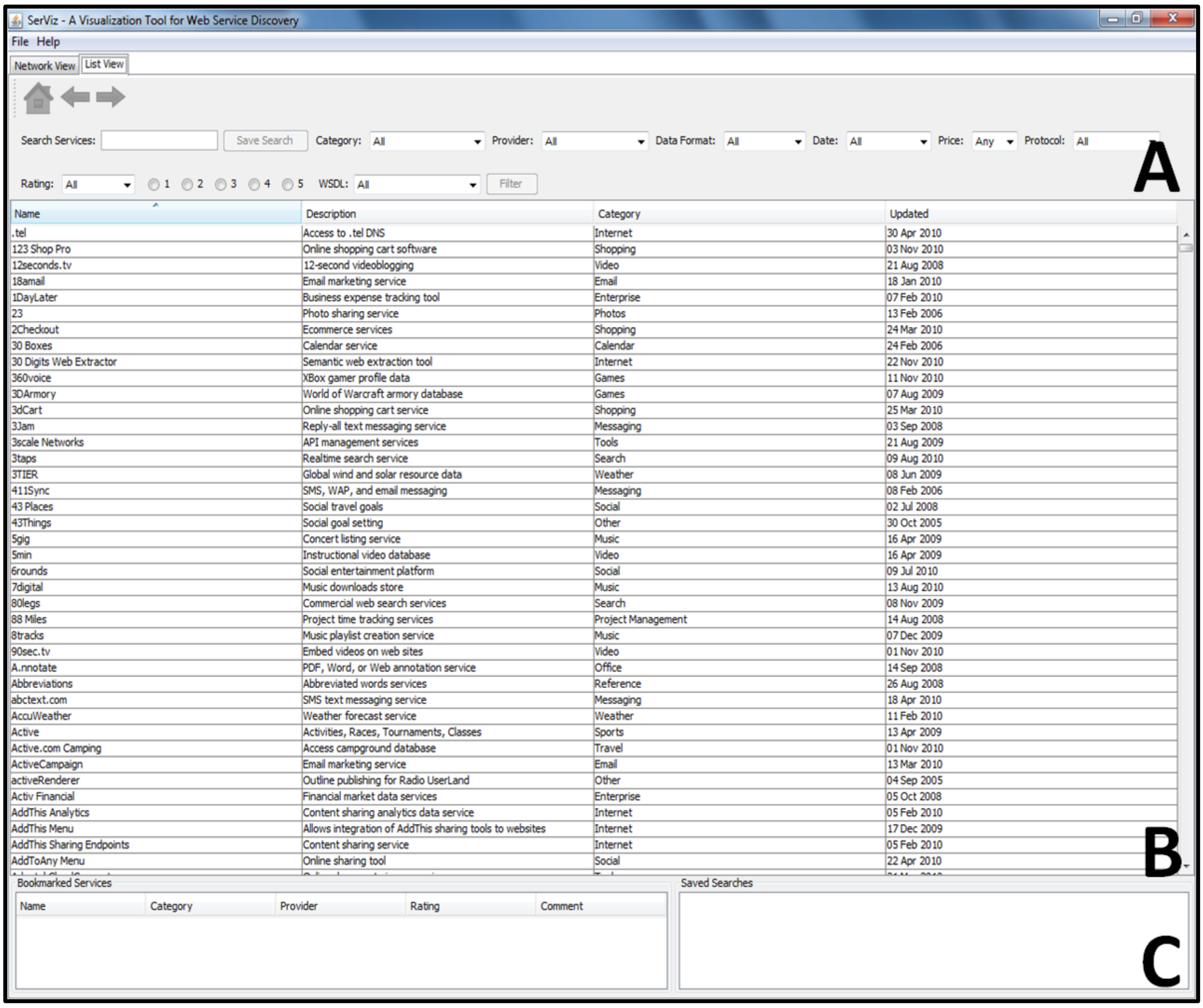


Figure 4-9 List View in SerViz

A scrollable list containing all the available web services in the collection is shown in Section B of Figure 4-9 with certain information initially displayed in the list also provided on the publication site, namely the web service name, a one-line description of the web service functionality, the category the web service belongs to and the date the web service was updated. This list represents the Overview as a user can view the entire collection of web services by scrolling the list. The list may not be entirely equivalent to the publication site as a user of the publication site would scroll the web page to view the collection of web services where a user would scroll the alphabetical list in SerViz.

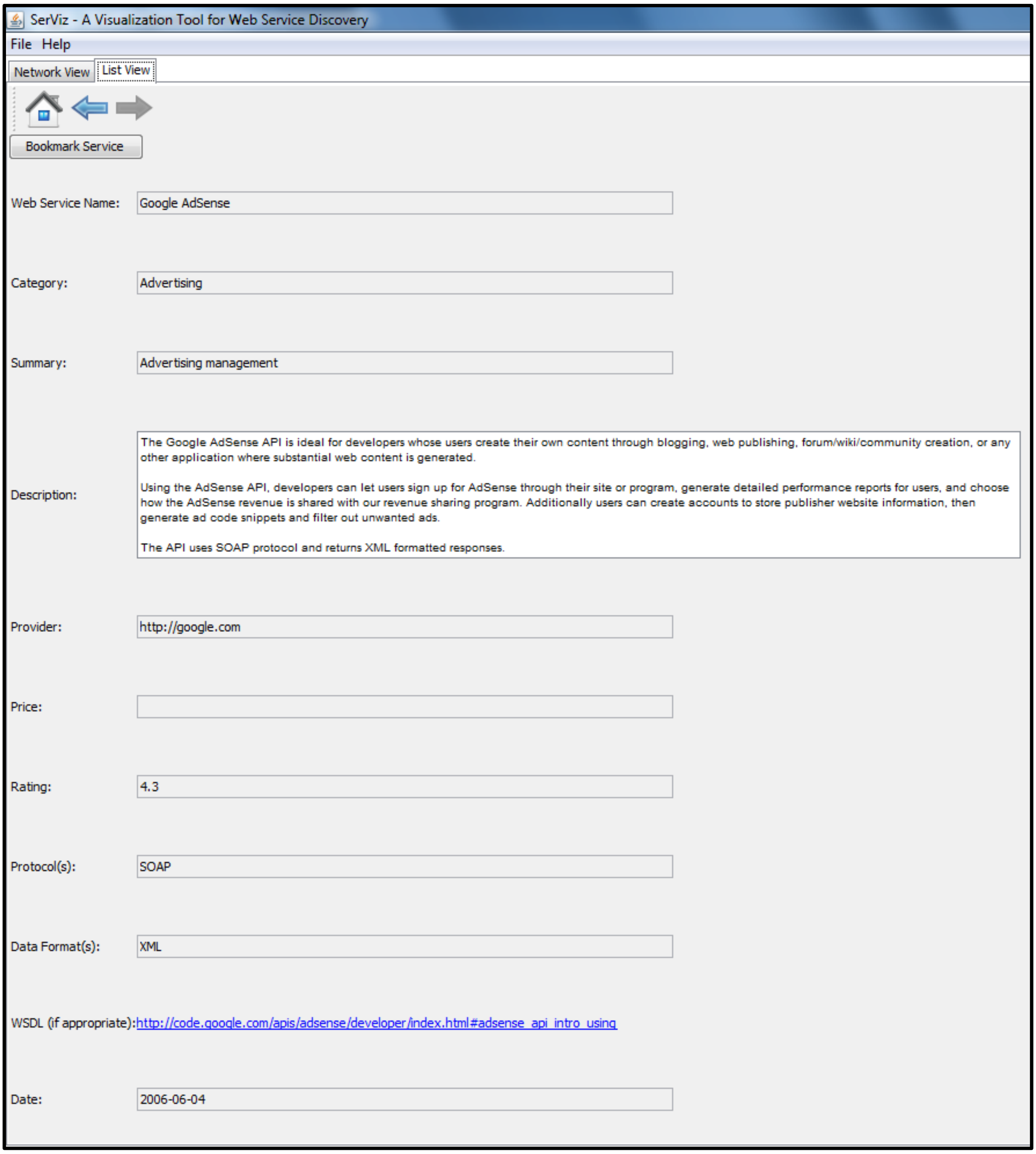


Figure 4-10 Details-on-Demand for the List Technique

The list view allows the user to search or browse for web services. A user can search for a web service using a keyword search and/or the filter facility (Section A of Figure 4-9). A user can filter on provider, data format, date, price, protocol, rating and whether the web service provides a WSDL file or not. Section C in Figure 4-9 illustrates the *Bookmarked Services* and *Saved Searches* lists. A user can view more information of a specific web service and return to the list to further discover web services. Figure 4-10 shows the details of the *Google*

AdSense web service supporting the details-on-demand task. A user can view the following detailed information for each web service:

- Web Service Name;
- Category;
- Summary or short one-line description;
- Detailed description of the web service functionality;
- Provider;
- Price;
- Rating;
- Protocol(s);
- Data Format;
- The link to the WSDL if it is provided; and
- Date.

A user can bookmark the web service at this stage. The user can return to the original list using either the Home or Back buttons.

4.3.3.2. Network View

The network view is based on the prototype shown in Figure 4-3, but was changed due to the design constraints of Prefuse and Java. The network view shown in Figure 4-11 initially displays a collection node connecting the category nodes in the web service collection. The web services contained in each category are initially hidden to minimize clutter on the screen. A user can explore the graph and browse for web services in this section by navigating and manipulating the graph. The size of the category nodes represent the number of web services contained in each category. The larger the node, the more web services are contained in that node calculated as a ratio to the total number of web services in the collection.

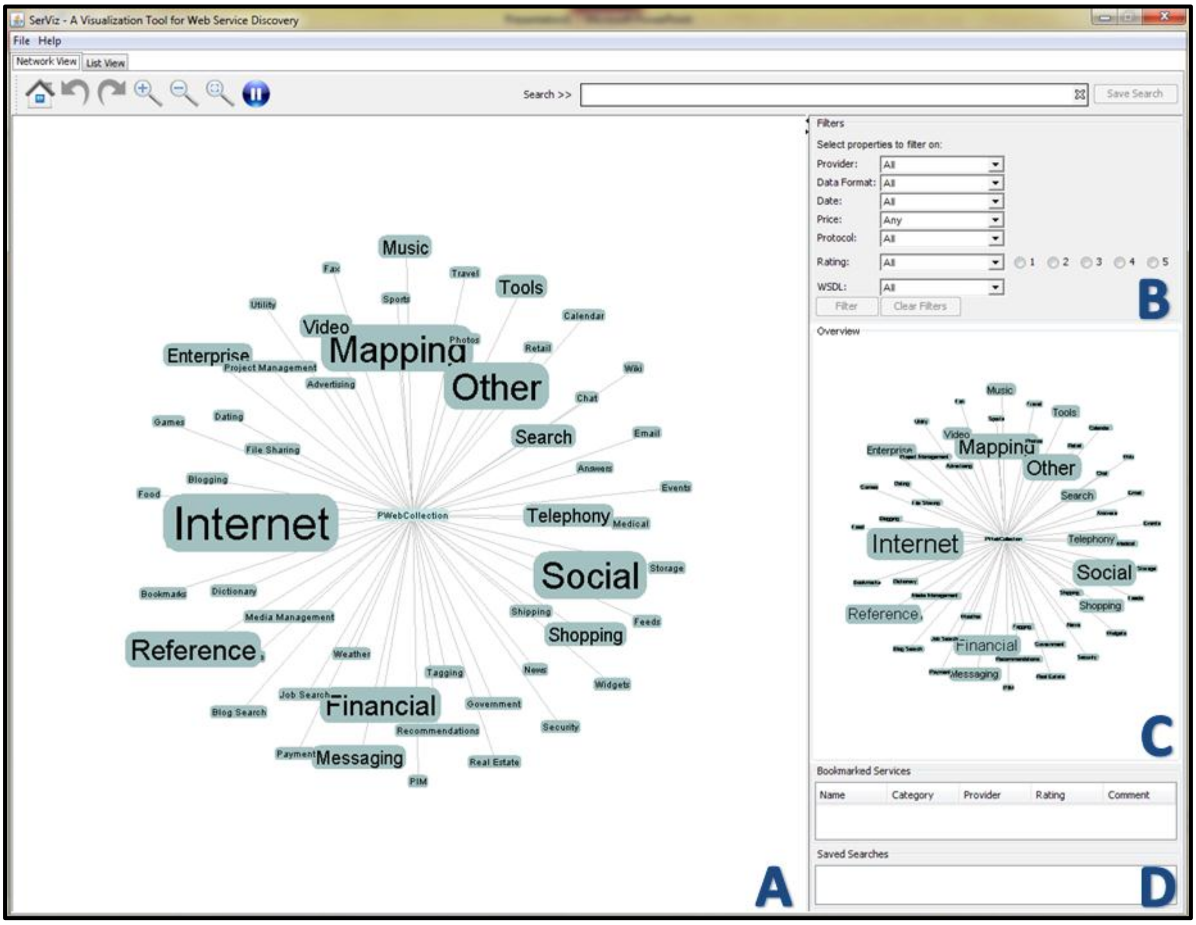


Figure 4-11 Network View in SerViz

The graph layout is controlled using a Force Directed Layout (FDL). The layout organizes the node positioning and collision detection. The FDL continuously recalculates node positions based on graph actions and thus the graph is displayed with an animation effect.

The use of colour-coding was chosen to support the blue-yellow contrast discussed by Heer and Boyd (2005). The blue-yellow contrast also considers colour-blind users. A collection node is represented in black to distinguish from other nodes (Figure 4-12 a). Category and web service nodes are coloured in blue (Figure 4-12 b and c). When a node is selected the node is coloured in light blue. Category nodes that contain web services that are part of the search and/or filtering results remained coloured in blue and the web services that meet the search and/or filtering criteria are coloured in orange. Irrelevant categories and web services not part of the results are de-saturated to light-grey as used in Perer and Shneiderman's SocialAction tool (2006). This enables the user to see the results more clearly. When the user cancels the search and filtering, the nodes return to the original blue colour.

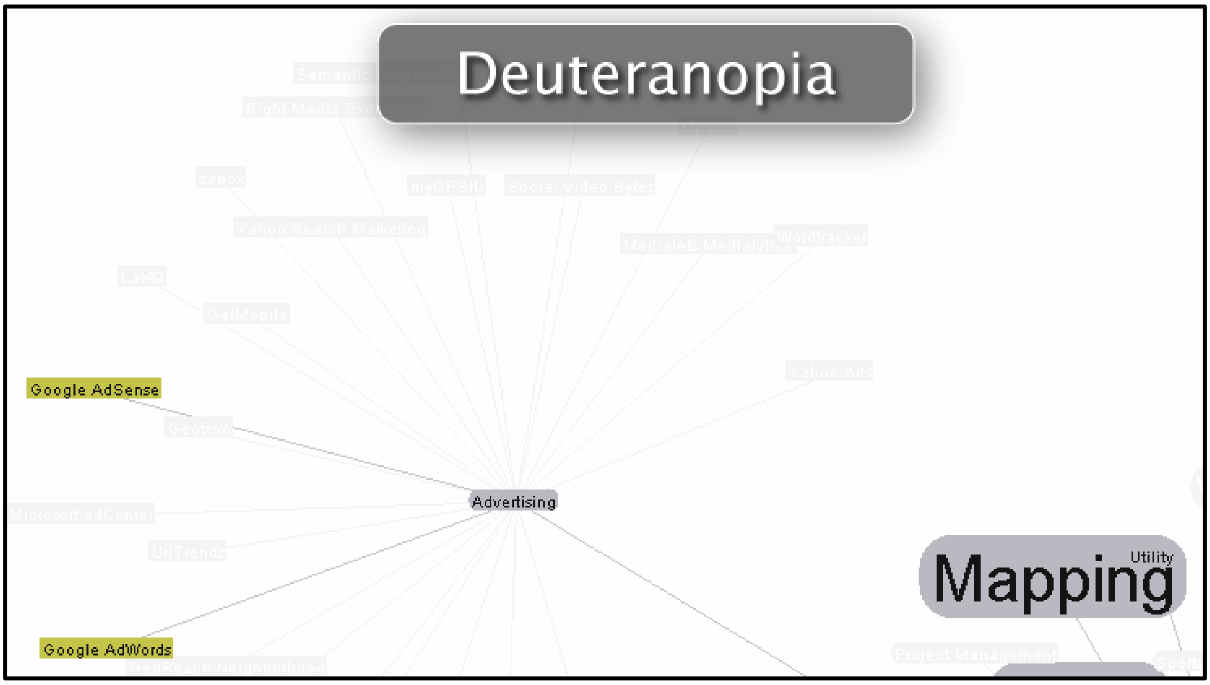
(a) (b) (c)

Figure 4-12 Types of Nodes in the Network View of SerViz (a) The Collection Node (b) A Category Node (c) A Web Service Node

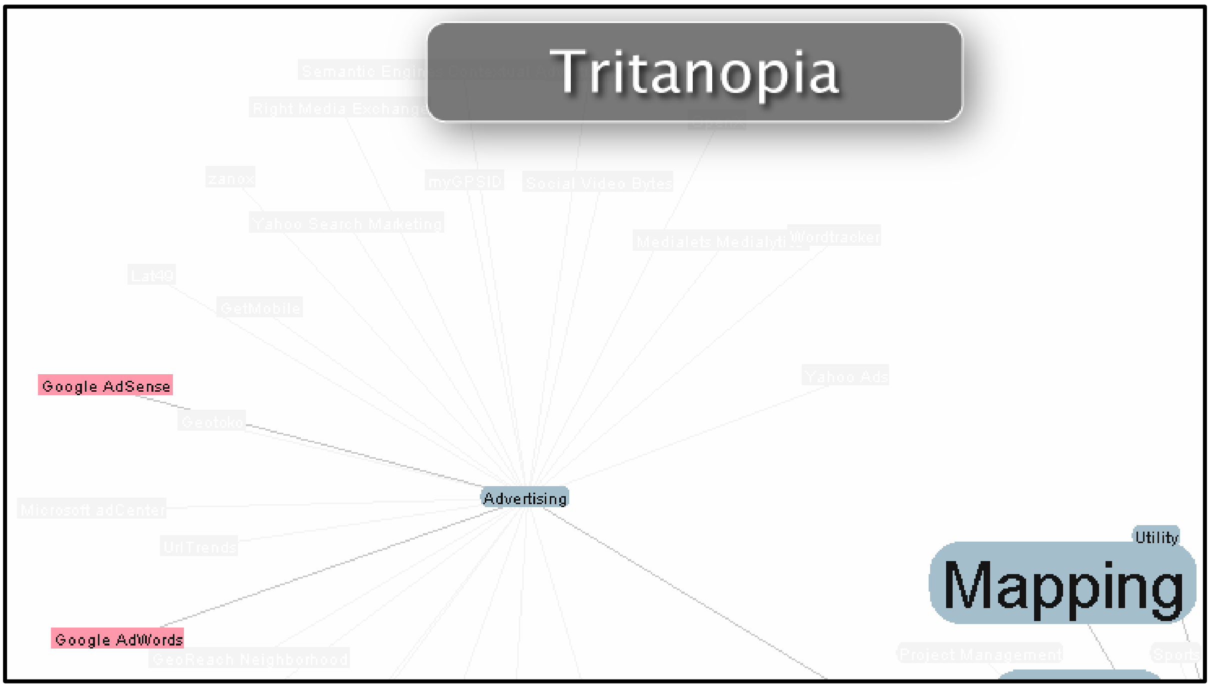
The colour-coding selected to distinguish between different types of nodes was chosen to support all users including those that suffer from colour-blindness. The most common form of colour-blindness is Deutranopia or Protanopia, commonly referred to as "red-green blindness" (Bernhard and Kelso 2007). Users that suffer from this type of colour-blindness have difficulties in distinguishing between red and green. An extremely rare form of colour-blindness includes Tritanopia, which is where users have a blue weakness. Bernhard and Kelso's research (2007) resulted in the development of a tool, called Color Oracle, used to visualize how colour-blind users see different colours. Color Oracle was used to determine to what extent the selected colour-coding in the network view supported colour-blind users shown in Figure 4-13. The results for Deutranopia and Protanopia were identical and so only the Deutranopia form is shown with Tritanopia. It was determined that the colour-coding sufficiently supported colour-blind users.

An example of browsing (and searching) the web service collection in the network view is shown in Section A of Figure 4-14. To find the *Google AdSense* web service, a user would search for the web service and then expand the *Advertising* category. The user can mouse-over the node to view a summary of the web service properties or select the web service to view more detailed information.

A panel is provided to the right of the graph that can be hidden if needed. A user can search for web services using the search facility at the top of the screen and by using the multiple filter selection criteria shown in Section B of Figure 4-14. The provision of search and filter facilities supports the Filter task. A user can search by keyword and filter on provider, data format, date, price, protocol, user rating and whether the web service provides a WSDL file or not.



(a)



(b)

Figure 4-13 How Colour Blind Users View the Network View (a) Deutranopia Colour Blindness and (b) Tritanopia Colour Blindness

An interactive overview is provided and shown in Section C of Figure 4-14. The overview is a minimized version of the graph which can be used to navigate the main graph. This is especially useful if the graph is large and many categories are expanded. A user can drag the white rectangle section, representing the graph's display space, to specific sections of the graph.

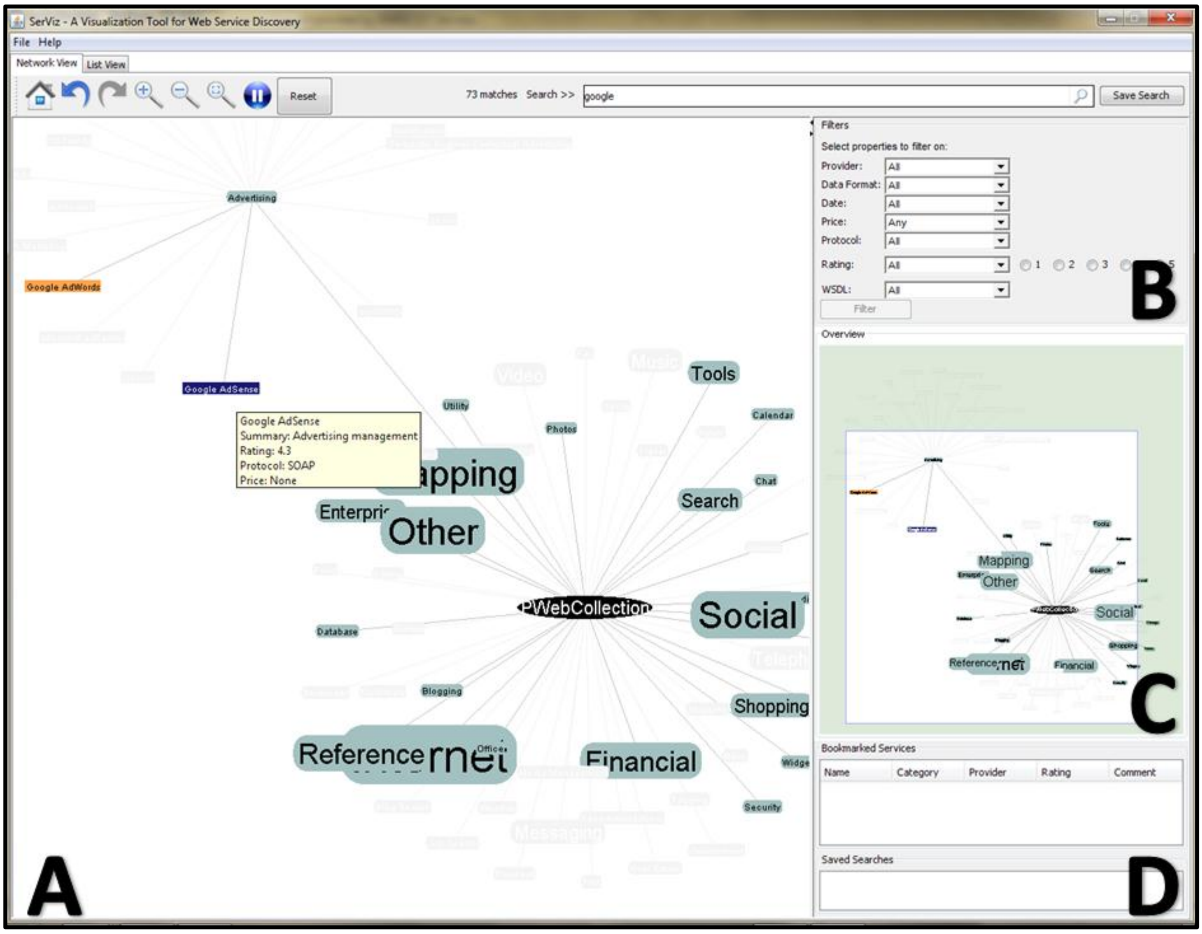


Figure 4-14 Searching and Browsing using the Network View

Section D in Figure 4-14 indicates how the network view in SerViz supports the History and Extract tasks. A user is given the option to bookmark a web service when viewing the details of the service (Figure 4-15).

A user can bookmark a web service to be used at a later stage in the *Bookmarked Services* list. A user can enter a comment as a note explaining the potential use of the web service to serve as a reminder when the user has completed the search for services. This prevents the user from having to search for the same web service each time he would like to use the web service. If the user selects the web service in the list, the application opens the website in the browser where more information can be found. A user can also save a search for web services in the *Saved Searches* list and return to the search after selection in this list.

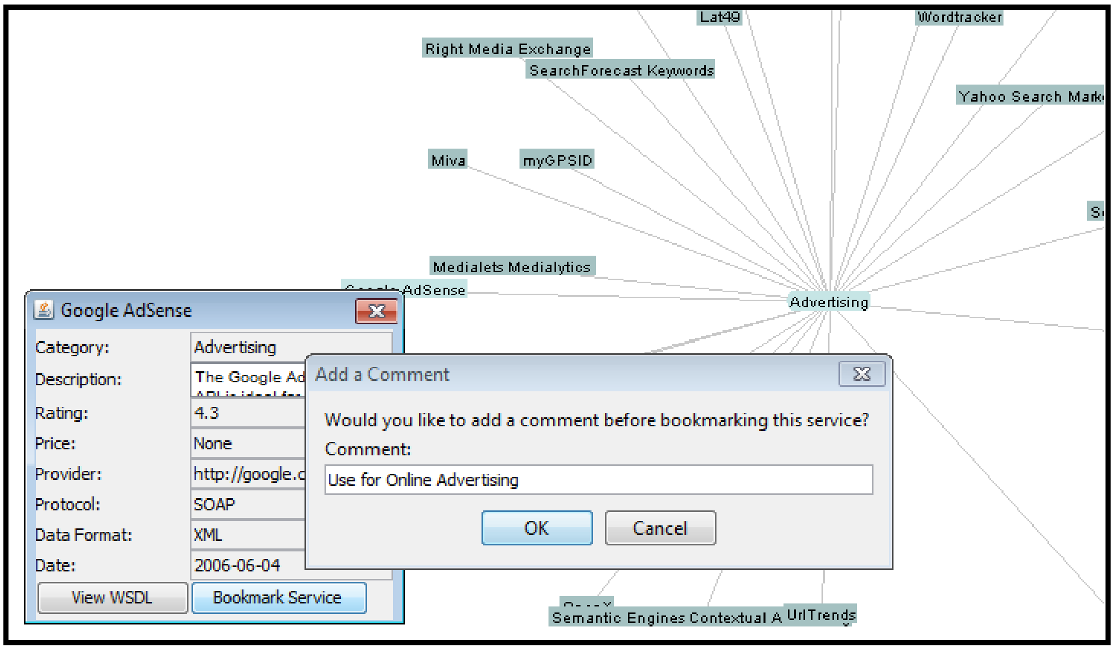


Figure 4-15 Bookmarking a Web Service using the Network View

A user can return to the display of the collection node and category nodes by selecting the *Home* button on the toolbar at the top-left of the screen (Figure 4-16). A user can undo and redo graph actions such as selecting the *Home* button, expanding or contracting category nodes, searching, filtering, viewing details, bookmarking a service and saving a search. Zooming in and out of the graph, fitting the graph to the screen space and allowing for pan and scroll controls also supports graph navigation. A user can also pause the animation effect provided by the FDL, but the layout forces the animation to continue when actions are performed that require node repositioning such as expanding a category node.

Figure 4-16 Controls for the Network View

4.4. Discussion

SerViz was developed in Java using the Prefuse data visualization toolkit according to the IV requirements for web service discovery as identified in Section 3.5. SerViz supports all the tasks outlined in Section 4.2.2. SerViz implements two views of the web service collection, namely the list and network views.

The list view was implemented for comparison purposes with the network view as the list view is used in current web service discovery methods. A comparison is needed to determine which view provides better support for web service discovery.

Prefuse provides extensive support for interactive data visualization. The process to convert the data to be visualized into the actual IV technique is relatively simple. All tasks listed in Section 4.2.2. could be implemented using this toolkit. The network IV technique provided by Prefuse responded well to the large dataset (n=2390) once the web service nodes were hidden. Prefuse automatically calculates layout and node positioning where other tools, such as Flare, do not provide sufficient support for this.

The FDL used for the network graph was useful for layout calculations although the continuous animation effect could possibly distract some users. The animation is an effect resulting from the limited support for the FDL to find the ideal position or convergence of each node. A limitation of using this layout for the network graph is that the expanding and contracting of nodes is not supported for the FDL and this had to be implemented separately. Additionally, a static minimized overview is provided by the toolkit and an interactive version of the overview needed to be developed. Although Shneiderman (1996) identified undoing and redoing of graph actions as a requirement for the History task, Prefuse does not provide support for this. Lastly, the details-on-demand task is not supported in Prefuse. A Java internal dialog was used to provide more detailed information on a web service and a tooltip was used to provide details on a category.

4.5. Conclusion

This chapter addressed the third research question listed in Chapter 1, namely how can IV techniques be applied to support web service discovery? The chapter answered this research question by outlining the design and implementation of SerViz – a prototype tool for interactive visualization of web service collections.

The ProgrammableWeb web service collection was used as the data source for SerViz. This web service collection is a large collection that provides approximately 2335 web services classified into 54 categories. Although this web service collection provides limited QoS (Quality-of-Service) information, information is provided for each web service including a web service name, description, rating, provider, data format, a link to the web site providing the service and the WSDL file (if appropriate). A limitation of this web service collection is that it is hierarchical in that a web service is contained in only one category. The ProgrammableWeb API was used to retrieve the web service collection and the collection was stored in an XML document. The web service collection retrieved needed to be customized to only include the relevant information. Appropriate web service attributes were extracted using XPath and XSLT resulting in a different XML document.

Several tasks were identified in Section 4.2.2., using the IV requirements for web service discovery outlined in the previous chapter. A list technique is the technique currently used by existing web service discovery methods. SerViz was developed using the Prefuse data visualization toolkit and implemented in Java. Although there are a few limitations with Prefuse, the toolkit provides enhanced support for interactive visualization compared to other existing toolkits such as Silverlight and Flare. SerViz visualizes the ProgrammableWeb web service collection using both the alphabetical list and network IV techniques. The XML document containing the ProgrammableWeb web service collection needed to be converted to GML to be visualized by SerViz. All of the tasks identified in Section 4.2.2. were implemented in both techniques in SerViz.

The next chapter will discuss a usability evaluation of SerViz to determine the usability of the prototype. The evaluation will also be used to determine if the network IV technique provides better support for web service discovery than the list technique.

Chapter 5: Evaluation

5.1. Introduction

This chapter will address the last research question defined in Chapter 1, namely *how effective are these IV techniques in supporting web service discovery?* This research question is answered by making use of an experimental research method to compare the alphabetical list technique and network IV (Information Visualization) technique implemented in SerViz as discussed in the previous chapter (Beets and Wesson 2011b). A usability evaluation was conducted to compare the two techniques and to identify any usability problems with SerViz. The results of the usability evaluation were used to determine what further steps need to be taken in this research.

The chapter discusses the research design and results of a usability evaluation used to compare the list and network IV techniques. Usability problems (if any) resulting from this evaluation are identified and relevant changes made to SerViz to address the problems are discussed. The chapter concludes with a discussion of the results of the study and what research is to follow.

5.2. Usability Evaluation

A usability evaluation was conducted to compare the list and network views. The evaluation captured performance metrics for each of the tasks completed for each view during the usability evaluation. Questionnaires were used to determine the user satisfaction after using each of the techniques. Usability problems that were identified from the results of the evaluation are discussed.

5.2.1. Evaluation Design

The experiment included six tasks for each technique evaluated. The experiment used a within subjects design making use of counterbalancing to avoid the learning effect.

