Using Information Visualisation to Support Visual Web Service Discovery

Simone Beets and Janet Wesson

Department of Computing Sciences , Nelson Mandela Metropolitan University
P. O. Box 77000, Port Elizabeth, 6031, South Africa
{*Simone.Beets2 @nmmu.ac.za, Janet.Wesson*}@nmmu.ac.za

This paper presents the design and evaluation of SerViz – a tool to support the visualisation of web service collections. The increasing number of web services available on the Internet highlights the need for an effective method of discovering these web services. Existing web service discovery methods do not, however, effectively support the user in finding suitable web services. Two information visualisation (IV) techniques, namely a network and tree IV technique, were used to visualise a large web service collection. A user study was conducted to compare these IV techniques. Participants provided high ratings for overall satisfaction for both IV techniques and low ratings for perceived cognitive load. Participants found both IV techniques easy to use and easy to learn. The results of the user study show that SerViz supports effective visual web service discovery. While the tree IV technique was faster for browsing for web services, the network IV technique was faster for searching.

Web Service Discovery, Web Service Collection, Information Visualisation, Prototype, User Study, Results.

1. INTRODUCTION

As the number of available web services on the Internet increases, finding suitable web services becomes a challenging task (Sabou and Pan 2007). Developing an effective method of discovering these services is thus essential. Web service discovery is concerned with searching for web services that are specific to a user's requirements.

Existing web service discovery methods do not effectively support the user in finding suitable web services (Beets and Wesson 2010). The current techniques provide textual lists that the user is required to explore and manually evaluate to select appropriate web services. Minimal information is made available to distinguish between web services of similar functionality. Web service discovery can thus become time-consuming and ineffective.

Information visualisation (IV) is widely accepted as 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). This paper investigates the use of IV to support visual web service discovery. This paper describes the results of a user study that was conducted comparing two IV techniques used to visualise the ProgrammableWeb web service collection.

Related work involving web service discovery and IV is discussed in Section 2. The design and implementation of SerViz, a prototype for interactive visualisation of web service collections, is described in Section 3 and 4. Section 5 discusses the method and results of the user study. Design recommendations for visual web service discovery are then presented in Section 6. The paper concludes by identifying the contribution of this paper and future work to be completed in Section 7.

2. RELATED WORK

This section provides a background to related work by discussing web service discovery and IV.

2.1. Web Service Discovery

The main methods of web service discovery are the Universal Description, Discovery and Integration (UDDI) standard, general-purpose search engines and web service publication sites (Beets and Wesson 2010). Several web service discovery criteria were identified including the provision of functional and non-functional web service properties, an effective graphical user interface (GUI), classification, ranking and sorting capability and the need for search and browse facilities. The main methods of web service discovery were

compared with these criteria to identify problems with existing web service discovery methods. The key problems identified were problems with searching and browsing and the poor presentation of web service collections.

Semantic web service (SWS) discovery is based on formalising semantic web service descriptions (Le Duy *et al.* 2010). Several semantic web service discovery approaches exist. The main SWS discovery approach is OWL-S, which uses ontologies and web service profiles to find suitable services. Web Service Modelling Ontology (WSMO) approaches base web service discovery on fulfilling a user's web service requirements. Semantic Annotations for WSDL (SAWSDL) and WSDL-S approaches provide semantic annotations of the Web Services Description Language (WSDL) of a web service to assist in discovering web services. These approaches focus on the accuracy of the web service discovery results. The focus of this research is on visualising web service collections to support web service discovery. This research could be used to enhance the user experience of semantic web service discovery in future. The main research question is how information visualisation can be used to effectively support web service discovery.