The usability metrics that were captured included effectiveness (task completion), efficiency (task completion combined with time-on-task) and user satisfaction. Eye-tracking was also used to identify scan paths and gaze patterns for each technique. The eye-tracking results were used to supplement the other usability results. Ten participants completed the evaluation using the SerViz prototype.

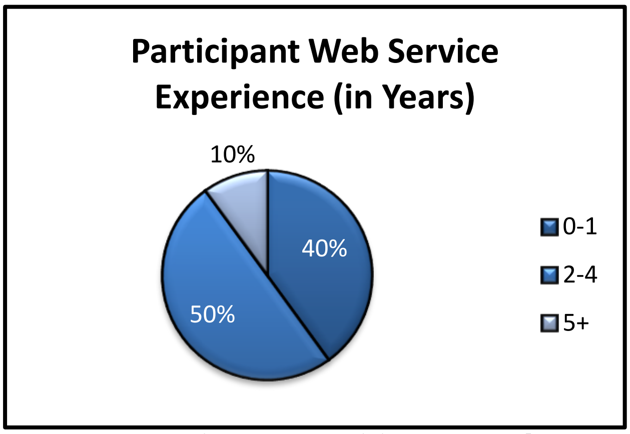
5.2.1.1. Evaluation Objectives

A list view and a network view were both implemented in SerViz to visualize web service collections. The aim of the usability evaluation was to compare the list and network views used to visualize web service collections in SerViz. The evaluation was also used to identify any usability problems in SerViz.

5.2.1.2. Participants

A representative sample consisting of ten (nine male, one female) post-graduate students from the Department of Computing Sciences at the Nelson Mandela Metropolitan University (NMMU) were used for the usability evaluation. According to Nielsen (2000), a single iteration of a usability evaluation requires five users and ten participants are able to identify most usability problems. Participants were in the age range of 20 – 39 years. The participants were required to have advanced computing knowledge. The participants were also required to have experience in using search engines, using a computer and the Internet. Each evaluation took approximately 45 minutes to complete.

Most of the participants had at least 10 years' experience using a computer. Although not required in order to participate in the evaluation, participants were asked whether they had any prior experience using IV tools and experience using web services (Figure 5-1). Sixty percent of participants had used an IV tool prior to taking part in the study and had at least two years' experience using web services.



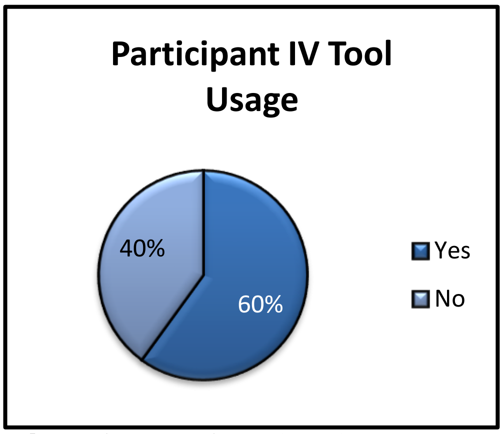


Figure 5-1 Participant IV Tool Usage and Web Service Experience (n=10)

5.2.1.3. Evaluation Metrics

Performance metrics were measured for each task completed by the participants. The following performance metrics were measured for each task:

- *Effectiveness:* Task success, i.e. whether the participant could complete the task or not.
- *Efficiency:* Task completion combined with the time taken to complete the task.

User satisfaction was also measured by means of questionnaires. This was captured using post-task and post-test questionnaires to determine the user reaction to the two views.

5.2.1.4. Questionnaires

Participants were required to have advanced computing knowledge and experience to complete the evaluation. A background questionnaire was provided to the participants at the start of the evaluation to capture the details of the user (Appendix B). This questionnaire was used to collect demographic and experience details for each participant.

A post-task questionnaire measuring cognitive load using a 7-point semantic differential scale and a 7-point Likert scale was provided to each participant after the evaluation of each view (Appendix D and E). A post-test questionnaire for user satisfaction was completed by each participant at the end of the evaluation (Appendix F).

The questionnaires were adapted from the Computer System Usability Questionnaire (CSUQ) (Lewis 1995) and the NASA - TLX form (Hart and Staveland 1988). The NASA - TLX form was used for measuring cognitive load to determine if the network view required any additional cognitive load. The CSUQ was used to measure standard usability.

5.2.1.5. Experimental Setup

The usability evaluation took place in the Usability Lab in the Department of Computing Sciences at NMMU. The usability lab consists of two rooms separated by one-way glass, namely the evaluation and control rooms. The participant was seated in the evaluation room and completed the tasks on the Tobii T60 Eye Tracker shown in Figure 5-2. The evaluator was positioned in the control room and was able to observe the participant during the evaluation. The evaluator and participant communicated via microphones and speakers provided in each room of the lab.

The participant completed the evaluation using the eye tracker computer and operating system. The evaluation took place using the Tobii Studio software which was used to track where the user was looking on the screen. Internet Explorer was also used if a user selected a service or viewed the WSDL (Web Service Description Language) file of a web service.

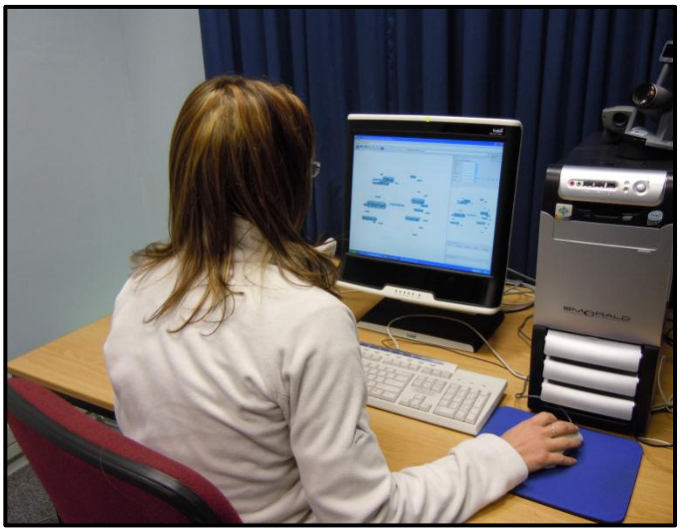


Figure 5-2 Evaluation Room of the NMMU Usability Lab

A screen recording was used to capture each evaluation. Two tests were created using the Tobii software, one where the participants used the network view first and the second using the list view first. Counterbalancing was used to reduce the learning effect as half the participants used the network view first and the other half of the participants used the list view first. Once the evaluation commenced, instruction screens were provided to the participants to guide them in what to do next. For example, an instruction would inform the participant to read a specific task. Once the participant was prepared to start a task he would select a key and the screen recording would display SerViz on the screen. The user would complete the task and select another key to end the task.

5.2.1.6. Tasks

Two similar but not identical sets of tasks for each view were given to each participant (Appendix C). The six tasks for each view are described in Table 5-1. A mapping was done to determine which of the IV tasks in Table 5-1 supported each evaluation task for the network view.

The Overview, Relate (a) and History (a) IV tasks were supported by all the evaluation tasks. The Relate (b) task was supported by SerViz, but not evaluated in this study as it was not part of the focus of the evaluation.

Table 5-1 Mapping of the Evaluation Tasks to the IV Tasks

Evaluation Tasks	IV Tasks Supported
1. Finding information in the web service collection by browsing the collection	Zoom Details-on-Demand (a)
2. Viewing details of a web service (Browsing only)	Zoom Details-on-Demand (b)
3. Searching for web services	Filter Details-on-Demand
4. Searching and filtering to find web services	Zoom Filter Relate (c)
5. Bookmarking a web service	Zoom Details-on-Demand (b) Filter (a) History (b) Extract (a)
6. Saving a search for web services	Zoom Filter (a) History (b) Extract (b)

5.2.1.7. Procedure

Each participant evaluated SerViz individually. Participants were required to complete one task at a time. The evaluation commenced by briefly explaining the procedure to the participant. The participant then provided consent to take part in the evaluation (Appendix A). Before evaluation of the prototype commenced, the participant completed a background questionnaire (Appendix B). The eye-tracker was then calibrated for the participant to allow it to accurately capture where the user was looking on the screen. After completing the tasks for each view, the participant completed a post-task questionnaire (Appendix D and E). Finally the participant completed a post-test questionnaire comparing the two views (Appendix F).

5.2.2. Evaluation Results

Three evaluation metrics were used in the usability evaluation, namely effectiveness, efficiency and user satisfaction. Eye-tracking was also used to supplement these results. This section describes the results of the evaluation.

5.2.2.1. Performance Results

This section discusses the results in terms of effectiveness and efficiency of the tasks for each view. The effectiveness results supported both views while the network view was faster in two of the six tasks as can be seen from the efficiency results.

Effectiveness

For each task in the task lists, participants were required to answer a question relating to the task. Effectiveness for each task was measured as a percentage of the number of correct answers.

Most tasks received 100% success rate, except for two tasks for the network view which received 80% (Task 1) and 90% (Task 6) success rates and one task for the list view which received a 70% (Task 6) success rate. The participants found the correct information in the web service collection for all three of these tasks, but answered incorrectly. Due to the high level of accuracy for both the network and list views, it can be concluded that both views effectively supported the users' tasks.

Efficiency

A *Reset* button was included in the toolbar of each view to reset the network or list views, clear the search fields and clear the filters. After each task the participant was required to select the *Reset* button. The time for each task was taken as the time from starting the task, when SerViz was displayed on the screen after reading the instruction, until the *Reset* button was selected. This duration included the participant reviewing the task instructions and answering the task. Table 5-2 outlines the task times for each task.

Using the Wilcoxon matched pairs non-parametric test, five of the six differences in task times were statistically significant, shown in bold in Table 5-2. Two were in the favour of the network view (Task 1 and Task 6) and three were in favour of the list view (Task 3, Task 4 and Task 5). The Wilcoxon matched pairs non-parametric test was used as the sample size of the usability evaluation was too small (n=10) to assume a normal distribution and the evaluation used a within subject design resulting in dependent samples (StatSoft Inc. 2003).

Table 5-2 Task Times in Seconds with Wilcoxon Test Statistics and p-values

Task	Network View	List View	Z	p-value
1	**33.02**	54.90	**2.19**	**0.03**
2	96.61	**45.82**	1.89	0.06
3	46.62	**26.75**	**2.40**	**0.02**
4	82.31	**48.21**	**2.80**	**0.01**
5	74.50	**57.67**	**2.19**	**0.03**
6	**48.47**	71.32	**2.80**	**0.01**

Task 1 involved browsing the web service collection to find a category without using the search or filter facilities. Task 6 involved saving a search for web services. The results showed that browsing the web service collection for a specific category and saving a search were significantly faster using the network view.

Task 1 involved browsing for a specific category where Task 2 required the participant to browse for a web service without being told which category the web service was in. The task times for Task 1 and Task 2 differed significantly. Task 2 involved first determining in which category the web service was classified and viewing the web service details. Although a list of the categories in the web service collection was provided to each participant beforehand, participants may still not have been aware of all the categories in the collection which could have contributed to the network view being slower in Task 2. An additional reason for Task 2 being slower than Task 1 for the network view could be because the category (in this case the *Medical* category) contained only a few web services and when the participant found the category it eventually became occluded by a larger category node positioned alongside this category.

A learnability factor could also be involved in these times. The list view is a familiar method of finding information where the network view is novel and so the participants had to learn the different aspects of the network view, such as navigation, while completing the tasks. Some tasks were also expected to be completed faster using the list view. For example, in Task 4 participants were required to search and filter the web service collection to find specific web services that meet the given criteria. When searching and filtering with the list view, the web service names were listed and so the user was not required to view further details of each web service. Using the network view, the participants were required to expand the category nodes to find the web services that met the given criteria which would take more time to complete. Given more time to learn the network view, the times for Tasks 2 – 5 could be improved.

5.2.2.2. Satisfaction Results

Satisfaction results were captured using the post-task questionnaires and the post-test questionnaire (Appendix D, E and F). The post-task questionnaires captured satisfaction in terms of different factors such as cognitive load and usability. A general section was also provided for the participants to note the most positive and negative aspects of each view. The post-test questionnaire captured preference between the two views. This section discusses these satisfaction results.

Satisfaction results were captured on the questionnaires using a 7-point semantic differential scale and a 7-point Likert scale. The satisfaction in the post-task questionnaire was broken down into three sections, namely cognitive load, overall satisfaction and usability.

Workload Results

Cognitive load for the network view was marginally lower than the list view shown in Figure 5-3 indicating 95% confidence intervals. There was no statistical significant difference between the two views for the workload ratings.

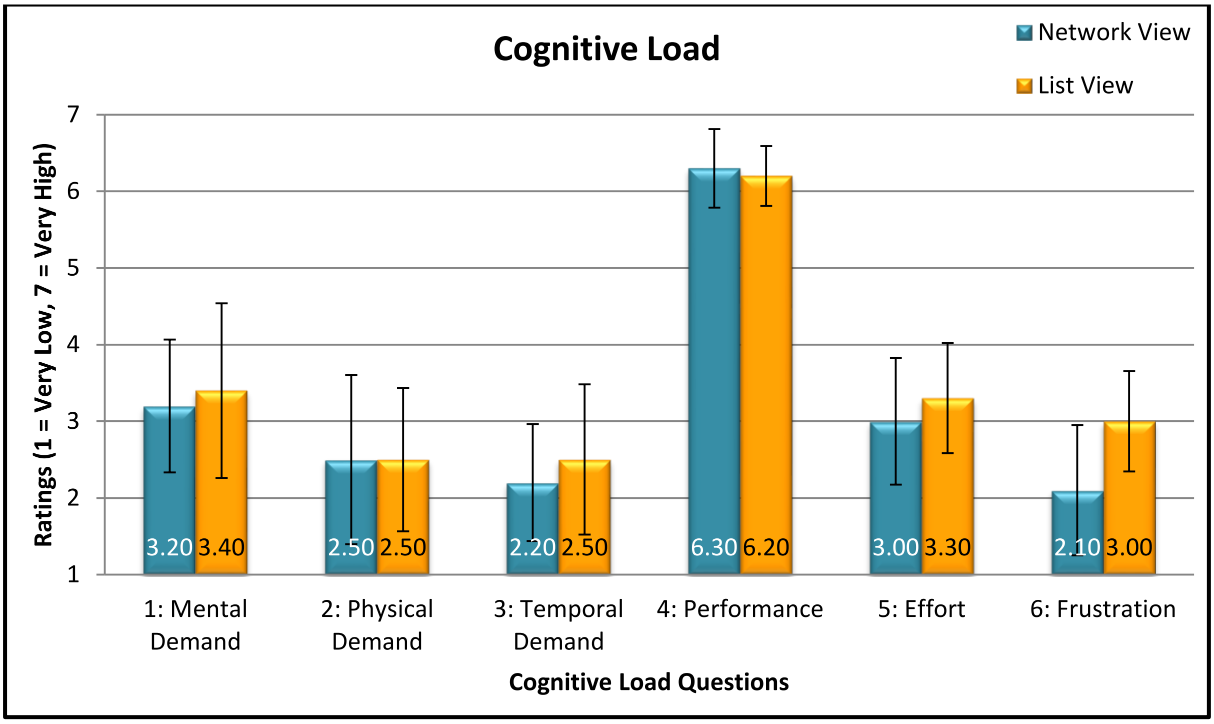


Figure 5-3 Cognitive Load using a 7-point Semantic Differential Scale (n=10)

Overall Satisfaction Results

The network view received higher ratings in all four satisfaction questions and the differences were statistically significant in all four of these questions (Table 5-3).

Table 5-3 Wilcoxon Matched Pairs Test for Satisfaction

Question	Z	p-value
1. Overall, I am satisfied with how easy it is to use View X	**2.37**	**0.02**
2. Overall, I am satisfied with View X	**2.37**	**0.02**
3. It was easy to learn to use View X	**2.37**	**0.02**
4. It was simple to use View X	**2.24**	**0.03**

The participants found the network view significantly simpler, easier to use and easier to learn. The participants were significantly more satisfied with the network view than the list view. Figure 5-4 highlights that the overall satisfaction was generally higher for the network view.

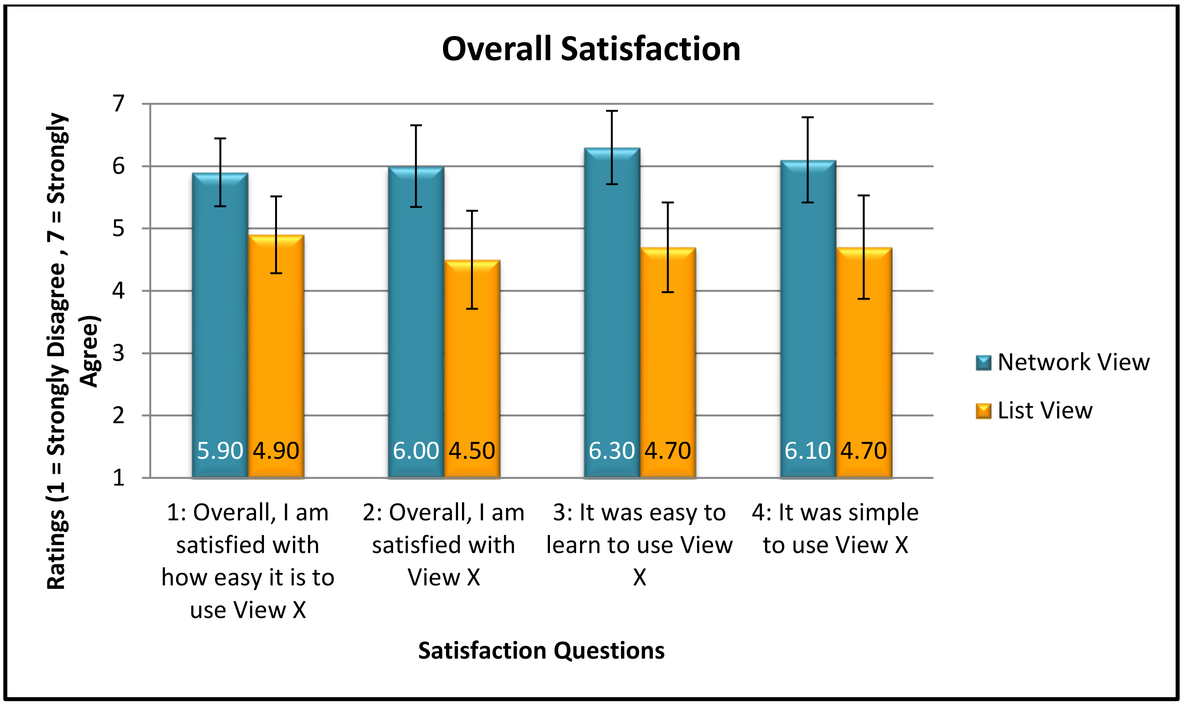


Figure 5-4 Overall Satisfaction using a 7-point Likert Scale (n=10)

Usability Results

The usability ratings were also all in favour of the network view. The differences in all questions except for questions 3 and 4 were statistically significant (see Table 5-4 and Figure 5-5).

Table 5-4 Wilcoxon Matched Pairs Test for Usability

Question	Z	p-value
1. I can effectively find web services using View X	**2.11**	**0.03**
2. I was able to find web services quickly using View X	**2.02**	**0.04**
3. I was able to efficiently find web services using View X	1.42	0.16
4. I became productive quickly using View X	1.68	0.09
5. View X has all functions and capabilities I expect from a web service discovery tool	**2.24**	**0.03**
6. I can effectively browse web service collections using View X	**2.80**	**0.01**
7. I was able to browse web service collections quickly using View X	**2.67**	**0.01**
8. I was able to efficiently browse web service collections using View X	**2.80**	**0.01**

Figure 5-5 illustrates the mean values for usability ratings for the eight questions. Participants perceived that they could effectively and efficiently discover web services using the network view, although the overall effectiveness for each view was identical and the network view was faster in only two of the six tasks. The perception that the participants could effectively and efficiently browse the web service collection using the network view supports the efficiency results that browsing was significantly faster using this view.

Generally, participants preferred the network view as a method to visualize web service collections. This supports the related work in which the type of data structure determined the visualization technique to be used for viewing web service collections (Shneiderman 1996).

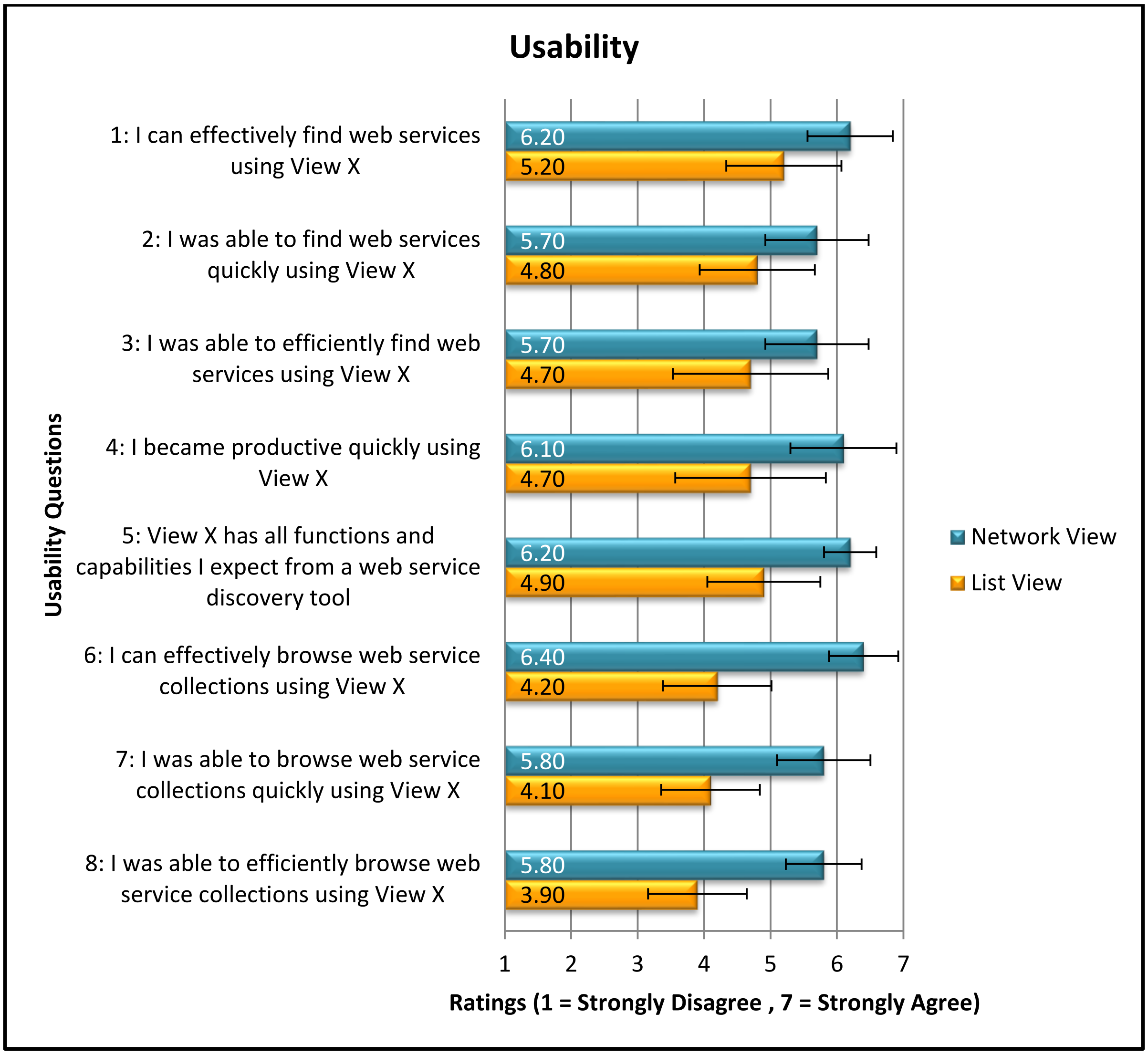


Figure 5-5 Usability Results using a 7-point Likert Scale (n=10)

Feedback was generally very positive for the network view. The comments regarding the most positive aspects of the network view are shown in Table 5-5 where n indicates the number of participants who made similar comments. Four participants found that the network view made effective use of screen space and three participants found that the categories were clearly displayed.

Table 5-5 Most Positive Aspects for the Network View

Description	n
1. Visualization makes good use of screen real-estate.	4
2. Categories are clearly shown.	3
3. Irrelevant data is faded out.	2
4. Easy to see numbers in collection & category.	2

One notable negative aspect that participants commented on was that the network view seemed slow as shown in Table 5-6. It was noticed upon experimental setup that the network view was distinctly slower than the list view. Running on a different computer, the network view performs as expected. Comparison of computer specifications between the eye-tracker computer and the other computer determined that the network view would be slower on the eye-tracker computer because of the graphics requirement. Other negative aspects of the network view related to on-screen clutter (n=3), occlusion issues (n=2) and the learnability effort involved (n=2). The negative comments relating to the network view are discussed in detail in Section 5.2.3.

Table 5-6 Most Negative Aspects for the Network View

Description	n
1. Can sometimes become too cluttered on the screen if a category has too many web services.	3
2. At first didn't realise I must drill down but quickly "learn" & then easier.	2
3. Some minor occlusion issues. Some smaller nodes almost too small to read.	2
4. Animation was a little slow on this PC.	2

General comments for the network view included positive comments regarding the view (n=3) and suggestions for improvement listed in Table 5-7. These suggestions are discussed further in Section 5.2.3.

Table 5-7 General Comments for the Network View

Description	n
1. The network view approach is novel and an efficient visualization technique.	3
2. I would recommend adding a "Hide" option for those greyed out services.	3
3. Maybe allow for functionality that expands the category for a particular search. This limits user intervention even less.	2
4. Make dialogs box regarding service information more attractive in terms of layout.	2

The list view received mixed comments. The positive comments related to the list view are shown in Table 5-8 where n indicates the number of participants who made similar comments. Participants found the list view to be simple and faster to use (n=3) and familiar (n=2). Participants also commented that all the web service information is displayed in the list view (n=2) and that the search facility made it easy to find web services (n=2).

Table 5-8 Most Positive Aspects for the List View

Description	n
1. The UI is simple & faster.	3
2. All information related to web service visible.	2
3. Familiar to what I've used before.	2
4. Search boxes made it easy to find web services.	2

Some of the negative comments of the list view are shown in Table 5-9. Participants found that scanning through the list was tedious (n=6).

Table 5-9 Most Negative Aspects for the List View

Description	n
1. Too much to read.	6
2. Viewing category information was more difficult. Would be useful for finding a specific web service, but less so if just searching for service to perform some function.	2

Participants made different comments in the general comments section in the post-task questionnaire. Participants found the assistance of the search facility in the list view helpful (n=2) as shown in Table 5-10.

Table 5-10 General Comments for the List View

Description	n
1. Search facility makes List view easy to use & efficient.	2

Post-Test Satisfaction Results

The post-test satisfaction results are shown in Figure 5-6. The post-test satisfaction questionnaire captured results using a 7-point Likert scale where a one represents preference for the list view and a seven represents preference for the network view.

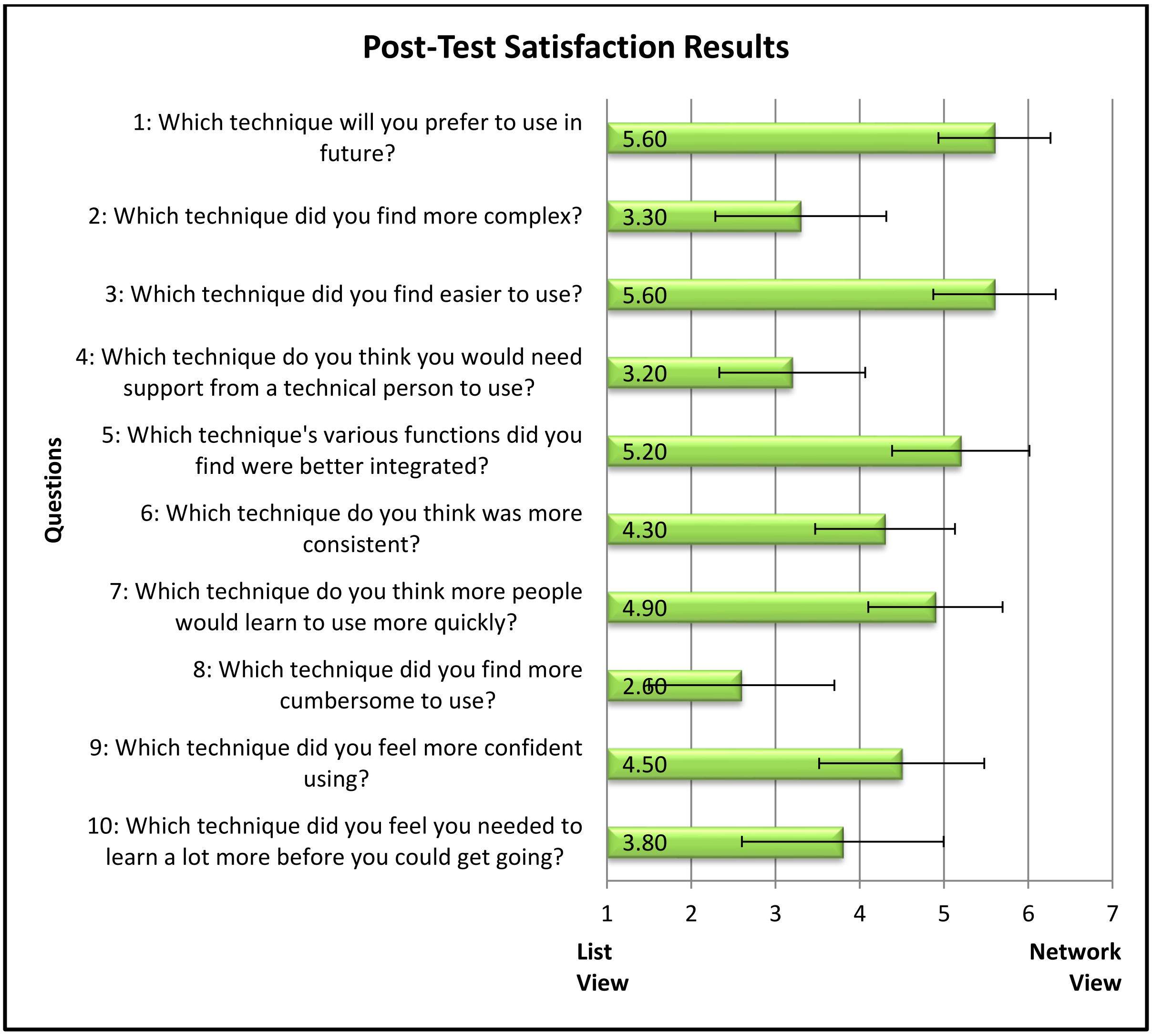


Figure 5-6 Post-Test Satisfaction Results using a 7-point Likert Scale (n=10)

Table 5-11 shows that the difference was statistically significant for questions 1, 3, 5, 7 and 8. Nine out of the ten participants would prefer to use the network view in future. The participants found the network view easier to use than the list view, better integrated and less cumbersome to use. The results also showed that the participants perceived the network view to be the view that would be the quickest to learn, even though a list is the existing method used to visualize web service collections.

Table 5-11 Wilcoxon Matched Pairs Test for the Post-Test Questions

Question	Z	p-value
1. Which technique will you prefer to use in future?	**2.60**	**0.01**
2. Which technique did you find more complex?	1.30	0.19
3. Which technique did you find easier to use?	**2.55**	**0.01**
4. Which technique do you think you would need support from a technical person to use?	1.62	0.11
5. Which technique's various functions did you find were better integrated?	**2.20**	**0.03**
6. Which technique do you think was more consistent?	0.63	0.53
7. Which technique do you think more people would learn to use more quickly?	**1.96**	**0.05**
8. Which technique did you find more cumbersome to use?	**1.96**	**0.05**
9. Which technique did you feel more confident using?	1.12	0.26
10. Which technique did you feel you needed to learn a lot more before you could get going?	0.34	0.74

General comments received from the post-test questionnaires also supported the network view. These comments are shown in Table 5-12 where n indicates the number of participants who made similar comments. Participants found SerViz to be well-integrated (n=3) and found that the network view provided a better organization of the web service collection (n=2) and made web service discovery easier (n=2).

Table 5-12 General Comments in the Post-Test Questionnaire

Description	n
1. Neat, clean, well-integrated system. Performed tasks easily, quickly and effectively. Lots of good functionality.	3
2. Network view offers better overview and organisation of contents of the service repository. Would've loved it if such an interface was available previously when I browsed web service collections.	2
3. The network view system makes web service discovery a lot easier.	2

5.2.2.3. Eye-Tracking Results

Eye-tracking was used to determine whether the participants noticed and made use of the different sections of the screen. Eye-tracking can also help to determine eye-gaze patterns (Namahn 2001). Although certain aspects of the network graph could not be analyzed using eye-tracking data due to the static image limitations, heat maps and gaze plots were captured for the network view as a whole. The heat maps were taken from the start of a task until before expanding a category. If the participants expanded a category, the graph would have changed as the category would have been centred. This would have led to a misrepresentation of the heat map as a heat map only represents a static image of the screen.

A heat map was taken of the first five participants for two of the six tasks. The heat maps were taken for only five of the ten participants because two tests were created using the Tobii Studio software to avoid the learning effect. One test was created for participants using the network view first and one test for participants using the list view first. It was not possible to combine multiple tests in the Tobii Studio program to obtain a heat map for all the participants.

The first task involved browsing the web service collection without using the search or filter facility. The heat map for this task is displayed in Figure 5-7 showing that emphasis was placed on the graph (Section A in Figure 5-7).

The users were required to find a specific category in the web service collection and write down the number of web services contained in this category. The specific category was positioned in the bottom-right of the graph. From the recordings and the heat map, the general process for completing this task was to look through all the categories to find the specific category and either expand the category or use the tooltip to complete the task. If a participant could not find the category easily, he would scan the rest of the interface to identify anything to help to find this category. The participant would then look through the categories once again and find the category. This task was the first time that the participants viewed the network view and this could also be the reason for scanning the interface.

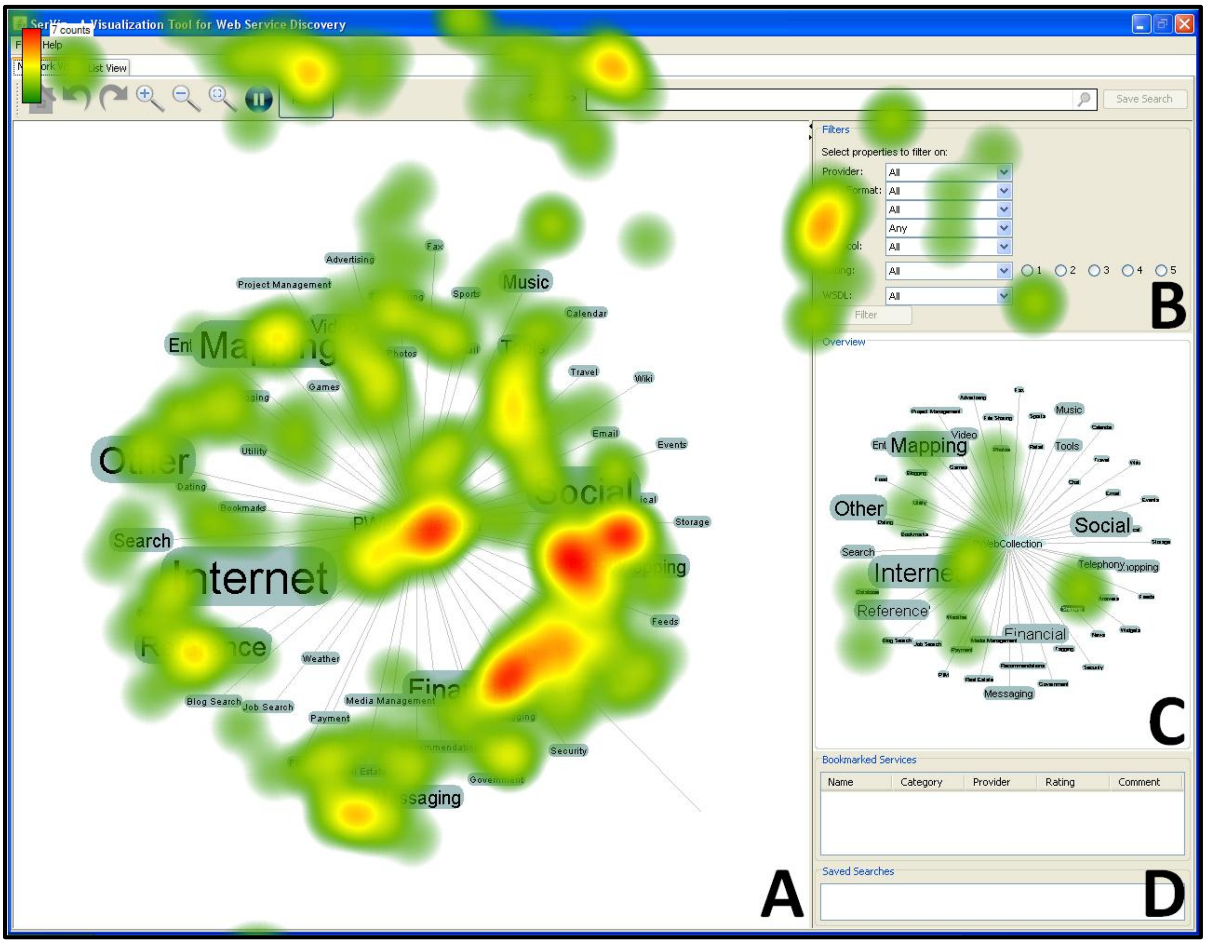


Figure 5-7 A Heat Map of Participants Browsing the Web Service Collection in the Network View (n=5)

The other task involved searching and filtering to find web services. The participants were required to write down the number of web services that satisfied the search and filter criteria. The categories that contained these web services were positioned in the top half of the graph. The heat map for this task can be seen in Figure 5-8.

Less emphasis was placed on the graph. The participants immediately used the search facility to find the web services (see Section B of Figure 5-8). The participants then looked through the list of filters to find the filter required by the task. The participants then used the graph to find the categories that contained the required web services.

It can be observed that the participants did not notice that the overview could be used to assist in navigating through the graph. Some participants clicked on the overview, but the overview requires dragging to move through the graph and does not move to a selected point.

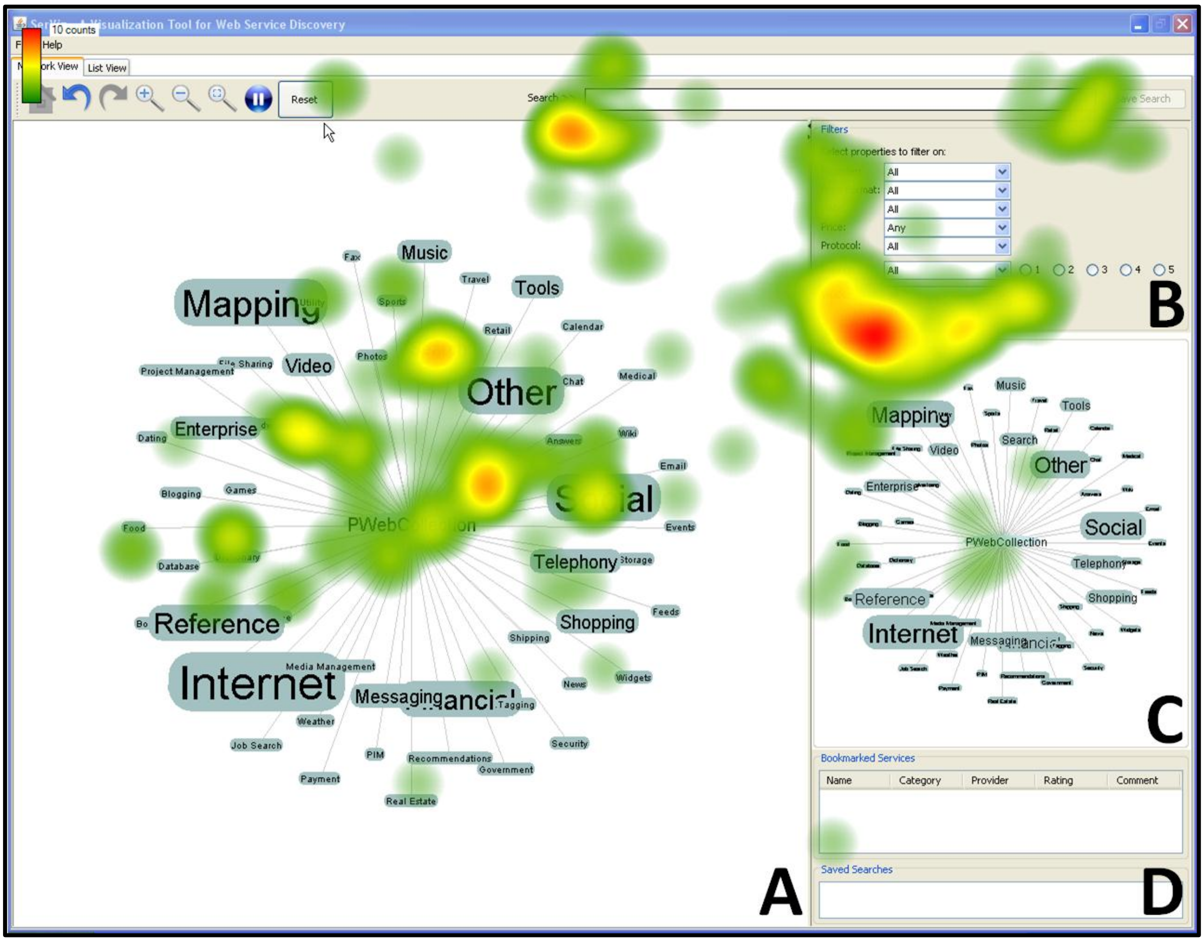


Figure 5-8 Heat Map of Participants using the Search and Filter Facilities in the Network View (n=5)

Figure 5-9 illustrates the heat map of the list view for the first task involving browsing for web services without using the search or filter facilities. The participants scanned the list to find information relating to the task. The task required the participants to find web services contained in a specific category. The participants generally first gazed at the web service names and then gazed at the category column in the list. The participants then sorted the list by category and completed the task. Emphasis was thus placed on the list.

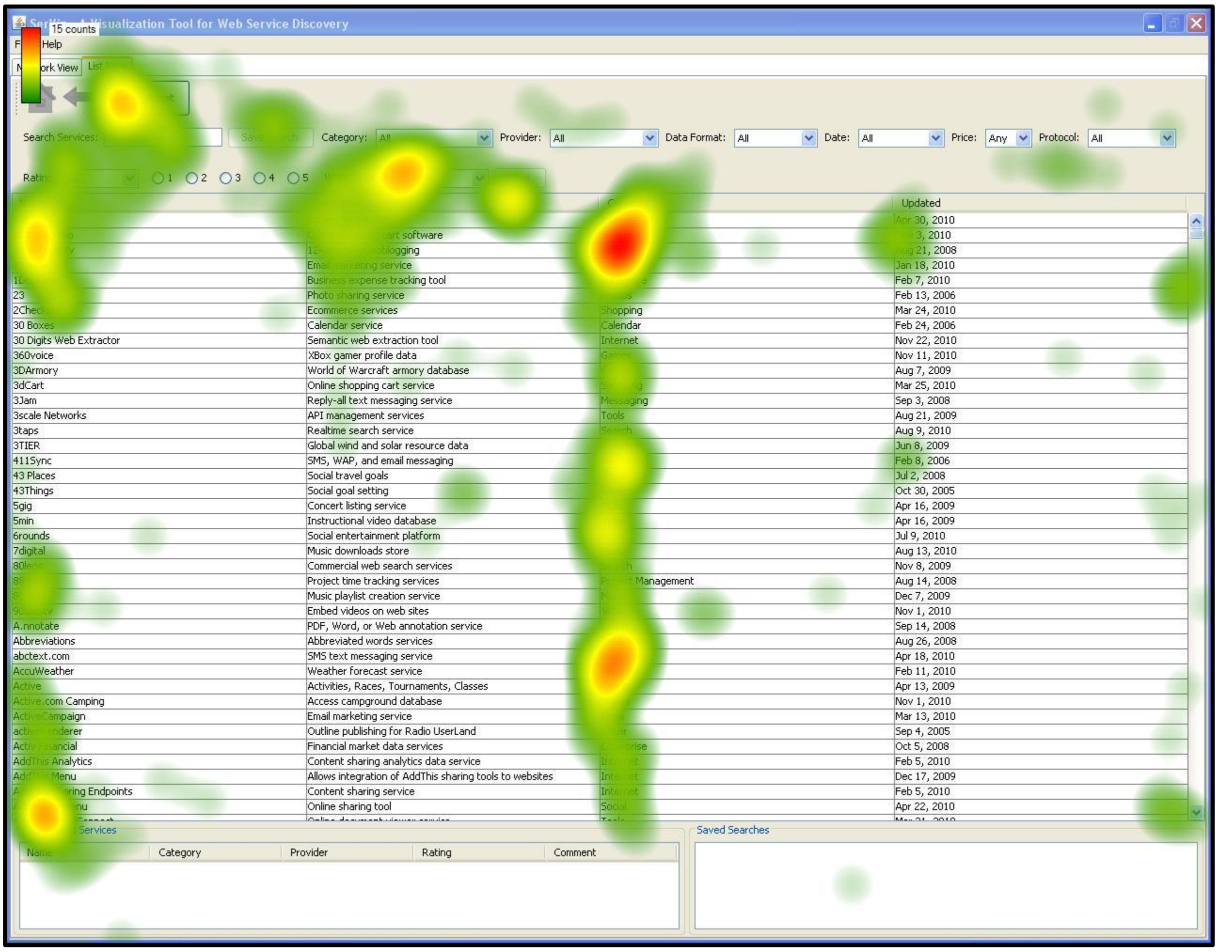


Figure 5-9 A Heat Map of Participants Browsing the Web Service Collection in the List View (n=5)

For Task 4 the participants were required to search and filter the web service collection to find web services that meet the specified criteria. Figure 5-10 shows that the participants focused mainly on the top section of the screen where the search and filter facilities were positioned and because the list was then filtered, with only a few results in the list, they only gazed at the top of the list. In contrast to Figure 5-9, emphasis was placed on the search and filter facilities.

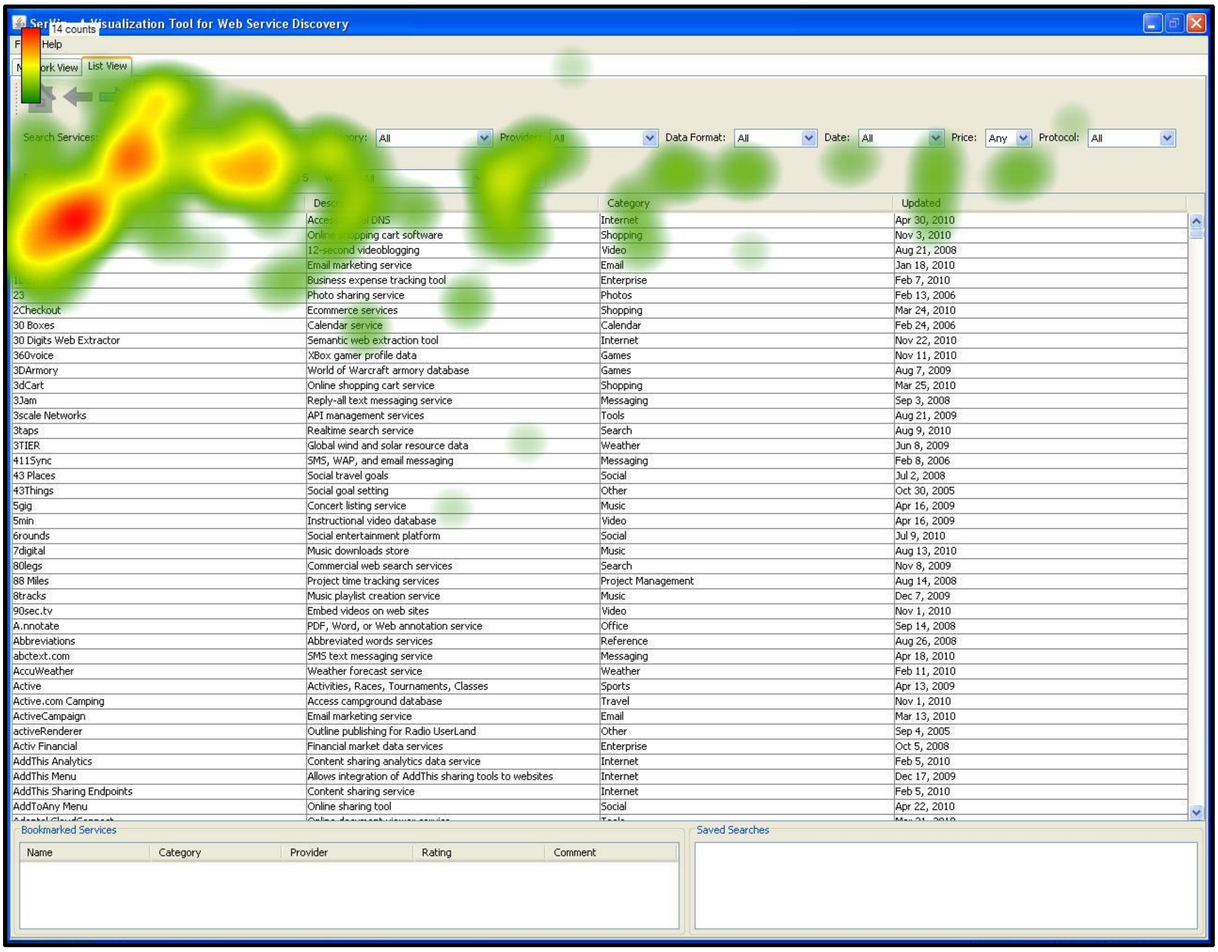


Figure 5-10 Heat Map of Participants using the Search and Filter Facilities in the List View (n=5)

5.2.3. Discussion

From the results it can be concluded that the network view was the preferred method to visualize web service collections. The principle that the data structure of the web service collection would determine which IV technique to use was supported by the participants' comments in the questionnaires regarding viewing the web services in their respective categories. The effectiveness results support both views. Although the task times do not conclusively support the network view, the satisfaction results were highly positive. Nine out of ten participants preferred to use the network view to visualize web service collections.

In addition to the questionnaires, screen recordings were used to identify any usability problems with the network view. The following usability problems were identified and will be addressed:

- The Overview does not encourage its usage in terms of graph navigation;

- The de-saturated nodes that are not part of the search or filter results clutter the screen;
- Animation continuously repositions the nodes which may lead to node occlusion or over-lapping;
- An improved way of distinguishing nodes in the web service collection is needed; and
- Expanding a category with many web services can lead to on-screen clutter.

Suggestions for improving the network view included the following:

- Provide an option to expand categories that contain web services included in search and/or filter results;
- Provide an option to hide irrelevant web services or categories excluded from search and/or filter results; and
- Provide an option to highlight which categories to include in the graph initially.

Limitations of this study include the fact that only ten participants were used for the evaluation. Additional limitations were that students were used for this study and that only one web service collection was visualized by SerViz.

5.2.4. Design Modifications

Modifications were implemented for the network view in SerViz following the results of the evaluation. The usability problems were addressed as follows:

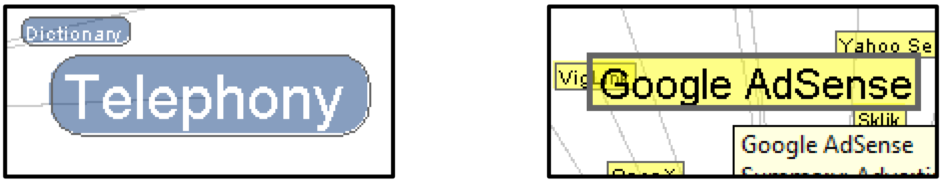
- **The Overview does not encourage its usage in terms of graph navigation.**

  Participants of the usability evaluation were not informed of the usage of the Overview in SerViz to determine if the participants look or make use of the Overview without any prior knowledge of its usefulness. The participants thus may not have been aware that the Overview was in fact interactive and could assist in graph navigation. Therefore participants should be informed in future of the availability of the Overview before an evaluation to ensure that the participants are aware of its usefulness in the network view.

- **The de-saturated nodes that are not part of the search or filter results clutter the screen.**

  An option was implemented in SerViz to provide users with the option of hiding nodes i.e. categories and web services that are not part of the search and/or filter results. If this option is checked, only the categories and web services that are part of the search and/or filter results will be visible. Un-checking the option will then display the de-saturated nodes again. Originally, to expand a category a user would just select the node. If a user accidently selected another category, that category would expand and the user would first need to contract the unintentionally expanded category and then expand the category that he wanted to expand initially. To expand a category now, a user will have to double-click on a category as a single click only selects the node implemented similarly to Song, Cai and Fu (2011).

- **Animation continuously repositions the nodes which may lead to node occlusion or over-lapping.**
  An animation timer was implemented into SerViz which calculates whether repositioning of nodes is required based on an overlapping threshold. This threshold was set to 20%. This means that the animation will continue to reposition the graph nodes until the overlapping percentage is less than 20%. The animation, or repositioning of nodes controlled by the layout, will stop and nodes will remain in the same position until another action is made on the graph, such as expanding a category. This is due to the fact that repositioning will again be required to position the new group of visible nodes and a new overlapping threshold will be calculated. Additionally, a certain level of transparency was added to the nodes in the network view so that users can easily view hidden or occluded nodes.
- **An improved way of distinguishing nodes in the web service collection is needed.**
  The colour-coding of the different nodes were changed to help distinguish between a collection node, the category nodes and web service nodes. Figure 5-11 illustrates the updated colour-coding in the network view.

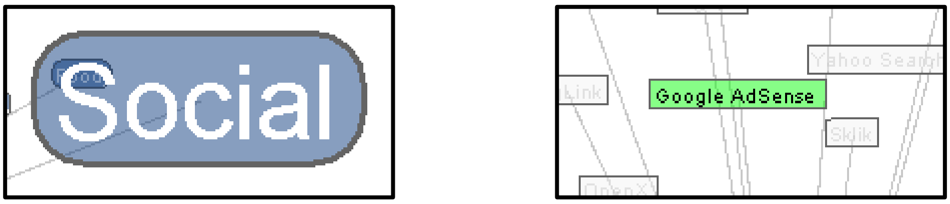


(a) **(b)**

Figure 5-11 The Updated Colour-Coding in the Network View (a) a Category and (b) a Web Service

A collection node is still represented in black with white text. A category node's colour was changed to a more intense blue with white text. The text of a category changes to black if the category is selected (by a single left click). A web service is represented by yellow to enhance the blue-yellow contrast identified by Heer and Boyd (2005).

When a user searches or filters to find web services, a web service part of the search and/or filter results will be coloured in green and its category node will remain blue as shown in Figure 5-12. Irrelevant categories and web services will still be de-saturated. When a user hovers over a node a dark-grey border appears around the node so that the user is still able to distinguish between different nodes. Additionally, a selected web service node's size is enlarged to distinguish the selected web service and other web services.



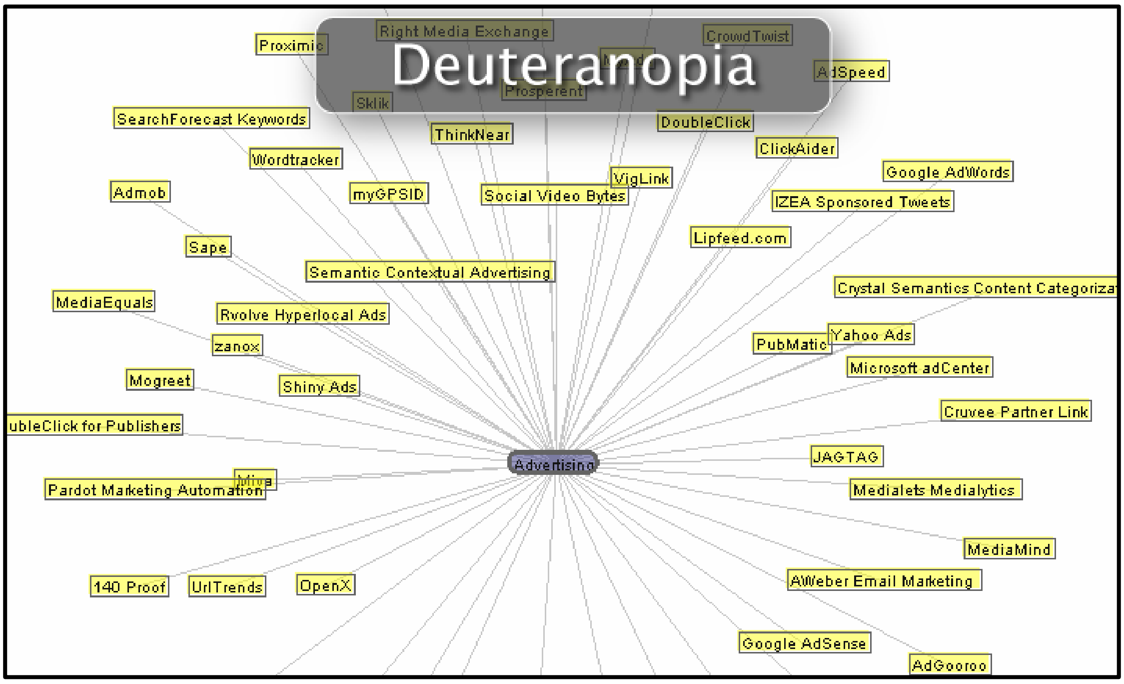
(a) **(b)**

Figure 5-12 Hovering and Searching with the Network View (a) Category Hover and (b) a Search Web Service

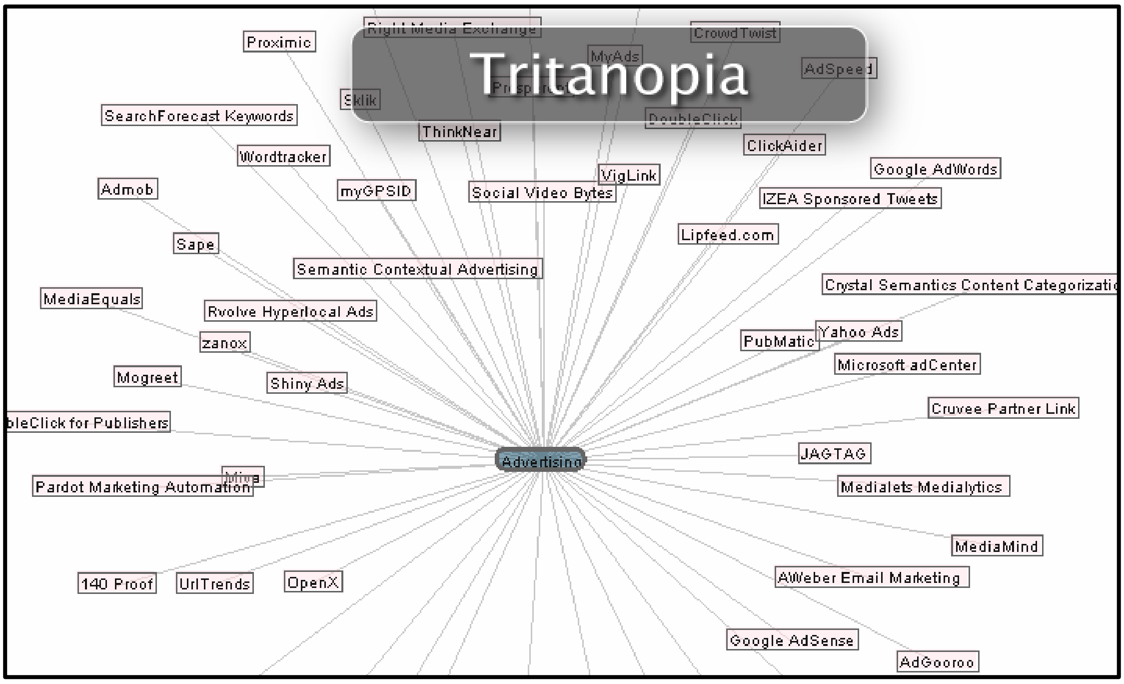
The updated colour-coding was tested with Color Oracle (Bernhard and Kelso 2007) to determine whether the colour-coding selected supported colour-blind users shown in Figure 5-13. The Deuteranopia and Protanopia results were identical similar to Section 4.3.3.2. It was determined that the updated colour-coding provided sufficient support for colour-blind users.

- **Expanding a category with many web services can lead to on-screen clutter.**

As the sample web service collection is large, the on-screen clutter when expanding is unavoidable unless web services are grouped according to name, although this will require an additional level in the hierarchy. An option was implemented which allows categories that have web services that are part of the search and/or filter results to be expanded automatically if the number of categories are less than five. This option, together with the hide option for irrelevant web services and categories, may reduce on-screen clutter.



(a)



(b)

Figure 5-13 How Colour Blind Users View the Network View Colour-Coding (a) Deutranopia Colour Blindness and (b) Tritanopia Colour Blindness

5.3. Conclusion

A usability evaluation was conducted with ten participants to compare the alphabetical list and network IV techniques. Tasks were included to evaluate each of the IV tasks discussed in

Chapter 4. The evaluation measured performance metrics, i.e. effectiveness, efficiency and user satisfaction for each technique. Eye-tracking was also used to complement these metrics.

The key result of the evaluation was that nine out of ten participants preferred the network IV technique. The list technique provided limited advantages over the network IV technique. The network IV technique was faster to browse web service collections for a specific category and to save a search. The overall questionnaire results supported the network IV technique.

The usability evaluation revealed that the network IV technique has several limitations. Certain usability problems were identified from the participants' comments and analysis of the screen recordings of each evaluation. The use of the Overview was not obvious, the large dataset led to on-screen clutter and the continuous animation led to node occlusion. It was also difficult for the participants to distinguish between different types of nodes (categories versus web services).

Modifications were implemented based on the usability problems identified in the network IV technique. An animation timer and the inclusion of node transparency was used to control node occlusion and an adjustment was made to the node colour-coding to assist users in distinguishing between nodes in the web service collection. Lastly, an option was provided to allow automatic category expansion when searching and/or filtering.

Chapter 6 will discuss an alternative IV technique selected to overcome the usability problems identified with the network IV technique. This alternative IV technique will then be compared with an improved version of the network IV technique to determine which IV technique is best suited to support visual web service discovery.

Chapter 6: An Alternative IV Technique

6.1. Introduction

The network information visualization (IV) technique was identified as the preferred technique to visualize web service collections in Chapter 5. The usability evaluation discussed in the previous chapter assisted in identifying several usability problems with the network IV technique. Due to these usability problems, an alternative IV technique was implemented in SerViz and compared with the network IV technique to determine which IV technique is best suited to visualize web service collections.

This chapter discusses the alternative IV technique implemented in SerViz. The design and implementation of this IV technique is discussed. A usability evaluation was conducted to compare the two IV techniques. The experimental design and results are detailed and discussed. The chapter concludes by outlining several design recommendations for IV techniques used for web service discovery.

6.2. Selection of an IV Technique

Two methods were used to visualize the ProgrammableWeb web service collection in SerViz, namely an alphabetical list-based method and network IV technique. The result of an evaluation used to compare these techniques was that the network IV technique was the preferred technique to visualize the ProgrammableWeb web service collection.

Certain usability problems were identified from the questionnaires completed by the participants together with analysis of the video recordings (Section 5.2.3.). These usability problems motivate the need to implement an alternative IV technique to the current network IV technique. These problems can also be used to identify requirements to determine the most suitable alternative IV technique to use, as this IV technique should overcome or avoid these problems.

The following sections identify requirements of the alternative IV technique and discuss the available hierarchical and network IV techniques. A discussion will be used to identify the most suitable IV technique to be implemented and compared with the current network IV technique in SerViz.

6.2.1. Requirements of IV Technique

A usability evaluation was conducted to compare the list and network IV techniques implemented in SerViz in Chapter 5. The following overall usability issues were identified for the network technique:

- Overlapping of nodes;
- Animation – the continuous repositioning of nodes; and
- Distinguishing between different types of nodes.

The alternative IV technique to be implemented in SerViz should overcome the above-mentioned usability issues. A structured IV technique that does not have an animation effect is a requirement for selection. This structure will additionally avoid the node overlap issues. The structure, colour and shape encodings should also facilitate distinguishing between nodes. If the collection is structured to display the different levels in the collection, it will be easier for a user to distinguish between the different types of nodes. Additionally, the IV technique should also support a large dataset and browsing and searching for web services. The selected IV technique must be supported by Prefuse. Prefuse techniques will be investigated so that the alternative IV technique can be incorporated in SerViz.

6.2.2. Hierarchical IV Techniques

The following IV techniques that support hierarchical or tree data in Prefuse are described in this section including the radial graph, tree map, node-and-link or tree view, balloon tree and adjacency diagram.