2.2. Information Visualisation

This paper proposes using information visualisation (IV) to address the shortcomings of the existing web service discovery methods. IV taxonomies have been identified to support the selection of suitable IV techniques (Shneiderman 1996). Thus, the data structure of the web service collection will be used to determine what IV techniques can be used for visual web service discovery. An analysis of several web service collections was conducted to identify the data structure of existing web service collections (Beets and Wesson 2011). Publication sites are becoming a popular method of web service discovery as public web service registries were recently developed to provide centralised domain-specific web service registries in the Life Sciences domain (Bhagat *et al.* 2010). Three publication sites, namely RemoteMethods (www.remotemethods.com), ProgrammableWeb (www.programmableweb.com) and Service-Finder (www.service-finder.eu), were analysed to determine each site's data structure. It was identified that a web service collection typically has a hierarchical or network data structure. A hierarchical web service collection contains web services that are classified in a single category, whereas a network web service collection contains web services that may be classified in one or more categories.

The node-link network IV technique was selected as the most appropriate IV technique to support visual web service discovery as hierarchical techniques are too limited to be used for visual web service discovery. The implementation of a network IV technique was discussed in an earlier paper (Beets and Wesson 2011). A pilot study was conducted comparing the network IV technique with a list technique. Ninety percent of the participants preferred the network IV technique for visual web service discovery. The network IV technique was also faster for browsing. Participants of the pilot study were highly satisfied with using the network IV technique to support visual web service discovery.

Several usability problems were, however, identified with the network IV technique in the pilot study (Beets and Wesson 2011). The key problems included occlusion due to the animation effect of the layout used for the network IV technique and difficulty in distinguishing between nodes, i.e. between a collection, category and web service node. This motivated a need to implement an alternative IV technique. The alternative IV technique selected was required to address the limitations of the network IV technique including animation issues, node occlusion and the difficulty in distinguishing between nodes. The selected alternative IV technique also needed to be supported by Prefuse, the toolkit used to implement SerViz. The node-link tree IV technique used to visualise hierarchical data was selected as the most suitable alternative IV technique based on these requirements. The tree IV technique was more structured than the network IV technique. This paper focuses on the design of the tree IV technique and a full user study to compare the network and tree IV techniques.

3. SERVIZ PROTOTYPE

The network and tree IV techniques were implemented in SerViz using Prefuse. This section describes the design of the SerViz prototype.

3.1. Data Source

The ProgrammableWeb web service collection was selected to be visualised by SerViz as it is a dynamic web service collection, which is growing constantly ($n=3506$ web services). Additionally, the ProgrammableWeb publication site provides an Application Programming Interface (API) that allows access to its web service collection. The retrieved web service collection data was stored in an Extensible Markup Language (XML) document.

3.2. Functionality

Previous research identified several functions that needed to be supported in SerViz (Beets and Wesson 2011). These functions were adapted from the visual information seeking mantra identified by Shneiderman (1996). These functions included providing an overview of the web service collection, zooming in or out of the collection by expanding or contracting categories, searching and filtering to find web services, viewing web service details, undoing and redoing graph navigation actions, bookmarking a web service and saving a search for web services.

3.3. Visualisation Techniques

The network and tree IV techniques were implemented as views in SerViz. The network view represents categories and web services in the web service collection in terms of nodes and edges. A collection root node, representing the web service collection, connects to category nodes within the collection. Each category node connects to the web services within the category with an edge.

The tree view represents the web service collection in terms of branches and leaf nodes. The web service collection node represents the root node which connects to category branches. Each category branch connects to the web service leaf nodes contained in the category.

4. SERVIZ IMPLEMENTATION

The network and tree views were then implemented in SerViz supporting the functions discussed in Section 3.2.

4.1. Implementation Tool

SerViz was developed using the Prefuse data visualisation toolkit (Heer *et al.* 2005) and implemented in Java. Prefuse provides enhanced support for interactive data visualisation.

4.2. Data

Relevant web service attributes, such as the web service name and description, were extracted from the XML document containing the ProgrammableWeb web service collection using Extensible Style-sheet Language Transformations (XSLT) and XPath. The resulting XML document was then converted to GraphML (GML) and TreeML (TML) files, which are used by Prefuse to visualise network and hierarchical data structures respectively.