Radial Graph

The Radial Graph is an example of the node-and-link hierarchical IV technique and is supported by the Prefuse visualization toolkit (Figure 6-1).

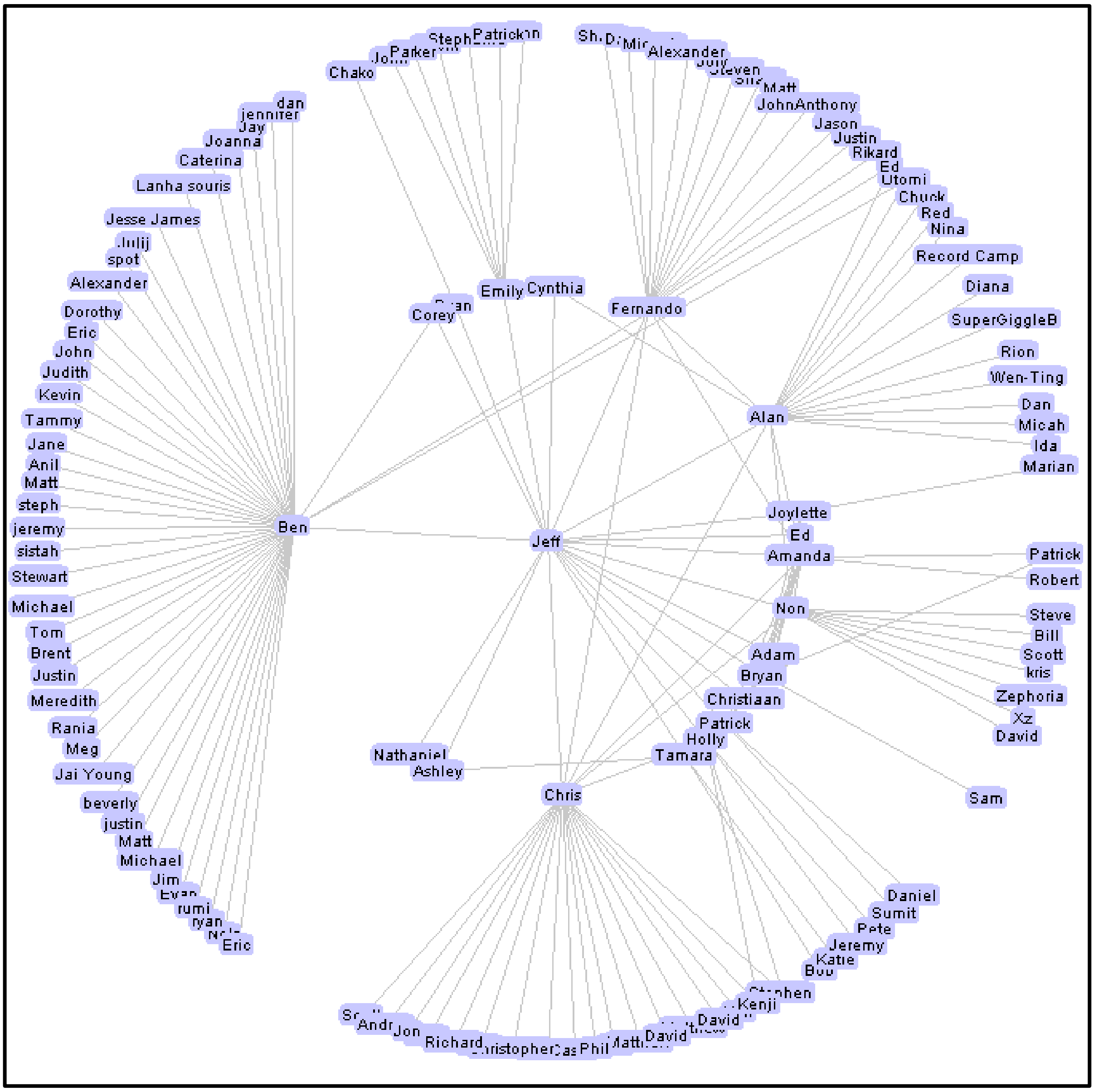


Figure 6-1 The Radial Graph in Prefuse

The Radial Graph, also referred to as a dendrogram or cluster algorithm by Heer, Bostock and Ogievetsky (2010), positions leaf nodes, in this case the web service nodes, at the diameter of the tree and thus on the same level. As the ProgrammableWeb web service collection array contains over 3300 web service nodes, this layout may not be suitable for large datasets as all the web service nodes may enlarge the diameter and thus the tree may result in a large screen space requirement and occlusion.

Tree Map

An example of an implementation of a tree map in Prefuse is illustrated in Figure 6-2. From Figure 6-2, it can be concluded that tree maps are used to provide a general overall view of the visualized information. It may be difficult to browse the tree map to find web services as each web service will represent an equal-sized square contained within a category. A user would then be required to hover over each square in a category to identify the web services in each category. While web services are contained in categories in a collection, the structure does not necessarily need to be visualized as nested subsets. With the current web service collection, there is only one level to be visualized by the tree map, namely the categories in the collection. Only the web services are visualized in each category and so one category of 100 web services will comprise of 100 squares that make up the category's space.

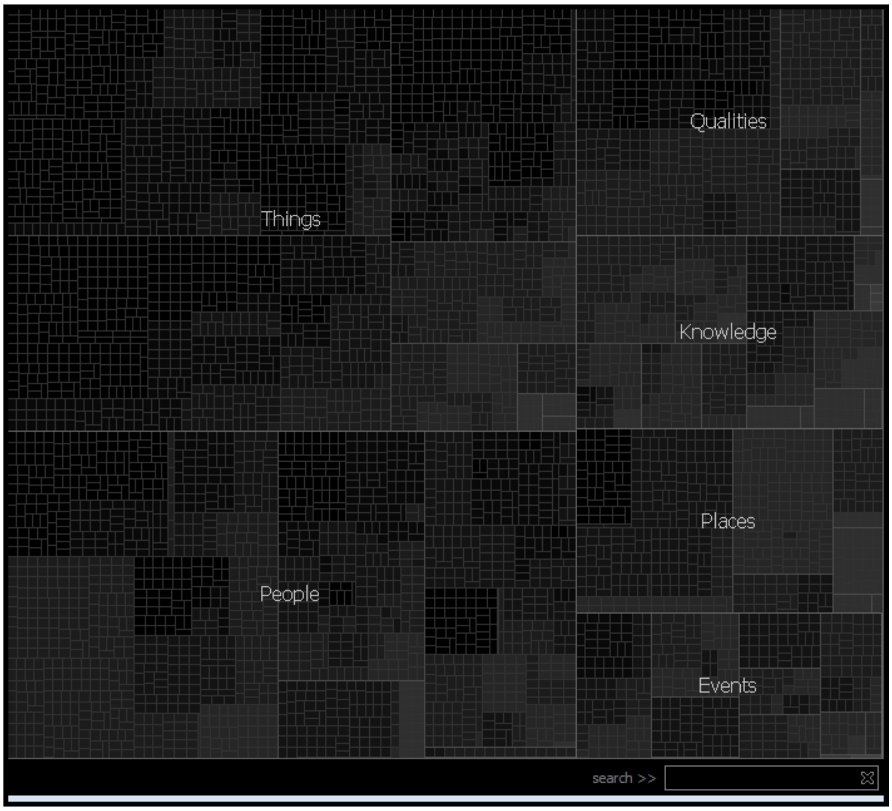


Figure 6-2 The Tree Map in Prefuse (Heer 2005)

Node-and-Link / Tree View

A Prefuse implementation of the node-and-link tree IV technique, or tree view as referred to in Prefuse, is shown in Figure 6-3. The Prefuse implementation of the node-and-link IV technique resembles the application of Windows Explorer shown in Figure 3-14 in Chapter 3. The difference between these techniques is that the Prefuse version includes an additional sub-layout that maximizes areas of interest in the tree and minimizes the remainder of the tree. Computer directory information can resemble web service collection information as a computer contains files (similar to the web service categories) and the files have information that describes them (the web service information). This IV technique can implement alphabetical sorting not supported by the currently implemented network IV technique. A disadvantage of this IV technique is that it would have a greater requirement for vertical screen space (Heer *et al.* 2010).

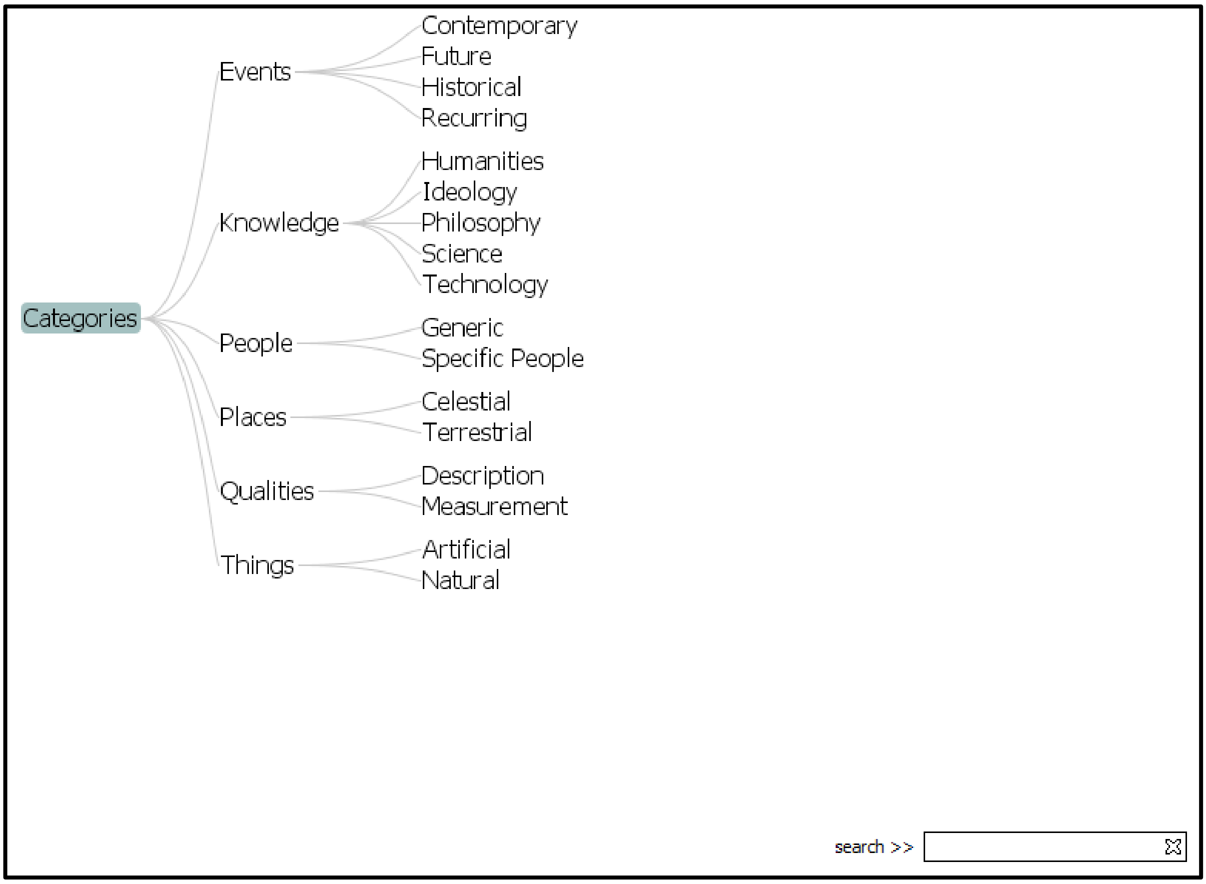


Figure 6-3 The Node-and-link or Tree View in Prefuse (Heer 2005)

Balloon Tree

This IV technique is similar to the node-and-link IV technique except the layout attempts to make use of the entire screen space by "ballooning" out the nodes and is less structured than the node-and-link IV technique (Figure 6-4). Node positioning in a space is an important aspect of trees as a tree would be unreadable otherwise (Card 2008, 527). Additionally, a usability problem identified with the network IV technique was that participants of the evaluation found it difficult to distinguish between the different types of nodes. These two issues are not sufficiently supported (or overcome) with this technique especially considering the size of the ProgrammableWeb web service collection (n=3562) used for SerViz.

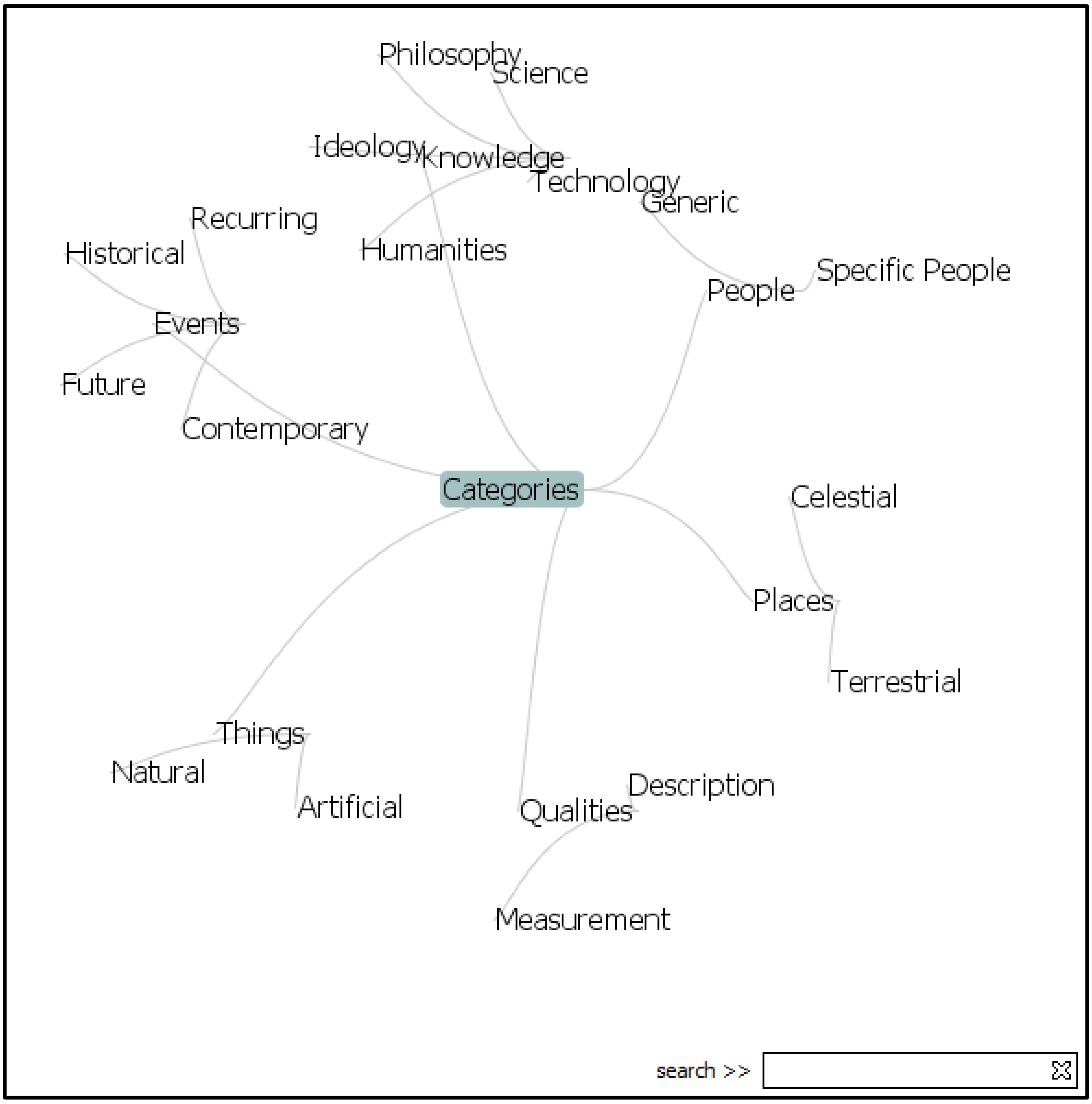


Figure 6-4 The Balloon Tree in Prefuse

Adjacency Diagram

The adjacency tree is a derivation of the node-and-link tree IV technique (Heer *et al.* 2010). The adjacency tree uses a "space-filling" method to visualize nodes as "solid areas". The position of the node represents the position in the hierarchy. An example of this technique includes DocuBurst used to visualize document content (Collins, Carpendale and Penn 2009) (Figure 6-5).

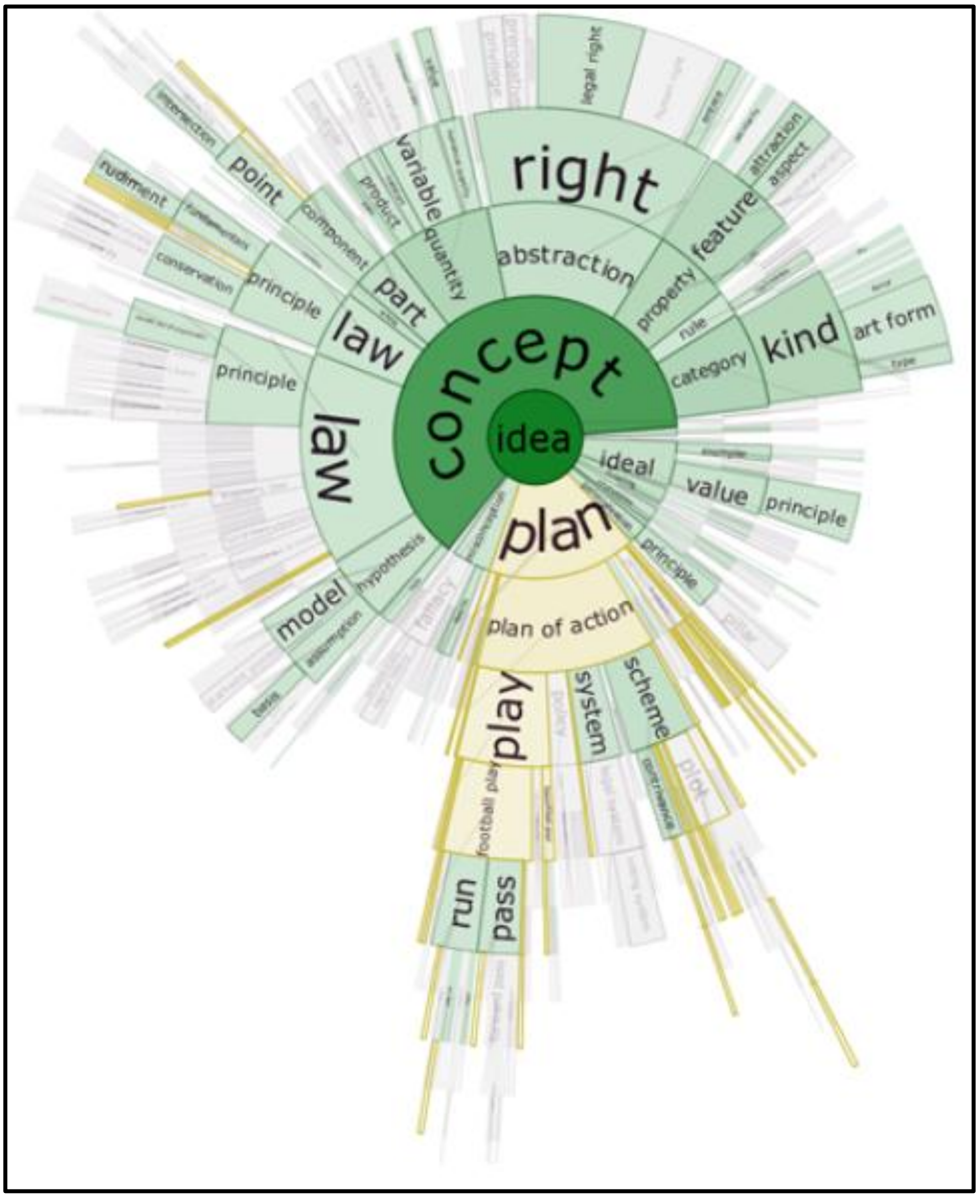


Figure 6-5 Starburst Technique (Collins *et al.* 2009)

As discussed in the Tree Map section, the web service collection information is not suited to a space-filling layout.

6.2.3. Network IV Techniques

The remaining IV technique in Prefuse that support network data is the Fruchterman Reingold network IV technique discussed below. The square matrix and arc diagram were excluded from consideration as these techniques are not supported by Prefuse or have already been discussed in Chapter 3.

Fruchterman Reingold

The Fruchterman Reingold network IV technique is another implementation of the node-and-link network IV technique using the force directed layout (FDL) (Figure 6-6). As this technique is similar to the current implementation of the network IV technique, this technique will not be considered as an alternative IV technique to overcome the usability problems identified from the implementation of the current network IV technique.

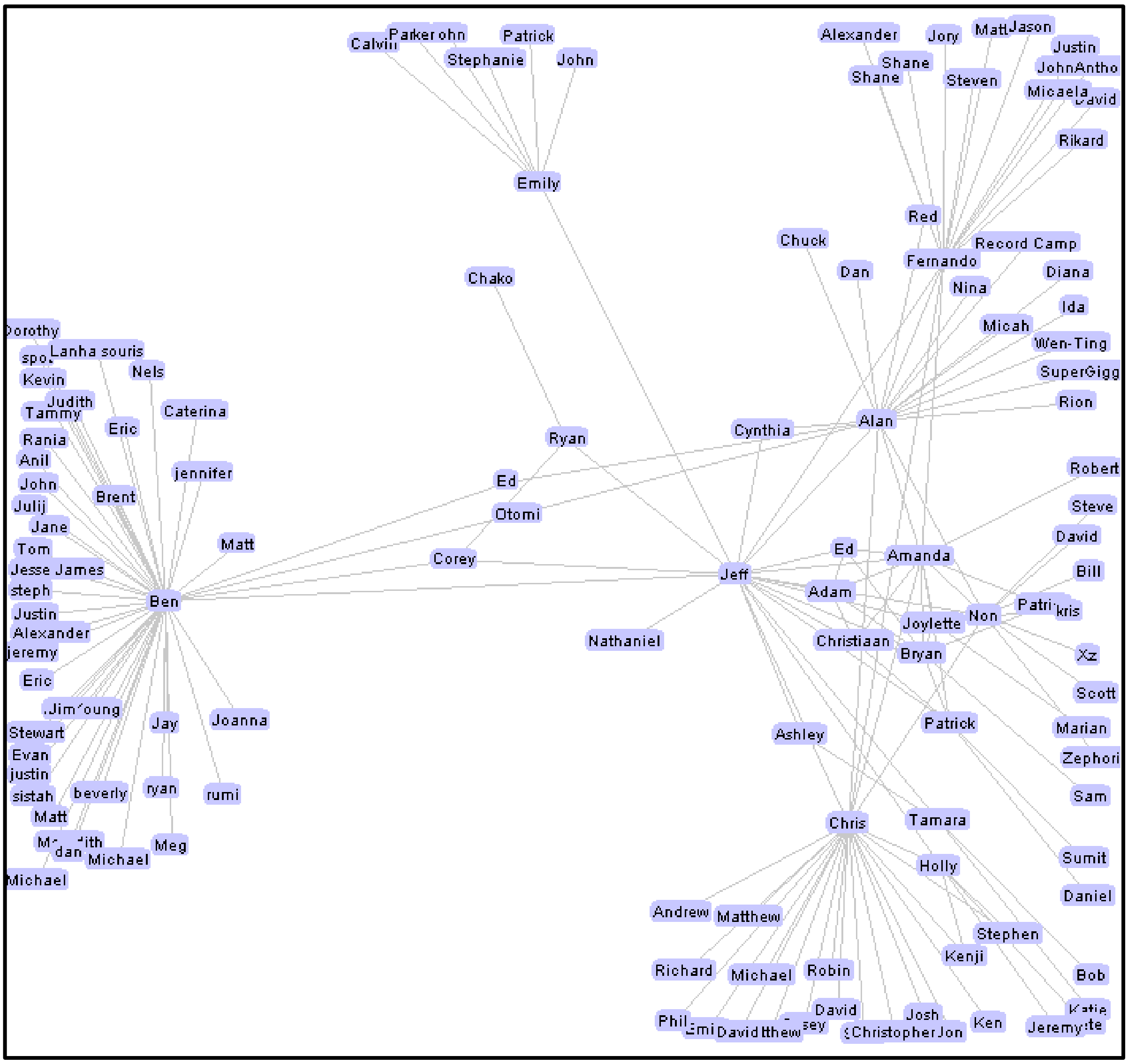


Figure 6-6 The Fruchterman Reingold Network in Prefuse

6.2.4. Discussion

The currently available network IV technique does not support the requirements identified in Section 6.2.1. and therefore network IV techniques will be excluded from consideration. The limitation of selecting a hierarchical IV technique as the alternative technique is that a tree can provide a link to only a single parent. The ProgrammableWeb web service collection has a hierarchical structure and can therefore be visualized by a hierarchical IV technique. This leads to a possible trade-off: a technique which allows for different types of web service collections with the identified usability problems or a technique that overcomes these problems but can only visualize hierarchical web service collections.

The hierarchical IV techniques that can be chosen as the alternative technique to be implemented into SerViz include the Radial Graph, Tree Map, Node-and-Link, Balloon Tree and Adjacency Diagram techniques. The Prefuse visualization toolkit supports these techniques. All these techniques can overcome the usability problems identified, but overlapping may become an issue with some of these techniques with the large web service collection that needs to be visualized.

The limitation of the Radial Graph is that it may not be suitable for the large dataset and navigating through the dataset with this technique may be difficult. Tree Maps provide a general overall view of the dataset and so it may not sufficiently support browsing the web service collection. The Tree Map IV technique may also not effectively represent the structure of the web service collection. As with the Tree Map technique, an adjacency diagram may not be the appropriate method to visualize web service collections as the information may not require “space-filling”.

The node-and-link IV technique is the most common IV technique to visualize hierarchical data. This technique resembles the Windows Explorer metaphor commonly used in the Windows operating system which could possibly be familiar to a user. Computer directories are somewhat similar to web service information. This technique uses a levelled structure which could help to distinguish between different types of nodes and does not have an animation effect except for the expansion of nodes. Alphabetical sorting is supported by this technique which could assist a user in finding a specific web service or category. The balloon tree IV technique is a derivation of the node-and-link technique but can use more screen space by “ballooning” out the nodes, but lacks the structured advantage provided by the node-and-link IV technique.

From this discussion, it can be concluded that the node-and-link IV technique to visualize hierarchical data should be implemented as an alternative technique in SerViz. This technique will be compared with an improved version of the currently implemented network IV technique to determine which IV technique provides the most support for web service discovery.

6.3. Design of Tree View

The node-and-link IV technique to visualize hierarchical data was selected as the most suitable alternative IV technique to visualize web service collections. This technique was implemented as an additional view in SerViz. The web service collection retrieved using the ProgrammableWeb API (Application Programming Interface) was converted to support the tree visualization in Prefuse.

6.3.1. Tree View in SerViz

The node-and-link IV technique, or tree IV technique, is referred to as the Tree View in SerViz. The list view was disabled in SerViz and the tree view was added as an additional tab shown in Figure 6-7.

The tree view was designed and implemented to support the functions listed in Section 4.2.2. By implementing this alternative IV technique in SerViz it is possible to compare the two IV techniques to determine which IV technique is best suited to visualize web service collections.

The tree view displays the collection node to the left connecting the category nodes listed below each other and to the right of the collection node (See Section A in Figure 6-7). Similarly to the network view, the web services are also initially hidden.

Instead of providing the Overview in this view, the tree view provides a fisheye strategy (Shneiderman 1996; Song *et al.* 2007). Variations of the fisheye strategy exist (Song *et al.* 2011), but a fisheye distortion technique that magnifies nodes of the tree as the cursor hovers over the nodes was selected for the tree view. A fisheye strategy provides an alternative method to support the overview task and may be a more suitable method to apply to the tree view due its hierarchical structure and the size of the sample web service collection used. Due to the implementation of the fisheye strategy, the category nodes are not sized according to the number of web services provided in each category, as in the network view. The fisheye strategy was implemented in the tree view similar to the implementation of Song *et al.* (2007).

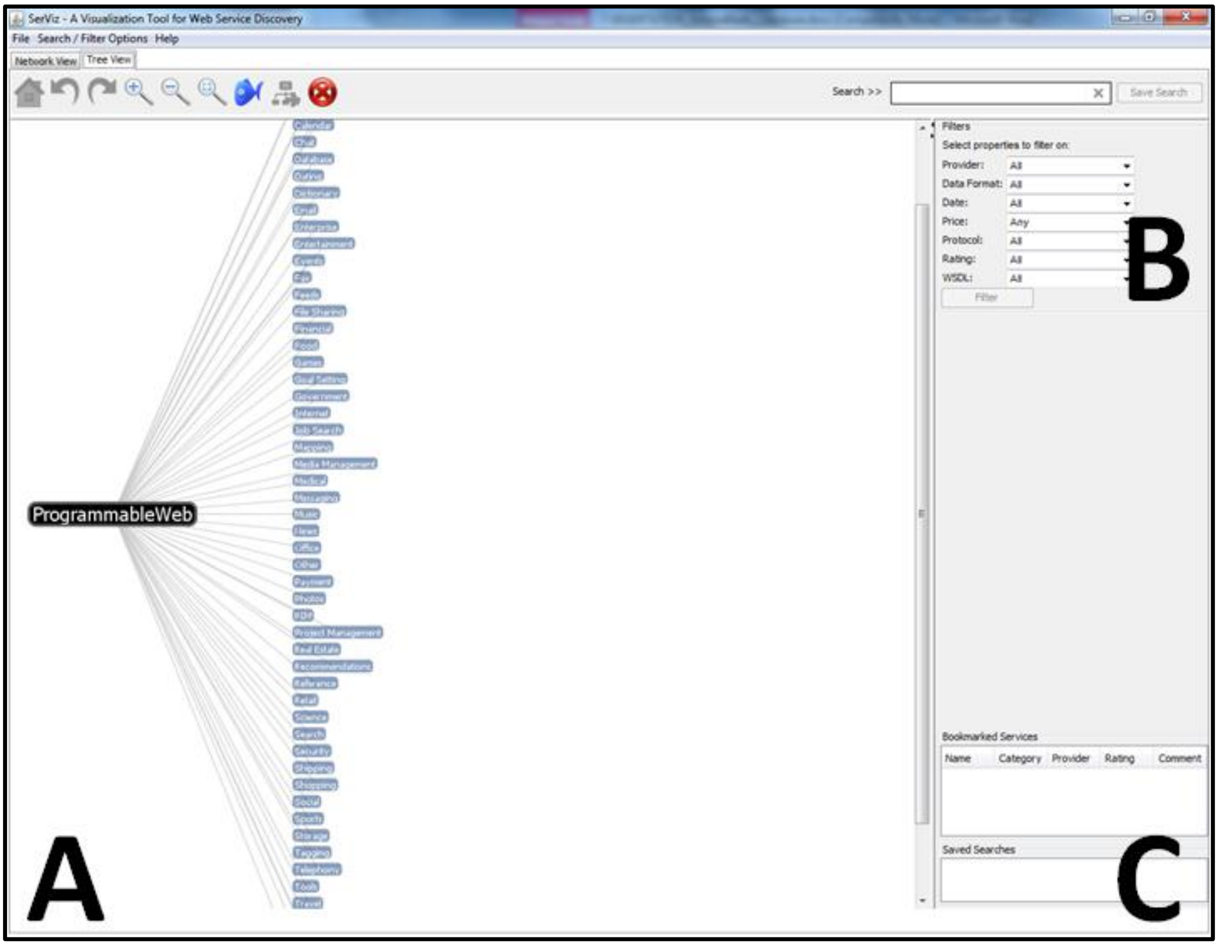


Figure 6-7 The Tree View in SerViz

The nodes do not require continuous re-positioning as in the network view and thus there is no animation on the tree view except for the fisheye strategy. The colour coding is identical to the modified colour coding in the network view for consistency (Section 5.2.4.).

An example of searching and browsing to find web services is shown in Figure 6-8. To find the *Google AdSense* web service using the tree view, a user can search for *Google* web services using the search bar at the top of the screen and expand the *Advertising* category. Selecting or hovering over a node gives it a grey border. The user can then double-click to view details of the web service or, as in Figure 6-8, hover over the web service to view web service information. The tree view also provides a filtering facility, bookmarking a web service and saving a search (Sections B and C in Figure 6-8).

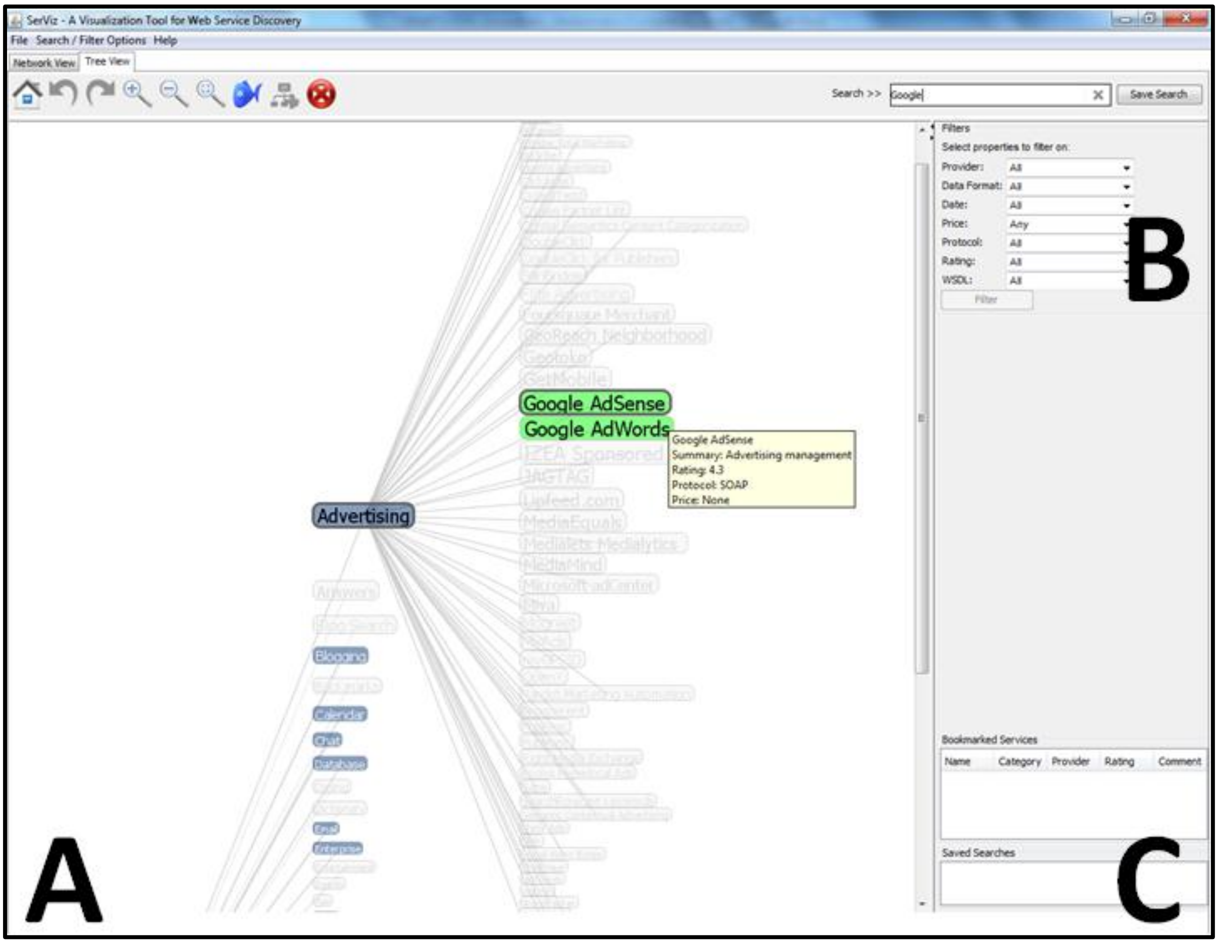


Figure 6-8 Searching and Browsing using the Tree View

Similar to the network view, the tree view provides a *Home* button, allows for undoing and redoing actions and zooming and panning (Figure 6-9). A user can also toggle between having the fisheye on or off. An additional button is provided in both views to allow for expanding categories that contain web services as part of search and/or filter results. The *Reset* button is provided for evaluation purposes to reset the tree, the search and the filters. A scroll bar is provided to scroll through the categories as the number of categories is too large to display on the screen with the default zoom level (Section A in Figure 6-7).

Figure 6-9 Controls in the Tree View

6.3.2. Data Conversion

The ProgrammableWeb web service collection was updated as the number of nodes in the collection increased from 2390 to 3562 since initial implementation. Three additional categories were included. A similar retrieval and conversion process was followed as described in Section 4.2.1.2. and illustrated in Figure 4-1.

The resultant XML (Extensible Mark-up Language) document was converted to TML (Tree Mark-up Language) as Prefuse provides a TML reader to read in the data in the document and convert the data to branch and leaf nodes for the hierarchical structure. Similarly to GML (Graph Mark-up Language), TML is structured according to nodes and edges. Similar steps were involved to successfully convert the XML document containing the web service collection to TML. A category node was defined as a branch and the web services contained in the category were nested under the category branch as leaf nodes. A snippet of the resultant TML document is shown in Figure 6-10.

```
<?xml version="1.0"?>
- <tree>
  - <declarations>
      <attributeDecl name="name" type="String"/>
      <attributeDecl name="WSCategory" type="String"/>
      <attributeDecl name="type" type="String"/>
      <attributeDecl name="link" type="String"/>
      <attributeDecl name="summary" type="String"/>
      <attributeDecl name="rating" type="String"/>
      <attributeDecl name="description" type="String"/>
      <attributeDecl name="protocol" type="String"/>
      <attributeDecl name="wsdl" type="String"/>
      <attributeDecl name="provider" type="String"/>
      <attributeDecl name="price" type="String"/>
      <attributeDecl name="date" type="String"/>
      <attributeDecl name="dataformats" type="String"/>
    </declarations>
  - <branch>
      <attribute name="name" value="ProgrammableWeb"/>
      <attribute name="type" value="COLLECTION"/>
    - <branch>
        <attribute name="name" value="Advertising"/>
        <attribute name="type" value="CATEGORY"/>

        ...
      - <leaf>
          <attribute name="name" value="140 Proof"/>
          <attribute name="WSCategory" value="Advertising"/>
          <attribute name="type" value="WEBSERVICE"/>
          <attribute name="link" value="http://www.programmableweb.com/api/140-proof"
          <attribute name="summary" value="Twitter advertising service"/>
          <attribute name="rating" value="5.0"/>
          <attribute name="description" value="The 140 Proof API gives users programmatic
            character brand messages postioned directly above the Twitter stream to targe
            data. The API uses RESTful calls and responses can be formatted in either XML
          <attribute name="protocol" value="REST"/>
          <attribute name="wsdl" value=""/>
          <attribute name="provider" value="http://www.140proof.com"/>
          <attribute name="price" value=""/>
          <attribute name="date" value="2011-01-12T03:40:26Z"/>
          <attribute name="dataformats" value="XML, JSON, HTML"/>
        </leaf>

        ...
    </branch>
</tree>
```

Figure 6-10 The Final TML Document

6.4. Evaluation Design

A usability evaluation was conducted to compare the two IV techniques implemented in SerViz. The evaluation was conducted similarly to the study described in Chapter 5. The evaluation included the six tasks used in the previous experiment (see Table 5-1). The evaluation also used a within subjects design. Identical usability metrics were captured including effectiveness, efficiency and user satisfaction. Eye-tracking was also used in this evaluation. Twenty participants completed this evaluation using SerViz.

6.4.1. Evaluation Objectives

The network and tree views were implemented in SerViz to visualize the ProgrammableWeb web service collection. The objective of this study was to compare the network and tree views to identify which IV technique is the more suitable IV technique to visualize web service collections.

6.4.2. Participants

None of the participants of the first usability evaluation were used for this evaluation. A larger sample was used for this evaluation consisting of a representative sample of twenty (sixteen male, four female) post-graduate students and staff from the NMMU (Nelson Mandela Metropolitan University) Department of Computing Sciences. Participants were in the age range of 21 – 50+ in years. Participants were required to have the same knowledge as the previous study. The evaluations took approximately an hour to complete as participants were provided with training tasks for the first four evaluation tasks (see Section 6.4.6.). All participants had at least six years' experience using a computer. Participants were asked whether they had any prior experience using IV tools and experience using web services (Figure 6-11). Seventy-five percent of participants had used an IV tool prior to taking part in the study and fifty-five percent of participants had at least two years' experience using web services.

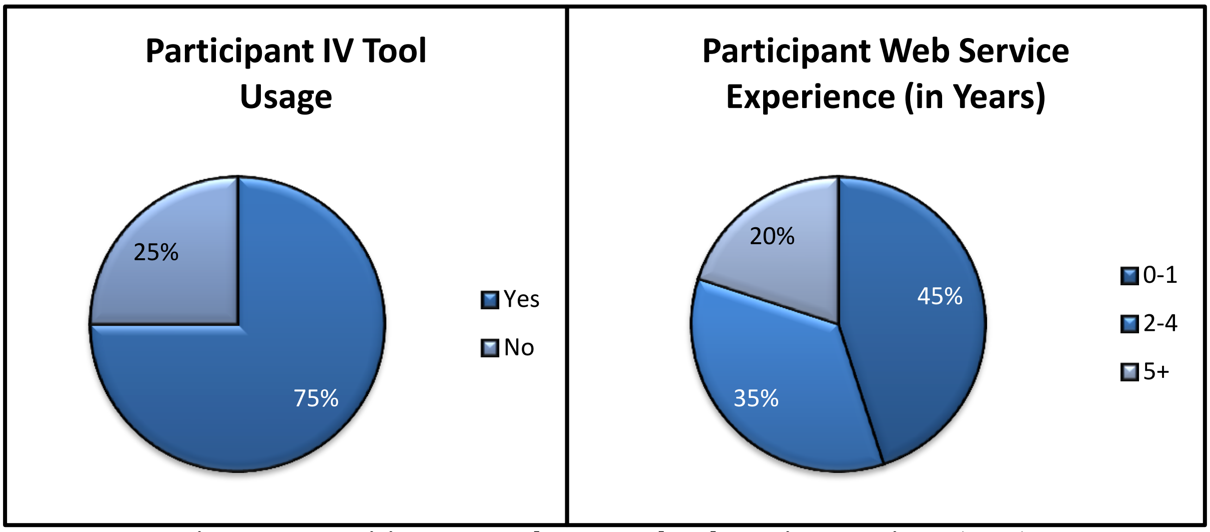


Figure 6-11 Participant IV Tool Usage and Web Service Experience (n=20)

6.4.3. Evaluation Metrics

The same metrics that were used in the previous study were used for this evaluation (see Section 5.2.1.3.). Performance metrics, i.e. effectiveness and efficiency were measured for each task and user satisfaction was also captured using similar questionnaires.

6.4.4. Questionnaires

An identical background questionnaire used in the previous study was provided to the participants at the start of the evaluation (Appendix B). The post-task and post-test questionnaires were re-used but an additional question was asked in each questionnaire (Appendix H, I and J). The question added to the post-task questionnaire related to the usefulness of the different methods of overview used for each technique, i.e. the fisheye strategy in the tree view and the overview in the network view. The additional question in the post-test questionnaire required the participants to rate which view would be more suitable to visualize a large web service collection (n=3562 nodes).

6.4.5. Experimental Setup

The procedure of the evaluation was conducted similar to the previous study. The only exception was that a different computer was connected to the eye-tracker monitor as the computer used in the previous study was slow for the network view. Participants made comments regarding this fact and so the eye-tracker monitor was connected to a computer with better specifications to avoid this.

6.4.6. Tasks

Two similar but not identical sets of tasks were provided to each participant to complete using both views (Section 5.2.1.6. and Appendix G). Due to the learning factor that could be involved when using new or unfamiliar techniques to find information, the first four tasks included a training task before the actual tasks. The training tasks were used to allow the participants to familiarize themselves with the views and so eliminate or reduce the learning factor in the evaluation results. The six tasks for each view include the following:

1. Finding information in the web service collection by browsing the collection
2. Viewing details of a web service (Browsing only)
3. Searching for web services
4. Searching and filtering to find web services
5. Bookmarking a web service
6. Saving a search for web services

6.4.7. Procedure

An identical procedure, followed in the previous study, was used for this evaluation (Section 5.2.1.7.). Each participant first provided consent to participate in the evaluation (Appendix A). The participant filled in the background questionnaire (Appendix B). The participant was then provided with a demonstration of the system and shown the two views. The eye-tracker was then calibrated for the participant. The participant completed a set of tasks using one view (Appendix G) and completed a post-task questionnaire on that view thereafter (Appendix H and I). The participant completed the second set of tasks using the second view and completed the post-task questionnaire for the second view (Appendix G, H and I). Counterbalancing was used to avoid the learnability effect by changing the order in which the views were used in the evaluation. The participant was required to conclude the evaluation by answering a post-test questionnaire comparing the two views (Appendix J).

6.5. Evaluation Results

The evaluation metrics that were measured in this evaluation include effectiveness, efficiency and user satisfaction. Eye-tracking was also used in this evaluation. The results of the evaluation will be discussed in this section.

6.5.1. Performance Results

Performance results were captured in terms of effectiveness and efficiency for each view. Both views received high levels of accuracy. The tree view received 100% success rates in Task 1, Task 3 and Task 6 while the network view received 100% success rates in Task 2, Task 3 and Task 5. The tree view was faster for browsing while the network view was faster for searching and filtering, bookmarking a web service and saving a search.

Effectiveness

As in the previous study, participants were required to provide answers to a question related to each task. The effectiveness of each task was calculated as the number of correct answers.

a) Network View

Three of the six tasks for the network view received 100% success rates (Task 2, Task 3 and Task 5). One task received a 95% success rate (Task 4) and the remaining two tasks received a 90% success rate (Task 1 and Task 6).

Task 4 involved searching and filtering to find web services. Only one participant answered incorrectly. The question required the participant to write down the names of the web services that met the search and filter criteria. The participant wrote down the categories and not the web services. This may have been due to reading the question incorrectly or not being able to distinguish between a category and a web service. Task 1 involved browsing for a category without using the search and filters to find the number of web services in the specific category. Participants could make use of the category's tooltip to easily find the number of web services within the category but some participants expanded the category and counted manually. Two of the participants that counted manually counted incorrectly leading to the 90% success rate for this task. Task 6 involved saving a search for web services. The participants were required to write down the number of categories that have web services as part of the saved search. There were three categories that contained relevant web services. One participant did not notice a smaller category in the top of the graph and the other participant did not notice a smaller category behind another de-saturated category and so both participants only found two categories.

b) Tree View

The tree view received 100% success rates in three of the six tasks (Task 1, 3 and 6). The other three tasks received 95% success rates (Task 2, 4 and 5). Task two involved browsing for a web service without using the search or filter facility and without informing the participants in which category the web service is contained. One participant answered incorrectly by providing the rating of the web service (required in the training task), instead of the upload date of the web service. The same participant who provided the incorrect answer for task four in the network view answered incorrectly for task four in the tree view. Task five involved bookmarking a web service. The participant was required to browse or search for a specific web service, bookmark the web service and provide the rating of the web service. The participant who answered incorrectly for this task bookmarked the web service, closed the details dialog of the web service and then accidently opened the web service above the specific web service and wrote down the rating of that web service without realizing it.

The participants could complete most of the tasks without difficulty. As with the previous evaluation's results, the high level of accuracy for both the network and tree view demonstrate that both views effectively supported the users' tasks.

Efficiency

The time for each task was calculated the same as in the previous study, i.e. from the start of the task until the Reset button was selected. Table 6-1 displays the task times.

Table 6-1 Task Times in Seconds with Wilcoxon Test Statistics and p-values

Task	Network View	Tree View	Z	p-value
1	29.88	**19.55**	**2.91**	**0.004**
2	52.38	**37.12**	**2.24**	**0.03**
3	**17.54**	21.59	**2.69**	**0.01**
4	**37.60**	40.79	1.87	0.06
5	**51.81**	55.07	0.60	0.55
6	**34.66**	37.50	0.56	0.58

The tree view was significantly faster for browsing (Tasks 1 and 2). The network view was faster for searching, searching and filtering, bookmarking a service and saving a search. Using the Wilcoxon matched pairs non-parametric test, three of the six differences in task times were statistically significant, shown in bold in Table 6-1. Task 1 and Task 2 were in favour of the tree view and Task 3 was in favour of the network view. The reason the tree view may be faster for browsing is that the categories are sorted in alphabetical order while the network view relies on the force directed layout to place the category nodes and is not sorted. The network view may be faster for searching due to the entire network (collection and categories) being visible on the screen, whereas the categories in the tree view cannot all fit on the screen.

6.5.2. Satisfaction Results

The satisfaction results for this evaluation were captured using the same post-task and post-test questionnaires similar to the previous evaluation (Appendix H, I and J). The satisfaction in the post-task questionnaire was divided into four sections (due to the addition of the overview question) namely cognitive load, overall satisfaction, usability and overview.

Workload Results

The results were similar for both views for cognitive load except that participants found that that the network view required significantly more effort (Figure 6-12).

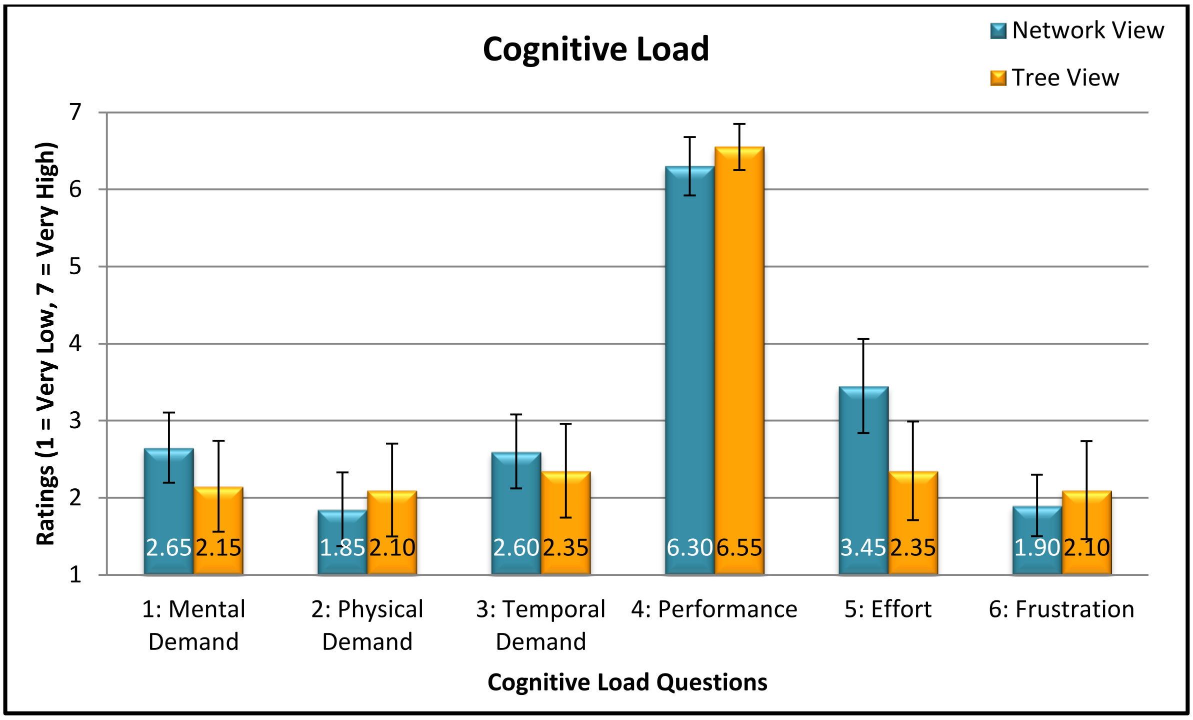


Figure 6-12 Cognitive Load using a 7-point Semantic Differential Scale (n=20)

The difference was statistically significant for the Effort item as can be seen in Table 6-2, significance shown in bold.

Table 6-2 Wilcoxon Matched Pairs Test for Cognitive Load

Question	Z	p-value
1: Mental Demand	1.69	0.09
2: Physical Demand	0.90	0.37
3: Temporal Demand	0.72	0.47
4: Performance	1.69	0.09
5: Effort	**2.45**	**0.01**
6: Frustration	0.16	0.88

Overall Satisfaction Results

The tree view received slightly higher ratings in three of the four questions for overall satisfaction as shown in Figure 6-13. There was no statistical significant difference between the two views for the overall satisfaction ratings.

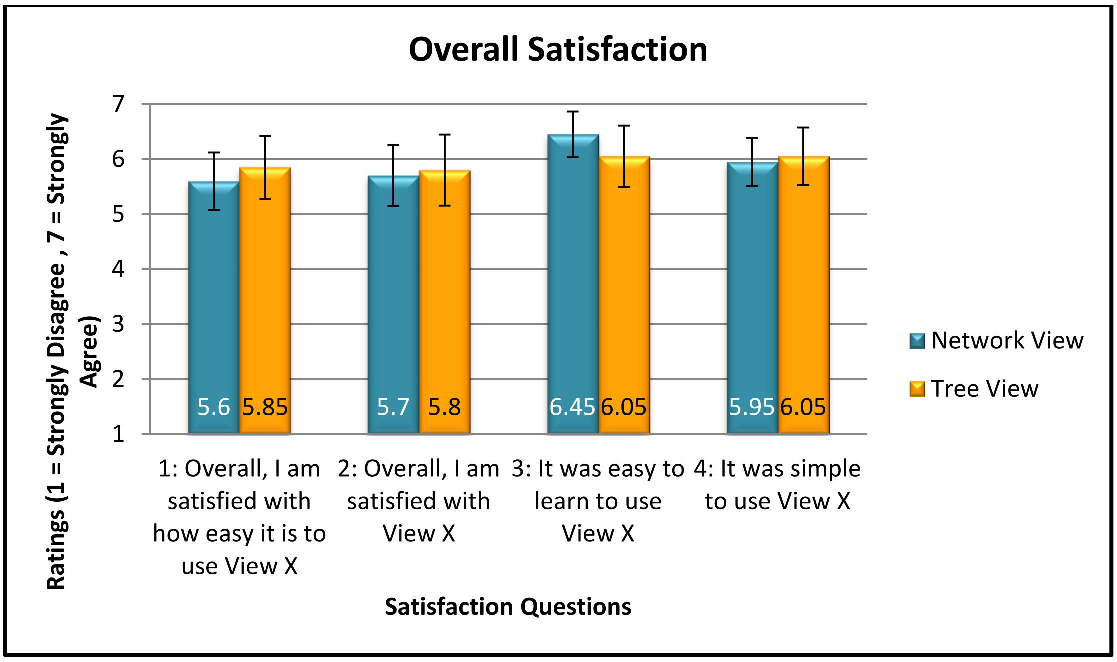


Figure 6-13 Overall Satisfaction using a 7-point Likert Scale (n=20)

Usability Results

The usability ratings were mainly in favour of the tree view except for question 5, which had very similar results (see Figure 6-14).

Table 6-3 Wilcoxon Matched Pairs Test for Usability

Question	Z	p-value
1. I can effectively find web services using View X	1.82	0.07
2. I was able to find web services quickly using View X	**2.53**	**0.01**
3. I was able to efficiently find web services using View X	1.05	0.29
4. I became productive quickly using View X	0.94	0.35
5. View X has all functions and capabilities I expect from a web service discovery tool	0.21	0.83
6. I can effectively browse web service collections using View X	0.56	0.58
7. I was able to browse web service collections quickly using View X	1.00	0.32
8. I was able to efficiently browse web service collections using View X	0.94	0.35

The difference was statistically significant in question 2 (see Table 6-3). The participants found the tree view significantly faster for finding web services which supports the efficiency result that the participants were faster in browsing the web service collection, although the network view was faster for searching (Question 2).

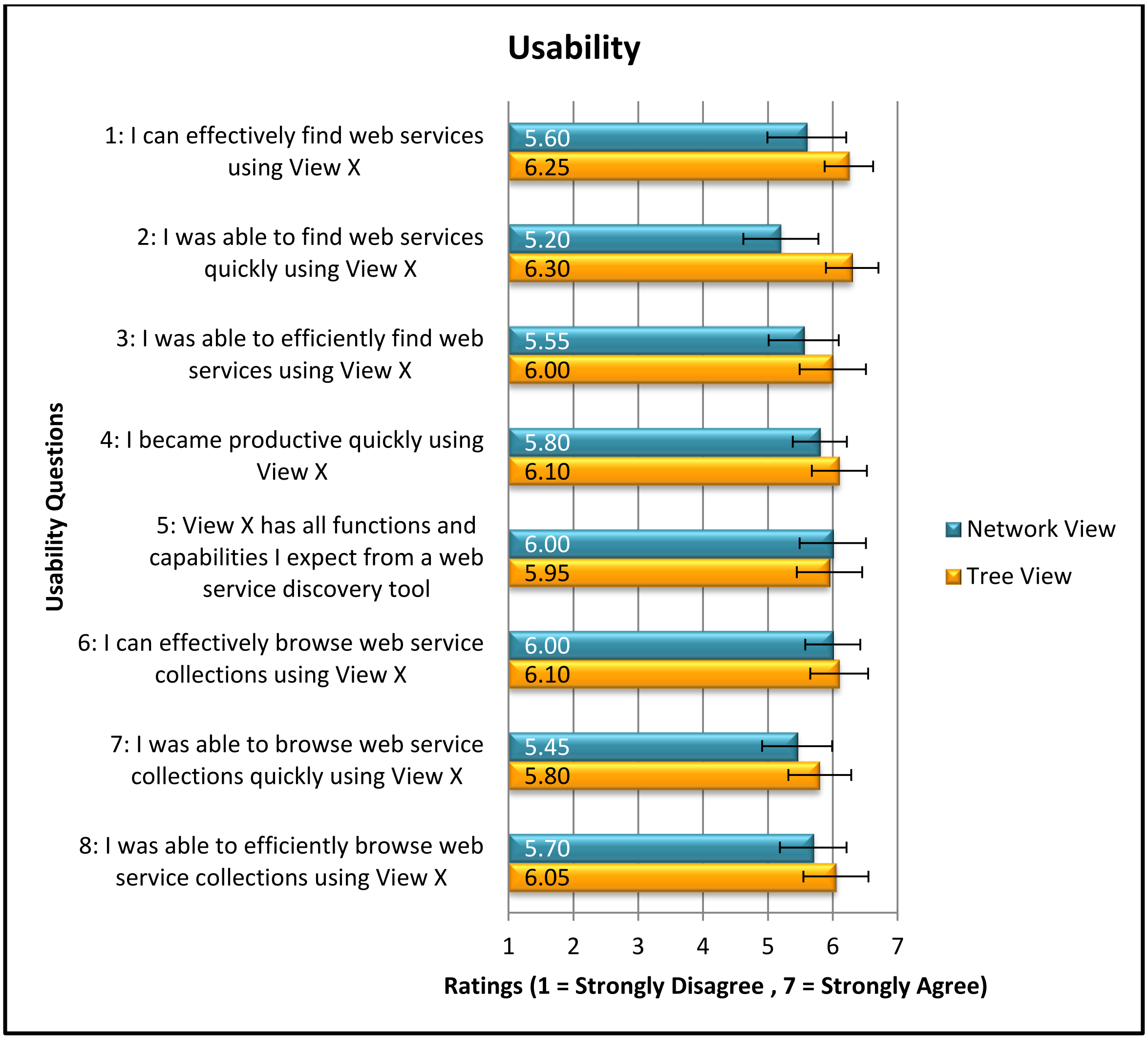


Figure 6-14 Usability Ratings using a 7-point Likert Scale (n=20)

An additional question was added to the post-task questionnaire to measure whether participants perceived the overview used in each of the views to be useful. Figure 6-15 shows that the participants considered the overview in the network view to be more useful than the fisheye strategy in the tree view. The difference between the mean rating and the neutral rating of 4 was significant for the network view (mean= 5.65, p=0.002). The mean rating for the usefulness of the overview was significantly higher for the network view (p=0.01).

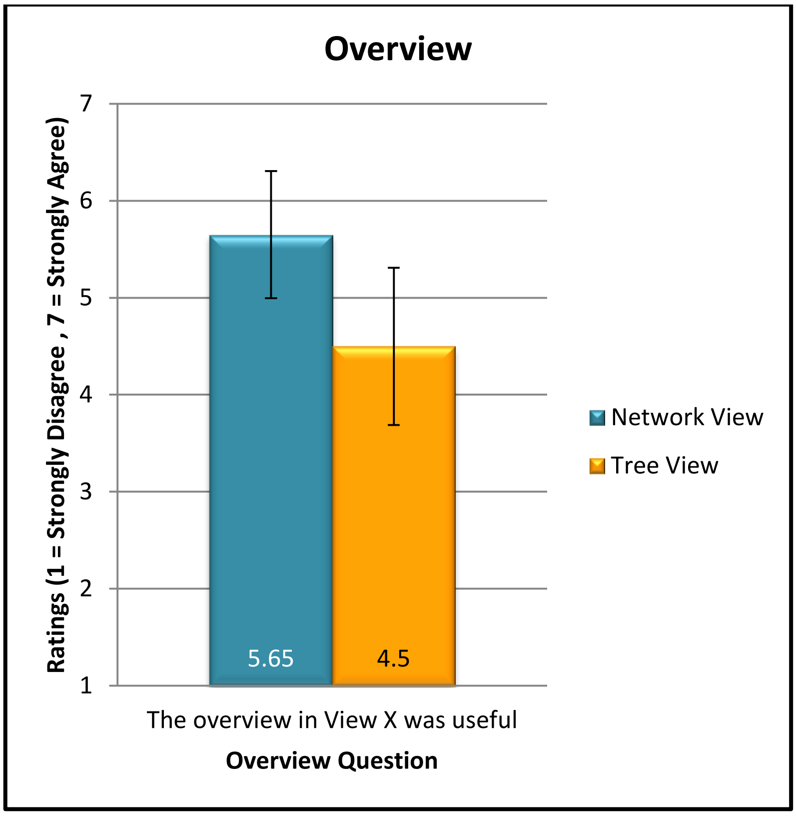


Figure 6-15 Overview using a 7-point Likert Scale (n=20)

Post-task questionnaires were also used to capture participants' comments regarding each view. Participants were asked to note the most positive and negative aspect of each view. The comments regarding the most positive aspect of the network view is shown in Table 6-4 where n indicates the number of participants who made similar comments.

Table 6-4 Most Positive Comments for the Network View

Description	n
1. Easy to learn.	5
2. Easy to find services; if unsure which service to look for, the categories are well-labelled and helps to guide you.	4
3. The physical size of a node to indicate number of services was good.	3
4. Easy to use the search box.	2
5. Aesthetically pleasing.	2
6. Not a lot animations / faster animations.	2
7. The non-linear organisation was a better use of space on the screen (I could scan left-right, not just up-down).	2
8. System managed moving to expanded section for me - took away anxiety of panning & scrolling.	2

Five participants found the network view to be easy to learn while four participants found it easy to find web services with the network view. The comments regarding the most negative aspect of the network view are shown in Table 6-5. Nine participants noted that they found it difficult to browse to find a category because of the network view not being sorted in any way.

Table 6-5 Most Negative Comments for the Network View

Description	n
1. Struggled to find categories without search - were not in any logical order (was expecting alphabetical).	9
2. Overlapping categories.	3
3. Sometimes hard to find a specific (small) category.	3
4. Difficult to find web services in a category if a specific category has numerous web services.	2

General comments involved a suggestion of including some sort of sorting in the network view (n=6) and that participants found the network view easy to use (n=4) as shown in Table 6-6.

Table 6-6 General Comments for the Network View

Description	n
1. Somehow implement some ordering of web service nodes.	6
2. Easy to use and operate.	4

Nine participants found the alphabetical ordering of the tree view to be the most positive aspect of this view as can be seen in Table 6-7. Seven participants found the tree view easy to use.

Table 6-7 Most Positive Comments for the Tree View

Description	n
1. The list was sorted alphabetically, making it quicker and easier to find a category.	9
2. Easy to use and very effective.	7
3. Fisheye / magnify helps in identifying where I am.	3

Nine participants found the fisheye strategy distracting and this could be the reason that the participants found the overview in the network view more useful. Seven participants found the tree view to be cluttered as shown in Table 6-8.

Table 6-8 Most Negative Comments for the Tree View

Description	n
1. Fisheye distracting.	9
2. Not easy to use with manually browsing → too much information / web services at one time.	7
3. Zooming with mouse wheel was a bit confusing at times.	2

General comments regarding the tree view included comments that the tree view was easy to use and easy to learn (n=6) (Table 6-9). A suggestion was made to reduce the method of expansion as it required more scrolling to find specific web services in a category (n=3).