4.3. Visualisation Techniques

4.3.1. Network View

A node-link IV technique to visualise network data was selected as the most appropriate IV technique to visualise web service collections. Modifications were implemented in the network view to address the usability issues identified in the pilot study and discussed in Section 2.2 (Beets and Wesson 2011). These modifications include the following:

- A level of transparency was assigned to the nodes to overcome node occlusion.
- Colour-encoding changes were made to assist users in distinguishing between collection, category and web service nodes. A collection node is shown in black, a category node in blue and a web service node in yellow to support the blue-yellow contrast discussed by Heer and Boyd (2005). The blue-yellow contrast also considers colour-blind users. A grey border is assigned to a selected or hovered over node to allow a user to be able to identify the type of node selected. When searching, web services relevant to the search change to green and the categories that contain these web services remain blue.
- The network layout makes use of a force-directed layout (FDL) to position the nodes in the collection. The FDL continuously repositions the nodes due to the expand / contract ability of the category nodes and so the layout results in an animation effect. An animation timer was implemented to pause the node repositioning of the FDL when the percentage of overlapping nodes is less than a certain threshold, in this case 20 per cent. The animation is forced to continue when a user performs graph actions that require node repositioning, such as expanding a category.

The collection node is positioned in the centre of the display. The category nodes surround the collection node and are sized according to the number of web services in each category. Web services are initially hidden to avoid on-screen clutter. The size of a category node represents the number of web services in the category.

The network view is shown in Figure 1 illustrating browsing and searching for the **Google Health** web service. When searching for the **Google Health** web service, the user can search for **Google** web services and expand the **Medical** category. The relevant web service(s) will be displayed and the user can view additional web service information by hovering over the web service, or as in Figure 1, by double-clicking on the web service. Irrelevant categories and web services are de-saturated to a light-grey similar to Perer and Shneiderman's Social Action tool (2006).

A user can use the filters to refine a search or filter separately to find web services. A user can filter on various web service properties including web service provider, data format, upload date, price, protocol, user rating and whether the web service provides a WSDL file.

Figure 1: Searching and browsing using the updated network view

The network view provides a minimised interactive overview to assist a user with graph navigation especially if many categories are expanded in the network. A user can also bookmark a service and save a search for web services. A toolbar is provided above the graph display to allow the user to return to the original display (Home), undo and redo graph actions such as expanding / contracting categories, zooming and panning and expanding relevant categories in a search. In addition to the animation timer, a user can pause the animation using the pause button in the toolbar. A reset button is provided for evaluation purposes to reset the network, the search and the filters.

4.3.2. Tree View

A node-link IV technique to visualise hierarchical data was selected as the most suitable alternative IV technique to visualise web service collections. This technique was implemented as an additional view in SerViz.

The tree view was designed and implemented to support the main system functions described in Section 3.2. 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 Figure 2). Similar to the network view, the web services are also initially hidden.

Instead of providing an Overview in this view, the tree view uses a fisheye strategy (Shneiderman 1996; Song *et al.* 2007). Several variations of the fisheye strategy exist (Song *et al.* 2011), but a

fisheye distortion technique that magnifies nodes as the cursor hovers over the nodes was selected for the tree view. 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 2: Searching and browsing using the tree view

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.

An example of searching and browsing to find web services is shown in Figure 2. 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. The user can 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 similar to the network view.

5. USER STUDY

A user study was conducted to compare the network and tree views implemented in SerViz. The method and results of the user study are discussed in this section.

5.1. Evaluation Method

The user study included six tasks evaluating the functions supported by SerViz. The user study used a within subjects design to measure preference between the views. The usability metrics that were captured included effectiveness, efficiency and user satisfaction. Eye-tracking was

also used in this user study. Twenty participants completed the user study using SerViz.

5.1.1. Evaluation Objectives

The objective of this study was to compare the network and tree views to identify which IV technique is the more suitable IV technique for visual web service discovery.

5.1.2. Participants

The participant sample consisted of twenty (16 male, four female) post-graduate students and staff from the Department of Computing Sciences at the Nelson Mandela Metropolitan University (NMMU). Participants were in the age range of 21 – 50+ years. The user study took approximately an hour to complete as participants were provided with training tasks for the first four evaluation tasks.

All participants had at least six years experience using a computer. Seventy-five percent of participants had used an IV tool prior to taking part in the study and 55 percent of participants had at least two years experience using web services.