Table 6-9 General Comments for the Tree View

Description	n
1. It was easy to use and easy to learn.	6
2. Reduce the expansion of the trees.	3
3. Scroll / zoom function acts the wrong way.	2

Post-Test Satisfaction Results

Post-test satisfaction results are shown in Figure 6-16. The results for the post-test questions were mixed but slightly favoured the tree view. The participants slightly preferred the tree view to visualize web service collections (mean = 3.65). Nine participants preferred the network view and one participant could not choose between the two. The participants preferred the network view as the more appropriate technique to visualize the sample web service collection of 3562 nodes (mean = 4.25).

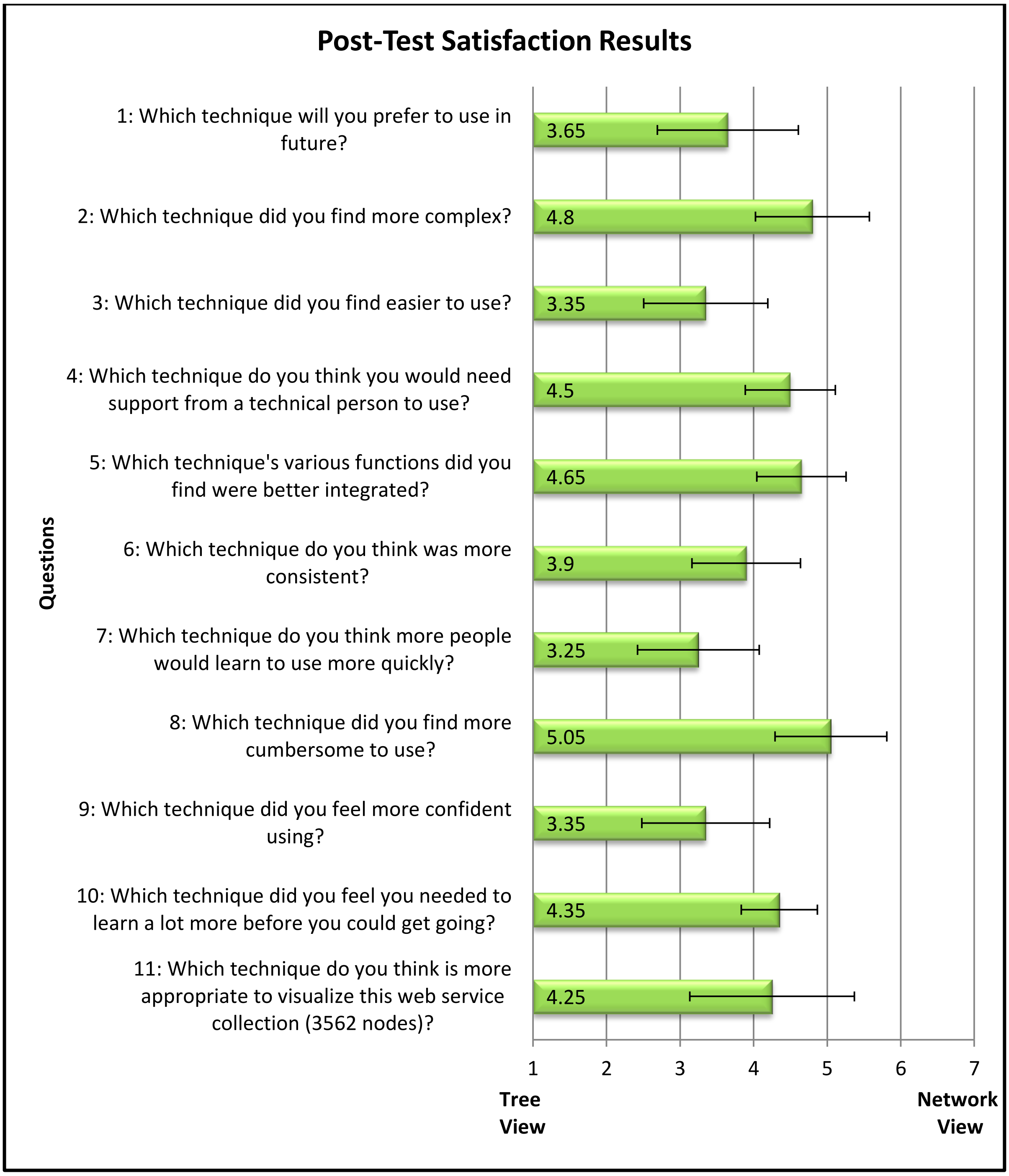


Figure 6-16 Post-Test Satisfaction Results using a 7-point Likert Scale (n=20)

The difference was statistically significant for question 8 as shown in Table 6-10. The participants found the network view significantly more cumbersome to use (mean =5.05, p=0.03).

Table 6-10 Wilcoxon Matched Pairs Test for the Post-Test Questions

Question	Z	p-value
1. Which technique will you prefer to use in future?	0.85	0.40
2. Which technique did you find more complex?	1.82	0.07
3. Which technique did you find easier to use?	1.29	0.20
4. Which technique do you think you would need support from a technical person to use?	1.44	0.15
5. Which technique's various functions did you find were better integrated?	1.94	0.05
6. Which technique do you think was more consistent?	0.21	0.83
7. Which technique do you think more people would learn to use more quickly?	1.59	0.11
8. Which technique did you find more cumbersome to use?	**2.12**	**0.03**
9. Which technique did you feel more confident using?	1.31	0.19
10. Which technique did you feel you needed to learn a lot more before you could get going?	1.35	0.18
11. Which technique do you think is more appropriate to visualize this web service collection (3562 nodes)?	0.47	0.64

General comments in the post-test questionnaire are shown in Table 6-11 where n indicates the number of participants who made similar comments. The comments received in the post-test questionnaire related to participants noting that SerViz provided good support for finding web services (n=11). Six participants made positive comments regarding the network view and five participants made positive comments regarding the tree view.

Table 6-11 General Comments in the Post-Test Questionnaire

Description	n
1. Works well in general & represents good search tool options.	11
2. The network view is awesome as it shows the most visited nodes larger than the least visited so it is more obvious. It will be more relevant to someone browsing the web than the tree view would be. The network view is also more interactive and fun to use. The network view is more frustrating to use without the search function, especially for the small nodes.	6
3. Like tree view's hierarchy as it sorts categories in alphabetical order.	5

6.5.3. Eye-Tracking Results

Heat maps were taken for the first and fourth task for the network view and only for the first task for the tree view. As discussed in Section 5.2.2.3., the heat maps were taken from the start of the task until a participant expanded a category to represent a static image. The first task involved browsing for a specific category to find the number of web services in the category.

The heat map for the first task of the network view is shown in Figure 6-17. The heat map was taken for fifteen participants as the other participants' graphs were not zoomed in to the default zoom level or the participants zoomed in or out while browsing the graph. The participants browsed the graph in the network view to find the required category towards the middle-left of the graph. The heat map shows that the participants browsed through all the categories to find the required category. This could explain why the browsing tasks were slower for the network view.

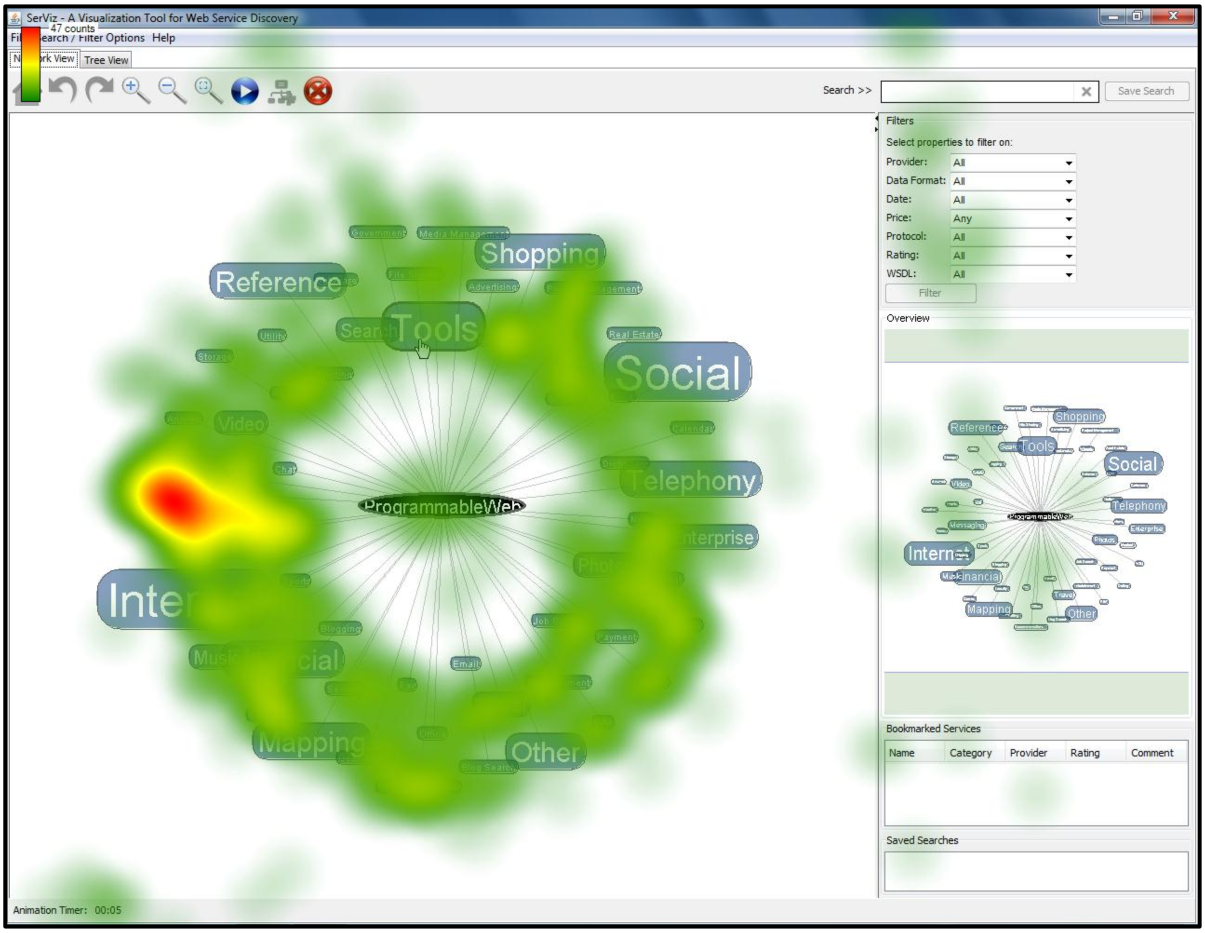


Figure 6-17 A Heat Map of Participants Browsing the Web Service Collection in the Network View (n=15)

Task four involved searching and filtering to find specific web services. The heat map for this task is shown in Figure 6-18. The heat map was taken for fifteen participants as the other participants zoomed in or panned the display of the graph while completing the task. The participants were required to enter a search for web services and use the filter facility to further refine the search. The participants were then required to expand the categories and write down the name(s) of the web services that met the search and filter criteria. The participants immediately searched and looked through the list of filters to find the specific filter required for the task. The participants then fixated on the category that contained a web service that met the search and filter criteria before expanding the category to find the web service. The category was positioned in the middle-left of the graph. The gaze on the overview shown in Figure 6-18 is fixation on the drop-down menu of the required filter rather than fixation on the top-left of the overview.

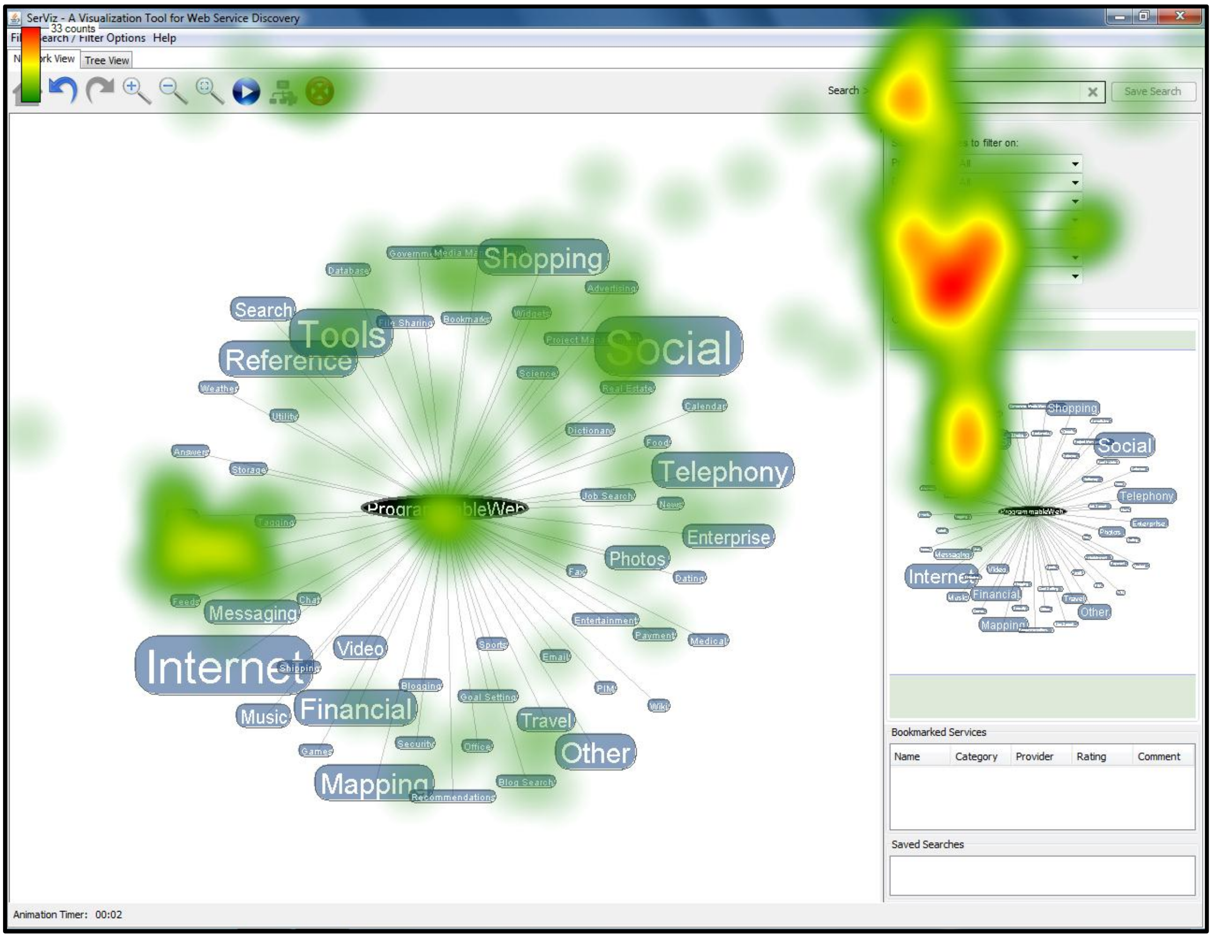


Figure 6-18 A Heat Map of Participants Searching and Filtering to find Web Services in the Network View (n=15)

The heat map taken for the first task of the tree view is shown in Figure 6-19. The participants immediately fixated on the required category due the sorted nature of the tree view. The comparison between the heat maps of the first task for each view shows that it was easier to browse the web service collection with the tree view due to the alphabetical sorting aspect.

It was not possible to obtain a heat map for the fourth task of the tree view as the participants either used the scroll bar to scroll to the category to be expanded or zoomed in or out during the task or the tree was not displayed at the default zoom level.

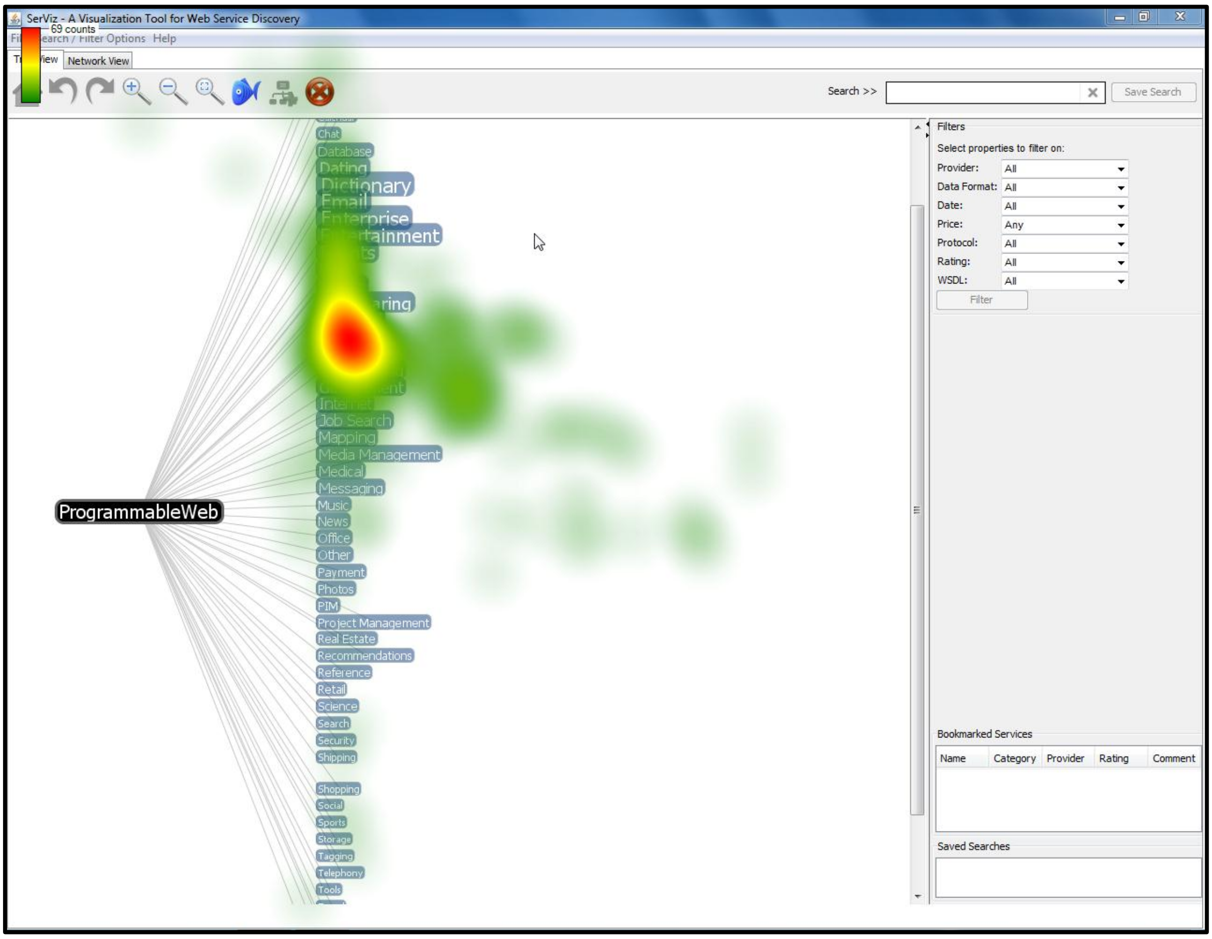


Figure 6-19 A Heat Map of Participants Browsing the Web Service Collection in the Tree View (n=19)

6.5.4. Discussion

The results show that the tree view was slightly preferred over the network view to visualize the web service collection. However the participants slightly favoured the network view as the most appropriate view to visualize the sample web service collection (Figure 6-16). Ten participants (50%) preferred the tree view, while nine participants preferred the network view and one participant could not choose between the two. The effectiveness results generally support both views.

The tree view was faster in terms of browsing as participants found that the alphabetical ordering of categories and web services assisted them in finding web services. The network view was faster for searching and filtering as searching and filtering required the participants to browse through the relevant categories and web services. Participants found the network view to be more cumbersome to use (Figure 6-16) and required more effort (Figure 6-12). This could be due to the fact that the network view does not provide any sorting when browsing categories and web services. Participants found both views easy to use and easy to learn. The participants perceived that they could quickly find web services using the tree view (Figure 6-14).

More participants made use of the overview compared to the previous study as the participants were aware of its use and found it to be useful. Some participants found the fisheye strategy to be useful while other participants found this type of overview to be distracting.

Based on the screen recordings of the evaluation and the evaluation results, some suggestions for improvement can be made:

- The network view requires some method of alphabetical ordering;
- Both views need a means for grouping web services to reduce on-screen clutter of large categories;
- The fisheye strategy for the tree view is too sensitive when attempting to select a node and provides too much animation;
- Shortcuts should be provided for buttons on the screen for both views; and
- The nodes should be resized to display the number of search items when searching and filtering in the network view.

A larger sample was used for this usability evaluation (n=20) compared to the previous evaluation (n=10), but the participants were limited to post-graduate students and staff.

6.6. Design Recommendations

The results of the evaluation were similar for both views. From this evaluation, certain design recommendations can be made when using an IV technique to visualize a web service collection. The following six recommendations can be made:

- **The IV technique needs to provide little or no animation.**

  The animation effect caused by the FDL's (Force Directed Layout) repositioning of nodes in the network view was found to be distracting to the participants of both usability evaluations. The second evaluation included a more controlled animation, but due to the size of the collection the nodes still overlapped. If many categories are expanded, the animation timer takes longer to fall under the overlapping threshold and thus it takes longer for the animation to stop. Also, the fisheye strategy provided animation that made it difficult for participants to select categories, with some participants even switching off the fisheye during the evaluation of the tree view.

- **The IV technique should provide a layout that makes it easy for a user to view all the nodes.**

  The FDL used to re-position the nodes in the network view does not provide support for overlapping nodes. The layout used for the selected IV technique either needs a near-fixed layout or should provide for collision detection between nodes. The participants of the usability evaluation commented that the tree IV technique did not make effective usage of effective screen space. If a tree IV technique is used to visualize web service collections, it could be useful to implement a multi-column layout for web services of a large category as discussed by Song *et al.* (2010).

- **The IV technique should provide alphabetical ordering of the nodes.**

  The results of the second evaluation showed that browsing was easier and faster for the tree view. This was because the tree view provided alphabetical sorting of the categories and the web services. The nodes should be sorted in some way to assist the user in browsing the web service collection.

- **The IV technique should provide a sufficient method of distinguishing between the collection, category and web service nodes.**

  From analysis of the screen recordings it was observed that participants found it difficult to distinguish between a category and a web service. The structure of the layout, the use of colour-coding and node shape should make it easy for a user to distinguish between the different types of nodes in the graph to avoid confusion.

- **For a large web service collection there needs to be an extra level in the hierarchy to group related web services.**

Some of the categories in the sample web service collection provided hundreds of web services. This can make it difficult for a user to browse a large category and leads to on-screen clutter. The web services need to be grouped somehow, e.g. by alphabetical ordering (A→G, H→N, O → U, V→Z). An option could also be provided to hide irrelevant categories and web services not part of the search and/or filter results instead of only de-saturating the nodes. Additionally, categories can be expanded that are part of the search and/or filter results to assist the user in finding certain web services and make the techniques less cumbersome to use.

- **The IV technique should provide an overview.**

  The overview used in the network view was shown to be more useful than the fisheye strategy used in the tree view. Participants of the evaluation comparing the two IV techniques found the fisheye strategy to be distracting.

Due to the similarity in the results of the evaluation comparing the network and tree views, it is recommended to provide both views to the user in order for the user to select which view he would like to use. The web service collection limitations will need to be considered however, as the tree view only supports hierarchical web service collections.

6.7. Conclusion

The usability problems identified with the network IV technique implemented in SerViz in the first usability evaluation identified a need for the implementation of an alternative IV technique. This IV technique was required to overcome the usability problems identified with the network IV technique. An analysis was done to select the most suitable IV technique and the node-and-link IV technique was selected to visualize hierarchical web service collection data. The main feature of this technique is that it is more structured than the network IV technique.

This alternative technique, referred to as the tree IV technique, was designed and implemented according to the functions outlined in Section 4.2.2. A fisheye strategy was implemented to support the overview function rather than the overview feature implemented in the network IV technique. A limitation of this technique is that this technique can only visualize hierarchical web service collections.

A usability evaluation was conducted to compare the network and tree IV techniques implemented in SerViz. The objective of the usability evaluation was to determine which IV technique would be more suitable to visualize web service collections. This evaluation was conducted similarly to the previous usability evaluation.

The participants (n=20) preferred the tree IV technique to visualize web service collections. The tree IV technique was faster for browsing while the network IV technique was faster for searching and filtering, bookmarking a web service and saving a search. Participants commented that the alphabetical ordering in the tree IV technique was a positive aspect of the technique and that the on-screen space usage was a positive aspect in the network IV technique. Participants also found the overview in the network IV technique more useful than the fisheye strategy in the tree IV technique. The participants preferred the network IV technique to visualize a large web service collection such as the ProgrammableWeb collection. The eye-tracking results showed that the alphabetical sorting characteristic of the tree IV technique made it easier to browse the web service collection.

Suggestions for improvement for both IV techniques were identified from the screen recordings and from the comments in the questionnaires. The network IV technique needs to have some method of alphabetical ordering to make it easier to browse web service collections. The fisheye strategy used for the tree IV technique is too sensitive and makes it difficult to select a category or web service. Both IV techniques require some way of reducing on-screen clutter and need short-cuts for buttons provided on the interface.

Based on the results of the comparative evaluation of the two IV techniques, several general design recommendations were made for using an IV technique to visualize web service collections. A further recommendation was made that both IV techniques should be provided to enable the user to choose which IV technique to use, if possible.

This usability evaluation further supports the thesis statement defined in Chapter 1 that IV techniques can be used to effectively support web service discovery. Participants of the first usability evaluation preferred using IV to visualize web service collections. The next chapter will discuss the contribution of this research and describe possible future work to be conducted.

Chapter 7: Conclusions

7.1. Introduction

The main goal of this research was to investigate the usage of information visualization (IV) techniques to support web service discovery. Problems of existing web service discovery methods were identified and IV was proposed to overcome the web service discovery issues. Appropriate IV techniques were identified and applied to a selected web service collection. A usability evaluation was conducted to compare the network IV technique to the commonly used list-based technique. From the results of this evaluation, an alternative IV technique, namely the tree IV technique, was selected for implementation. A comparative evaluation was used to determine which IV technique is more suitable to visualize web service collections.

This chapter concludes the research by summarizing the findings of the research and outlining the contributions of the work. Limitations of the research and problems encountered are described. The chapter closes by discussing recommendations and suggestions for future work.

7.2. Achievements of Research Objectives

This section revisits the research objectives defined in Chapter 1 and discusses the achievements of the research in terms of these objectives. A summary concludes the section.

7.2.1. Review of Research Objectives

The thesis statement for this research was defined in Chapter 1 as follows:

> *Information visualization techniques can be used to effectively support web service discovery.*

The main research objective for this research was to determine how IV techniques can be used to effectively support web service discovery. The sub-objectives of this research used to address the main research objective were as follows:

1. To identify the existing problems with web service discovery;

2. To identify the IV techniques that can be used to support web service discovery;
3. To develop a web service discovery tool using suitable IV techniques; and
4. To evaluate the effectiveness of the proposed IV techniques

The extent to which these research objectives were met is discussed in the next section.

7.2.2. Research Achievements

This research has shown that IV can be successfully used to support web service discovery. This outcome was determined by addressing the above-mentioned research objectives.

The first research question was addressed in Chapter 2. Web services and web service discovery were discussed to gain an insight into the problem domain. Several problems were identified with existing web service discovery methods by reviewing the different methods of web service discovery, namely the Universal Description, Discovery and Integration (UDDI), search engines and publication sites. These three methods are considered to be the main web service discovery methods. The results of the comparison involved problems with the search and browse facilities and the poor presentation of web service collections.

The number of web services available over the Web is increasing rapidly. Web service discovery is a method for finding specific web services. Several criteria were proposed to compare each web service method, including the provision of functional and non-functional web service properties, an effective graphical user interface (GUI), a classification facility, a search and browse facility, a ranking facility and a sorting facility. The three existing methods of web service discovery were compared using these criteria to identify problems. The key problems identified were problems with searching and browsing, and the poor presentation of web service collections.

Chapter 3 addressed the second research question. Several IV techniques were described and analyzed. The node-and-link network IV technique was selected as the most appropriate IV technique to visualize web service collections. This IV technique was selected as a result of analyzing the data structure of three web service collections, namely RemoteMethods, ProgrammableWeb and Service-Finder.

The three publication sites were compared to determine which had the most support for the web service discovery criteria discussed in Chapter 2. RemoteMethods and ProgrammableWeb contain web service collections with a hierarchical data structure, while Service-Finder provides a web service collection with a network data structure. The node-and-link network IV technique was selected as hierarchical techniques were considered to be too limited to visualize most web service collections.

IV requirements for web service discovery were also identified in Chapter 3. These requirements were adapted from the visual information seeking mantra proposed by Shneiderman (1996) for web service discovery. These requirements included supporting the following tasks: overview, zoom, details-on-demand, filter, relate, history and extract.

Two existing applications incorporating IV to visualize web service collections were discussed and compared using the criteria identified in Chapter 2. Several shortcomings were identified with these applications including a lack of a search facility.

The third research objective was discussed and addressed in Chapter 4 which involved the design and implementation of SerViz, a tool for the interactive visualization of web service collections. SerViz incorporated the network IV technique and an alphabetical list-based technique to visualize the ProgrammableWeb web service collection. The ProgrammableWeb web service collection was selected to be visualized by SerViz as it is a dynamic web service collection which is growing constantly. Additionally, the ProgrammableWeb publication site provides an API (Application Programming Interface) that allows access to its web service collection. The list-based technique is the most commonly used method for web service discovery and was implemented for comparison purposes.

SerViz was developed using the Prefuse data visualization toolkit and implemented in Java. Other possible toolkits included Microsoft Silverlight and Flare, a Flash-based implementation of Prefuse. Microsoft Silverlight provides limited support for interactive graph visualization. A comparison of Prefuse and Flare determined that, while Flare provides better UI (User Interface) controls, Prefuse provides enhanced support for interactive data visualization. The XML (Extensible Mark-up Language) document containing the ProgrammableWeb web service collection was converted to a GraphML (GML) file, which is used by Prefuse to visualize network data structures. Several functions were identified and derived from the IV requirements for web service discovery discussed in Chapter 3 and implemented in both techniques in SerViz. This research has also shown that Prefuse can be used to interactively visualize web service collections. While Prefuse provided support for most of the required functionality such as layout and node positioning, some functionality such as undoing and redoing actions and details-on-demand was not supported and these functions were implemented separately.

The last research objective was addressed in Chapter 5 and Chapter 6. Chapter 5 described a usability evaluation of SerViz comparing the network IV technique with the existing list-based technique. The usability evaluation was conducted using 10 participants in the Usability Lab in the Department of Computing Sciences at the Nelson Mandela Metropolitan University (NMMU). Efficiency, effectiveness and user satisfaction were captured for the usability evaluation. Two sets of similar but not identical sets of tasks were provided to each participant to evaluate each IV technique. These tasks evaluated the browsing, searching, filtering, bookmarking a web service and saving a search aspects of each IV technique.

The main result of the study was that most participants (90%) preferred the network IV technique to visualize web service collections. The network IV technique was faster for browsing the web service collection and for saving a search for web services. Due to the high level of accuracy received for both techniques, it was determined that both techniques supported the user's tasks. Participants were significantly more satisfied with the network IV technique. Participants perceived that they could effectively and efficiently discover web services using the network IV technique, even though the overall effectiveness for each technique was identical and the network IV technique was faster in only two of the six tasks. The perception that the participants could effectively and efficiently browse the web service collection using the network IV technique supports the efficiency result that browsing was significantly faster using this IV technique.

The post-test questionnaire results generally supported the network IV technique. The participants found the network IV technique easier to use than the list technique, better integrated and less cumbersome to use. The results also showed that the participants perceived the network IV technique to be the technique that would be quicker to learn, although a list is the existing method used to visualize web service collections. Post-test questionnaire comments generally favoured the network IV technique. Eye-tracking results showed that different processes were followed when browsing and searching the web service collection. The participants focused on the network graph when browsing, but focused on the search and filter facilities and the results in the graph when searching.

The usability evaluation described in Chapter 5 was used to identify any usability problems with the two techniques. Several usability problems were identified with the network IV technique. The overview was not determined as being useful as the participants did not make use of the overview during the evaluation. Other problems included difficulties with on-screen clutter, the continuous animation of the layout, node occlusion and difficulty in distinguishing between the collection, category and web service nodes. Modifications were implemented in SerViz to address the identified usability problems. These improvements included a hide option when searching to hide irrelevant categories and web services; changing the single click to expand a category to a double-click to reduce on-screen clutter; an animation timer to reduce the continuous node re-positioning; implementation of a certain level of transparency to the node colour to view overlapped or hidden nodes; an improved colour-coding to assist with distinguishing between collection, categories and web services; and an expand option to automatically expand relevant categories when searching and filtering to discover web services.

The results from the usability evaluation and usability problems identified with the network IV technique discussed in Chapter 5 provided motivation for the investigation of an alternative IV technique in Chapter 6. Requirements were identified to assist in selecting an appropriate alternative IV technique to overcome or avoid the usability problems identified with the network IV technique. The design, implementation and evaluation of the alternative technique were discussed in Chapter 6. The selection process compared the possible IV techniques to be considered. The node-and-link hierarchical IV technique (referred to as the tree IV technique in Chapter 6), was selected as the alternative IV technique to be implemented in SerViz. This technique was selected as it is more structured than the network IV technique. A limitation of this technique is that it is restricted to visualizing hierarchical web service collections. The XML document containing the ProgrammableWeb web service collection was then converted to a TreeML (TML) file supported by Prefuse and used by the toolkit to visualize hierarchical data structures. This conversion process was similar to the XML-GML conversion used for the network IV technique using XSLT (Extensible Style-sheet Language Transformations) and XPath. The tree IV technique was designed and implemented similar to the network IV technique in terms of functionality and the tasks outlined in Chapter 4. A fisheye strategy was implemented instead of the overview used in the network IV technique as it offers an alternative method to support the overview task and was considered as an appropriate overview method for the tree IV technique.

A comparative usability evaluation was conducted to compare the two IV techniques to determine which IV technique is more suitable for visualizing web service collections. Twenty participants completed the usability evaluation following a similar procedure to the first evaluation. Identical metrics were captured for this evaluation and similar questionnaires were used from the first usability evaluation. Similar task lists were used in the evaluation with the addition of training tasks to the first four tasks to avoid the learnability factor involved when using novel techniques to complete a task. The main result of the evaluation was that half of the participants (n=10) preferred the tree IV technique and most of the remaining participants (n=9) preferred the network IV technique and one participant was unable to choose between the two. The tree IV technique was faster for browsing, due to the sorted nature of the technique, while the network IV technique was faster for searching and filtering. Both IV techniques received high levels of accuracy for effectiveness. Post-test questionnaire results slightly favoured the tree IV technique. The most positive aspect of the tree technique was the alphabetical ordering of the nodes, while the most positive aspect of the network technique was the better screen space usage. Participants found the overview to be more useful than the fisheye strategy. The participants preferred the network IV technique to visualize the ProgrammableWeb web service collection due to the size of the collection (3562 nodes). The eye-tracking results showed that participants focused on the entire graph in the network IV technique when browsing and on the required category in the tree IV technique. This was due to the sorted nature of the tree IV technique which supported the results that the tree IV technique was faster for browsing the web service collection.

Suggestions for improvement were identified for both techniques from the participants' comments and screen recordings. The network IV technique needs to support alphabetical ordering of nodes; the fisheye strategy in the tree IV technique was too sensitive; and both techniques need some method of reducing on-screen clutter. Chapter 6 also included general design recommendations for using IV techniques for web service discovery. A recommendation was also made to provide both IV techniques to the users in order to enable them to choose which IV technique to use for web service discovery, if the web service collection is hierarchical.

7.2.3. Summary

This research supported the research objectives defined in Chapter 1. The participants of the first usability evaluation preferred using IV techniques for web service discovery and the second evaluation confirmed these results. This result supports the thesis statement that IV techniques can be used to support web service discovery. The research achievements include the following:

- Identification of several problems with existing web service discovery methods;
- Proposal of web service discovery criteria;

- Definition of IV requirements for web service discovery;
- Selection of the most appropriate IV technique(s) to visualize web service collections (network and tree IV techniques);
- Development of a prototype tool for interactive visualization of web service collections;
- Determination of the usefulness of using IV techniques to support web service discovery; and
- Proposal of design recommendations for using IV techniques for web service discovery.

7.3. Summary of Contributions

The contribution of the research is discussed in terms of the theoretical and practical contributions. The theoretical contributions relate to the general use of IV to support web service discovery. The practical contributions relate to the implementation of the IV techniques.

7.3.1. Theoretical Contribution

Several theoretical contributions were made by this research. The results of the first usability evaluation conducted comparing the network IV technique to the list-based technique supported the principle that the structure of the data to be visualized can be used to determine the appropriate IV technique to be used (Shneiderman 1996).

Criteria were identified for web service discovery in Chapter 2 that can be used to compare several different methods of web service discovery or to determine which specific method provides the most support for effective web service discovery. IV requirements for web service discovery were derived from Shneiderman's visual information seeking mantra adapted for web service discovery (Section 3.5.). This can assist in further development of IV techniques for web service discovery. Finally, general design recommendations for using IV techniques to visualize web service collections were proposed (Section 6.6.). The main recommendation was that both IV techniques should be available to allow the user to select his preferred option, if possible.

7.3.2. Practical Contribution

The practical contribution of this research includes the SerViz tool and the IV techniques that were developed and implemented in SerViz. Two selected IV techniques were implemented in SerViz to support the IV requirements for web service discovery identified in Chapter 3. Most of the participants of the first usability evaluation preferred the network IV technique to the list-based technique, while half of the participants preferred the tree IV technique to the network IV technique in the second evaluation. The tree IV technique was faster for browsing while the network IV technique was faster for searching and filtering.

Results of the usability evaluations provide a practical contribution as researchers can use these results for comparison purposes. SerViz was developed as a tool for interactive visualization of web service collections. SerViz can be used to incorporate more IV techniques for future comparisons and experimentation or to visualize different web service collections.

7.4. Limitations and Problems Encountered

Limitations include the fact that only one web service collection (ProgrammableWeb) was visualized in SerViz. An additional limitation is that only post-graduate students and staff were used for both experiments.

SerViz was implemented as a desktop application, although it would be preferable to have a web application for easier access to the tool. SerViz was implemented in Java due to the limited support for interactive graph drawing provided by other tools and toolkits. Although there are limitations in using Java, it was determined that Prefuse can be used to effectively visualize web service collections. There was also limited access to public web service collections and thus the visualization of one web service collection was implemented.

Problems were encountered with visualizing the web service collection in Prefuse. The different visualizations in Prefuse required different data structures, namely GML and TML. Two separate conversion processes were thus required to convert the XML document containing the ProgrammableWeb web service collection to the required data structures. Additionally, several calls were required using the API provided by ProgrammableWeb to retrieve its web service collection as each call only retrieved 20 web services at a time. Each retrieval call had to append web services to the XML document containing the web service collection. A difficulty encountered during the implementation phase was that certain required functionality was not supported by Prefuse, such as the expansion and contraction of category nodes with the force directed layout (FDL) used for the network IV technique, undoing and redoing actions and details-on-demand.

Other problems encountered with the evaluations included the limitations of the eye-tracking results. This was due to heat maps representing static images and the IV techniques being interactive.

7.5. Future Research

Several opportunities for future research are possible from this research. Due to the SerViz prototype being limited to a desktop application, it may be useful to develop an interactive graph visualization toolkit in a more suitable library or plug-in such as Microsoft Silverlight. This toolkit could follow the IV requirements for web service discovery and design recommendations to make the toolkit specific for web service discovery. Due to the ubiquitous nature of computing today, it may also be useful to extend this research to a mobile platform or investigate collaborative web service discovery on a multi-touch surface platform.

The SerViz prototype can also be extended to enhance its features. The prototype could provide a means for updating the web service collection easily if new web services are added to a specific web service collection. The prototype could be used to visualize more than one web service collection or even multiple web service collections at the same time. The prototype could possibly crawl the Web to compile its own web service collection with the necessary web service information and classification of web services. Minor prototype extensions could include improving the prototype according to the comments and usability problems identified from the two usability evaluations, such as reducing the sensitivity of the fisheye strategy in the tree IV technique and the inclusion of alphabetical ordering in the network IV technique.

It could be feasible to conduct a further study in industry to obtain a different perspective of the usefulness of IV techniques to support web service discovery. Minor evaluation extensions include determining the usefulness of the Hide and Expand features implemented in SerViz which were not evaluated in the second usability evaluation as these features were not part of the main focus of the usability evaluation.

List of References

Al-Masri, E. and Mahmoud, Q. H. (2007a): Discovering the Best Web Service. In *Proceedings of 16th International Conference on World Wide Web*, Banff, Alberta, Canada: pages 1257-1258.

Al-Masri, E. and Mahmoud, Q. H. (2007b): WSCE: A Crawler Engine for Large-Scale Discovery of Web Services. In *Proceedings of 2007 IEEE International Conference on Web Services (ICWS)*, Salt Lake City, UT: pages 1104 - 1111.

Al-Masri, E. and Mahmoud, Q. H. (2008a): Discovering Web Services in Search Engines. *IEEE Internet Computing*,, vol. 12 (no. 3): pages 74-77.

Al-Masri, E. and Mahmoud, Q. H. (2008b): Investigating Web Services on the World Wide Web. In *Proceedings of 17th international conference on World Wide Web*, Beijing, China: pages 795-804.

Al-Masri, E. and Mahmoud, Q. H. (2008c): Toward Quality-Driven Web Service Discovery. *IT Professional*, vol. 10 (no. 3): pages 24-28.

Al-Masri, E. and Mahmoud, Q. H. (2009a): A Broker for Universal Access to Web Services. In *Proceedings of 2009 Seventh Annual Communication Networks and Services Research Conference*, Moncton, NB: pages 118-125.

Al-Masri, E. and Mahmoud, Q. H. (2009b): Web Service Discovery and Client Goals. *Computer*, vol 42 (no. 1): pages 104-107.

Al-Masri, E. and Mahmoud, Q. H. (2009c): Identifying Client Goals for Web Service Discovery. In *Proceedings of 2009 IEEE International Conference on Services Computing*, Bangalore: pages 202-209.

Al Hunaity, M. A. (2008): Towards an Efficient Quality Based Web Service Discovery Framework. In *Proceedings of 2008 IEEE Congress on Services - Part I*, Honolulu, HI: pages 261-264.

Amar, R. A. and Stasko, J. T. (2004): A Knowledge Task-Based Framework for Design and Evaluation of Information Visualizations. In *Proceedings of IEEE Symposium on Information Visualization (INFOVIS '04)*, Austin, TX: pages 143-150.

Atkinson, C., Bostan, P., Hummel, O. and Stoll, D. (2007): A Practical Approach to Web Service Discovery and Retrieval. In *Proceedings of 2007 IEEE International Conference on Web Services (ICWS)*, Salt Lake City, UT: pages 241-248.

Bachlechner, D. and Fink, K. (2008): Semantic Web Service Research: Current Challenges and Proximate Achievements. *International Journal of Computer Science and Applications*, vol. 5 (no. 3b): pages 117-140. Technomathematics Research Foundation.

Bachlechner, D., Siorpaes, K., Fensel, D. and Lausen, H. (2006): Web Service Discovery - A Reality Check. In *Proceedings of Demos and Posters of the 3rd European Semantic Web Conference (ESWC)*, Budva, Montenegro.

Badr, Y., Abraham, A., Biennier, F. and Grosan, C. (2008): Enhancing Web Service Selection by User Preferences of Non-functional Features. In *Proceedings of 4th International Conference on Next Generation Web Services Practices (NWESP '08)*, Seoul: pages 60-65.

Bashir, S., Javed, M. Y., Khan, H., Khan, A. and Khiyal, M. S. H. (2010): Indexer Based Dynamic Web Services Discovery. *International Journal of Computer Science and Information Security (IJCSIS)*, vol. 7 (no. 2): pages 153-159.

Beets, S. and Wesson, J. (2011a): SerViz: A Tool for Interactive Visualization of Web Service Collections. In *Proceedings of 2011 Southern Africa Telecommunication Networks and Applications Conference (SATNAC)*, East London International Convention Centre (ELICC), East London: pages 387-392.

Beets, S. and Wesson, J. (2011b): Using Information Visualization to Support Web Service Discovery. In *Proceedings of 2011 Annual Research Conference of the South African Institute for Computer Scientists and Information Technologists (SAICSIT 2011)*, Cape Town, South Africa: pages 11-20.

Berners-Lee, T. (2009): *Web Services* [online]. Available at: http://www.w3.org/DesignIssues/WebServices.html. [Accessed 24 February 2010].

Bernhard, J. and Kelso, N. V. (2007): Color Design for the Color Vision Impaired. *Cartographic Perspectives* (no. 58): pages 61-67.

Bhagat, J. , Tanoh, F., Nzuobontane, E., Laurent, T., Orlowski, J. , Roos, M., Wolstencroft, K., Aleksejevs, S., Stevens, R., Pettifer, S., Lopez, R. and Goble, C. A. (2010): BioCatalogue: A Universal Catalogue of Web Services for the Life Sciences. *Nucleic Acids Research* (no. 38; 2):W689-W694.

Borzo, J. (2004): *Get the Picture* [online]. Available at: https://panopticon.com/images/stories/in_the_news/2004_01_wall_street_journal_visualization_software_cuts_clutter.pdf. [Accessed 12 July 2010].

Cañas, A. J. , Carff, R., Hill, G., Carvalho, M., Arguedas, M. , Eskridge, T. C., Lott, J. and Carvajal, R. (2005): Concept Maps: Integrating Knowledge and Information Visualization. *Knowledge and Information Visualization: Searching for Synergies, Lecture Notes in Computer Science Springer-Verlag*: pages 205-219. Heidelberg/NY: Springer.

Card, S. (2008): Information Visualization. In *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications, Human Factors and Ergonomics*: pages 509-543. A. Sears and J.A. Jacko (Eds.) Boca Raton, Florida, CRC Press.

Card, S. K., Mackinlay, J. and Shneiderman, B. (1998): *Readings in Information Visualization - Using Vision to Think*. San Francisco.

Carpineto, C., Osinski, S., Romano, G. and Weiss, D. (2009): A Survey of Web Clustering Engines. *ACM Computing Surveys*, vol. 41 (no. 3): pages 1-38.

Carr, D. A. (1999): Guidelines for Designing Information Visualization Applications. In *Proceedings of Ericsson Conference on Usability Engineering (ECUE)*, Stockholm, Sweden: pages 1-7.

Cavanaugh, E. (2006): Web services: Benefits, Challenges, and a Unique, Visual Development Solution (White Paper). Available at http://www.altova.com/whitepapers/webservices.pdf. Altova Inc.

Chen, C. (2005): Top 10 Unsolved Information Visualization Problems. *IEEE Computer Graphics and Applications,* vol. 25 (no. 4): pages 12-16. IEEE Computer Society Press.

Chen, C. (2010): Information Visualization. *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 2 (no. 4): pages 387-403. John Wiley & Sons, Inc.

Chua, F. F., Yuan, H. and Kim, S. D. (2007): A Visualization Framework for Web Service Discovery and Selection Based on Quality of Service. In *Proceedings of IEEE 2007 Asia-Pacific Service Computing Conference*, Tsukuba Science City: pages 312-319.

Cockburn, A., Karlson, A. and Bederson, B. B. (2008): A Review of Overview+Detail, Zooming, and Focus+Context Interfaces. *ACM Computing Surveys*, vol. 41 (no. 1): pages 1-31. ACM.

Collins, C., Carpendale, S. and Penn, G. (2009): DocuBurst: Visualizing Document Content using Language Structure. *Computer Graphics Forum (Proceedings of Eurographics/IEEE-VGTC Symposium on Visualization (EuroVis '09))*, volume 28 (no. 3): pages 1039-1046.

Cook, K., Earnshaw, R. and Stasko, J. (2007): Guest Editors' Introduction: Discovering the Unexpected. *IEEE Computer Graphics and Applications*, vol. 27 (no. 5): pages 15-19. IEEE.

D'Mello, D. A., Ananthanarayana, V. S. and Narasimhan, V. L. (2010): Challenges (Research Issues) in Web Services. In *Proceedings of 2010 International Conference on Advances in Computer Engineering (ACE), ,* Bangalore, Karnataka, India: pages 343-345

Deligiannidis, L., Kochut, K. J. and Sheth, A. P. (2006): User-Centered Incremental Data Exploration and Visualization. *Technical Report, LSDIS Lab and Computer Science*. Available at http://lsdis.cs.uga.edu/projects/semvis/PGV/PGV.pdf.

Dustdar, S. and Treiber, M. (2005): A View Based Analysis on Web Service Registries. *Distributed Parallel Databases,* vol. 18 (no. 2): pages 147-171. Kluwer Academic Publishers.

Fekete, J., Wijk, J. J. , Stasko, J. and North, C. (2008): The Value of Information Visualization. In *Information Visualization: Human-Centered Issues and Perspectives*: pages 1-18. Springer-Verlag Berlin, Heidelberg.

Gomadam, K., Ranabahu, A., Nagarajan, M., Sheth, A. P. and Verma, K. (2008): A Faceted Classification Based Approach to Search and Rank Web APIs. In *Proceedings of 2008 IEEE International Conference on Web Services (ICWS '08)*, Beijing: pages 177-184.

Guinard, D., Trifa, V., Karnouskos, S., Spiess, P. and Savio, D. (2010): Interacting with the SOA-Based Internet of Things: Discovery, Query, Selection, and On-Demand Provisioning of Web Services. *IEEE Transactions on Services Computing*, vol. 3 (no. 3): pages 223-235. IEEE.

Hagemann, S., Letz, C. and Vossen, G. (2007): Web Service Discovery - Reality Check 2.0. In *Proceedings of Third International Conference on Next Generation Web Services Practices (NWeSP 2007)*, Seoul pages 113-118.

Hart, S.G. and Staveland, L. E. (1988): Development of NASA-TLX (Task Load Index): Result of Empirical and Theoretical Research. In *Human Mental Workload*:pages 239-250. P.A. Hancock and N. Meshkati (Eds.) Amsterdam, Elsevier Science Publishers.

Heer, J. (2005): *Prefuse Visualization Gallery* [online]. Available at: http://www.prefuse.org/gallery/. [Accessed 4 December 2011].

Heer, J., Bostock, M. and Ogievetsky, V. (2010): A Tour Through the Visualization Zoo. *Communications of the ACM*, vol. 53 (no. 6): pages 59 - 67. ACM Media.

Heer, J. and Boyd, D. (2005): Vizster: Visualizing Online Social Networks. In *Proceedings of 2005 IEEE Symposium on Information Visualization (InfoVis), ,* Minneapolis, MN vol. 5: pages 32–39. IEEE Computer Society.

Heer, J. and Card, S. K. (2004): DOITrees Revisited: Scalable, Space-Constrained Visualization of Hierarchical Data. In *Proceedings of Working Conference on Advanced Visual Interfaces*, Gallipoli, Italy: pages 421-424.

Heer, J., Card, S. K. and Landay, J. A. (2005): Prefuse: A Toolkit for Interactive Information Visualization. In *Proceedings of SIGCHI Conference on Human Factors in Computing Systems*, Portland, Oregon, USA: pages 421-430. ACM.

Hofstee, E. (2009): *Constructing a Good Dissertation: A Practical Guide to Finishing a Master's, MBA or PhD on Schedule*. Johannesburg, South Africa: EPE.

Hornbæk, K. and Hertzum, M. (2011): The Notion of Overview in Information Visualization. *International Journal of Human-Computer Studies*, vol. 69 (no. 7-8): pages 509-525. Elsevier.

Ismaili, F., Shishedjiev, B., Zenuni, X. and Raufi, B. (2010): GHSOM-Based Web Service Discovery. In *Proceedings of 4th Conference on European Computing Conference (ECC'10)*, Stevens Point, Wisconsin, USA: pages 207-211.

Keim, D. A. (2002): Information Visualization and Visual Data Mining. *IEEE Transactions on Visualization and Computer Graphics*, vol. 8 (no. 1): pages 1-8. IEEE.

Keim, D. A., Andrienko, N., Fekete, J., Görg, C., Kohlhammer, J. and Melançon, G. (2008): Visual Analytics: Definition, Process, and Challenges. In *Information Visualization: Human-Centered Issues and Perspectives*: pages 154-175. Springer-Verlag Berlin, Heidelberg.

Keim, D. A., Kohlhammer, J., Santucci, G., Mansmann, F., Wanner, F. and Schaefer, M. (2009): Visual Analytics Challenges. In *Proceedings of Echallenges 2009*, Istanbul, Turkey: page 8.

Keim, D. A., Mansmann, F., Oelke, D. and Ziegler, H. (2008): Visual Analytics: Combining Automated Discovery with Interactive Visualizations. In *Proceedings of Discovery Science*, vol. 5255: pages 2-14. Springer-Verlag Berlin, Heidelberg.

Keim, D. A., Mansmann, F., Schneidewind, J., Thomas, J. and Ziegler, H. (2008): Visual Analytics: Scope and Challenges. In *Visual Data Mining: Theory, Techniques and Tools for Visual Analytics*: pages 76-90. Springer-Verlag Berlin / Heidelberg, Lecture Notes In Computer Science.

Keim, D. A., Mansmann, F., Schneidewind, J. and Ziegler, H. (2006): Challenges in Visual Data Analysis. In *Proceedings of Conference on Information Visualization*, London, England: pages 9-16. IEEE Computer Society.

Kosara, R. (2007): Visualization Criticism - The Missing Link Between Information Visualization and Art. In *Proceedings of 11th International Conference Information Visualization*, Zurich: pages 631-636. IEEE Computer Society.

Krus, D. J. (2002): *Imaging Higher Transcendental Functions in 3-Dimensions* [online]. Available at: http://www.visualstatistics.net. [Accessed 13 September 2010].

Ladner, R., Warner, E., Petry, F., Gupta, K. M., Moore, P. G., Aha, D. W. and Shaw, K. (2006): Case-Based Classification Alternatives to Ontologies for Automated Web Service Discovery and Integration. In *Proceedings of SPIE Defense & Security Symposium*, Orlando (Kissimmee), FL, USA, vol. 6201: pages 620117-1-620117-8.

Landre, E. and Wesenberg, H. (2007): Rest versus Soap: As Architectural Style for Web Services. In *Proceedings of 5th International OOPSLA Workshop on SOA & Web services Best Practices*, Montréal, Québec, Canada: pages 931–940.

Lewis, J. (1995): IBM Computer Usability Satisfaction Questionnaire: Psychometric Evaluation and Instructions for Use. *International Journal of Human-Computer Interaction*, vol. 7 (no. 1): pages 57-78. L. Erlbaum Associates Inc. Hillsdale, NJ, USA.

Li, W. and Takatsuka, M. (2004): Adding Filtering to Geometric Distortion to Visualize a Clustered Graph on Small Screens. In *Proceedings of 2004 Australasian Symposium on Information Visualisation - Volume 35*, Christchurch, New Zealand: pages 71-79. Australian Computer Society, Inc.

Liang, J. and Huang, M. L. (2010): Highlighting in Information Visualization: A Survey. In *Proceedings of 2010 14th International Conference Information Visualisation (IV)*, London: pages 79-85.

Lu, J. and Yu, Y. (2007): Web Service Search: Who, When, What and How. In *Proceedings of Web Information Systems Engineering - WISE 2007 Workshops*, Nancy, France: pages 284-295.

Makris, C., Panagis, Y., Sakkopoulos, E. and Tsakalidis, A. (2005): Efficient and Adaptive Discovery Techniques of Web Services Handling Large Data Sets. *The Journal of Systems and Software 79*: pages 480-495. Elsevier Science Inc. New York, NY, USA

Masui, T. (1998): LensBar - Visualization for Browsing and Filtering Large Lists of Data. In *Proceedings of 1998 IEEE Symposium on Information Visualization*, North Carolina: pages 113-120. IEEE Computer Society.

Microsoft (2011): *Microsoft Silverlight* [online]. Available at: http://www.silverlight.net/. [Accessed 05 April 2011].

Namahn (2001): *Using Eye Tracking for Usability Testing* [online]. Available at: http://www.namahn.com/resources/documents/note-eyetracking.pdf. [Accessed 4 December 2011].

Newmarch, J. (2003): *Web Service Structure* [online]. Available at: http://jan.newmarch.name/webservices/tutorial.html. [Accessed 31 March 2010].

Nielsen, J. (2000): *Why You Only Need to Test with 5 Users* [online]. Available at: http://www.useit.com/alertbox/20000319.html. [Accessed 1 December 2011].

North, C. (2005): Information Visualization. In *Handbook of Human Factors and Ergonomics*: pages 1222-1246. G. Salvendy (Ed.) New York, John Wiley & Sons.

OASIS (1993-2010): *uddi.xml.org - Online community for the UDDI Oasis standard: UDDI 101* [online]. Available at: http://uddi.xml.org/uddi-101. [Accessed 31 March 2010].

Olivier, M. S. (2009): *Information Technology Research: A Practical Guide for Computer Science and Informatics*. Pretoria, South Africa: Van Schaik Publishers, 3rd Edn.

Pacific Northwest National Laboratory (1998): *PictureThis, Connex* [online]. Available at: http://picturethis.pnl.gov/picturet.nsf/by+id/SMAA-45WQ5G. [Accessed].

Parastatidis, S. and Webber, J. (2005): MEP SSDL Protocol Framework. *Technical Report, CS-TR-900*. School of Computing Science, University of Newcastle, Newcastle upon Tyne, UK.

Pedrinaci, C., Liu, D., Maleshkova, M., Lambert, D., Kopecky, J. and Domingue, J. (2010): iServe: A Linked Services Publishing Platform. In *Proceedings of Ontology Repositories and Editors for the Semantic Web Workshop at the 7th Extended Semantic Web*, Heraklion, Greece.

Perer, A. and Shneiderman, B. (2006): Balancing Systematic and Flexible Exploration of Social Networks. *IEEE Transactions on Visualization and Computer Graphics (InfoVis)*, vol. 12 (no. 5): pages 693–700. IEEE Computer Society.

Pettifer, S., Ison, J., Kalaš, M., Thorne, D., McDermott, P., Jonassen, I., Liaquat, A., Fernández, J. M., Rodriguez, J. M., INB-Partners, Pisano, D. G., Blanchet, C., Uludag, M., Rice, P., Bartaseviciute, E., Rapacki, K., Hekkelman, M., Sand, O., Stockinger, H., Clegg, A. B., Bongcam-Rudloff, E., Salzemann, J., Breton, V., Attwood, T. K., Cameron, G. and Vriend, G. (2010): The EMBRACE Web Service Collection. *Nucleic Acids Research* (no. 38): W683–W688.

Plaisant, C. (2004): The Challenge of Information Visualization Evaluation. In *Proceedings of Working Conference on Advanced Visual Interfaces*, Gallipoli, Italy: pages 109-116. ACM.

Plaisant, C. (2005): Information Visualization and the Challenge of Universal Usability. In *Exploring Geovisualization*: pages 53-82. J. Dykes, A.M. MacEachren and M.J. Kraak (Eds.) Amsterdam, Elsevier.

Plaisant, C., Grosjean, J. and Bederson, B. B. (2002): SpaceTree: Supporting Exploration in Large Node Link Tree, Design Evolution and Empirical Evaluation. In *Proceedings of IEEE Symposium on Information Visualization (InfoVis'02)*, Boston, Massachusetts, USA: pages 57-64. IEEE Computer Society.

Platzer, C., Rosenberg, F. and Dustdar, S. (2009): Web Service Clustering using Multidimensional Angles as Proximity Measures. *ACM Transactions on Internet Technology*,, vol. 9 (no. 3): pages 1-26. ACM New York, NY, USA.

ProgrammableWeb (2009): *ProgrammableWeb Publication Site* [online]. Available at: http://www.ProgrammableWeb.com. [Accessed 10 May 2011].

Rabelo, E., Dias, M., Franco, C. and Pacheco, R. C. S. (2008): Information Visualization: Which Is the Most Appropriate Technique to Represent Data Mining Results? In *Proceedings of 2008 International Conferences on Computational Intelligence for Modelling, Control and Automation; Intelligent Agents, Web Technologies and*

Internet Commerce; and Innovation in Software Engineering, Vienna: pages 1228-1233.

Rajasekaran, P., Miller, J., Verma, K. and Sheth, A. (2004): Enhancing Web Services Description and Discovery to Facilitate Composition. In *Proceedings of SWSWPC 2004: International Workshop on Semantic Web Services and Web Process Composition*, San Diego, CA, USA: pages 55–68.

Ram, S., Hwang, Y. and Zhao, H. (2006): A Clustering Based Approach for Facilitating Semantic Web Service Discovery In *Proceedings of 15th Annual Workshop on Information Technologies & Systems (WITS)*, Dallas, TX, USA.

RemoteMethods (1999): *RemoteMethods Publication Site* [online]. Available at: http://www.remotemethods.com. [Accessed 10 May 2011].

Risch, J. S., Rex, D. B., Dowson, S. T., Walters, T. B., May, R. A. and Moon, B. D. (1997): The STARLIGHT Information Visualization System. In *Proceedings of IEEE Conference on Information Visualization (IV '97)*, London: pages 42 - 49.

Rivadeneira, W. and Bederson, B. B. (2003): A Study of Search Result Clustering Interfaces: Comparing Textual and Zoomable User Interfaces *University of Maryland Human-Computer Interaction Lab (HCIL) Technical Report (HCIL-2003-36)*.

Sabou, M. and Pan, J. (2007): Towards Semantically Enhanced Web Service Repositories. *Web Semantics: Science, Services and Agents on the World Wide Web*, vol.5 (no. 2): pages 142-150. Elsevier Science Publishers B. V. Amsterdam, The Netherlands.

Schulz, H-J. (2011): Treevis.net: A Tree Visualization Reference. *IEEE Computer Graphics and Applications*, vol. 31 (no. 6): pages 11-15. IEEE Computer Society Press Los Alamitos, CA, USA.

Sellami, M., Tata, S., Maamar, Z. and Defude, B. (2009): A Recommender System for Web Services Discovery in a Distributed Registry Environment. In *Proceedings of 2009 Fourth International Conference on Internet and Web Applications and Services*: pages 418-423. IEEE Computer Society.

Service-Finder (2008): *Service-Finder Publication Site* [online]. Available at: http://www.service-finder.eu. [Accessed 29 March 2011].

Sha, L., Shaozhong, G., Xin, C. and Mingjing, L. (2009): A QoS Based Web Service Selection Model. In *Proceedings of 2009 International Forum on Information Technology and Applications (IFITA '09)*, Chengdu, vol. 3: pages 353-356.

Shneiderman, B. (1996): The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. In *Proceedings of IEEE Symposium on Visual Languages*, Boulder, CO, USA: pages 336-343.

Shneiderman, B. (Ed.) (1998): *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, Reading, MA: Addison-Wesley Publishing.

Shneiderman, B. (2001): Inventing Discovery Tools: Combining Information Visualization with Data Mining. *Discovery Science, Lecture Notes in Computer Science*, vol. 2226/2001: pages 17-28. Springer Berlin / Heidelberg.

Silobreaker Ltd. (2009): *SiloBreaker* [online]. Available at: http://www.Silobreaker.com. [Accessed 14 September 2010].

Song, H., Cai, X. and Fu, Y. (2011): Rectangular Tree Browser: a Navigation and Visualization Tool for Large Hierarchies. *Journal of Information & Computational Science*, vol. 8 (no. 2): pages 354-361.

Song, H., Kim, B., Lee, B. and Seo, J. (2010): A Comparative Evaluation on Tree Visualization Methods for Hierarchical Structures with Large Fan-outs. In *Proceedings of CHI 2010*, Atlanta, Georgia, USA: pages 223-232.

Song, H., Qi, Y., Tian, X. and Xu, D. (2007): Navigating and Visualizing Long Lists with Fisheye View and Graphical Representation. In *Proceedings of Second Workshop on Digital Media and its Application in Museum & Heritage (DMAMH 2007)*, Chongqing: pages 123-128.

Stasko, J., Görg, C. and Liu, Z. (2008): Jigsaw: Supporting Investigative Analysis through Interactive Visualization. *Information Visualization*, vol. 7 (no. 2): pages 118-132. Palgrave Macmillan.

StatSoft Inc. (2003): *Nonparametric Statistics* [online]. Available at: http://egret.psychol.cam.ac.uk/statistics/local_copies_of_sources_Cardinal_and_Aitken_ANOVA/stnonpar.html. [Accessed 1 December 2011].

Steinmetz, N., Kerrigan, M. , Lausen, H., Tanler, M. and Sirbu, A. (2008): Simplifying the Web Service Discovery Process. In *Proceedings of 1st Workshop on Semantic Metadata Management and Applications (SeMMA) at the 5th European Semantic Web Conference (ESWC 2008)*, Tenerife, Spain: pages 31-45.