5.1.3. Evaluation Metrics

Effectiveness, efficiency and user satisfaction were measured for each task. Effectiveness was measured by task success, i.e. whether the participant could complete the task or not. Efficiency was measured by task completion in combination with the time taken to complete the task. Post-task and post-test questionnaires were used to measure user satisfaction.

5.1.4. Questionnaires

User satisfaction was measured using a post-task questionnaire, measuring cognitive load using a 7-point semantic differential scale and a 7-point Likert scale, and a post-test questionnaire. The post-task questionnaire was provided to each participant after each evaluation. The post-test questionnaire was provided to each participant measuring preference between the two views and was completed at the end of the evaluation.

The questionnaires that were used for the user study were adapted from the Computer System Usability Questionnaire (CSUQ) and the NASA-TLX form. The NASA-TLX form was used for measuring workload to determine if the views required any additional cognitive load. Standard usability was measured using the CSUQ. A question was added to the post-task questionnaire relating 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.

5.1.5. Experimental Setup

The Usability Lab in the Department of Computing Sciences at NMMU was used for the user study.

This lab comprises an evaluation and a control room which are separated by one-way glass. The tasks were completed on the Tobii T60 Eye Tracker. The eye tracker was used to track the eye gaze of each participant. Two tests were created using the software, each test representing an evaluation of a single view. The participants were guided through the evaluation using instruction screens.

5.1.6. Tasks

Two similar but not identical sets of tasks were provided to each participant to complete using both views. These tasks were derived from the functional requirements described in Section 3.2. Browsing, searching, bookmarking a web service and saving a search for web services were determined to be the most important web service discovery tasks, based on the visual information seeking mantra identified by Shneiderman (1996) and adapted for web service discovery. 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 six tasks for each view were the following:

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

5.1.7. Procedure

Each participant completed an evaluation for the network and tree views. The tasks were completed one task at a time. The learnability effect was avoided using counterbalancing. The order in which the views were evaluated was swapped.

5.2. 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 per cent success rates for Task 1, Task 3 and Task 6 while the network view received 100 per cent success rates for 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.

5.2.1. Effectiveness

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 provided.

(i) Network View

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

Task 4 involved searching and filtering to find web services. The question required the participant to write down the names of the web services that met the search and filter criteria. Only one participant answered incorrectly. 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 find the number of web services within the category but some participants expanded the category and counted manually. Two of the participants who counted manually counted incorrectly leading to the 90 per cent 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. 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.

(ii) Tree View

The tree view received 100 per cent success rates for three of the six tasks (Task 1, Task 3 and Task 6). The other three tasks received 95 per cent success rates (Task 2, Task 4 and Task 5). Task 2 involved browsing for a web service without using the search or filter facility and without informing the participants in which category the web service was 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 4 in the network view answered incorrectly for Task 4 in the tree view. Task 5 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 closed the web service's details dialog and accidentally opened the dialog of another web service.

The participants completed most of the tasks without difficulty. The high level of accuracy for both the network and tree view demonstrates that both views effectively supported visual web service discovery.

5.2.2. Efficiency

The time for each task was calculated from the start of the task until the Reset button was selected. Table 1 displays the task times. The tree view was significantly faster for browsing (Tasks 1 and 2). The network view was significantly faster for searching, and faster for 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 1.

Table 1: Task times in seconds with Wilcoxon test statistics and p-values (n=20)

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

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 FDL to place the category nodes and is not sorted. The network view may be faster for searching due to the entire network (collection and category nodes) being visible on the screen, whereas the categories in the tree view cannot all fit on the screen.

5.3. Satisfaction Results

Satisfaction results were captured using the post-task and post-test questionnaires. The satisfaction in the post-task questionnaire was divided into four sections, namely cognitive load, overall satisfaction, usability and overview.

5.3.1. Workload Results

The results were similar for both views for cognitive load except that participants found that the network view required significantly more effort (Figure 3).

The difference was statistically significant for the Effort item ($Z=2.45$; $p=0.01$).

Figure 3: Cognitive load using a 7-point semantic differential scale ($n=20$)

5.3.2. Overall Satisfaction Results

The tree view received slightly higher ratings in three of the four questions for overall satisfaction as shown in Figure 4. There was no statistical significant difference between the two views for the overall satisfaction ratings, which were both very high (>5.5).

Figure 4: Overall satisfaction using a 7-point Likert scale ($n=20$)

5.3.3. Usability Results

The usability ratings were generally high (>5.0), with the majority in favour of the tree view except for question 5 (see Figure 5). The difference was statistically significant for question 2 ($Z=2.53$; $p=0.01$). The participants found the tree view significantly faster for finding web services which supports the efficiency results (Question 2).

5.3.4. Overview

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. Participants considered the overview in the network view (mean = 5.65) to be more useful than the fisheye strategy (mean = 4.5) in the tree view. The mean rating for the usefulness of the overview was significantly higher for the network view ($p=0.01$). Post-task questionnaires were also

used to capture participants' comments regarding each view.

Figure 5: Usability ratings using a 7-point Likert scale ($n=20$)

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 2 where n indicates the number of participants who made similar comments. Five (25 per cent) participants found the network view to be easy to learn while four (20 per cent) participants found it easy to find web services with the network view. The most negative aspect of the network view was that nine (45 per cent) participants found it difficult to browse to find a category because of the network view not being sorted.

Table 2: 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

Nine (45 per cent) participants found the alphabetical ordering of the tree view to be the most positive aspect of this view (Table 3). Seven (35 per cent) participants found the tree view easy to use.

Table 3: 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 helps in identifying where I am.	3

Nine (45 per cent) participants found the fisheye strategy distracting. Seven (35 per cent) participants found the tree view to be cluttered.

5.4. Post-Test Satisfaction Results

Post-test satisfaction results are shown in Figure 6. The results for the post-test questions were mixed but slightly favoured the tree view. The participants slightly preferred the tree view to visualise web service collections (mean = 3.65). Nine participants preferred the network view and one participant could not choose between the two views. The participants preferred the network view as the more appropriate technique to visualise the sample web service collection of 3562 nodes (mean = 4.25). The difference was statistically significant for question 8 ($Z=2.12$; $p=0.03$). The participants found the network view significantly more cumbersome to use (mean = 5.05, $p=0.03$).

Figure 6: Post-test satisfaction results using a 7-point Likert scale (n=20)

General comments in the post-test questionnaire are shown in Table 4 where n indicates the number of participants who made similar comments. Most of the positive comments received in the post-test questionnaire related to participants noting that SerViz provided good support for finding web services ($n=11$). Six (30 per cent) participants made positive comments regarding the network view and five (25 per cent) participants made positive comments regarding the tree view.

Table 4: 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. The network view is also more interactive and fun to use.	6
3. Like tree view's hierarchy as it sorts categories in alphabetical order.	5

5.5. Eye-Tracking Results

Heat maps were taken for the first task of each view. 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 7. The heat map was taken for fifteen participants as the other participants' graphs were not zoomed into 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.

Figure 7: Heat map of participants browsing the web service collection in the network view (n=15)

The heat map taken for the first task of the tree view is shown in Figure 8. The participants immediately fixated on the required category due to 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 feature.

Figure 8: Heat map of participants browsing the web service collection in the tree view (n=19)

5.6. Discussion

Both views received high ratings for overall satisfaction and usability and low ratings for cognitive load. The results show that the tree view was slightly preferred over the network view (Figure 6). Ten participants (50 per cent) preferred the tree view, while nine participants preferred the network view and one participant could not choose between the two. The effectiveness results generally supported both views.

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 5). 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 as it required the participants to only browse through the relevant categories and web services. Participants found the network view to be more cumbersome to use (Figure 6) and required more effort (Figure 3). This could be due to the fact that the network view does not provide any sorting when browsing categories and web services. A limitation of this user study is that no weighting, in terms of task importance, was assigned to the different evaluation tasks.

Based on the screen recordings of the user study and the user study 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;

- the nodes should be resized to display the number of search items when searching and filtering in the network view.

From the results of the user study, both views received similar positive results. Due to the similarity in results, it may be useful in future to provide both views to allow the user to select which view s/he would like to use.

6. DESIGN RECOMMENDATIONS

From this user study, certain design recommendations can be made when using an IV technique to visualise a web service collection:

- ***It is recommended to provide both views to the user to select which view s/he would like to use.***

The web service collection type will need to be considered however, as the tree view only supports hierarchical web service collections. The tree view is also limited to the maximum number of categories and web services that can be displayed in the available screen space.

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

The animation effect caused by the repositioning of nodes in the network view was found to be distracting to participants. Also, the fisheye strategy provided animation that made it difficult for participants to select categories, with some participants 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 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 study commented that the tree view did not make effective usage of available screen space. If a tree IV technique is used to visualise large 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 user study showed that browsing was easier and faster with the tree view. This was because the tree view provided alphabetical sorting of the categories and 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 and 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 contained 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, for example by alphabetical ordering (A→G, H→N, O→U, V→Z).

- **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 user study found the fisheye strategy to be distracting.

7. CONCLUSION

This paper discussed SerViz – an interactive tool to support visual web service discovery. Two IV techniques, namely the network and tree IV techniques were implemented in SerViz to determine which IV technique better supports visual web service discovery.

The user study determined that IV techniques can be used to effectively support web service discovery. The overall satisfaction of both IV techniques was high and low ratings were obtained for cognitive load. These results demonstrate that SerViz is easy to learn, minimises frustration and supports finding suitable web services. The results of the user study show that SerViz supports effective visual web service discovery. SerViz can be used to manipulate large web service collections in a simple and easy to use manner. Several design recommendations were made for effective visual web service discovery. Future work could extend this research to a multi-touch surface platform to investigate collaborative visual web service discovery incorporating more web service collections.

8. ACKNOWLEDGMENTS

Acknowledgements are due to the NMMU / Telkom Centre of Excellence for funding this research.

9. REFERENCES

- Beets, S. and Wesson, J. (2010) Can Information Visualization Techniques be used to Support Web Service Discovery? In *Proceedings of SATNAC*, Stellenbosch, South Africa, pp. 1-2.
- Beets, S. and Wesson, J. (2011) SerViz: A Tool for Interactive Visualization of Web Service Collections. In *Proceedings of SATNAC*, East London, South Africa, pp. 387-392.
- Beets, S. and Wesson, J. (2011) Using Information Visualization to Support Web Service Discovery. In *Proceedings of SAICSIT*, Cape Town, South Africa, pp. 11-20.
- Bhagat, J., et al. (2010) BioCatalogue: A Universal Catalogue of Web Services for the Life Sciences. *Nucleic Acids Research*, pp. W689-W694.
- Fekete, J., Wijk, J. J., Stasko, J. and North, C. (2008) *The Value of Information Visualization*, in *Information Visualization: Human-Centered Issues and Perspectives*. Springer-Verlag Berlin, Heidelberg. pp. 1-18.
- Heer, J. and Boyd, D. (2005) Vizster: Visualizing Online Social Networks. In *Proceedings of InfoVis*, pp. 32-39. IEEE Computer Society.
- 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*, pp. 421-430. ACM.
- Le Duy, N., Kirchberg, M. and Kanagasabai, R. (2010) Review of Semantic Web Service Discovery Methods. In *Proceedings of 2010 6th World Congress on Services (SERVICES-1)*, Miami, Florida, pp. 176-177.
- Perer, A. and Shneiderman, B. (2006) Balancing Systematic and Flexible Exploration of Social Networks. *Visualization and Computer Graphics*, 12, pp. 693-700.
- Sabou, M. and Pan, J. (2007) Towards Semantically Enhanced Web Service Repositories. *Web Semantics: Science, Services and Agents on the World Wide Web*, 5, pp. 142-150.
- 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, pp. 336-343.
- Song, H., Cai, X. and Fu, Y. (2011) Rectangular Tree Browser: a Navigation and Visualization Tool for Large Hierarchies. *Journal of Information & Computational Science*, 8, pp. 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*, Atlanta, Georgia, USA, pp. 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 DMAMH*, pp. 123-128.