Stollberg, M. and Kerrigan, M. (2007): Goal-based Visualization and Browsing for Semantic Web Services In *Proceedings of Web Information Systems Engineering - WISE 2007 Workshops*, Nancy, France: pages 236-247.

The Internet Society (2006): *The Atom Publishing Protocol* [online]. Available at: http://bitworking.org/projects/atom/draft-ietf-atompub-protocol-12.html. [Accessed 29 March 2011].

ThinkMap Inc. (1998): *The Visual Thesaurus* [online]. Available at: http://www.visualthesaurus.com/. [Accessed 05 April 2011].

Tilsner, M., Hoeber, O. and Fiech, A. (2009): CubanSea: Cluster-Based Visualization of Search Results. In *Proceedings of IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technologies (WI-IAT '09)*, Milan, Italy: pages 108-112.

TouchGraph (2007): *TouchGraph* [online]. Available at: http://www.touchgraph.com. [Accessed 14 September 2010].

UC Berkeley Visualization Lab. (2009): *Flare* [online]. Available at: http://flare.prefuse.org/. [Accessed 05 April 2011].

Uddam, C. (2008): A Framework for Web Service Discovery: Service's Reuse, Quality, Evolution and User's Data Handling. In *Proceedings of 2nd SIGMOD PhD Workshop on Innovative Database Research*, Vancouver, Canada: pages 13-18.

Vande Moere, A. (2008): Beyond the Tyranny of the Pixel: Exploring the Physicality of Information Visualization. In *Proceedings of 2008 12th International Conference Information Visualisation*, London: pages 469-474. IEEE Computer Society.

Viégas, F. B. and Wattenberg, M. (2007): Artistic Data Visualization: Beyond Visual Analytics. In *Proceedings of 2nd International Conference on Online Communities and Social Computing*, Beijing, China: pages 182-191. Springer-Verlag.

Viegas, F. B., Wattenberg, M., van Ham, F., Kriss, J. and McKeon, M. (2007): ManyEyes: A Site for Visualization at Internet Scale. *IEEE Transactions on Visualization and Computer Graphics*, vol. 13 (no. 6): pages 1121-1128. IEEE Educational Activities Department.

W3C Web Services Architecture Working Group (2004a): *Web Services Architecture* [online]. Available at: http://www.w3.org/TR/ws-arch/#wsdisc. [Accessed 29 March 2010].

W3C Web Services Architecture Working Group (2004b): *Web Services Glossary* [online]. Available at: http://www.w3.org/TR/ws-gloss/. [Accessed 29 March 2010].

W3C Web Services Architecture Working Group (2008): *Extensible Markup Language (XML) 1.0* [online]. Available at: http://www.w3.org/TR/xml/#sec-terminology. [Accessed 29 March 2011].

w3schools (2010): *Web Services Basics* [online]. Available at: http://www.w3schools.com/webservices/. [Accessed 31 March 2010].

w3schools (2011a): *XPath Tutorial* [online]. Available at: http://www.w3schools.com/xpath/default.asp. [Accessed 29 March 2011].

w3schools (2011b): *XSLT Tutorial* [online]. Available at: http://www.w3schools.com/xsl/. [Accessed 29 March 2011].

Wu, C. and Chang, E. (2007): Searching Services "on the Web": A Public Web Services Discovery Approach. In *Proceedings of Third International IEEE Conference on Signal-Image Technologies and Internet-Based System (SITIS '07)*, Shanghai: pages 321-328.

Xu, Z., Martin, P., Powley, W. and Zulkernine, F. (2007): Reputation-Enhanced QoS-based Web Services Discovery. In *Proceedings of IEEE International Conference on Web Services (ICWS 2007)*, Salt Lake City, UT: pages 249-256.

Yeom, G., Yun, T. and Min, D. (2006): A QoS Model and Testing Mechanism for Quality-Driven Web Services Selection. In *Proceedings of The Fourth IEEE Workshop on Software Technologies for Future Embedded and Ubiquitous Systems and the Second International Workshop on Collaborative Computing, Integration, and Assurance (SEUS 2006/WCCIA 2006)*, Gyeongju: pages 199-204.

Yi, J. S., Kang, Y. A., Stasko, J. and Jacko, J. (2007): Toward a Deeper Understanding of the Role of Interaction in Information Visualization. *IEEE Transactions on Visualization and Computer Graphics*, vol. 13 (no. 6): pages 1224-1231. IEEE Educational Activities Department.

Yi, J. S., Kang, Y. A., Stasko, J. and Jacko, J. (2008): Understanding and Characterizing Insights: How Do People Gain Insights Using Information Visualization? In *Proceedings of 2008 Conference on BEyond time and errors: novel evaLuation methods for Information Visualization*, Florence, Italy: pages 1-6. ACM.

Zhuge, H. and Liu, J. (2004): Flexible Retrieval of Web Services. *Journal of Systems and Software*, vol. 70 (no. 1-2): pages 107-116. Elsevier Inc.

Appendix A: Informed Consent Form

NELSON MANDELA METROPOLITAN UNIVERSITY

INFORMATION AND INFORMED CONSENT FORM

RESEARCHER'S DETAILS	
Title of the research project	**Using Information Visualization Techniques to Support Web Service Discovery**
Reference number	
Principal investigator	**Simone Beets**
Contact telephone number (private numbers not advisable)	**041 504 2094**

A. DECLARATION BY OR ON BEHALF OF THE PARTICIPANT		**Initial**
I, the participant and the undersigned	(full names)	

A.1. HEREBY CONFIRM AS FOLLOW		**Initial**
I, the participant was invited to participate in the above-mentioned research project		
that is being undertaken by	Simone Beets	
from	Department of Computing Sciences	
Of the Nelson Mandela Metropolitan University		

A.2 THE FOLLOWING ASPECTS HAVE BEEN EXPLAINED TO ME, THE PARTICIPANT						**Initial**
	Aim	The investigators are studying how IV techniques can be used to support web service discovery The information will be used to/for research purposes				
	Procedures	I understand that I am required to use a system to evaluate different information visualization techniques for web service discovery while the eye-tracking equipment monitors the tasks that I complete and captures where I am looking on the screen				
	Risks	I understand that there are no risks involved in participating in this process				
	Confidentiality	My identity will not be revealed in any discussion, description or scientific publications by the investigators				
	Voluntary participation / refusal / discontinuation	My participation is voluntary	YES	NO		
		My decision whether or not to participate will in no way affect my present or future career/employment/lifestyle	TRUE	FALSE		

	No pressure was exerted on me to consent to participate and I understand that I may withdraw at any stage without penalisation	

	Participation in this study will not result in any additional cost to myself	

I HEREBY VOLUNTARILY CONSENT TO PARTICIPATE IN THE ABOVE-MENTIONED PROJECT:	
Signed/confirmed at on 20	
Signature	Signature of the witness:
	Full name of witness:

Appendix B: Background Questionnaire

1	Gender	Male	Female			
2	Age	18-20	21-29	30-39	40-49	50+
3	Education	Under-Graduate Degree	Honours Degree	Masters Degree	PhD Degree	
4	Occupation	Student	Computing Sciences Staff	Information Technologist (Industry)		
5	Computer Experience (In Years)	3-5	6-9	10+		
6	Computer Expertise	Novice	Intermediate	Expert		
7	Internet Experience (In Years)	2-5	6-9	10+		
8	Search Engine Experience (In Years)	2-5	6-9	10+		
9	Web Service Experience (In Years)	0-1	2-4	5+		
10	Have you used any Information Visualization tools before?	Yes	No			
	If so, then which tools: ______________________________ ______________________________					

Appendix C: First Evaluation Task List

A. List View Tasks

Task 1: Find the Number of Web Services in a Category	
1.1.	Without using the search or filter facilities, how many web services does the *Answers* category have?

Answer: ___________________________________

1.2.	Click the Reset button.

Task 2: View Details of a Web Service	
2.1.	Without using the search facility, browse the list to find the *Yahoo Ads* web service.
2.2.	Double-click on the web service to view its detail window.
2.3.	What is the rating of this service?

Answer: ___________________________________

2.4.	Click the Reset button.

Task 3: Searching for Web Services	
3.1.	Enter *TwitterBrite* in the search box at the top of the screen to search for this web service.
3.2.	Click the *Filter* button.
3.3.	In what category is this service?

Answer: ___________________________________

3.4.	Click the Reset button.

Task 4: Searching and Filtering to Find Web Services	
4.1.	Enter *Yahoo* in the search box at the top of the screen to search for these web services.
4.2.	By the *Rating* filter criteria select Exactly in the dropdown list and select 3.
4.3.	Click the *Filter* button.
4.4.	Write down the names of the web services that meet this criteria?

Answer: ___________________________________

4.5.	Click the Reset button.

Task 5: Bookmarking a Web Service	
5.1.	Browse the list or use the search facility to find the *Facebook Ads* web service. • To search: Enter *Facebook Ads* in the search box at the top of the screen to search for these web services and click the *Filter* button.
5.2.	Double-click the web service to view its detail window.
5.3.	Click the *Bookmark Service* button.
5.4.	Enter a comment *"Use for advertising on Facebook"* in the comment field when prompted to add a comment before bookmarking the service.
5.5.	Click OK.
5.6.	Click the *Home* button to return to the list.
5.7.	What is the rating of the service?

Answer: ________________________________

5.8.	Click the Reset button.

Task 6: Saving a Web Service Search	
6.1.	Enter *Mobile* in the search box at the top of the screen to search for these web services.
6.2.	Click the *Filter* button.
6.3.	Click the *Save Search* button.
6.4.	Clear the search box at the top of the screen with the *Clear Filters* button.
6.5.	Double click on the *Mobile* term in the *Saved Searches* list.
6.6.	How many categories have Mobile web services?

Answer: ________________________________

6.7.	Click the Reset button.

Switch to the Network View Tab

B. Network View Tasks

Task 1: Find the Number of Web Services in a Category	
1.1.	Without using the search or filter facilities, how many web services does the *Shipping* category have?

Answer: ________________________________

1.2.	Click the Reset button.

Task 2: View Details of a Web Service	
2.1.	Without using the search facility, browse the categories to find the *Google Health* web service. (Left-click to expand a category, right-click to contract the category)
2.2.	Left click on the web service to view its detail window.
2.3.	What is the rating of this service?

Answer: ________________________________

2.4.	Click the Reset button.

Task 3: Searching for Web Services	
3.1.	Enter *ProgrammableWeb* in the search box at the top of the screen to search for this web service.
3.2.	Click the search icon to the right of the search box.
3.3.	In what category is this service?

Answer: ________________________________

3.4.	Click the Reset button.

Task 4: Searching and Filtering to Find Web Services	
4.1.	Enter *Google* in the search box at the top of the screen to search for these web services.
4.2.	Click the search icon to the right of the search box.
4.3.	By the *WSDL* filter criteria under Filters select *Only those with WSDL* in the dropdown list.
4.4.	Click the Filter button.
4.5.	Write down the names of the web services that meet these criteria? (Left-click to expand a category, right-click to contract the category)

Answer: ________________________________

4.6.	Click the Reset button.

Task 5: Bookmarking a Web Service	
5.1.	Browse the network view or use the search to find the *PayPal* web service. • To search: Enter *PayPal* in the search box at the top of the screen & click the search icon to search • To Browse: Left-click to expand a category, right-click to contract the category
5.2.	Left click on the web service to view its detail window.
5.3.	Click the *Bookmark Service* button.
5.4.	Enter a comment *"Use for online payment solutions"* in the comment field when prompted to add a comment before bookmarking the service.
5.5.	Click OK.
5.6.	What is the rating of the service?

Answer: ________________________________

5.7.	Click the Reset button.

Task 6: Saving a Web Service Search	
6.1.	Enter *Multimedia* in the search box at the top of the screen to search for these web services.
6.2.	Click the search icon to the right of the search box.
6.3.	Click the *Save Search* button.
6.4.	Click the *X* button to the right of the search box to cancel the search.
6.5.	Double click on the *Multimedia* term in the *Saved Searches* list.
6.6.	How many categories have multimedia web services?

Answer: ________________________________

6.7.	Click the Reset button.

Appendix D: First Evaluation Post-Task Questionnaire: List View

Post-Task Questionnaire – List View									
A. Cognitive load									
1. Mental demand: How mentally demanding were the tasks?									
	Very Low	1	2	3	4	5	6	7	Very High
2. Physical demand: How physically demanding were the tasks?									
	Very Low	1	2	3	4	5	6	7	Very High
3. Temporal demand: How hurried or rushed was the pace of the tasks?									
	Very Low	1	2	3	4	5	6	7	Very High
4. Performance: How successful were you in accomplishing what you were asked to do?									
	Very Low	1	2	3	4	5	6	7	Very High
5. Effort: How hard did you have to work to accomplish your level of performance?									
	Very Low	1	2	3	4	5	6	7	Very High
6. Frustration: How insecure, discouraged, irritated, stressed, and annoyed were you?									
	Very Low	1	2	3	4	5	6	7	Very High

B. Overall satisfaction									
1. Overall, I am satisfied with how easy it is to use the List View									
	Strongly Disagree	1	2	3	4	5	6	7	Strongly Agree
2. Overall, I am satisfied with the List View									
	Strongly Disagree	1	2	3	4	5	6	7	Strongly Agree
3. It was easy to learn to use the List View									
	Strongly Disagree	1	2	3	4	5	6	7	Strongly Agree

4. It was simple to use the List View									
	Strongly Disagree	1	2	3	4	5	6	7	Strongly Agree
C. Usability									
1. I can effectively find web services using the List View									
	Strongly Disagree	1	2	3	4	5	6	7	Strongly Agree
2. I was able to find web services quickly using the List View									
	Strongly Disagree	1	2	3	4	5	6	7	Strongly Agree
3. I was able to efficiently find web services using the List View									
	Strongly Disagree	1	2	3	4	5	6	7	Strongly Agree
4. I became productive quickly using the List View									
	Strongly Disagree	1	2	3	4	5	6	7	Strongly Agree
5. The List View has all functions and capabilities I expect from a web service discovery tool									
	Strongly Disagree	1	2	3	4	5	6	7	Strongly Agree
6. I can effectively browse web service collections using the List View									
	Strongly Disagree	1	2	3	4	5	6	7	Strongly Agree
7. I was able to browse web service collections quickly using the List View									
	Strongly Disagree	1	2	3	4	5	6	7	Strongly Agree
8. I was able to efficiently browse web service collections using the List View									
	Strongly Disagree	1	2	3	4	5	6	7	Strongly Agree
D. General									
1. Identify the most positive aspect of the technique									
2. Identify the most negative aspect of the technique									
3. Please provide any general comments or suggestions for improvement									

Appendix E: First Evaluation Post-Task Questionnaire: Network View

Post-Task Questionnaire – Network View									
E. Cognitive load									
1. Mental demand: How mentally demanding were the tasks?									
	Very Low	1	2	3	4	5	6	7	Very High
2. Physical demand: How physically demanding were the tasks?									
	Very Low	1	2	3	4	5	6	7	Very High
3. Temporal demand: How hurried or rushed was the pace of the tasks?									
	Very Low	1	2	3	4	5	6	7	Very High
4. Performance: How successful were you in accomplishing what you were asked to do?									
	Very Low	1	2	3	4	5	6	7	Very High
5. Effort: How hard did you have to work to accomplish your level of performance?									
	Very Low	1	2	3	4	5	6	7	Very High
6. Frustration: How insecure, discouraged, irritated, stressed, and annoyed were you?									
	Very Low	1	2	3	4	5	6	7	Very High

F. Overall satisfaction									
1. Overall, I am satisfied with how easy it is to use the Network View									
	Strongly Disagree	1	2	3	4	5	6	7	Strongly Agree
2. Overall, I am satisfied with the Network View									
	Strongly Disagree	1	2	3	4	5	6	7	Strongly Agree
3. It was easy to learn to use the Network View									
	Strongly Disagree	1	2	3	4	5	6	7	Strongly Agree

4. It was simple to use the Network View									
	Strongly Disagree	1	2	3	4	5	6	7	Strongly Agree
G. Usability									
1. I can effectively find web services using the Network View									
	Strongly Disagree	1	2	3	4	5	6	7	Strongly Agree
2. I was able to find web services quickly using the Network View									
	Strongly Disagree	1	2	3	4	5	6	7	Strongly Agree
3. I was able to efficiently find web services using the Network View									
	Strongly Disagree	1	2	3	4	5	6	7	Strongly Agree
4. I became productive quickly using the Network View									
	Strongly Disagree	1	2	3	4	5	6	7	Strongly Agree
5. The Network View has all functions and capabilities I expect from a web service discovery tool									
	Strongly Disagree	1	2	3	4	5	6	7	Strongly Agree
6. I can effectively browse web service collections using the Network View									
	Strongly Disagree	1	2	3	4	5	6	7	Strongly Agree
7. I was able to browse web service collections quickly using the Network View									
	Strongly Disagree	1	2	3	4	5	6	7	Strongly Agree
8. I was able to efficiently browse web service collections using the Network View									
	Strongly Disagree	1	2	3	4	5	6	7	Strongly Agree
H. General									
1. Identify the most positive aspect of the technique									
2. Identify the most negative aspect of the technique									
3. Please provide any general comments or suggestions for improvement									

Appendix F: First Evaluation Post-Test Questionnaire

Post-Test Questionnaire

A. General									
1. Which technique will you prefer to use in future?									
	List View	1	2	3	4	5	6	7	Network View
Reason for Selection: ____________________ ____________________									
2. Which technique did you find more complex?									
	List View	1	2	3	4	5	6	7	Network View
Reason for Selection: ____________________ ____________________									
3. Which technique did you find easier to use?									
	List View	1	2	3	4	5	6	7	Network View
Reason for Selection: ____________________ ____________________									
4. Which technique do you think you would need support from a technical person to use?									
	List View	1	2	3	4	5	6	7	Network View
Reason for Selection: ____________________ ____________________									
5. Which technique's various functions did you find were better integrated?									
	List View	1	2	3	4	5	6	7	Network View

Reason for Selection:

____

____

6. Which technique do you think was more consistent?

	List View	1	2	3	4	5	6	7	Network View

Reason for Selection:

____

____

7. Which technique do you think more people would learn to use more quickly?

	List View	1	2	3	4	5	6	7	Network View

Reason for Selection:

____

____

8. Which technique did you find more cumbersome to use?

	List View	1	2	3	4	5	6	7	Network View

Reason for Selection:

____

____

9. Which technique did you feel more confident using?

	List View	1	2	3	4	5	6	7	Network View

Reason for Selection:

____

____

10. Which technique did you feel you needed to learn a lot more before you could get going?

	List View	1	2	3	4	5	6	7	Network View

Reason for Selection:

____

____

B. General
General comments regarding the system:

Appendix G: Second Evaluation Task List

A. Tree View Tasks

Task 1: Find the Number of Web Services in a Category

1.1.	
1.1.1.	Without using the search or filter facilities, how many web services does the *Shipping* category have?

Answer: ________________________________

1.1.2.	Click the button and press F10.

1.2.	
1.2.1.	Without using the search or filter facilities, how many web services does the *Food* category have?

Answer: ________________________________

1.2.2.	Click the button and press F10.

Task 2: View the Details of a Web Service

2.1.	
2.1.1.	Without using the search facility, browse the categories to find the *AccuWeather* web service. (Double Left-click to expand a category, right-click to contract the category)
2.1.2.	Double Left click on the web service to view its detail window.
2.1.3.	What is the rating of this service?

Answer: ________________________________

2.1.4.	Click the button and press F10.

2.2.	
2.2.1.	Without using the search facility, browse the categories to find the *Interfax* web service. (Double Left-click to expand a category, right-click to contract the category)

2.2.2.	Double Left click on the web service to view its detail window.
2.2.3.	What date was the web service uploaded?

Answer: ___________________________________

2.2.4.	Click the  button and press F10.

Task 3: Searching for Web Services

3.1.	
3.1.1.	Enter *ProgrammableWeb* in the search box at the top of the screen to search for this web service.
3.1.2.	In what category is this service?

Answer: ___________________________________

3.1.3.	Click the  button and press F10.

3.2.	
3.2.1.	Enter *LiveTimer* in the search box at the top of the screen to search for this web service.
3.2.2.	In what category is this service?

Answer: ___________________________________

3.2.3.	Click the  button and press F10.

Task 4: Searching and Filtering to Find Web Services

4.1.	
4.1.1.	Enter *StrikeIron* in the search box at the top of the screen to search for these web services.
4.1.2.	In the *Price* filter criteria select *Paid* in the dropdown list.
4.1.3.	Click the *Filter* button.
4.1.4.	Write down the name(s) of the web service(s) that meet these criteria? (Double Left-click to expand a category, right-click to contract the category)

Answer: ___________________________________

4.1.5.	Click the  button and press F10.

4.2.	
4.2.1.	Enter *Yahoo* in the search box at the top of the screen to search for these web services.
4.2.2.	In the *Data Format* filter criteria under Filters select *HTML* in the dropdown list.
4.2.3.	Click the *Filter* button.
4.2.4.	Write down the name(s) of the web service(s) that meet these criteria? (Double Left-click to expand a category, right-click to contract the category)

Answer: ________________________________

4.2.5.	Click the  button and press F10.

Task 5: Bookmarking a Web Service	
5.1.	Browse the tree view or use the search to find the *Photobucket* web service. • To search: Enter *Photobucket* in the search box at the top of the screen • To Browse: Double Left-click to expand a category, right-click to contract the category
5.2.	Double Left click on the web service to view its detail window.
5.3.	Click the *Bookmark* button.
5.4.	Enter a comment *"Use for photo sharing"* in the comment field when prompted to add a comment before bookmarking the service.
5.5.	Click *OK*.
5.6.	What is the rating of the service?

Answer: ________________________________

5.7.	Click the  button and press F10.

Task 6: Saving a Web Service Search	
6.1.	Enter *Multimedia* in the search box at the top of the screen to search for these web services.
6.2.	Click the *Save Search* button.
6.3.	Click the *X* button to the right of the search box to cancel the search.
6.4.	Double click on the *Multimedia* term in the *Saved Searches* list.
6.5.	How many categories have multimedia web services?

Answer: ________________________________

6.6.	Click the  button.
6.7.	Switch to the Network View Tab and press F10.

B. Network View Tasks

Task 1: Find the Number of Web Services in a Category

1.1.	
1.1.1.	Without using the search or filter facilities, how many web services does the *Wiki* category have?

Answer: ________________________________

1.1.2.	Click the button and press F10.

1.2.	
1.2.1.	Without using the search or filter facilities, how many web services does the *Weather* category have?

Answer: ________________________________

1.2.2.	Click the button and press F10.

Task 2: View the Details of a Web Service

2.1.	
2.1.1.	Without using the search facility, browse the categories to find the *Google Health* web service. (Double Left-click to expand a category, right-click to contract the category)
2.1.2.	Double Left click on the web service to view its detail window.
2.1.3.	What is the rating of this service?

Answer: ________________________________

2.1.4.	Click the button and press F10.

2.2.	
2.2.1.	Without using the search facility, browse the categories to find the *Yahoo Ads* web service. (Double Left-click to expand a category, right-click to contract the category)
2.2.2.	Double Left click on the web service to view its detail window.
2.2.3.	What date was the web service uploaded?

Answer: ________________________________

2.2.4.	Click the button and press F10.

Task 3: Searching for Web Services

3.1.	
3.1.1.	Enter *TwitterBrite* in the search box at the top of the screen to search for this web service.
3.1.2.	In what category is this service?

Answer: ___________________________________

3.1.3.	Click the button and press F10.

3.2.	
3.2.1.	Enter *Swoogle* in the search box at the top of the screen to search for this web service.
3.2.2.	In what category is this service?

Answer: ___________________________________

3.2.3.	Click the button and press F10.

Task 4: Searching and Filtering to Find Web Services

4.1.	
4.1.1.	Enter *Google* in the search box at the top of the screen to search for these web services.
4.1.2.	In the *WSDL* filter criteria under Filters select *Only those with WSDL* in the dropdown list.
4.1.3.	Click the *Filter* button.
4.1.4.	Write down the name(s) of the web service(s) that meet these criteria? (Double Left-click to expand a category, right-click to contract the category)

Answer: ___________________________________

4.1.5.	Click the button and press F10.

4.2.	
4.2.1.	Enter *Adobe* in the search box at the top of the screen to search for these web services.
4.2.2.	In the *Protocol* filter criteria under Filters select *SOAP* in the dropdown list.
4.2.3.	Click the *Filter* button.
4.2.4.	Write down the name(s) of the web service(s) that meet these criteria? (Double Left-click to expand a category, right-click to contract the category)

Answer: ___________________________________

4.2.5.	Click the button and press F10.

Task 5: Bookmarking a Web Service	
5.1.	Browse the network view or use the search to find the *SecondLife* web service. • To search: Enter *SecondLife* in the search box at the top of the screen • To Browse: Double Left-click to expand a category, right-click to contract the category
5.2.	Double Left click on the web service to view its detail window.
5.3.	Click the *Bookmark* button.
5.4.	Enter a comment *"Use for web page maps"* in the comment field when prompted to add a comment before bookmarking the service.
5.5.	Click *OK*.
5.6.	What is the rating of the service?

Answer: ___________________________________

5.7.	Click the button and press F10.

Task 6: Saving a Web Service Search	
6.1.	Enter *Mapping* in the search box at the top of the screen to search for these web services.
6.2.	Click the *Save Search* button.
6.3.	Click the *X* button to the right of the search box to cancel the search.
6.4.	Double click on the *Mapping* term in the *Saved Searches* list.
6.5.	How many categories have mapping web services?

Answer: ___________________________________

6.6.	Click the button and press F10.

Appendix H: Second Evaluation Post-Task Questionnaire: Tree View

Post-Task Questionnaire – Tree View									
A. Cognitive load									
1. Mental demand: How mentally demanding were the tasks?									
	Very Low	1	2	3	4	5	6	7	Very High
2. Physical demand: How physically demanding were the tasks?									
	Very Low	1	2	3	4	5	6	7	Very High
3. Temporal demand: How hurried or rushed was the pace of the tasks?									
	Very Low	1	2	3	4	5	6	7	Very High
4. Performance: How successful were you in accomplishing what you were asked to do?									
	Very Low	1	2	3	4	5	6	7	Very High
5. Effort: How hard did you have to work to accomplish your level of performance?									
	Very Low	1	2	3	4	5	6	7	Very High
6. Frustration: How insecure, discouraged, irritated, stressed, and annoyed were you?									
	Very Low	1	2	3	4	5	6	7	Very High

B. Overall satisfaction									
1. Overall, I am satisfied with how easy it is to use the Tree View									
	Strongly Disagree	1	2	3	4	5	6	7	Strongly Agree
2. Overall, I am satisfied with the Tree View									
	Strongly Disagree	1	2	3	4	5	6	7	Strongly Agree
3. It was easy to learn to use the Tree View									
	Strongly Disagree	1	2	3	4	5	6	7	Strongly Agree

4. It was simple to use the Tree View									
	Strongly Disagree	1	2	3	4	5	6	7	Strongly Agree
C. Usability									
1. I can effectively find web services using the Tree View									
	Strongly Disagree	1	2	3	4	5	6	7	Strongly Agree
2. I was able to find web services quickly using the Tree View									
	Strongly Disagree	1	2	3	4	5	6	7	Strongly Agree
3. I was able to efficiently find web services using the Tree View									
	Strongly Disagree	1	2	3	4	5	6	7	Strongly Agree
4. I became productive quickly using the Tree View									
	Strongly Disagree	1	2	3	4	5	6	7	Strongly Agree
5. The Tree View has all functions and capabilities I expect from a web service discovery tool									
	Strongly Disagree	1	2	3	4	5	6	7	Strongly Agree
6. I can effectively browse web service collections using the Tree View									
	Strongly Disagree	1	2	3	4	5	6	7	Strongly Agree
7. I was able to browse web service collections quickly using the Tree View									
	Strongly Disagree	1	2	3	4	5	6	7	Strongly Agree
8. I was able to efficiently browse web service collections using the Tree View									
	Strongly Disagree	1	2	3	4	5	6	7	Strongly Agree
D. Overview									
1. The fisheye strategy in the Tree View was useful									
	Strongly Disagree	1	2	3	4	5	6	7	Strongly Agree
E. General									
1. Identify the most positive aspect of the technique									
2. Identify the most negative aspect of the technique									

3. Please provide any general comments or suggestions for improvement

Appendix I: Second Evaluation Post-Task Questionnaire: Network View

Post-Task Questionnaire – Network View

F. Cognitive load									
1. Mental demand: How mentally demanding were the tasks?									
	Very Low	1	2	3	4	5	6	7	Very High
2. Physical demand: How physically demanding were the tasks?									
	Very Low	1	2	3	4	5	6	7	Very High
3. Temporal demand: How hurried or rushed was the pace of the tasks?									
	Very Low	1	2	3	4	5	6	7	Very High
4. Performance: How successful were you in accomplishing what you were asked to do?									
	Very Low	1	2	3	4	5	6	7	Very High
5. Effort: How hard did you have to work to accomplish your level of performance?									
	Very Low	1	2	3	4	5	6	7	Very High
6. Frustration: How insecure, discouraged, irritated, stressed, and annoyed were you?									
	Very Low	1	2	3	4	5	6	7	Very High

G. Overall satisfaction									
1. Overall, I am satisfied with how easy it is to use the Network View									
	Strongly Disagree	1	2	3	4	5	6	7	Strongly Agree
2. Overall, I am satisfied with the Network View									
	Strongly Disagree	1	2	3	4	5	6	7	Strongly Agree
3. It was easy to learn to use the Network View									
	Strongly Disagree	1	2	3	4	5	6	7	Strongly Agree

4. It was simple to use the Network View									
	Strongly Disagree	1	2	3	4	5	6	7	Strongly Agree
H. Usability									
1. I can effectively find web services using the Network View									
	Strongly Disagree	1	2	3	4	5	6	7	Strongly Agree
2. I was able to find web services quickly using the Network View									
	Strongly Disagree	1	2	3	4	5	6	7	Strongly Agree
3. I was able to efficiently find web services using the Network View									
	Strongly Disagree	1	2	3	4	5	6	7	Strongly Agree
4. I became productive quickly using the Network View									
	Strongly Disagree	1	2	3	4	5	6	7	Strongly Agree
5. The Network View has all functions and capabilities I expect from a web service discovery tool									
	Strongly Disagree	1	2	3	4	5	6	7	Strongly Agree
6. I can effectively browse web service collections using the Network View									
	Strongly Disagree	1	2	3	4	5	6	7	Strongly Agree
7. I was able to browse web service collections quickly using the Network View									
	Strongly Disagree	1	2	3	4	5	6	7	Strongly Agree
8. I was able to efficiently browse web service collections using the Network View									
	Strongly Disagree	1	2	3	4	5	6	7	Strongly Agree
I. Overview									
1. The overview in the Network View was useful									
	Strongly Disagree	1	2	3	4	5	6	7	Strongly Agree
J. General									
1. Identify the most positive aspect of the technique									
2. Identify the most negative aspect of the technique									

3. Please provide any general comments or suggestions for improvement

Appendix J: Second Evaluation Post-Test Questionnaire

Post-Test Questionnaire

C. General									
1. Which technique will you prefer to use in future?									
	Tree View	1	2	3	4	5	6	7	Network View
Reason for Selection:									
2. Which technique did you find more complex?									
	Tree View	1	2	3	4	5	6	7	Network View
Reason for Selection:									
3. Which technique did you find easier to use?									
	Tree View	1	2	3	4	5	6	7	Network View
Reason for Selection:									
4. Which technique do you think you would need support from a technical person to use?									
	Tree View	1	2	3	4	5	6	7	Network View
Reason for Selection:									
5. Which technique's various functions did you find were better integrated?									
	Tree View	1	2	3	4	5	6	7	Network View

Reason for Selection:

____

____

6. Which technique do you think was more consistent?

	Tree View	1	2	3	4	5	6	7	Network View

Reason for Selection:

____

____

7. Which technique do you think more people would learn to use more quickly?

	Tree View	1	2	3	4	5	6	7	Network View

Reason for Selection:

____

____

8. Which technique did you find more cumbersome to use?

	Tree View	1	2	3	4	5	6	7	Network View

Reason for Selection:

____

____

9. Which technique did you feel more confident using?

	Tree View	1	2	3	4	5	6	7	Network View

Reason for Selection:

____

____

10. Which technique did you feel you needed to learn a lot more before you could get going?

	Tree View	1	2	3	4	5	6	7	Network View

Reason for Selection:

____

____

11. Which technique do you think is more appropriate to visualize this web service collection (3562 nodes)?

	Tree View	1	2	3	4	5	6	7	Network View

Reason for Selection:

____

____

D. General

General comments regarding the system: