# A Novel Method for Visualising and Interacting With Hierarchical Datasets in 3D

**Sebastian Reiter**

Thesis submitted to

The University of Gloucestershire

According to the requirements of the degree

Doctor of Philosophy (PhD)

School of Computing and Engineering

June, 2024

Word Count: 55,958

## Abstract

The area of data visualisation and computer graphics in general is evolving constantly. There remain numerous challenges to solve in the context of data visualisation in VR or AR. In this work, I argue that 2D and text will be at the core of data visualisation for the foreseeable future. Hence, new visualisation techniques for 2D, 3D non-VR or AR applications may support overview and navigation tasks using novel approaches that combine both 2D and 3D techniques. Also, they should be able to scale arbitrarily from a design perspective to prevent visual clutter and a technical perspective to load only the relevant parts of the tree. Moreover, they may be designed utilising a user-centred approach, which does include an attempt to understand the needs of potential users.

This thesis describes a user-centred design, development and evaluation of such a new technique to visualise and interact with hierarchical data. In comparison with existing techniques for visualising hierarchical data my novel method is the first to utilise a user-centred design approach derived from user experience (UX) design. Based on the identified user needs, I introduce a new design metaphor and a novel design derived from it. The novelty of my design stems from the fact that it visualises nodes in detail while also being scalable without introducing visual clutter. Also, it is the first technique for hierarchical data to be designed from the start for both mobile and PC devices. It also introduces a novel concept to offer a classic list based 2D navigation view by dynamically fixating the camera in a native 3D view. This approach combines the benefits of a 3D overview with 2D navigation in just one view where a switch between the modes is simply realised by moving the camera and scaling the node labels.

The thesis presents two main practical contributions. The first one is the identification of actual user needs in the context of viewing and interacting with hierarchical data focussing on file and folder management. The second one is the design and development of a scalable technique to visualise and interact with hierarchical data.

**Author's Declaration**

I declare that the work in this thesis was carried out in accordance with the regulations of the University of Gloucestershire and is original except where indicated by specific reference in the text. For copyrighted material, permissions have been received before submission. No part of the thesis has been submitted as part of any other academic award. The thesis has not been presented to any other education institution in the United Kingdom or overseas.

Any views expressed in the thesis are those of the author and in no way represent those of the University.

Signed [redacted] Date 16th of June 2024 ……………………….

DOI:10.46289/GMZF4569

# Acknowledgements

I would like to thank my supervisor Professor Kamal Bechkoum for his support, advice and guidance during my PhD studies. This endeavour would not have been possible without his support. I also would like to thank my second supervisor Professor Shujun Zhang for his feedback. I am also grateful to all the people who helped me during my studies. This includes every research participant and staff of the University of Gloucestershire.

I dedicate this work to my mother and father, who enabled me to go to university, my wife and daughter, who supported and motivated me every day throughout this program. Without them I would not be where I am today.

## Table of Contents

## List of Figures

## List of Tables

# Chapter 1 Introduction

## 1.1 Project Background

The visualisation of data and interaction with those datasets is an area which is continuously progressing. At the time of writing, research is mainly focussing on techniques for virtual reality (VR) and augmented reality (AR) devices. For example, Belter and Lukosch (2022) carried out a user study to evaluate the suitability of VR games for usage in educational environments. Hence, according to Park and Kim (2022) the *Metaverse* is expanding rapidly. They described the Metaverse as a reference to a virtual world where a user is represented by an *avatar,* which acts as their alter ego in this world.

Also, there are techniques exploring the realisation of 3D virtual desktops for more traditional 2D output devices, for example, the display of a PC monitor. To exemplify, Yin and Gao (2016) presented such a technique implemented to run on a tablet computer supporting pen and touch interaction.

Hence, there are numerous methods available to visualise and interact with data. However, most techniques are very specific in their scope. For example, Teo et al. (2017) presented a technique for VR to visualise and query an extensive set of images including their meta data.

When viewed in a more general context one has to first differentiate between the various data types available to be visualised. In the context of their *type by task taxonomy* for information visualisations, Shneiderman (1996) described seven data types and seven tasks. Shneiderman (1996) proposed their taxonomy to sort potential prototypes and to provide guidance to researchers identifying new opportunities. They described the following data types:

- 1-dimensional: linear data, for example, text or a list of names
- 2-dimensional: map, planar data, for instance, maps, newspaper layouts

- 3-dimensional: projections of the real world, for example, buildings, human body
- temporal: similar to 1-dimensional data with the difference that items have a time component (start-/finish time, overlap of items), for example, project management systems
- multidimensional: data with n-attributes, for example, relational databases
- tree: data items have a single link to one parent item (except root), for example, computer-directory file management
- network: similar to tree but with no restriction regarding the number of links to other items, for example, World Wide Web (Shneiderman, 1996)

Although there are countless interesting challenges to solve in the context of data visualisation in VR or AR, this research focuses on the visualisation and interaction with hierarchical (tree) data on traditional PC monitors and mobile devices. Worth noting that, one can differentiate between the following basic types of tree visualisation: “node-link, … nested, … adjacency, … indented list and … matrix” (Graham & Kennedy, 2010, p. 239). Authors like Schulz et al. (2011) categorise methods to visualise hierarchies into explicit (node-link) and implicit (for instance, nested) representations.

In general, my choice to focus my research on techniques to visualise and interact with tree data has been informed by my initial literature review and previous research. Hence, I asked myself why techniques to visualise hierarchical data like for example the 3D *Cone Tree* (see Section 2.4.4) introduced by Robertson et al. (1991) never replaced the common tree view (see Section 2.3.1), for example, when navigating in a file system. I argue that 2D, and in general text-based content, is not a thing of the distant past. I argue that 2D and text will be at the core of what we actually do for the foreseeable future. For one there are individuals like Ken Kutaragi who do not advocate the usage of headsets as they isolate one from the real world (Mochizuki & Furukawa, 2022). Moreover, regarding the Metaverse there are still critical challenges to be solved, for example, concerning privacy and security (Park & Kim, 2022). Also, I argue that the technical capabilities of computing have increased over time, for example, by utilising current graphics processors, which may allow for new techniques to visualise more data in real time.

In short, new visualisation techniques may support linked 2D and 3D representations to support precise manipulation of 2D data and also overview orientated tasks in 3D (Marriott et al., 2018). Such new visualisations should consider utilising a user-centred design approach to be perceived as meaningful and effective by the user (Shen et al., 2019). Also, specific to types of tree visualisations, new tools may only load the required parts of a graph into memory and utilise 3D space to reduce visual clutter (Pavlopoulos et al., 2010).

### 1.2 Research aim and Objectives

The overall aim of this thesis is to propose, design and develop a novel method to visualise and interact with hierarchical data utilising conventional 2D input and output devices like touch input and computer monitors. Moreover, the thesis aims to describe the design and evaluation of this potential novel method. Consequently, my research objectives were:

- To carry out a critical investigation into existing methods to visualise and interact with hierarchical data.
- To design a novel method to visualise and interact with hierarchical data according to the identified requirements, for example, gaps in existing techniques challenging the predominance of the common tree view.
- To develop a software prototype for the novel method to identify technical challenges and limitations. Worth noting that, a software prototype does also include non-usable artifacts to demonstrate, for example, usage of the novel method in a simulated scenario.
- To evaluate the method from a technical perspective, for example, rendering performance and a user-centred perspective.

## 1.3 Contributions

This thesis presents two main practical contributions. The first one is the identification of actual user needs in the context of viewing and interacting with hierarchical data focussing on file and folder management. This user-centred design approach may lead to a method which is perceived as meaningful and effective by the user when compared to other existing techniques (Shen et al., 2019). The second one is the design and development of a scalable technique to visualise and interact with hierarchical data. Scalability in this context means for one that visual clutter may not occur when the tree grows in size. Moreover, the method is scalable from a technical perspective by offering the possibility to render only parts of the whole tree. This novel method may be utilised in commercial scenarios, for example, to visualise organisational charts. Also, it may be useful in academia, for example, when visualising data clusters.

In short, the main contributions are:

- Identification of actual user needs in the context of viewing and interacting with hierarchical data focussing on file and folder management.
- Design and develop a novel method to visualise and interact with hierarchical data utilising a user-centred design approach explicitly challenging the predominance of the common tree view.

## 1.4 Structure

This thesis has the following structure:

Chapter 1, “Introduction”, briefly describes the background of my research, my motivation, research aim and objectives, and contribution.

Chapter 2, “Literature Review”, describes various techniques to visualise and interact with hierarchical data. The techniques are discussed and are compared in a task analysis. Also, opportunities for new techniques are discussed.

Chapter 3, “Research Methodology”, contains the utilised methodology of my project, including a critical comparison of potential methods and tools.

Chapter 4, “Understanding User Needs”, focuses on the identification of user requirements for the novel method.

Chapter 5, “Design of a Novel Method”, describes the whole design process starting from initial sketches to wireframes

Chapter 6, “Development and Evaluation of a Novel Method”, outlines the whole process to evaluate my novel method. It also contains benchmark results and a final user study for evaluation.

Chapter 7, “Conclusion and Further Work”, concludes the thesis highlighting the main research results and the potential for future studies.

# Chapter 2 Literature Review

## 2.1 Introduction

In this chapter I provide an overview of techniques used to visualise hierarchical data. Note, most of the references I cite in this chapter are rather old. At the time of writing, they are relevant for my research because the terminology and techniques are still the latest ones regarding the visualisation of hierarchical data.

Section 2.2 describes various general visualisation approaches and types to provide context for the following sections.

Sections 2.3 (Desktop - 2D), 2.4 (Desktop - 3D) and 2.5 (Mobile) provide a detailed description of a subset of available visualisation techniques for hierarchical data. Note, as hierarchical visualisations and their variations are numerous, I only detail a subset of state-of-the-art techniques. These are layouts which informed future solutions, were recently developed, or introduced a novel approach. Worth noting that my selection is subject to bias because I chose the techniques to include. For instance, other researchers might not share my perception of novelty. To mitigate this, I aimed to cover a wide range of different techniques.

Where available I reviewed scholarly resources about the described techniques focusing on the following criteria:

- Node limit visual clutter (explicit)
- Node limit performance/usability (explicit)
- User study - quantitative results (for example, task performance)
- User study - qualitative results (for instance, interviews, focus groups)
- Interaction (for example, scroll, zoom)
- Additional remarks (not only from the original creator of the technique)

These criteria are used by researchers to evaluate (hierarchical) visualisation techniques, for example, node limits, user studies. For transparency, Table 2.1 shows a complete overview of the references I have found covering these criteria.

**Table 2.1**

*Availability Mapping for Scholarly Resources*

| | Node limit visual clutter | Node limit performance/usability | User study - quantitative | User study - qualitative | Interaction | Additional remarks |
|---|---|---|---|---|---|---|
| 2D - Tree view | N | N | Y | Y | Y | N |
| 2D - Treemap | N | N | Y | Y | Y | Y |
| 2D - Circle Packing | N | N | N | Y | Y | N |
| 2D - Hierarchical Edge Bundles | N | Y | N | Y | Y | N |

| | Node limit visual clutter | Node limit performance/usability | User study - quantitative | User study - qualitative | Interaction | Additional remarks |
|---|---|---|---|---|---|---|
| 2D - InfoSky Visual explorer | N | Y | Y | N | Y | N |
| 2D - Hierarchical Point Placement | N | N | N | N | Y | N |
| 2D - STREAMIT | N | Y | N | N | Y | N |
| 2D - Ordered and Quantum Treemaps | N | N | Y | N | Y | N |
| 2D - Textual Fisheye Tree Views | N | N | N | N | Y | Y |
| 2D - Hyperbolic browser | N | N | Y | Y | Y | N |
| 3D - Degree-of-Interest Trees | N | Y | N | N | Y | N |
| 3D - Document Lens | N | N | N | N | Y | N |
| 3D - Perspective Wall | N | N | N | N | Y | N |
| 3D - Cone Tree | Y | Y | Y | Y | Y | Y |
| 3D - H3 Layout | N | Y | N | N | Y | Y |
| 3D - HotSauce | N | N | N | N | Y | Y |
| 3D - Information Cube | N | Y | N | N | Y | Y |
| 3D - SemNet | N | N | N | N | Y | Y |
| 3D - Cam Tree (derived from Cone Tree) | N | N | N | N | N | Y |
| 3D - Information Landscape | N | N | N | N | Y | Y |
| 3D - Beamtree | N | Y | Y | N | Y | N |
| 2D - Enhanced Radial Edgeless Tree | N | N | Y | N | Y | N |
| 2D - RECTANGULAR VIEW | N | N | N | N | Y | N |
| 2D - MAGIC EYE VIEW for mobile handhelds | N | N | N | N | Y | N |

| | Node limit visual clutter | Node limit performance/usability | User study - quantitative | User study - qualitative | Interaction | Additional remarks |
|---|---|---|---|---|---|---|
| 3D - Space Manager | N | N | N | Y | Y | N |
| 2D - PaisleyTree | N | Y | N | N | Y | N |
| 2D – Mobile Tree Browser | N | N | Y | N | Y | N |

*Note*. List of the covered visualisation techniques mapped to the availability of scholarly resources available for the different criteria. "Y" (yes) indicates that I have found relevant references for the criteria while "N" (no) indicates that I have not found any relevant references.

Also, I reviewed each method according to the following criteria:

- Screen space utilisation
- Conveyance of structural information
- Support for long text labels (visual clutter)
- Overview tasks
- Support for precise navigation

The reason why I chose to focus on these review criteria is because these topics are often discussed in the literature linked to hierarchical data visualisation and interaction.

The following Section, 2.6, presents various selected references focussing on comparative studies and critical discussions for future visualisation techniques. Also, I present a task analysis where I mapped the visualisation techniques (see Section 2.3, 2.4 and 2.5) to tasks outlined by Shneiderman (1996).

The last Section, 2.7, draws conclusions to define a potential scope for new visualisation methods.

**2.2 Contextualisation**

According to Shneiderman (1996) a combination of an overview and detail view is also called *context plus focus*.

Another system to display an overview (contextual) and detail view is called *overview+detail*. According to Cockburn et al. (2009) these systems in contrast to context plus focus interfaces explicitly separate the overview and detail view from each other through spatial separation. Also, they described that techniques implementing zooming are viewed as another approach to support both focused and contextual views. Moreover, they outlined the *cue* approach. They explained that this approach describes views which modify how items are displayed, for example, to highlight items in search results. Also, they highlighted that a lot of systems utilise a combination of techniques, for example, overview+detail with zoom.

Worth noting that, when a technique utilises a context+focus or overview+detail approach I explicitly mention the utilised approach in the title. However, this was not applied to zooming or highlighting (cue) approaches in a similar way. I argue that zooming and highlighting are implemented in some form for almost all available software applications, for example, in code editors, text processors or file navigation.

Some of the described techniques utilise *directed graphs*. According to Zhou et al. (2005) a directed graph (G) is a finite set V of vertices with E edges. They explained that an edge of the directed graph consists of two graph vertices (u,v) which form an ordered pair. According to Andrews et al. (2002) a *directed acyclic graph* is a directed graph where cycles are not allowed.

As described by Marriott et al. (2018) I too consider a visualisation technique to be 3D when it maps information to three different spatial dimensions. As detailed by them the output device (for example, 2D screen) and whether these mappings are abstract or not was not relevant for the classification. Worth noting that, I did not consider techniques built for virtual or augmented reality devices.

Some visualisation techniques based their design on a metaphor. One of the earliest design metaphors was the *Desktop metaphor* as an analogy with real office spaces (Johnson et al., 1989).

Various techniques utilise a *fisheye* lens. According to Furnas (1986) this type of lens visualises areas near the centre of the lens in great detail while the more remote areas are successively displayed in less detail. They explained that for interface design these fisheye views require the use of a *Degree of Interest (DOI)* function which calculates a number for each point in a structure to rate the user's interest in seeing that very point.

## 2.3 2D Visualisation Techniques for Hierarchical Data

### 2.3.1 Tree View

According to Tominski et al. (2006) common tree views have a vertical indentation-based tree layout. They explained that most common tree views immediately hide or show child nodes when a parent is expanded or collapsed.

This tree style is also called *indentation*. According to Graham and Kennedy (2010) it is the most common form to display a tree in graphical user interfaces, for instance, it can be seen in the Microsoft Windows Explorer for folder navigation.

Regarding potential node limits for visual clutter, I argue that there is no such limit due to scrolling and the design structure of the technique. Hence, based on the design occlusion cannot occur. When the dataset grows the part of the tree which cannot be visualised is usually hidden and accessible by scrolling.

For potential node limits regarding usability there are implementations available which imply that there is no such limit, for example *lazy loading* (see Bozhanov, 2020).

In quantitative user studies the tree view is often used as a baseline for comparison to other techniques. Cockburn and McKenzie (2000) compared a tree view interface to a Cone

Tree when measuring, for instance, task performance for folder and file navigation. They concluded that the participants were able to complete the tasks faster using the tree view. They also asked the participants for comments about the interfaces after they completed all of the tasks. Several research participants stated that tree views provide a worse feel for the structure of the data-space in comparison to a Cone Tree.

According to Kobsa (2004) implementations like the Windows Explorer shielded very good overall performance with regard to speed, correctness and user satisfaction. They carried out a between-subjects experiment comparing different information visualisation systems including measuring task performance accuracy, task completion times and user satisfaction. They outlined that their dataset (test hierarchy) was based on items from eBay containing five levels and 5,799 nodes while their employed tasks focused on various topics like navigation and also attribute based questions.

Andrews and Kasanicka (2007) compared different hierarchy browsers including the tree view, *Treemap* (see Section 2.3.2), *information pyramids* (see Section 2.4.10) and *hyperbolic browser* (see Section 2.3.10) in a user study collecting task completion time, subjective ratings and overall preference data. They concluded that there were no significant variations regarding task performance between the different hierarchy browsers including the tree view implementation except for a single task. However, they noted that users had a preference towards the tree view browser. They argued that this preference may be the result of a form of bias as users utilise programs like the Windows Explorer regularly.

According to Tominski et al. (2006) common interactions for a tree view are to expand or collapse tree nodes and scrolling the entire tree. Kobsa (2004) described that as part of the Windows Explorer, sorting, searching and a file detail view are available.

I consider the tree view to be a good option for precise navigation, a detail view and to convey structural information. Through its utilisation of scrolling, it is able to display text to varying detail, for example to visualise long folder names. Due to scrolling it may lack overview capabilities but for navigation this may also be its strength. Especially when using a familiar dataset, a user may be able to identify their data of interest faster without the potential distraction of an overview. Also, the option to collapse and to expand tree nodes may support object selection because the user can explore potential paths without losing their orientation by collapsing paths not relevant to them. Also, this feature supports the utilisation of the available screen space by enabling to user to hide unnecessary information. The indented layout may help to convey structural information, for example, to show which files belong to a folder.

However, I agree that its wide adoption may be due to bias. For me its success is closely linked to the success of the Desktop metaphor, for example, Microsoft Windows and others. To exemplify, according to Johnson et al. (1989) the Xerox Star models the task of file organisation similar to the real world, for instance, by providing the option to put files into a folder.

### 2.3.2 Treemap

The Treemap is an implicit (nested) visualisation technique (Schulz et al., 2011). It shares attributes with "circle diagrams attributed to Venn and Euler" (Baron, 1969, p. 117). In a Treemap the relationship between a parent and a child node is visualised by nesting the child inside the parent (Schulz et al., 2011). The method was first introduced by Johnson and Shneiderman (1991).

According to Kobsa (2004) Treemap users excelled in task solution times and correctness of their answers. They also highlighted the comparatively high user satisfaction ratings of the Treemap.

Based on video analysis Kobsa (2004) also concluded that the colour coding and filtering capabilities were the Treemap's most valuable features. However, their video analysis also confirmed concerns that a Treemap visualisation is challenging when solving global structure tasks, for example, like identifying the maximum depth of a hierarchy.

In their comparative study, Andrews and Kasanicka (2007) described that there were no significant differences between the four techniques including the Treemap browser regarding task performance except for a single task. However, the Treemap browser was consistently rated significantly lower than the other techniques by the participants.

Known options for interaction include the already mentioned filtering feature implemented by Kobsa (2004) in their Treemap solution for file and folder navigation.

According to Bederson et al. (2002) Treemaps can hardly be used to visualise images because they do not guarantee a specific aspect ratio for their result. Consequently, images with a fixed aspect ratio do not fit well in the Treemap's rectangles.

Van Ham and Van Wijk (2002) highlighted two challenges for standard Treemaps. For one they argued that Treemaps often lead to long and thin rectangles (high aspect ratios) which can result in, for example, interaction problems. Secondly, by design a Treemap hides the hierarchical layers of the tree which makes it difficult to reconstruct the hierarchical information, for example, the tree layout.

According to Rekimoto and Green (1993) the Treemap utilises the available screen space effectively but the nested visualisation is difficult to understand because the structural information of the data is not as apparent.

Tu and Shen (2007) demonstrated the flexibility of the Treemap layout by introducing a novel spiral Treemap layout to visualise alterations in hierarchical datasets.

Stasko and Zhang (2000) introduced a radial space-filling technique they called *Sunburst*. They described their technique as a radial alternative to the rectangular Treemap. Yang et al. (2003) presented a radial space-filling visualisation system called *InterRing* focusing on interactive operations for hierarchical data. Woodburn et al. (2019) introduced the *sundown chart* as a semi-circular alternative to the Sunburst technique optimised for the wide aspect ratios of modern displays and to reduce disorientation while zooming.

Kruskal and Landwehr (1983) presented a technique called *Icicle Plots*. According to Macquisten et al. (2020) it utilises the adjacency of nodes when visualising hierarchies. They described it as an implicit technique where the tree is visualised vertically with the leaf nodes at the bottom, child nodes are above them and root is at the top. Also, the width of the nodes visualises a frequency value, for example, a child node with six included leaves is wider than one with only two leaves while root is always the widest.

Worth noting that, Macquisten et al. (2020) carried out a usability study to compare five techniques including the Treemap, Icicle Plots and Sunburst method. They aimed to investigate how well the techniques represent the data visually. They concluded that Icicle Plots are among the best to compare node sizes in large hierarchies while Sunburst is better when comparing colour. However, for small hierarchies the Treemap and Icicle Plots are among the overall best options. Also, when leaf node information is important or the hierarchy contains a large number of leaf nodes the Treemap may be the most appropriate choice as it is more space efficient.

I agree that the Treemap utilises the available screen space very well due to its space filling layout. Also, it conveys the proportions of a dataset, for example, when visualising the amount of child nodes. However, the structural information of the data is hard to understand. Also, depending on the output window or screen size the rectangles may not offer enough space to show long text labels, for example, full size text labels. Therefore, it is very well suited to support overview tasks but not for precise object selection or a detail view. For example, if one wants to visualise 1,000 folders in a Treemap, navigation would no longer be possible because some of the rectangles would become too small to select them.

### 2.3.3 Circle Packing (Overview+Detail)

Wang et al. (2006) introduced the method *Circle Packing* to visualise large hierarchies. They described that their method was in part inspired by Treemap visualisations but they use nested circles instead of rectangles to visualise the hierarchy.

According to Wang et al. (2006), Figure 2.1 shows a directory overview for 284 folders and 3,355 files containing seven levels. Directories are represented by white circles with labels at their bottom. The files are shown as coloured circles. The radius of one coloured circle reflects the file's size. They also implemented a tree view on the top-right of the interface which is linked to the main view (left). For instance, when a user clicks on an item in the tree view the associated subitems are shown on the left (main view).

I argue that this tree view as a type of map adds a form of overview+detail functionality to their method.

Wang et al. (2006) also implemented an option to view the result in 3D as nested cylinders (directories) and spheres (files).

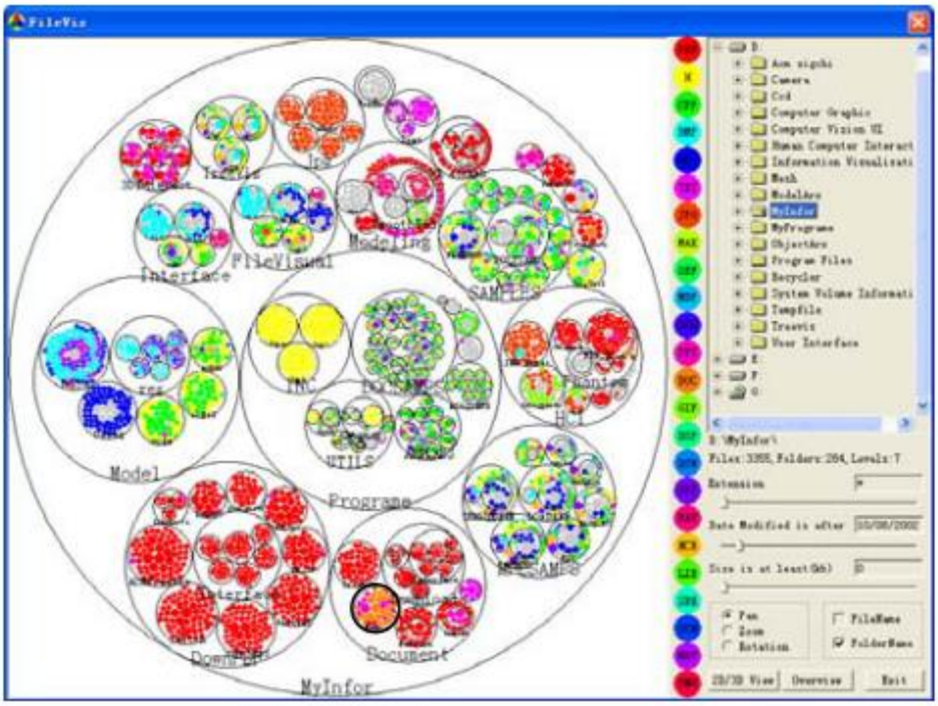



Figure 2.1: User Interface and the Overview of “D:\MyInfor”

*Note*. From “Visualization of large hierarchical data by circle packing” by W. Wang, H. Wang, G. Dai and H. Wang, 2006, *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, p. 519 (https://doi.org/10.1145/1124772.1124851). Copyright 2006 by the Association for Computing Machinery. Reprinted with permission.

According to Wang et al. (2006) their file visualisation tool was used by twenty-one students in their laboratory. They received various comments from the students, for example, that their system is robust and efficient even for thousands of files, their user interface is easy to use and their interface needs less operation in comparison to traditional file management systems.

Wang et al. (2006) implemented filtering (for example, by filename extension, file date, file size), zooming. They explained that zoom is applied when a user clicks on a folder to show the included objects in more detail.

I consider the Circle Packing method to be a novel alternative to Treemaps. Because of its circular layout it may not be an ideal choice to utilise the available screen space. However, in comparison to a Treemap it provides structural information through their nested circles approach. However, I argue that similar to a Treemap there are node limits to this approach. For example, if one wants to visualise 1,000 folders, navigation would no longer be possible because some of the circles (folders) would become too small to select them. Also, it is questionable if in this scenario the visualisation would still provide meaningful insights into the data, for example, are text labels still readable, small or uniformly sized circles.

The technique may be used to support overview orientated tasks based in its layout. However, in comparison to a Treemap, the nested circles and their colours may be distracting to a user.

Their method may show a more detailed view through their implementation of zoom. However, the level of detail is limited by the available screen space and the number of objects (circles). Hence, for some cases it may not be possible to show longer text labels. However, this is somehow mitigated by their implementation of an additional tree view.

Concerning object selection, it may still be a better alternative compared to the Treemap. Hence, with Circle Packing a user may be able to quickly identify their objects (circles) of interest to select them.

### 2.3.4 Hierarchical Edge Bundles

Holten (2006) introduced *Hierarchical Edge Bundles*. They described that their method integrates with existing tree visualisation techniques, reduces visual clutter when displaying a large number of adjacency (additional, non-hierarchical relations) edges and offers flexible parameterisation to control bundling strength. Also, they explained that their core idea was to visually bundle adjacency edges together similar to, for example, network cables.

According to Holten (2006), Figure 2.2 shows the result of an increasing bundling strength value β from left to right. When the value increases the visualisation provides high level information by implicitly bundling the respective adjacency paths.

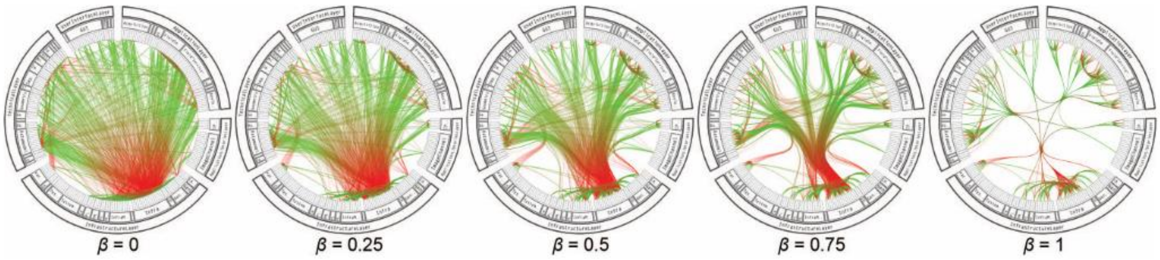



Figure 2.2: Using the Bundling Strength β to Provide a Trade-Off Between Low-Level and High-Level Views of the Adjacency Relations

*Note*. From "Hierarchical Edge Bundles: Visualization of Adjacency Relations in Hierarchical Data" by D. Holten, 2006, *IEEE transactions on visualization and computer graphics*, *12*(5), p. 746 (https://doi.org/10.1109/tvcg.2006.147). © 2006 IEEE. Reprinted with permission.

Regarding potential node limits for visual clutter, I assume that there is no such limit based on the aim of the method. Hence, it is an implicit method to reduce visual clutter.

According to Holten (2006), their prototype application offered real time performance with at least 10 frames per second on their development (ATI Radeon X300 GPU and Pentium 4 3.0GHz CPU with 1GB of RAM) system. They visualised up to 1,011 adjacency relations representing module-to-module calls of a hierarchically organised software system and associated call graph.

Holten (2006) organised informal user studies to demonstrate their method and invited participants from the industry and academia to examine their method interactively. They described that a majority of the participants found the method to be useful to gain quick insights in the adjacency relations of a hierarchy. The participants especially valued the flexible parameterised approach for the bundling strength β to gain multiple levels of insights into the data, for example, to be able to switch from a low-level node-to-node connectivity to a high-level visualisation. Also, the participants highlighted that the visualisation, more specifically the bundles gave them an impression of the organisation of the hierarchical data.

Holten (2006) implemented a filtering feature, for example to adjust bundling strength.

I consider the method to be an interesting approach to reduce the visual complexity of adjacency edges. The applicability of Hierarchical Edge Bundles depends on the choice of the tree visualisation technique. For some techniques the visual shift when adjusting the bundling strength parameter may lead to a user becoming disorientated.

Note, given its purpose this method is not part of my further review. Hence, it is not a standalone technique to visualise hierarchical data. However, adjacency edges may become relevant when designing my method.

### 2.3.5 InfoSky Visual Explorer (Overview+Detail)

Andrews et al. (2002) introduced the *InfoSky* system. According to them InfoSky enables users to explore large hierarchies of document collections. They described that they utilized a real-world telescope metaphor, “zooming galaxy of stars” (Andrews et al., 2002, p. 167), for the design of their system by placing objects (documents) of similar content close together and visualise them as stars. Their system allows the user to seamlessly switch between an overview and detail view with zooming and animation. They also enabled the mapping of metadata, for example, document size. Moreover, the creation of queries is possible to visually highlight matches.

According to Andrews et al. (2002) Figure 2.3 shows the interface using a dataset of approximately 109,000 news articles from a German daily newspaper with a (manually) classified hierarchy of around 6,900 (sub-)collections and 15 levels of depth. The interface consists of a control panel for history, bookmarks and search features. A tree view with expandable folders on the left mirrors the selection of the telescope view on the right side.

The telescope view on the right shows the galaxy at the user selected location and magnification. A small overview of the whole galaxy (dataset) is provided in the upper left corner of the telescope view to indicate the current location and magnification.

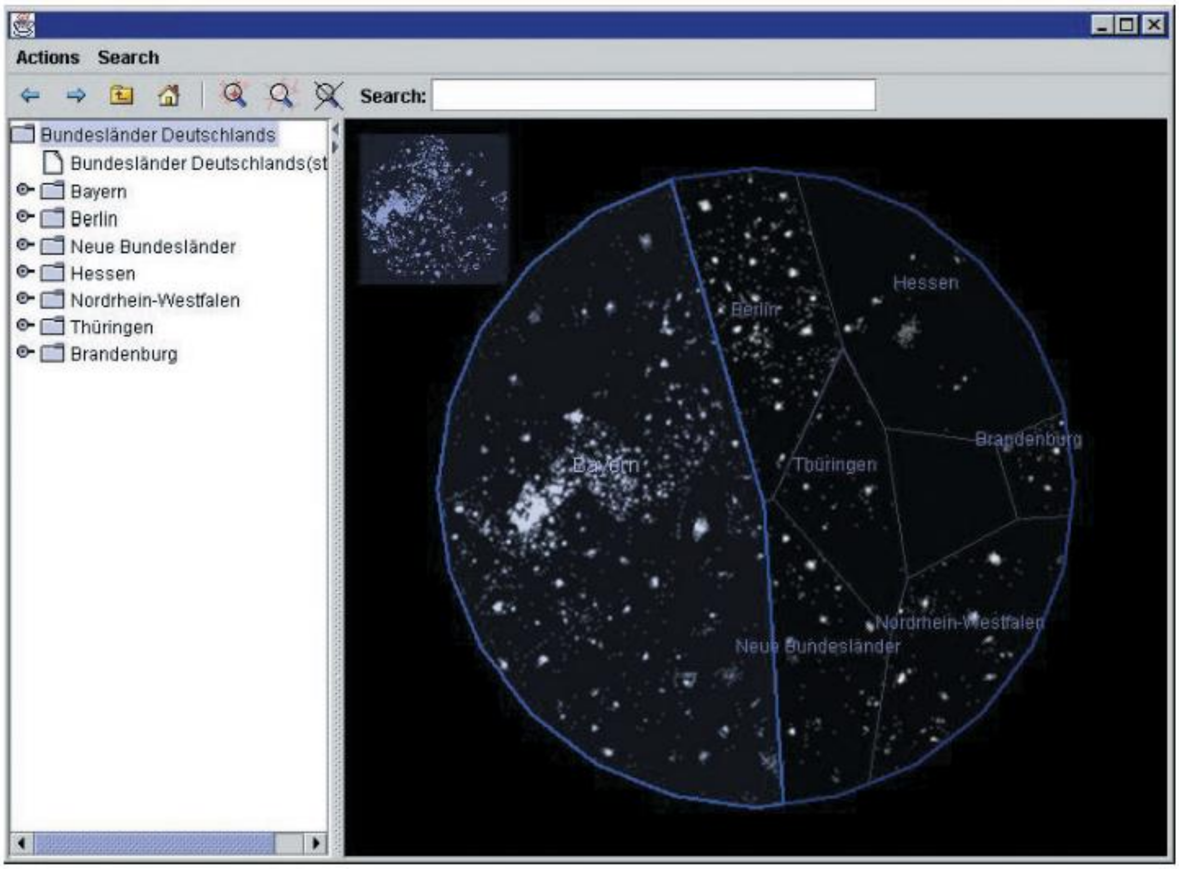



Figure 2.3: The User Interface of the InfoSky Visual Explorer

*Note*. From "The InfoSky visual explorer: Exploiting Hierarchical Structure and Document Similarities" by K. Andrews, W. Kienreich, V. Sabol, J. Becker, G. Droschl, F. Kappe, M. Granitzer, P. Auer and K. Tochtermann, 2002, *Information Visualization*, *1*(3-4), p. 168 (https://doi.org/10.1057/palgrave.Ivs.9500023). Copyright 2002 by SAGE Publishing. Reprinted with permission.

Regarding potential node limits for visual clutter, I assume there is no such limit based on the design of the method (zooming).

Based on their set requirements there is also no node limit regarding usability: “Visualise very large (hundreds of thousands, if not millions of entities)” (Andrews et al., 2002, p. 166).

Andrews et al. (2002) carried out a formal experiment to measure task performance between the InfoSky tree browser (see Figure 2.3, tree view) and the InfoSky telescope (see Figure 2.3, telescope view) view/browser. They explained that in this stage both views were separated into two different applications for comparison. The dataset from the German daily newspaper was used (see Figure 2.3). They concluded that their InfoSky telescope view performed slower than the tree browser.

Andrews et al. (2002) implemented a control panel (for example, history, bookmarks and search), a tree view (expand and collapse folders), zoom and sliding (panning). They described that it is also possible to select and centre single documents.

According to Andrews et al. (2002), their method expects a *collection hierarchy* for data input. They explained that their collection hierarchy is a directed acyclic graph where entities, for example, documents are allowed to have multiple parents.

Worth noting that, similar documents (stars) are placed closer together to form visual clusters of stars (Andrews et al., 2002).

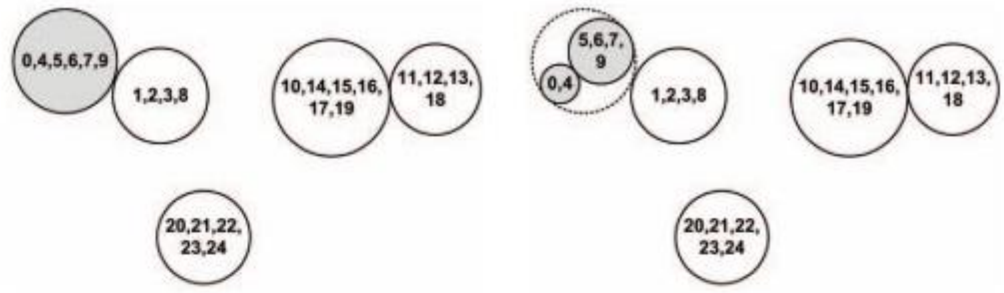
Albeit an interesting metaphor I argue that the method does not utilise the available screen space very well caused by, for example, the circular cut-out of their telescope and the integration of an additional map for orientation. The method conveys structural information very well. This is achieved by their nested approach and by placing similar stars (documents) closer together.

Especially their integration of a map supports overview orientated tasks. However, for long text labels I do not consider this method to be very suitable. The clustering of stars may not leave enough screen space to show long text labels. Therefore, it may also be challenging for users to select individual objects (documents). During navigation a user may be constantly forced to zoom and pan to identify and select their objects of interest. However, this is somehow mitigated by their integration of a tree view which mirrors the selection of the telescope view.

### 2.3.6 Hierarchical Point Placement

Paulovich and Minghim (2008) developed a technique they named *Hierarchical Point Placement (HiPP)*. They explained that the technique visualises a hierarchal cluster tree where elements depending on the context can represent individual data objects and clusters of objects. Also, when a cluster is selected the respective node gets replaced with its children on the layout. Furthermore, they used colour coding to visualise additional relationships between the clusters. For instance, they showed a document map where the different colours of the clusters indicate the most frequent subject areas of the contained documents.

According to Paulovich and Minghim (2008), Figure 2.4 shows an example of the visualisation technique based on a hierarchical cluster tree for a bi-dimensional (2D) data set. On the left (a) the root node is displayed where the size of each circle corresponds to the number of data objects in it. When a user selects a node (highlighted in grey) this node gets replaced by its children (b). In this example two (sub-)clusters in 0,4,5,6,7,9. The numbers shown in the circles are a reference to the included objects. The position of these clusters is projected in a way to preserve the neighbourhood relationships of the included children (for example, object 4) as much as possible.




(a) Projection of *ROOT* node. (b) Projection of *(0,4,5,6,7,9)* node.

Figure 2.4: Example of Projecting a Hierarchy

*Note*. From "HiPP: A Novel Hierarchical Point Placement Strategy and its Application to the Exploration of Document Collections" by F. V. Paulovich and R. Minghim, 2008, *IEEE transactions on visualization and computer graphics*, *14*(6), p. 1231 (https://doi.org/10.1109/tvcg.2008.138). © 2008 IEEE. Reprinted with permission.

Regarding potential node limits for visual clutter, Paulovich and Minghim (2008) noted that because almost all point placement techniques map and visualise every data instance at once there is such a limit depending on the visual area available. They explained that when there is not enough space for the visualisation overlap will occur. Further they highlighted the risk of cognitive overload when all data instances are displayed at once.

Paulovich and Minghim (2008) compared their method to other similar point placement projection techniques to assess the different layouts regarding the preservation of the data objects neighbourhood. Further, they described the computational requirements for their method. No user studies have been carried out, for example to measure task performance or error rates.

According to Paulovich and Minghim (2008) they implemented a feature to select nodes. Also, they added a form of filtering because the user can select multiple clusters and join them into a new one.

I consider this technique to be very effective for visualising hierarchical data when it is important to preserve neighbourhood relationships of leaves. However, because it tries to preserve neighbourhood relationship it may not utilise the available screen space very well depending on the dataset. Also, it is prone to overlap (visual clutter) for large datasets.

Similar to a Treemap the structural information may be difficult to understand due to the nested design approach.

The technique may be used to support overview orientated tasks based on its layout. Their usage of colour to visualise additional relationships also supports the overview. However, because the method tries to preserve neighbourhood relationships it is not well suited for long text labels. Hence, when two objects are close together (very similar) there may be not enough screen space to show even a short text label. The method may be suitable for object selection. Especially when their mechanism to show and hide (sub-)clusters is viewed as an alternative to expand or collapse folders in a tree view. Therefore, similar to a tree view the user may then explore potential paths without losing their orientation by collapsing (hide) paths not relevant to them. However, for a large group of objects this may not be possible because the children may become too small to view or select them.

### 2.3.7 STREAMIT

Alsakran et al. (2011) introduced an interactive visualisation system to enable a user to explore constantly streaming-in text documents. They called their prototype for the system *STREAMIT*. According to them their system utilises a force-directed simulation where text documents are continuously inserted. They explained that each document is visualised as a moving mass particle. Similar documents (particles) are moved closer together based on their pairwise text similarity to form data clusters and outliers. When new documents are inserted, the system adjusts its visualisation automatically to reduce the risk of change blindness regarding new emerging patterns. Also, they optimised their method by utilising a similarity grid where new documents are placed in a cell of the grid with the most similar documents. Therefore, the new document reaches its preferred location more quickly during simulation.

According to Alsakran et al. (2011) Figure 2.5 shows the STREAMIT interface with its components. They explained that the main window is shown on the top left where each document is visualised as a circle positioned based on their similarity to each other. The visualisation dynamically changes as the simulation continues. A grey scale indicates the age of a document, for example, dark grey is older. The user can set an attribute of the documents to be reflected in the circle’s sizes. To reduce the risk of overlap the user can define a minimum distance between two documents. An animation control panel (3) is at the bottom of the visualisation. This can be used to examine a buffer window (recent documents buffered in a moving time window). At the top right a keyword table (1) is shown which lists all the keywords of the currently visualised documents including attributes like their frequencies. Document tables (2) are displayed at the bottom right which show various attributes of the documents, for example, title, authors. Each document table (tab) shows a different (sub-)set of the documents. For instance, one tab only shows all the documents currently on display in the main window while another one shows all of the buffered documents.

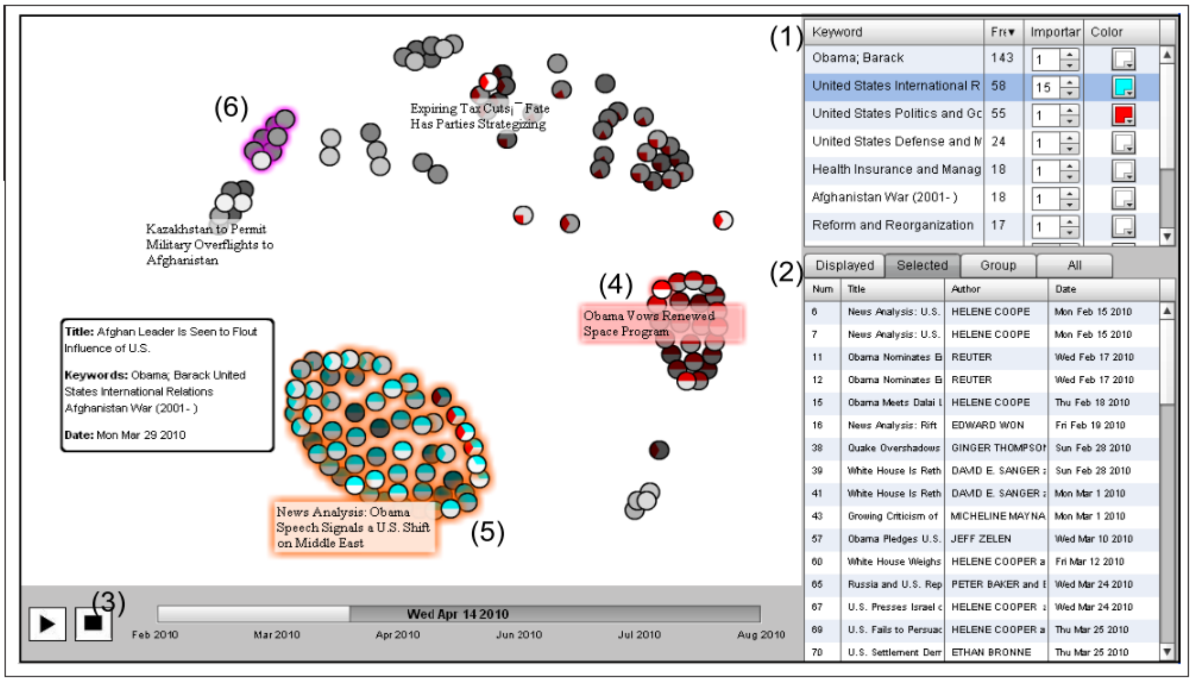



Figure 2.5: STREAMIT Interface

*Note*. From "STREAMIT: Dynamic visualization and interactive exploration of text streams" by J. Alsakran, Y. Chen, Y. Zhao, J. Yang and D. Luo, 2011, *2011 IEEE Pacific Visualization Symposium*, p. 135 (https://doi.org/10.1109/pacificvis.2011.5742382). © 2011 IEEE. Reprinted with permission.

Regarding node limits for visual clutter, Alsakran et al. (2011) stated that their system is indeed subject to visual clutter when the number of documents to visualise gets very large.

According to Alsakran et al. (2011) STREAMIT offers real time performance (25-30 frames per second). They conducted their performance experiments using text streams with news documents from the New York Times and synthetic data on consumer graphics cards and PCs (NVidia Quadro NVS 295 GPU with 2GB texture memory, Intel Core2 1.8GHz CPU with 2GB RAM).

Alsakran et al. (2011) carried out two case studies with different data sets.

According to Alsakran et al. (2011), a user can filter group documents, for example. Also, a user can select single or multiple documents in the document tables. They explained that the user can also select documents based on keywords in the keyword table to be highlighted in the main window (see Figure 2.5, point 5). Additionally, selected documents can be saved for later examination (history, replay feature).

I consider the method to be very valuable when working with streaming data. It does utilise the available screen space very efficiently because the system adjusts itself when new data is read. However, it is subject to visual clutter for large datasets. Structural information is conveyed based on the clustered layout.

It may also be an effective method to support overview orientated tasks. Not only does the system visualise emerging clusters their data streaming approach also may mitigate the risk of change blindness.

The method is not well suited to display long text labels because it aims to group similar objects close together. This is somehow mitigated by their implementation of spatially separated detail windows, the document tables.

Also, it may be used for object selection because of the document and keyword tables. However, there may be not enough screen space available to visualise objects (documents) with many different attributes in the document tables.

### 2.3.8 Ordered and Quantum Treemaps

Bederson et al. (2002) introduced a new *strip* Treemap algorithm. They explained that their algorithm is based on the *Squarified Treemap algorithm* (see Bruls et al., 2000). Continuing, their algorithm is processing the Treemap rectangles in order placing them in strips (vertical or horizontal). Bederson et al. (2002) stated that all Treemap algorithms in theory can be quantized. One variant they described is a quantized version of their strip treemap. It differs from the strip treemap when computing the rectangle dimensions as these get reduced or are expanded to lay them out in a grid for the number of objects.

Bederson et al. (2002) used quantum treemaps to develop a photo browser application they named *PhotoMesa* which visualises thumbnails of photos clustered by their metadata.

According to Bederson et al. (2002) Figure 2.6 shows their PhotoMesa application using their quantized version of a strip treemap.

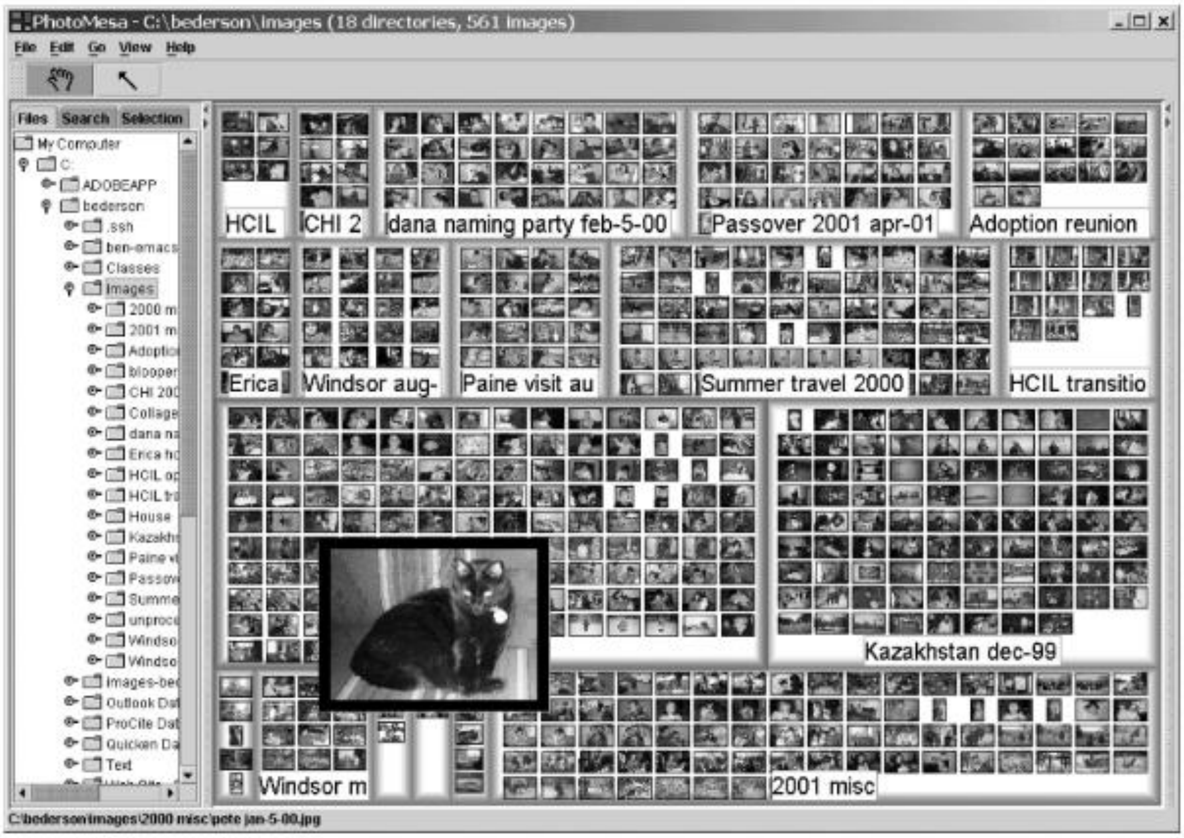



Figure 2.6: PhotoMesa Using Quantum Strip Treemaps to Group 556 Images in 17 Directories

*Note*. From "Ordered and quantum treemaps: Making effective use of 2D space to display hierarchies" by B. B. Bederson, B. Shneiderman and M. Wattenberg, 2002, *ACM Transactions on Graphics*, *21*(4), p. 851 (https://doi.org/10.1145/571647.571649). Copyright 2002 by the Association for Computing Machinery. Reprinted with permission.

Bederson et al. (2002) carried out a user study to measure task performance (time based) comparing their strip algorithm against an implementation of the Squarified Treemap and *pivot* (see Shneiderman & Wattenberg, 2001) algorithm. Their dataset consisted of 100 rectangles with random sizes from a uniform distribution. All participants carried out 10 tasks for each algorithm. For each task a random new Treemap was generated where the number of the rectangle was shown as a label reflecting the input order. The participants had to find and select the correct rectangle corresponding to a target number shown at the bottom of the application, for example, 12. After completing all of the tasks, the 20 participants were asked to rate the three algorithms. They concluded that for their time measurements the strip and pivot implementations performed very similar with both being faster than the Squarified version. Similarly, most of the participants chose either the strip or pivot version as their personal preference.

According to Bederson et al. (2002), a user can smoothly zoom into a region of PhotoMesa when a rectangle (region) is selected or zoom out by right-clicking.

I consider their Treemap algorithm a valid enhancement to the Treemap. Similar to a Treemap their algorithms utilise the available screen space very well due to their space filling layout.

The PhotoMesa application using their quantized version of a strip treemap provides more structural information than a Treemap. However, the structural information may be lost if the number of images grows too large for one rectangle. Similar to a Treemap it is well suited to support overview orientated tasks. However, for large datasets the rectangles may become too large to visualise the whole tree because they are scaled to be laid out in a grid. The method may be able to provide support for long text labels because of the rather fixed input object dimensions, for example, 160 pixels by 90 pixels and zoom operations. Therefore, it may also be a useful approach when visualising text-based nodes instead of pictures. Also, the fixed input object dimensions combined with zooming may enable a user to quickly identify and select their objects of interest.

### 2.3.9 Textual Fisheye Tree Views (Context+Focus)

Tominski et al. (2006) introduced *Textual Fisheye Tree Views* to enhance a tree view with overview+detail and context+focus techniques. According to them they preserved the vertical indentation-based layout of a common tree view including its interaction methods, for example, scroll, expand or collapse nodes. However, in their technique they utilised animation when nodes are collapsed or expanded. Additionally, they implemented zoom operations controlled by the mouse's scroll wheel (overview). To restore readability when zoomed out they applied a 1D fisheye transformation. Consequently, the label at the users focus point gets magnified. The amount of magnification can be set by the user. They stated that all of these added features can be used on demand.

Tominski et al. (2006) also applied the interaction techniques they described (like zoom, fisheye) to another tree layout. They argued that this *Non-Textual Fisheye Tree View* may be more effective than the Textual Fisheye Tree View when navigating deep hierarchy trees.

According to Tominski et al. (2006), Figure 2.7 shows the effect of the applied fisheye transformation for their Textual Fisheye Tree View. The amount of magnification can be set by the user. The vertical and horizontal scroll bars remain operable.

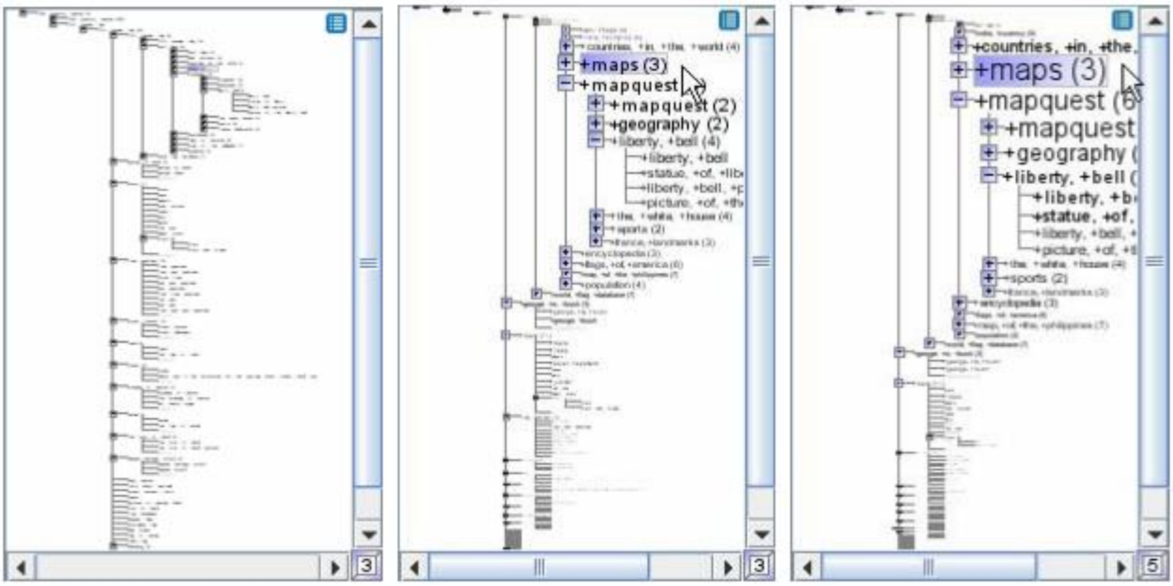


Figure 2.7: Showing the Fisheye Tree View Zoomed out (left) and With Different Amounts of Fisheye Magnification Applied (Middle and Right)

*Note*. From "Fisheye Tree Views and Lenses for Graph Visualization" by C. Tominski, J. Abello, F. Van Ham and H. Schumann, 2006, *Tenth International Conference on Information Visualisation (IV'06)*, 3.1. Textual Fisheye Tree View section (https://doi.org/10.1109/iv.2006.54). © 2006 IEEE. Reprinted with permission.

Regarding potential node limits for visual clutter, I argue that there is no such limit due to scrolling and the described layout of the technique (see Section 2.3.1).

According to Tominski et al. (2006) a user can collapse and expand nodes. Also, they implemented common vertical and horizontal scrolling. Optionally a user can apply zoom using the mouse scroll wheel. Moreover, a 1D fisheye transformation allows the user to magnify regions of interest. They described that the amount of magnification can be adjusted by the user.

Abello et al. (2006) developed a node-link-based graph visualisation system called *ASK-GraphView*. As a part of this system, they provided an overview of the entire hierarchy utilising a technique similar to a Non-Textual Fisheye Tree View. Also, they stated that their system has not been formally evaluated yet.

I do not agree with Tominski et al. (2006) regarding their technique providing overview+detail functionality. Based on my understanding, an overview+detail technique has to provide spatially separated views for overview and detail, for example, in the form of a map.

In general, I consider their method to be very similar to a tree view regarding its capabilities (see Section 2.3.1). In contrast to a tree view their technique does support overview orientated tasks by enabling a user to apply zoom. When zoomed out object selection is still possible through their utilisation of a fisheye view. However, a user may temporarily lose the option to efficiently visualise long textual nodes because of the distortion caused by the fisheye view. Therefore, it may be difficult for users to identify their preferred balance between readability and overview when setting their zoom level and amount of fisheye magnification.

### 2.3.10 Hyperbolic Browser (Context+Focus)

Lamping et al. (1995) introduced the hyperbolic browser a context+focus (fisheye) visualisation technique to display large hierarchies. They explained that their technique projects the hierarchy uniformly on a hyperbolic plane to map it onto a circular display region. According to them the combination of a fisheye distortion with the ability to uniformly embed a growing structure were the key aspects (Escher woodcut) motivating their design. They stated that their hyperbolic browser can display more nodes and longer text labels (1,000 nodes) in comparison to a standard 2D hierarchy browser (100 nodes) using a 600 pixel by 600 pixel window. They implemented animated transitions when navigating the tree to maintain visual object constancy and to support a user’s assimilation when changing views.

The figure with the title “Figure 3: Changing the focus.” (Lamping et al., 1995, p. 402) shows, according to Lamping et al. (1995), a smooth transformation when changing focus. They explained that the available space for a node is reduced continuously based on its distance from the centre of the view. Also, a user can change the focus by clicking on a visible point. Consequently, the selected region is moved to the centre of the view. Alternatively, the user can drag a visible point to any other position in the view. Nodes close to the centre are magnified while regions toward the edge are minified. According to them one challenge of the hyperbolic browser is to preserve orientation. They stated that in the figure this challenge can be observed as nodes keep rotating during navigation.

Regarding potential node limits for visual clutter, after evaluation Lamping et al. (1995) noted that the amount of text to display was a challenge for the hyperbolic browser due to nature of the test data, for example, ill-structured data, long links causing visual overlap.

Lamping et al. (1995) carried out an experiment to compare their hyperbolic browser against a conventional 2D scrolling browser with a horizontal tree structure. According to them they utilised four World-Wide-Web hierarchies as the dataset. To measure task performance the four participants had to locate and double click on particular nodes in the hierarchy. The experiment applied a within-subject design. Their measurements for task performance showed no significant differences concerning time or the number of carried out user actions between the two methods.

Lamping et al. (1995) also carried out a post experimental survey for qualitative results. They explained that overall, all of the four participants preferred their hyperbolic browser. Three of the participants valued the ability to see more nodes at once. Two participants highlighted the more efficient use of screen space and the ability to view different structural attributes.

Worth noting that, in their comparative study, Andrews and Kasanicka (2007) described that there were no significant differences between the four techniques including the hyperbolic browser regarding task performance except for a single task. However, based on their subjective ratings the participants consistently preferred to use the tree view browser compared to the other techniques.

According to Lamping et al. (1995), a user can click on a visible point to change the focus of the visualisation. Also, they enabled the user to drag a visible point to any other position. Moreover, a user can select individual nodes.

I consider their technique to be a good option to support overview orientated tasks and for object selection in general. However, I argue that areas of the available screen space are not utilised because of their circular display region. Also, for text-based datasets (labels) their method may be subject to visual clutter. Therefore, this technique may be more efficient for labels with a fixed aspect ratio like image thumbnails. Still, their projection approach enables the user to view structural information while retaining enough information for precise object selection. Also, their novel interpretation of a fisheye view supports overview orientated tasks because a user can view a large section if not the whole tree at once.

## 2.4 3D Visualisation Techniques for Hierarchical Data

### 2.4.1 Degree-of-Interest Trees (Context+Focus)

Card and Nation (2002) proposed *Degree-of-Interest trees* which they described as a type of tree using context+focus visualisation techniques and degree-of-interest calculations. They explored a strategy for an *Attention-reactive User Interface (AUI)* which according to them consists of two parts. One part is a method to continuously predict a user's Degree of Interest (DOI) while the other part is a visualisation technique to show information based on the DOI calculation, for example, to show or hide certain objects.

Card and Nation (2002) propose a Degree-of-Interest tree solution that utilises a method to keep the trees visualisation within a predefined bounding box and combines various context+focus techniques. They explained that their method computes a user's DOI estimates utilising their enhanced DOI calculation, for instance, also considering order distance from the focus node. It omits nodes with a low DOI and scales nodes geometrically according to DOI. Also, nodes are scaled semantically according to the node size while large unexpanded branches of the tree are visualised as clusters. Moreover, they used animated transitions to highlight changes in the tree.

According to Card and Nation (2002), Figure 2.8 shows their Degree-of-Interest tree visualising an uniform tree. They detailed that in this example, coming from node 1, node 3 has been selected which changed the DOI calculation for the nodes. Consequently node 1 is now visually smaller while the size of node 3 increased. The nodes below node 3 also increased in size based on the DOI calculations. The transitions are animated to support the user, for example, to keep their orientation.

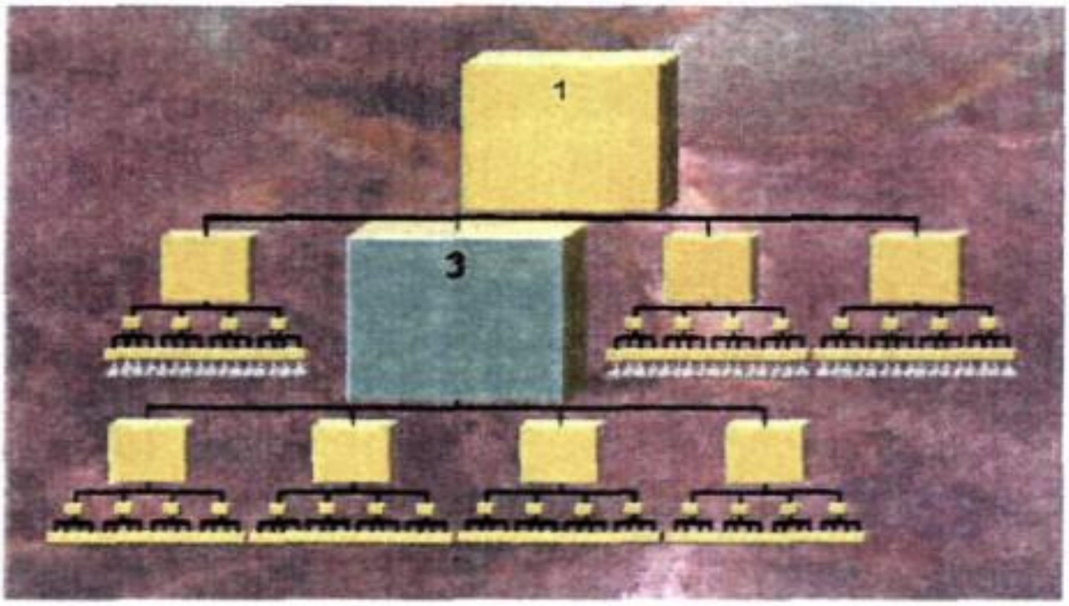



Figure 2.8: Example of a Degree-of-Interest Tree Focusing on Node 3

*Note*. From "Degree-of-interest trees: a component of an attention-reactive user interface" by S. K. Card and D. Nation, 2002, *Proceedings of the Working Conference on Advanced Visual Interfaces*, p. 239 (https://doi.org/10.1145/1556262.1556300). Copyright 2002 by the Association for Computing Machinery. Reprinted with permission.

According to Card and Nation (2002) they additionally utilised compression in x and y direction when the tree is too large to visualise. However, they did not describe precise node limits for visual clutter.

Card and Nation (2002) used their technique to visualise a 7,000 node taxonomic database. However, they did not evaluate if their Degree-of-Interest tree was still usable with this amount of nodes.

Card and Nation (2002) described that they implemented semantic zoom. They explained that when a node gets selected the content of it and surrounding nodes (DOI) is shown in varying detail. For instance, smaller nodes may only display some data but not all of it omitting certain entries or show just word abbreviations. Additionally, users can rotate nodes (rendered as 3D boxes) to display more information by dragging the mouse horizontally. According to them they allowed a maximum of three faces for one node (front, left, right). Also, they implemented a search feature to show search results in context to their surrounding nodes.

I consider their technique to be a valid option to support overview tasks and object selection. Their technique does utilise the available screen space very well because of their DOI approach in combination with its space filling layout. The method enables a user to view structural information for the whole tree while retaining enough information for precise object selection when a region is in focus. Because the user can view the whole tree and its structure it does support overview orientated tasks very well.

The method shows node labels in varying detail based on the DOI. Still, depending on the use case word abbreviations for long text labels may not always be suitable to convey their meaning. However, they offer the option for a detail view by allowing a user to rotate nodes for additional information to be shown. I argue that this type of rotation can be interpreted as a type of scrolling to show more information if necessary.

Regardless, for large datasets the technique may no longer be a viable option because the structural information may get lost, for example, when nodes become too small.

### 2.4.2 Document Lens (Context+Focus)

Robertson and Mackinlay (1993) introduced an interaction technique to understand structurally unknown paper documents. They described that they utilised 3D rendering to more effectively use the available screen space. Also, they aimed to use interactive animation and to show a combined detail and overview (context+focus) visualisation. They called their technique *Document Lens*. Robertson and Mackinlay (1993) explained that existing techniques like a traditional magnifying lens face various challenges, for example, they obscure parts of a document. Also, they noted that fisheye lenses tend to distort the text decreasing its readability.

Robertson and Mackinlay (1993) described their Document Lens as a rectangular magnifying lens with elastic sides that pull surrounding text toward the lens resulting in a cropped pyramid. They highlighted that their technique allows for the whole text document to be viewed as a whole while magnifying a rectangular area of interest.

According to Robertson and Mackinlay (1993) Figure 2.9 shows their technique where the whole document is visible while the focus area is magnified. They highlighted that the text in and near the lens remains readable. Also, they noted that the highlighted regions remain visible to preserve context. Moreover, they highlighted the efficient use of screen space.

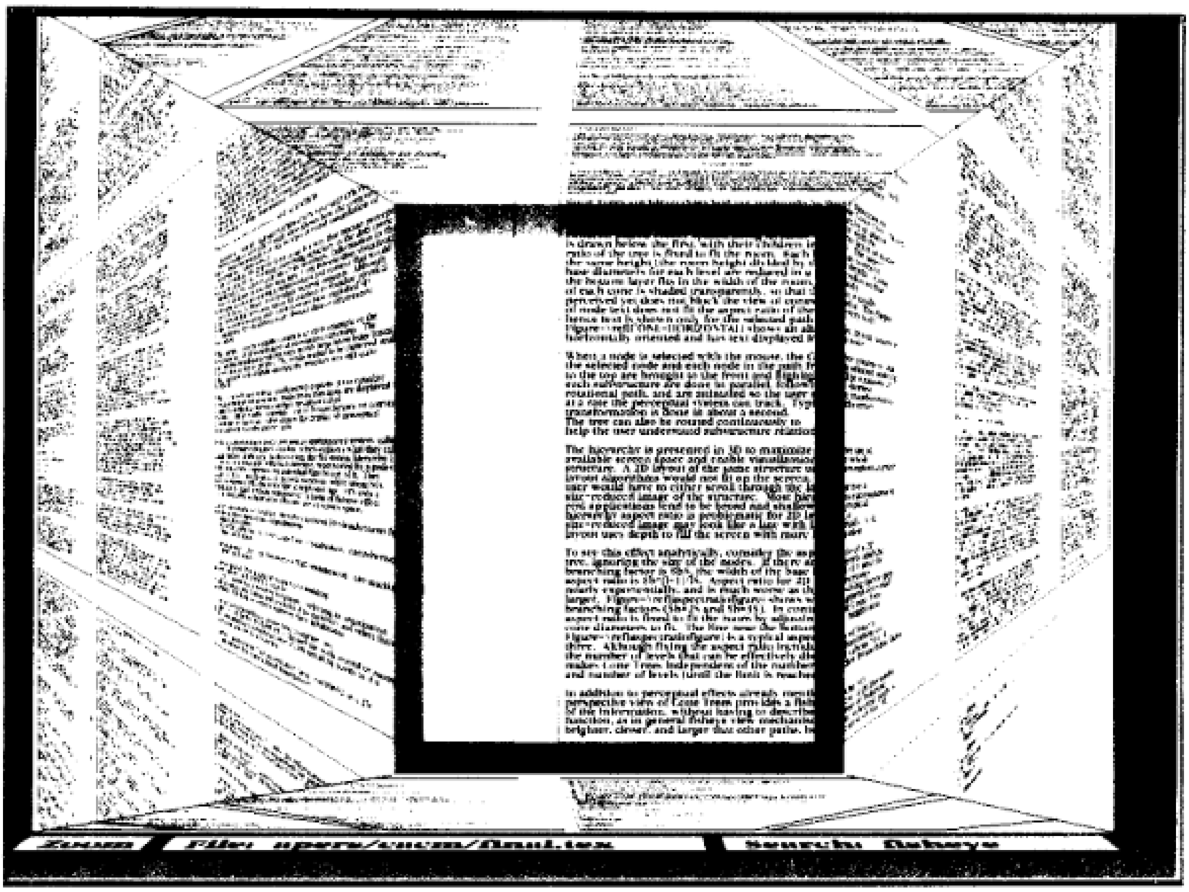


Figure 2.9: Document Lens With Lens Pulled Toward the User

*Note*. From "The Document Lens" by G. G. Robertson and J. D. Mackinlay, 1993, *Proceedings of the 6th annual ACM symposium on User interface software and technology*, p. 105 (https://doi.org/10.1145/168642.168652). Copyright 1993 by the Association for Computing Machinery. Reprinted with permission.

Robertson and Mackinlay (1993) described that they enabled mouse interaction to move the lens horizontally and vertically. Additionally, the keyboard can be used to move the lens in the z plane (forward, backward). They highlighted that the lens's movement utilises interactive animation. Moreover, the size of the lens is fixed in their implementation.

For potential usage scenarios Robertson and Mackinlay (1993) described the visualisation of search results. When using their technique search results highlighted by colour enable the user to identify patterns in the whole document. Additionally, they described various technical challenges, for example, the need to couple lens and viewpoint movement so the lens does not move out of sight. Also, they highlighted that their Document Lens is applicable to more scenarios than just viewing documents. They explained that it could also be used for 2D graphs like a diagram.

Based on the methods layout I argue that it does utilise the available screen space very well. However, the ideal size of the lens and surrounding regions may differ depending on the use case or dataset. When visualising a node-link type tree like the tree view (see Section 2.3.1) adding this technique may convey structural information very well because the whole tree and its links may be visualised. Such a scenario may also support overview orientated tasks while the lens offers enough support for precise object selection. Still, based on the usage scenario and dataset a user (if technically possible) may be forced to adjust the size and zoom level of the lens regularly to identify a setup most usable to them. Hence, this challenge is quite similar to identifying the ideal setup of a Textual Fisheye Tree View (see Section 2.3.9). Also, the support for long text labels depends on the size of the lens and zoom level, for example, a label may no longer be readable when it does not fit within the region of the lens.

### 2.4.3 Perspective Wall (Context+Focus)

Mackinlay et al. (1991) introduced the *Perspective Wall* a 3D visualisation technique for linear data. According to them their technique folds a 2D layout into a 3D wall visualisation. Their wall integrates one region to view details and perspective regions for overview (context+focus). They explained that they utilised a physical metaphor of folding to transform or distort a 2D layout into a 3D one for their design.

According to Mackinlay et al. (1991) the *Bifocal Display* (see Spence & Apperley, 1982) is a 2D conceptual ancestor to their Perspective Wall. They identified various challenges when using a Bifocal Display like no smooth integration of context+focus, the requirement to create multiple versions of an item (distorted, detail view).

According to Mackinlay et al. (1991) Figure 2.10 shows their Perspective Wall. They explained that it does not require specially scaled versions of items and retains task specific features. The centre panel displays details while the two perspective panels (left, right) show context (overview). The cards in the figure represent files structured by their modification date (horizontal axis) and file types (vertical axis). Because of the perspective projection cards closer to the centre (detail) panel appear larger. When an item is selected by the user the view gets centred on it. These transitions are animated.

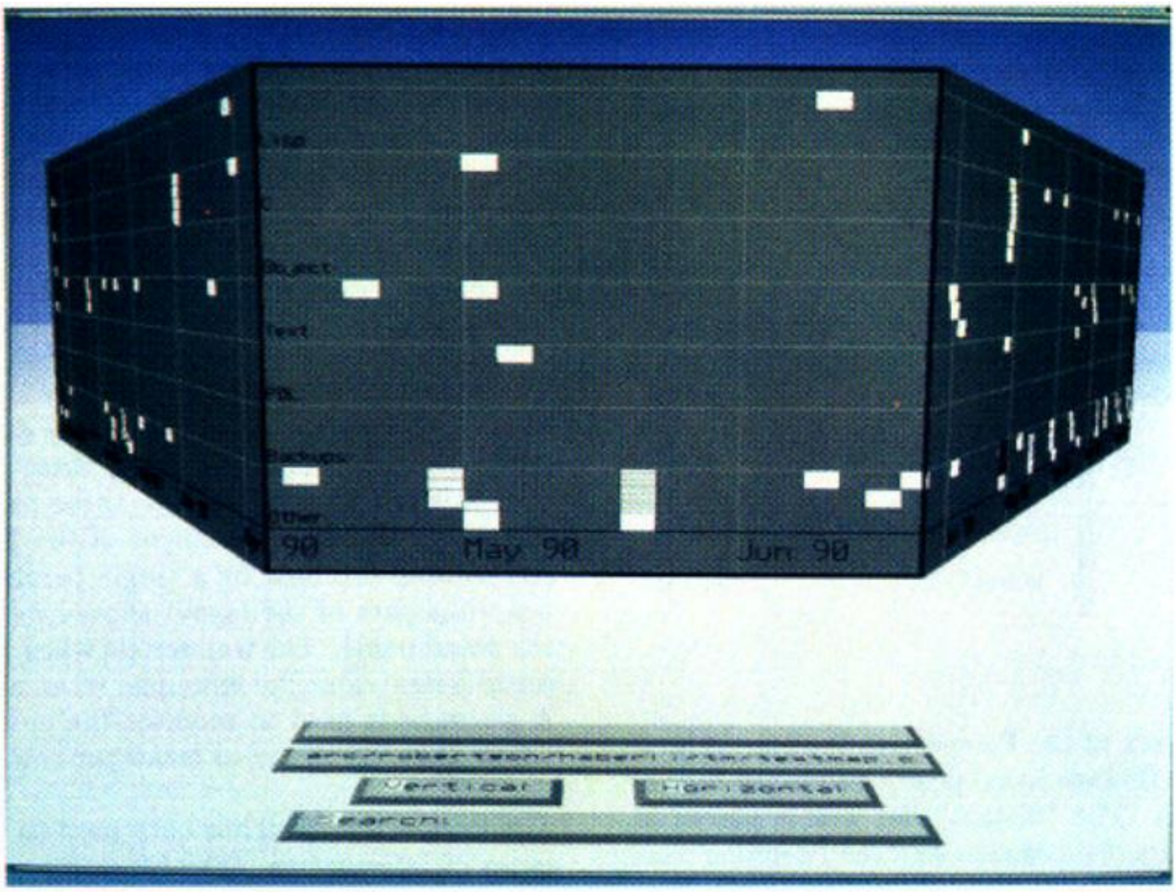



Figure 2.10: Mackinlay Plate 1

*Note*. From "The Perspective Wall: Detail and Context Smoothly Integrated" by J. D. Mackinlay, G. G. Robertson and S. K. Card, 1991, *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, p. 177 (https://doi.org/10.1145/108844.108870). Copyright 1991 by the Association for Computing Machinery. Reprinted with permission.

According to Mackinlay et al. (1991) a user is able to select an item. Consequently, the selected item is moved to the centre panel (detail). Also, they allowed the user to adjust the ratio between the centre and perspective panels. They explained that this feature can be used to increase the screen space for the centre (detail) panel when needed.

Mackinlay et al. (1991) concluded that their Perspective Wall allows for an efficient use of available screen space. Also, they argued that the view can be smoothly and intuitively adjusted.

Worth noting that, even though the Perspective Wall is a visualisation technique (described) for linear data its core folding metaphor applies to arbitrary 2D layouts (Mackinlay et al., 1991).

I argue that this method does not utilise the available screen space very well because of the large rather empty areas surrounding the actual visualisation. In general, I consider this method to be very similar to the Document Lens regarding its capabilities (see Section 2.4.2). The essential difference being that instead of a lens this method utilises panels to achieve a similar effect.

### 2.4.4 Cone Tree (Context+Focus)

Robertson et al. (1991) described a technique they called Cone Tree to visualise hierarchical data in 3D. According to them they rendered the tree in 3D to effectively use the available screen space (utilise depth) and to visualise the whole tree. They explained that the usage of a 3D perspective view also provides a form of a fisheye lens effect without the need for a DOI function. They argued that the selected path in their visualisation appears brighter, closer and larger than other (not selected) paths because they used a 3D perspective view, colouring and lightning. Additionally, they implemented selective rotation to focus on objects of interest. Also, they added shadows for the cones and nodes casted onto the (virtual) floor to add depth and convey structural information regarding the tree. Moreover, they utilised animation with the aim to reduce cognitive load, to support the user's understanding of the data structure and for entertainment.

Robertson et al. (1991) explained that their method is not very well suited to display node text given the aspect ratio of the single nodes (cards). Therefore, their technique only shows text for the user selected path.

According to Robertson et al. (1991) Figure 2.11 shows a simple Cone Tree visualising a user's directory tree showing their folders as individual nodes. They described that the nodes are rendered as index cards. The root node of the hierarchy is displayed at the top and forms the apex for the first cone with its children's nodes. Each vertical layer has cones of the same height. This universal height is calculated by dividing the (virtual) rooms height by the hierarchy's depth. The overall aspect ratio of the tree is fixed derived from the (virtual) room's size. The diameters of the cones for each layer are calculated in a way so that the content of the bottom layer still fits in the width of the (virtual) room. The cones are transparent to outline their structure and to prevent occlusion. When the user selects a node with the mouse this node is rotated to the front of the view. Consequently, all other nodes in the selected path are moved to the front and are highlighted. These rotations utilise animation to reduce cognitive load.

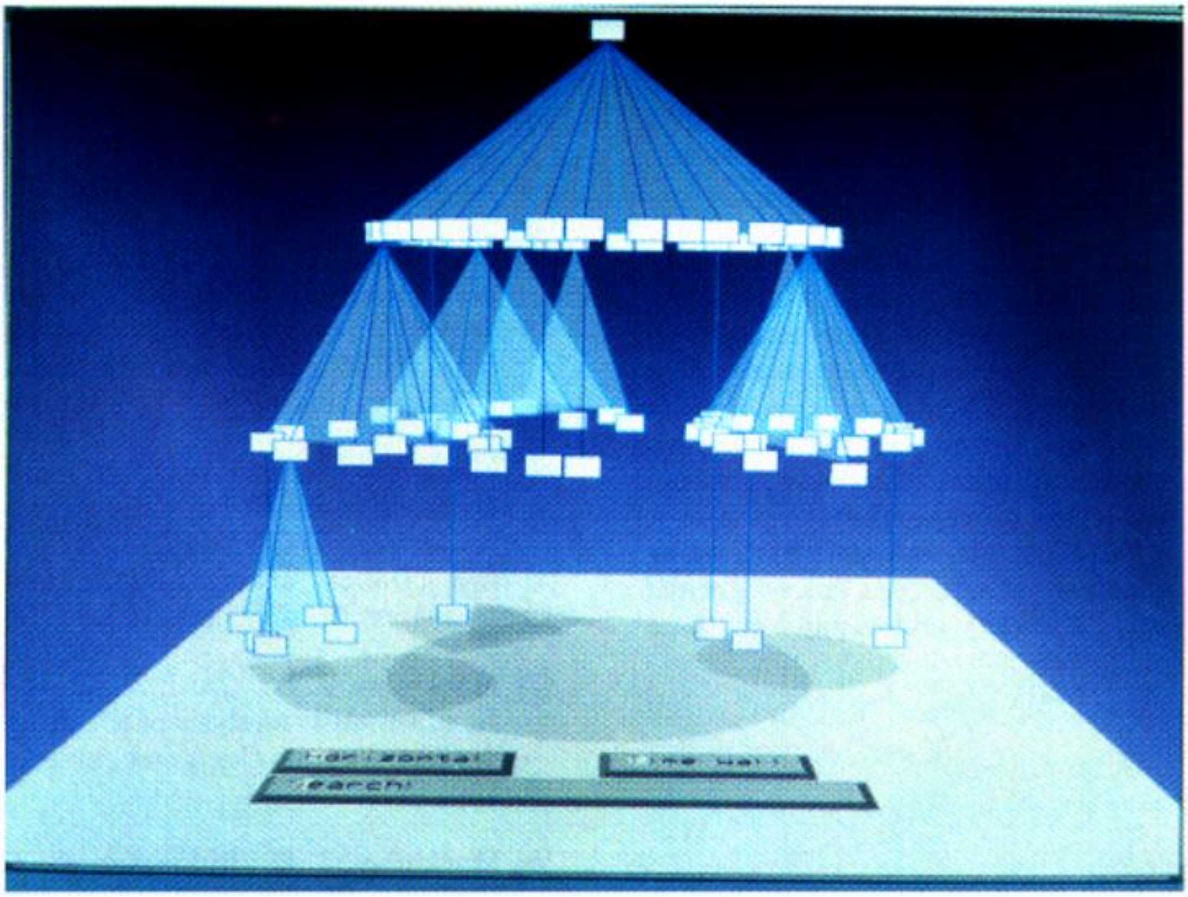


Figure 2.11: Robertson Plate 1

*Note*. From "Cone Trees: Animated 3D Visualizations of Hierarchical Information" by G. G. Robertson, J. D. Mackinlay and S. K. Card, 1991, *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, p. 193 (https://doi.org/10.1145/108844.108883). Copyright 1991 by the Association for Computing Machinery. Reprinted with permission.

Carriere and Kazman (1995) enhanced the Cone Tree approach by augmenting it with graphical and interaction techniques like animated zooming and usage-based filtering. They called their enhanced version of the Cone Tree *fsviz*. They described their system, as an example, in the context of a Unix file system hierarchy visualisation and interaction tool. They also introduced a feature to automatically coalesce nodes whenever a subtree is at a threshold distance from the user’s focus node to reduce visual clutter and improve rendering performance.

Sancho-Chavarria and Mata-Montero (2018) utilised the Cone Tree technique to visualise and manage hierarchies of biological taxonomies. They called their web-based 3D tool *Den3D*. They described that their system also utilises animation to support, for example, the user’s mental model during navigation.

According to Robertson et al. (1991) their Cone Tree becomes visually cluttered when datasets with more than 1,000 nodes, 10 layers or a branching factor over 30 are visualised.

Carriere and Kazman (1995) stated that their version (fsviz) can visualise up to 5,000 nodes without visual clutter. However, they explained that their fsviz implementation suffers from poor interactive performance (usability) when hierarchies larger than approximately 2,000 nodes are visualised.

Sancho-Chavarria and Mata-Montero (2018) carried out volume tests to Den3D. They concluded that trees with approximately 4,000 nodes still provided reasonable performance in regards to loading, creation and navigation (usability).

Carriere and Kazman (1995) carried out a small user test to measure task performance. They compared their version of the Cone Tree against the usage of Unix shell utilities for file system hierarchy navigation. The five users had to carry out three tasks for file navigation. They concluded that they see some indication that their enhanced Cone Tree might be useful for performing specific tasks concerning hierarchical data, for example, identify patterns of usage.

Sancho-Chavarria and Mata-Montero (2018) carried out a user study for quantitative and qualitative results to evaluate if their Den3D tool could assist in teaching biodiversity topics. They described that the participants were 12 high school science or biology teachers from different schools. They took part in a four-part workshop including a tutorial for Den3D, the creation of a new taxonomy with the tool, the design of a lesson plan for their respective students in groups and answering a questionnaire. The questionnaire consisted of nine quantitative questions and one qualitative (open-ended) one for suggestions. Based on their user study they concluded that Den3D might be a suitable tool for biology education. They argued that most participants considered their tool to be a good choice for hierarchy visualisation, an efficient educational tool and that it has the capacity to support biodiversity education.

According to Robertson et al. (1991) a user can select nodes with a mouse. Consequently, the selected node and all other nodes along the selected path are rotated to the front and visually highlighted. They described that a user can also rotate the tree continuously to better understand the relationship between different nodes.

Robertson et al. (1991) implemented a feature where a user can flick a node towards the top node to hide all of its child nodes. Consequently, this cut off node is modified by adding a *grow tab* below it. They described that when a user selects the grow tab or flicks the node away from the top the children are brought back into the visualisation. They also implemented a *prune others* menu operation to hide (prune) all the "siblings of the selected node" (Robertson et al., 1991, p. 191). Also, they enabled the user to drag and drop a node and its respective children to a new position. The respective target cone is highlighted while dragging to provide visual feedback. According to them they also implemented a search feature where nodes are highlighted with a red bar to indicate their search score. They explained that after the search the node with the highest search score gets selected.

According to Carriere and Kazman (1995) their enhanced version of the Cone Tree allows drag and drop functionality to move and copy subtrees. Also, they combined control panel buttons and selections in the tree to offer search and directory creation features. Moreover, they enabled the user to select nodes in the tree with a mouse. They explained that when a file is selected details for it are shown in a file information browser contained in the control panel. If the selected node has child nodes additional options are enabled to rotate the subtree. Additionally, they explained that the user can also magnify, minify or collapse subtrees. They also enabled zooming, for example, to a selected node. Also, to shift focus to a subtree the user can nominate any node to appear as the new root node in the visualisation.

According to Sancho-Chavarria and Mata-Montero (2018) the user can navigate in Den3D using keyboard commands providing the notion of moving in 3D space. Also, the user is able to zoom and to bring nodes to the front of the screen. Additionally, they provided a feature to temporarily hide or unhide subtrees. They explained that users can select nodes to show an image and the name of the taxon. Moreover, they implemented two more levels of detail when required, for example, including textual descriptions (details-on-demand). They explained that users can also use a search feature to export the search results in Microsoft Word format (extract).

Additionally, Sancho-Chavarria and Mata-Montero (2018) implemented various editing features to, for instance, store various trees (history), apply prune and snip operations for subtrees, create and edit nodes.

Robertson et al. (1991) highlighted the importance of animation in their technique to reduce a user’s processing load. They claimed that a Cone Tree is more suitable for unbalanced hierarchies. They argued that a rotating balanced tree might result in a uniform appearance of the subtrees and is therefore hard to follow from a user’s perspective.

Carriere and Kazman (1995) highlighted their contribution to Cone Trees. They argued that they were able to remove visual clutter because of their focus on node layout. Additionally, they argued that their feature to automatically coalesce nodes ensures the preservation of distant information.

Sancho-Chavarria and Mata-Montero (2018) highlighted their Den3D feature to divide trees into subtrees to overcome the 4,000 node threshold.

I consider the Cone Tree to be a very novel approach to visualise hierarchical data primarily because of their utilisation of 3D to utilise the available screen space. Therefore, it is well suited to support overview orientated tasks. However, regardless of features to coalesce nodes I argue that the visualisation is still subject to visual clutter when the number of nodes to visualise increases. This potential for visual clutter may also limit its capabilities to convey structural information, for example, when links are occluded by other objects. Also, there may be not enough screen space to visualise long text labels without overlap. Because of the layout, I do not consider the Cone Tree to be an ideal option when focussing on object selection. Hence, there is very little space left for textual information (node labels). Also, because of overlap it may be difficult for a user to select the object of interest.

### 2.4.5 H3 Layout (Context+Focus)

Munzner (1997) introduced their *H3 layout* technique, which can visualise large datasets (directed graphs) as node-link diagrams in 3D hyperbolic space. According to them they chose directed graphs as their visualisation target because many structures can be represented as node-link graphs, for example, directory structures. They explained that many directed graphs have an inherent hierarchical structure when domain specific knowledge is utilised. They call such graphs *hierarchical graphs*. Also, they highlighted that their technique works well with trees as a subset of hierarchical graphs. According to them hyperbolic space is infinite. Therefore, it allows the allocation of the same amount of space for each node in a tree regardless of its depth. Moreover, even though hyperbolic space is infinite it can still be projected onto a finite volume in euclidean space enabling a context+focus view. They argued that this view enables a user to see details near a current node of interest while distant features are still visible albeit distorted.

According to Munzner (1997) Figure 2.12 shows an example for a hierarchical graph rendered in 3D hyperbolic space. They detailed that text labels for nodes are culled automatically.

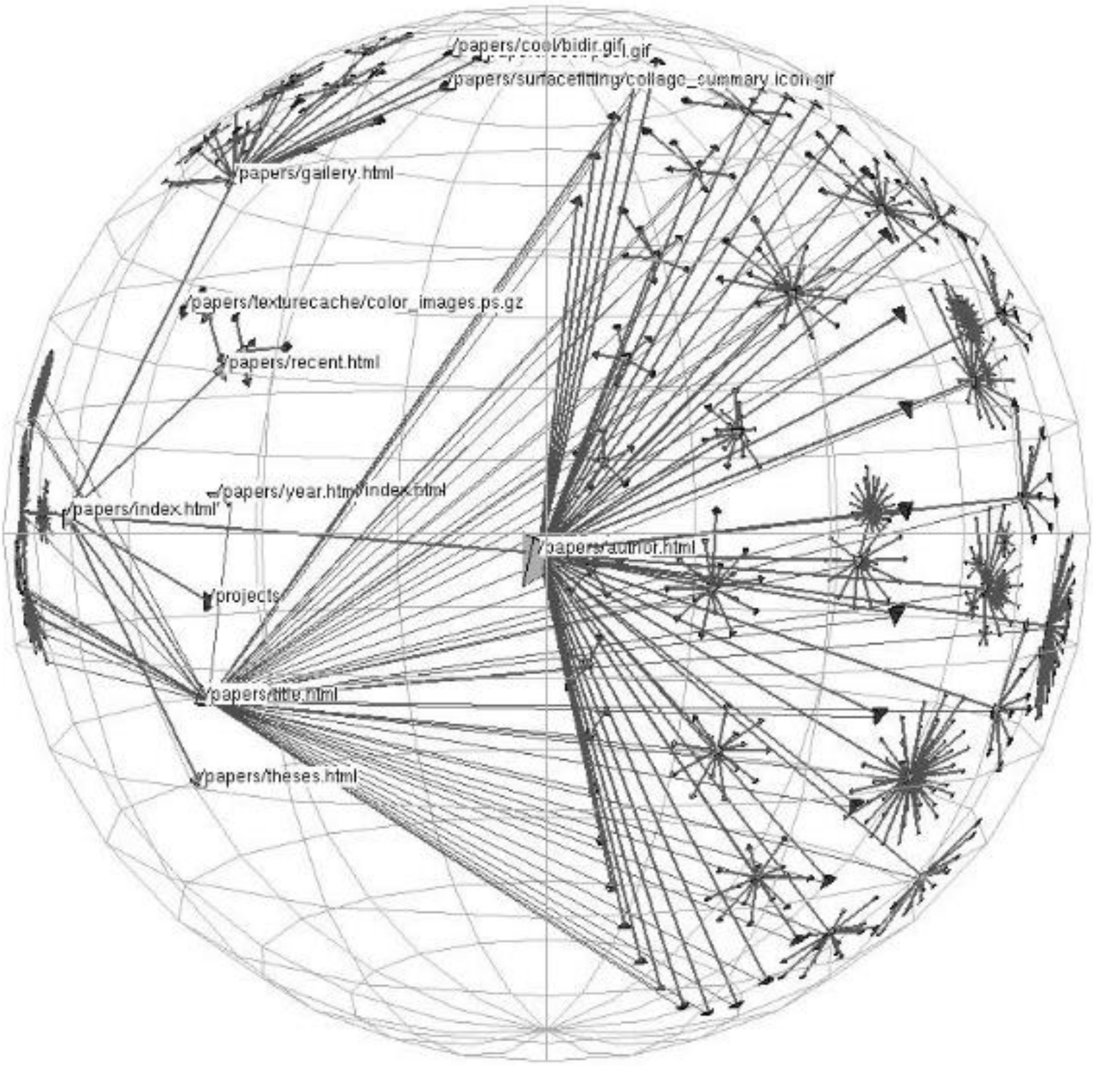



Figure 2.12: Example Showing a Part of the Stanford Graphics Group in 3D Hyperbolic Space Visualising Over 20,000 Nodes

*Note*. From "H3: laying out large directed graphs in 3D hyperbolic space" by T. Munzner, 1997, *Proceedings of VIZ '97: Visualization Conference, Information Visualization Symposium and Parallel Rendering Symposium*, 2.1 2D Graph and Tree Drawing section (https://doi.org/10.1109/infvis.1997.636718). © 1997 IEEE. Reprinted with permission.

Munzner (1997) tested their technique by visualising graphs with over 20,000 nodes. They explained that when a node is selected, it is translated to the centre of the sphere utilising animation. Additionally, rotation is applied to show the ancestors of the selected node on the left and its descendants on the right. Also, they implemented a feature to show or hide non tree links which leave or enter selected nodes using a toggle control.

According to Munzner (1997) their technique can effectively show aspects of a large graph, for example, overview and relationships but is not well suited for other tasks like selecting items from a linear list.

Hughes et al. (2004) introduced the idea to visualise a *phylogenetic tree* in 3D hyperbolic space. According to Fang et al. (2010) phylogeny describes the history of species while a phylogenetic tree also shows the relationships between different species. Hughes et al. (2004) concluded that 3D visualisations of phylogenetic trees in hyperbolic space may be a valid augmentation to other visualisations.

In general, I consider this method to be very similar to the Hyperbolic browser regarding its capabilities (see Section 2.3.10). However, I argue that the utilisation of 3D allows the H3 layout to show more structural information and objects. Therefore, it may be an even better option to support overview orientated tasks. However, similar to other 3D node-link visualisations it may be prone to visual clutter, for example, when links are occluded by other objects. This may get even worse when long text labels are part of the visualisation. Therefore, this technique may be more suitable for labels with a fixed aspect ratio like image thumbnails. However, because of the described risk of object occlusion I do not consider this method to be a good choice for selecting individual objects. Also, I argue that similar to the Hyperbolic browser the available screen space is not utilised very well because of the methods spherical display region.

### 2.4.6 HotSauce

According to Dodge (n.d.) Ramanathan V. Guha developed *HotSauce* while working for Apple Research in the mid-1990s. Continuing, Guha explained that HotSauce began as an experiment to visualise their large hierarchical *MCF* data structures in 3D. The MCF, short for Meta Content Framework, was also developed by Guha and is an approach to describe and organise information using object metadata. Moreover, when a user selects a MCF enabled website in their browser the HotSauce plugin renders the structure of this web page in 3D and enables a first-person perspective view.

The figure with the title “An example of the HotSauce interface, showing the Sailmaker Software website.” (Dodge, n.d., Introduction section) shows according to Dodge (n.d.) the HotSauce interface visualising a website. They explained that in this videogame like view the individual web pages are visualised a coloured labels in an infinite black space. The user is able to fly through this space using the mouse (direction of flight) and buttons (to move forward, backward). Individual web pages are visualised as rectangles including their page title. Broader topics are rendered as rectangles with rounded corners. The various hierarchy levels are highlighted by colour and depth. In this example, the green rectangle is the root node while the red ones represent the second level of the hierarchy.

According to Dodge (n.d.) the user is able to interact with the HotSauce interface by using the mouse to change the view direction and buttons to move forwards and backwards (variant of zooming). Additionally, users are able to double click on nodes (rectangles) to access those web pages.

Guha stated that one strength of their HotSauce interface is for entertainment: “it was ‘fun to use.’” (Dodge, n.d., Making the Web Flyable section). However, according to Dodge (n.d.) it is difficult to find web pages (content of interest) and to move towards those pages. Also, a user can easily lose orientation while navigating.

Johnson (1997/1999) described their attempt to use HotSauce to replace their file system. They found that the initial usage was very entertaining. However, it took them too much effort and attention to fly through the interface. Still, they concluded that the interface provides a glimpse into the future for (genuine) spatial systems.

Guha highlighted two key challenges they encountered while developing HotSauce (Dodge, n.d.). These were the creation of a non-linear layout of the hierarchical data in 3D and developing algorithms for fast text drawing.

I argue that the method does utilise the available screen space very well because it uses 3D to fan out the data in all three dimensions. However, it is prone to visual clutter because of object occlusion. Therefore, it may not be very useful when supporting overview orientated tasks. Also, it does not convey much structural information given its layout. Structural information is only conveyed through the colour of the nodes. The method does not support the visualisation of long text labels very well. Hence, long text labels may increase the risk of object occlusion. Therefore, this technique may be more suitable for labels with a fixed aspect ratio like image thumbnails. Regardless, because of their support for relative movement I still consider it a valid option for object selection. With this kind of control mechanism, a user may be able to fly through the data quite efficiently. However, to support a user's orientation the integration of some form of a map may prove beneficial (overview+detail). Also, to support precise navigation some form of gravitation may be implemented, for example, at a certain distance the users view may lock on to the closest node.

### 2.4.7 Information Cube

Rekimoto and Green (1993) introduced the *Information Cube* a 3D visualisation technique for hierarchical data. According to them their technique utilises transparency and nesting to visualise hierarchies. They explained that the large outer cube (the largest one) represents the root node while the next level nodes are visualised as cubes within it. Also, they utilised a design metaphor of a physical box or container.

According to Rekimoto and Green (1993) Figure 2.13 shows their Information Cube visualising around 1,500 files and 50 directories from a Unix's directory structure. They explained that text labels are shown on surfaces. The result can be viewed using conventional

2D monitors or head-mounted displays. All cubes are translucent allowing the user to see inside the cubes. A cube can contain various types of 3D objects like a 3D graph. They argued that their usage of semi-transparent rendering can help to reduce the complexity of the information visualised because it offers depth of information while maintaining an understandable structure.

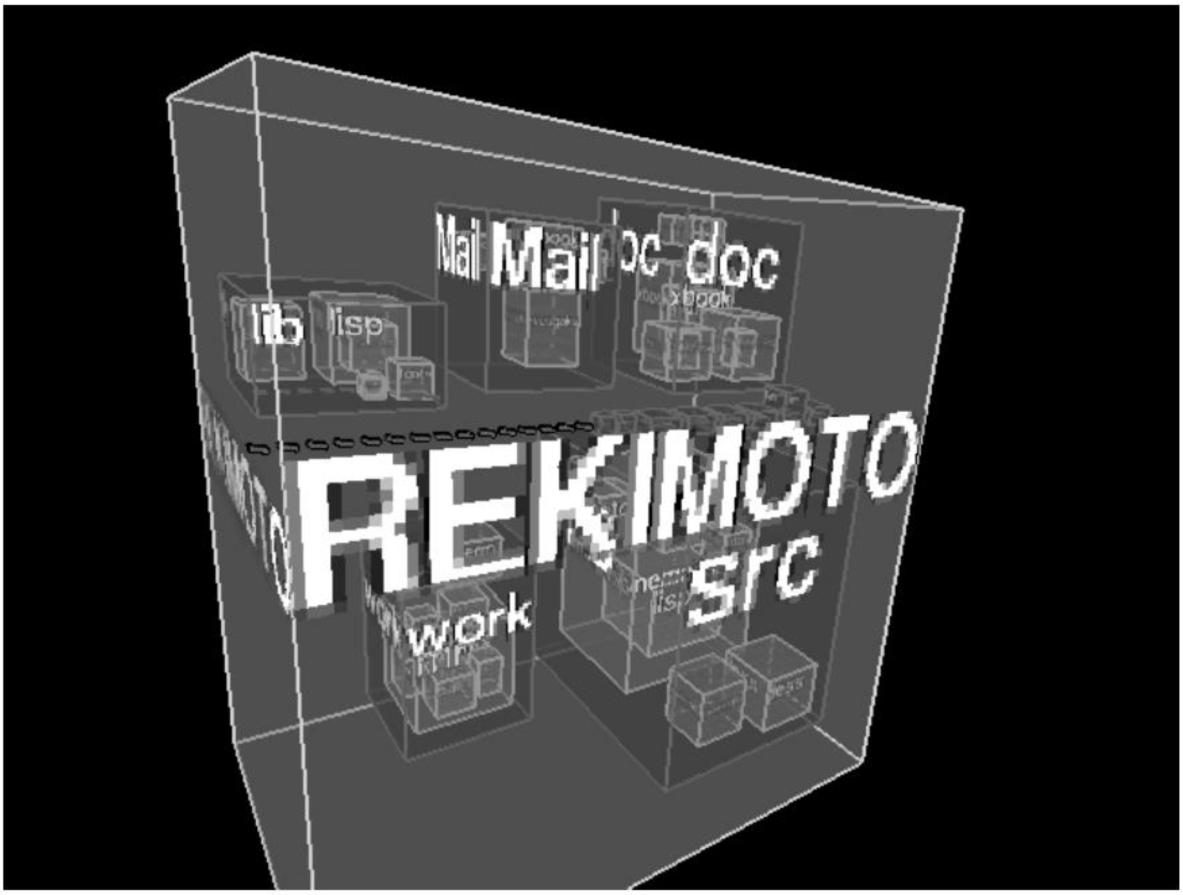



Figure 2.13: Displaying a Unix Directory Structure Using the Information Cube

*Note*. From "The Information Cube: Using Transparency in 3D Information Visualization" by J. Rekimoto and M. Green, 1993, *Proceedings of the Third Annual Workshop on Information Technologies & Systems (WITS'93)*, p. 127. Copyright 1993 by J. Rekimoto and M. Green. Reprinted with permission.

Regarding node limits for visual clutter, when comparing their method to the Cone Tree Rekimoto and Green (1993) argued that even 1,000 children nodes (sub objects) can be reasonably visualised with their Information Cube while the same number of objects results in an unnaturally spreading shape using the Cone Tree.

Regarding node limits for usability, when assessing the suitability for different datasets Wiss et al. (1998) concluded that for their implementation of the Information Cube the size of a cube does not represent the size of its contents very well (excess space) if there are fewer than $\lceil \sqrt[3]{n} \rceil^3$ children or if the children vary in size. Additionally, they found that when there is a large difference between the largest and the smallest subhierarchy the smallest child node (cube) can hardly be seen. They described the ideal data set for the Information Cube to be a hierarchy where all leaves are at the same level.

According to Rekimoto and Green (1993) a user can rotate and select objects using a *DataGlove*. A DataGlove is a physical glove to be used with a computer to project a real-time 3D model of the hand for the manipulation of computer-generated objects (Zimmerman et al., 1986). According to Rekimoto and Green (1993) they restricted the rotation to only the z axis to improve the users sense of orientation. Moreover, they described that when using a conventional 2D monitor, for example, CRT, as the output device they also utilised a 6D input device called *3SPACE ISOTRAK* to control the eyes position and orientation. The 3SPACE ISOTRAK by Polhemus Inc. is an instrument capable to measure relative 3D joint rotation (McGill et al., 1997). According to Rekimoto and Green (1993) for both output devices (2D monitor, head-mounted display) their method requires the use of the DataGlove. Also, they explained that when an object (cube) is selected the user moves (animated transition) towards it (zoom).

Wiss et al. (1998) stated that it is not possible to maintain global context (overview) while zooming in using the Information Cube.

I consider the Information Cube with its nested approach to be very similar to the Treemap (see Section 2.3.2) or Circle Packing (see Section 2.3.3).

Their 3D layout and usage of transparency supports conveying structural information. Additionally, because of their usage of 3D the method utilises the available screen space quite well. However, I agree with Wiss et al. (1998) that the visualisation may become unstable (excess space) depending on the dataset. Regardless, because of its nested approach combined with transparency the method may still be a well-suited option to support overview orientated tasks. However, depending on the sizes of the cubes there may be not enough space to show long text labels. Consequently, support for object selection strongly depends on the dataset to visualise. Also, it may be beneficial to add a form of map to support a user maintaining global context while navigating (overview+detail).

### 2.4.8 SemNet (Context+Focus and Overview+Detail)

Fairchild et al. (1988) introduced *SemNet* a 3D interface to interact with large knowledge bases. Knowledge bases are “large data bases of symbolic knowledge” (Fairchild et al., 1988, p. 202).

Fairchild et al. (1988) stated that their design target for SemNet was to aid users in understanding the structure of large knowledge bases. Therefore, their solution renders individual entities of the data as simple rectangles with labels while their relationships are shown as coloured arcs. Additionally, they highlighted their utilisation of heuristics to optimise object placement. Moreover, they applied various fisheye techniques to convey the logical structure of the data.

The figure with the title “Figure 3-1: The complete knowledge base of Prolog modules with nodes assigned to random positions” (Fairchild et al., 1988, p. 208) shows according to Fairchild et al. (1988) the relationships between Prolog modules. They explained that in this example each element was randomly assigned to a position, labels were removed from nodes and the nodes were reduced in size. According to them a map is shown with two 2D parts one for the x-y plane and the other one for the y-z plane. Also, the viewpoint position is visualised with an asterisk in both parts of the map. Moreover, the user can change the position of the viewpoint using their mouse by moving the asterisk in either part (plane) of the map.

According to Fairchild et al. (1988) they implemented a map to support a user's orientation (overview+detail). Also, they described their support for relative movement where a user can rotate the viewpoint and move forward and backward in the views direction using a mouse and keyboard. However, they outlined that users lost orientation when using this type of (helicopter) movement. They speculated that this was caused by a lack of visual cues. Also, they described that this method of movement was slow because a user has to adjust multiple controls to reach their destination. They speculated that their relative movement implementation would be an efficient way for navigation when other input devices are used, for example, joystick and accelerator.

Also, Fairchild et al. (1988) supported absolute movement when a user points to a location on the map. According to them this method is faster and simple to use compared to their relative movement implementation. However, they described two disadvantages when using absolute movement in SemNet. For one there is a lack of precision when navigating and second there is as an additional cognitive demand on the user caused by the different views (2D map of 3D space).

According to Fairchild et al. (1988) there is a function to temporarily move all connected nodes of a selected knowledge object to its position. They explained that the moved nodes are positioned around the selected knowledge object. Alternatively, they highlighted that SemNet also allows the user to rotate the knowledge space while the viewpoint does not change. Additionally, they described that a user can revisit recently visited knowledge elements by selecting them from a 2D menu (history).

Concerning SemNet Robertson et al. (1991) commented that visualisation results tend to be cluttered and it is difficult to understand the structure of the data visualised.

Worth noting that, even though SemNet was described as a system to visualise and interact with knowledge bases (network data type) Fairchild et al. (1988) implied more potential use cases like browsing hypermedia documents or visualising the structure of large projects.

I argue that the system does utilise the available screen space very well because of its usage of 3D and space filling approach. However, I agree with Robertson et al. (1991) that the visualisation may be subject to visual clutter. Still, similar to other 3D node-link visualisations it may be able to visualise more objects and convey more structural information than 2D alternatives. Also, the systems map may support overview orientated tasks. Consequently, I consider it to be a good option to support overview orientated tasks. However, it is not a well-suited option for visualising long text labels because those may cause even more visual clutter by occluding other objects. I do not consider SemNet to be a well-suited option for object selection due to visual clutter and its outdated control schemes, for example, the helicopter movement.

### 2.4.9 Cam Tree (Context+Focus)

In addition to the Cone Tree Robertson et al. (1991) also introduced the *Cam Tree*. According to them it is an alternative horizontal layout compared to the vertical layout of the Cone Tree. They explained that with this horizontal layout they were able to display text for each node in the tree.

Robertson et al. (1991) described that the Cam Tree uses a simple 2D projection to render shadows to convey information regarding the hierarchy. In contrast they utilised the shadows of the Cone Tree to convey information regarding the clusters of a hierarchy.

According to Robertson et al. (1991) Figure 2.14 shows a Cam Tree after a node was selected. They explained that nodes are visually highlighted using colour. Also, they stated that this figure shows a user’s directory tree visualising their folders as individual nodes.

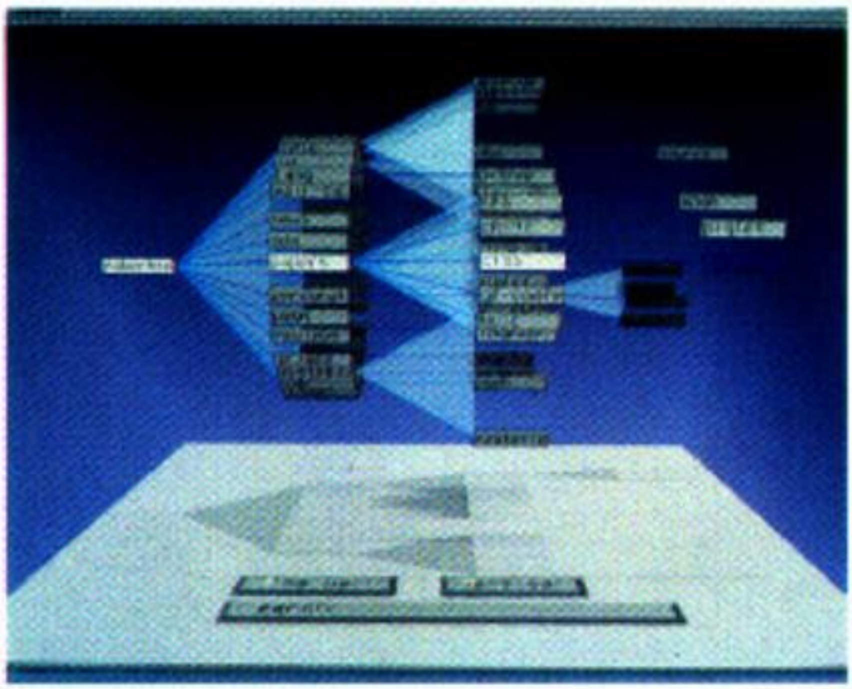


Figure 2.14: Robertson Plate 4

*Note*. From "Cone Trees: Animated 3D Visualizations of Hierarchical Information" by G. G. Robertson, J. D. Mackinlay and S. K. Card, 1991, *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, p. 193 (https://doi.org/10.1145/108844.108883). Copyright 1991 by the Association for Computing Machinery. Reprinted with permission.

Worth noting that, albeit not explicitly stated I assume that the node limits for visual clutter described by Robertson et al. (1991), their methods for interaction, rendering (for example, animation, rotation) and general remarks are the same as for the Cone Tree (see Section 2.4.4).

Regarding node limits for visual clutter, when assessing the suitability for different datasets Wiss et al. (1998) concluded that for their implementation of the Cam Tree datasets with many levels and subhierarchies are prone to occlusion. They explained that for both of their visualised datasets many subtrees were occluded. Consequently, they stated that viewing data is a challenge even for their small and balanced dataset.

In general, I consider this method to be very similar to a Cone Tree regarding its capabilities (see Section 2.4.4). In contrast to the Cone Tree, I consider the Cam Tree to be a better option for selecting objects when visualising text-based datasets because of its horizontal layout. However, depending on the depth of the hierarchy and the length of the text labels to visualise the screen space may not be enough to visualise the whole tree. Potential mitigating measures may be the usage of word abbreviations instead of full text.

### 2.4.10 Information Landscape (Overview+Focus)

Tesler and Strasnick (1992, as cited in Andrews, 1995) at Silicon Graphics developed the "File System Navigator (FSN, or 'Fusion')" (Tesler & Strasnick, 1992, as cited in Andrews, 1995, p. 98), short *FSN*, to visualise a file system as a landscape of information. According to Tesler and Strasnick (1992, as cited in Andrews, 1995) in this system directories are rendered as blocks on a plane with their height indicating the overall size of the incorporated files. They explained that blocks laid upon the directory blocks represent the individual files of the directory. Again, the height of each (file) block indicates the size of the file.

Similar to FSN Andrews (1995) developed a (client) visualisation technique they called *Harmony's Information Landscape*. They explained that at start-up FSN renders the whole hierarchy once. In contrast, their solution constructs (renders) the landscape incrementally during navigation. Additionally, they highlighted their utilisation of colours to indicate individual document types. Also, they coupled the 3D Information Landscape to a 2D browser display so changes in one display are mirrored onto the other (overview+focus).

Worth noting that, Andrews (1995) also described a data model for their *Hyper-G Internet information management system* (see Andrews, 1995) including structures like hierarchical collections and hyperlinks.

According to Wiss et al. (1998) Figure 2.15 shows their implementation of an Information Landscape. They explained that nodes are visualised as blocks on a flat surface.

The lines are connecting the blocks forming a tree. The leaves of a node are visualised as boxes placed upon the blocks. The surface area of the blocks gets adjusted based on the number of their children (boxes). Also, the height of the individual boxes visualises an attribute like the data size of the element.

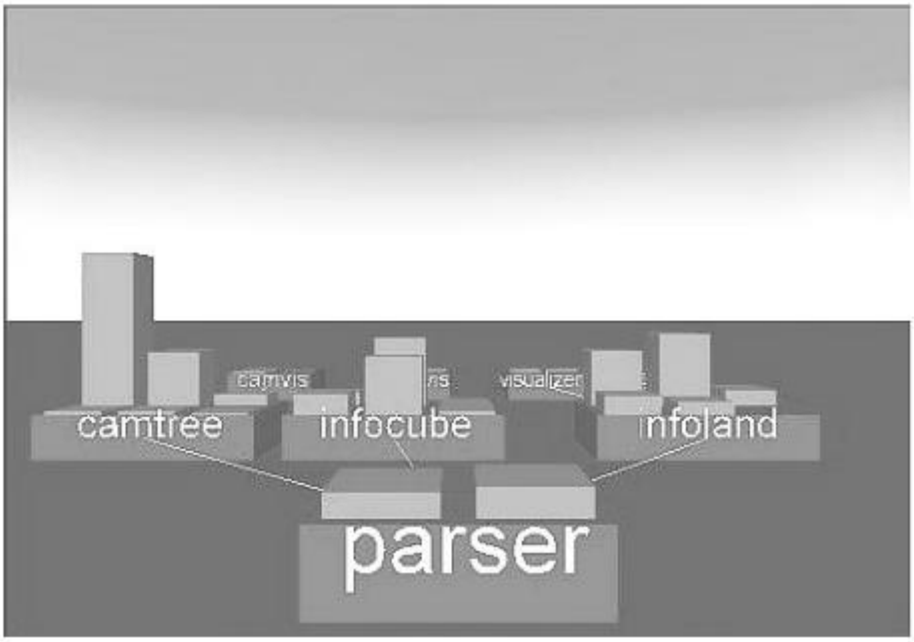



Figure 2.15: Implementation of the Information Landscape Visualizing the File System Data set

*Note*. From "Evaluating three-dimensional information visualization designs: a case study of three designs" by U. Wiss, D. Carr and H. Jonsson, 1998, *Proceedings. 1998 IEEE Conference on Information Visualization. An International Conference on Computer Visualization and Graphics (Cat. No.98TB100246)*, 4.2 The Client Application section (https://doi.org/10.1109/iv.1998.694211). 

Regarding node limits for usability, when assessing the suitability for different datasets Wiss et al. (1998) concluded that for their implementation of the Information Landscape datasets with subhierarchies varying in size are difficult to visualise. They explained that this kind of setup results in a wide arrangement where subtrees can no longer be viewed as a whole (lack of available screen space).

According to Andrews (1995) a user can fly across the landscape to identify features of interest. Additionally, they explained that they implemented a map shown in the top right to support a user's orientation (overview+focus). However, they did not specify how the user interacts with their interface, for example, by using a mouse or keyboard.

The implementation from Wiss et al. (1998) allows the selection of blocks or boxes with a mouse and to fly up to a position near the selected item.

Andrews (2002) also introduced a similar technique called information pyramids which grow upward as the hierarchy gets deeper to form a pyramid like structure. They explained that the main difference is that there is only one large plateau to visualise the root node while each child node gets stacked upon it, for example, sub directories. Worth noting that, in their comparative study, Andrews and Kasanicka (2007) described that there were no significant differences between the four techniques including the information pyramid browser regarding task performance except for a single task.

Regardless, when focusing on the Information Landscape I claim it is an interesting method. However, I argue that it does not utilise the available screen space very well because it fills the landscape with objects but leaves the sky quite empty. Also, their method may be subject to visual clutter depending on the camera angle as objects may occlude each other. The method conveys structural information very well. This is achieved by showing clear links between nodes and combining this with a form of nested approach for the leaves. However, I do not consider their technique to be a valid option to support overview orientated tasks. The potential low camera angle occludes much of the structure. Also, I agree with Wiss et al. (1998) that for datasets with subhierarchies varying in size the visualisation may become challenging. Moreover, I argue that when a node has many leaves (boxes) the surface area of the blocks (node) may become too large to visualise without losing almost all structural information. In contrast, the visualisation of long text labels may be a challenge when the block is too narrow. One mitigation strategy may be to set the width of the text label to a fixed size. However, this may result in overlap or larger areas of unused screen space. Still, for object selection the method may be a valid option because of its consistent layout and the option to quickly identify attributes of interest, for example, file size.

In conclusion, when the method is viewed as a metaphor to navigate through data similar to moving around a landscape or city in real life, I consider it to be a very novel approach indeed. However, I argue that there may be more potential utilising this metaphor, for example, a city map may support a user's orientation (overview+detail) or clouds in the sky may show additional objects.

### 2.4.11 Beamtree

Van Ham and Van Wijk (2002) introduced a new technique to visualise large sets of hierarchical data called the *beamtree*. According to them their solution was inspired by Treemaps. However, they aimed to visualise the data structure explicitly with a new algorithm. They explained that their beamtree technique utilises overlap to indicate relationships between items (parent-child). Their solution visualises this overlap as a stack of rectangles with applied shading in 3D space. They utilised shading and 3D to convey a stronger perception of depth for the hierarchical structure.

According to Van Ham and Van Wijk (2002), Figure 2.16 shows their approach to construct a beamtree. They explained that in part b of the figure, a Treemap is shown where each node is related to a rectangle region. Next, in contrast to a Treemap every node of the tree is visualised by scaling their rectangles. When the scaling factor decreases the leaves are visualised as thin rectangles (d). To address this challenge, a subdivision algorithm is applied to the leaves instead of scaling them (e). Additionally, the child nodes get sorted so all leaves are at the left side or top of their parent node (f). Also, they highlighted their usage of a constrained length scale factor to remove the touching edges from non-leaf nodes. Further they applied shading to improve readability.

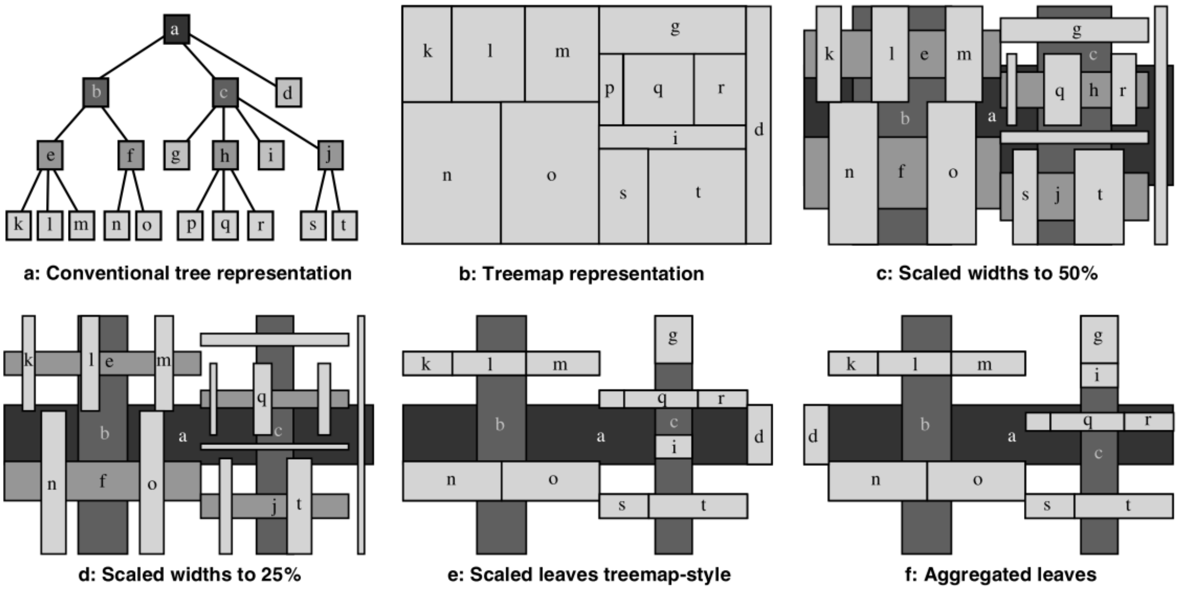



Figure 2.16: Scaling a Treemap to a Beamtree

*Note*. From "Beamtrees: compact visualization of large hierarchies" by F. Van Ham and J. J. Van Wijk, 2002, *IEEE Symposium on Information Visualization, 2002. INFOVIS 2002.*, 3. Beamtrees section (https://doi.org/10.1109/infvis.2002.1173153). © 2002 IEEE. Reprinted with permission.

According to Van Ham and Van Wijk (2002) they were able to render a beamtree with more than 25,000 nodes offering real time performance. They described that they used 1D texture maps instead of 3D models to visualise leaves using a PC with a GeForce2 GPU.

Van Ham and Van Wijk (2002) carried out a (small pilot) user study comparing their beamtrees (separate 2D and 3D version) with two other Treemap techniques. They measured task performance and error rates. They explained that their five tasks covered topics like identifying the largest leaf nodes and indicating the total depth of the tree (number of levels). They explained that each of the 12 participants (their co-workers) had to carry out all five tasks for a small (around 200 nodes) and a large tree (around 1,000 nodes) for all four different visualisations. Also, they chose to hide node labels (text) so participants had to focus on the visualisation. However, they described that they added a popup tooltip feature to view detailed size information for a node at the mouse’s current position (mouse over). At the end of their tests, they asked the participants to rate the different methods based on personal preference.

According to Van Ham and Van Wijk (2002) the results showed an indication that 3D beamtrees may perform significantly better than usual Treemaps for understanding tree structures (global structural information), for example, to identify the maximum depth or balance. Also, they highlighted that most participants had a strong personal preference towards the 3D beamtree.

According to Van Ham and Van Wijk (2002) a user can rotate the beamtree structure. Also, they implemented a button to return to a top-down view. Moreover, they added a feature for the user to change their viewpoint in 3D space. Furthermore, they explained that they implemented a layered view where each beam is visualised at a depth proportional to its depth in the tree. Also, they highlighted that their solution utilises animation when a user enters or exits this layered view.

I consider their method to be a novel alternative to Treemaps. Because of its scaling approach it may not be an ideal choice to utilise the available screen space. However, in comparison to a Treemap it provides structural information through its layout. Still, I argue that similar to a Treemap there are node limits to this approach, for example, rectangles to become very small. Regardless, it is suited to support overview tasks because of its primarily nested visualisation. However, it may be challenging to visualise long text labels for nodes because of the methods usage of scaling. Hence, there may be not even enough space to show any text labels at all. Concerning object selection, I do not consider this method to be a good choice because for large datasets the rectangles may become too small to select them. Also, it may be difficult for a user to even identify items of interest in such a scenario.

## 2.5 Visualisation Techniques on Mobile Devices for Hierarchical Data

### 2.5.1 Enhanced Radial Edgeless Tree

Chhetri et al. (2015) presented *ERELT (Enhanced Radial Edgeless Tree)*, their visualisation technique for hierarchical data on mobile devices. According to Chhetri et al. (2015) most hierarchical structures are visualised as lists on mobile devices. They described that their technique aims to resolve two challenges for visualising and navigating hierarchies on mobile devices. These are the optimal (maximal) usage of available screen space and interaction methods to enable fast navigation and exploration.

Their ERELT technique is an enhanced version of the *Radial Edgeless Tree (RELT)* method (Chhetri et al., 2015). RELT was first introduced by Hao et al. (2007).

According to Chhetri et al. (2015) RELT occasionally produces nodes with a strange shape, for example, for a tree with only one or two children.

According to Chhetri et al. (2015) their ERELT technique produces a more coherent shape for nodes and their labels. They described that they implemented (threshold-based) mechanisms to hide parts of the tree to support usability, for instance, to prevent node labels to become unreadable. Moreover, their ERELT algorithm draws only a sub-tree at a time.

Chhetri et al. (2015) utilised a colouring algorithm to colour nodes dynamically. They applied different colours to support the visual structure of the result and to avoid confusion. Also, they explained that their design consists of visually long and narrow nodes. Therefore, the text labels can utilise the nodes length to provide a consistent style supporting the user scanning the hierarchy.

According to Chhetri et al. (2015) Figure 2.17 shows a sample music library visualised with their ERELT technique. They explained that their technique divides the screen into multiple rectangles. Moreover, added radial lines separate individual nodes.

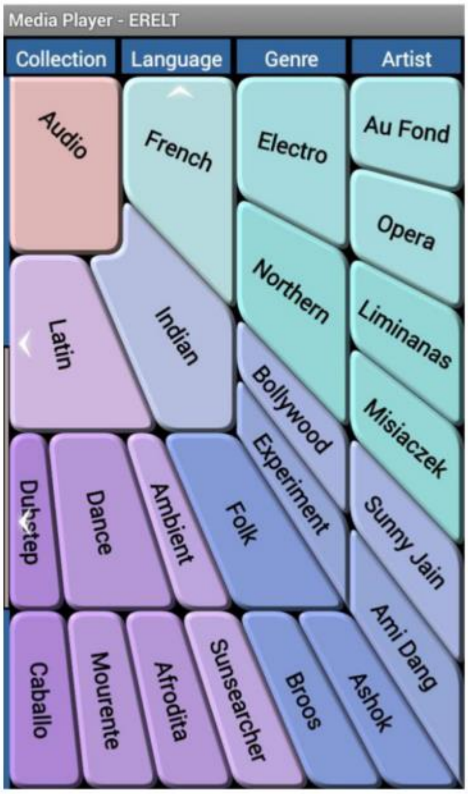



Figure 2.17: ERELT Visualization of a Structure

*Note*. From "A mobile interface for navigating hierarchical information space" by A. P. Chhetri, K. Zhang and E. Jain, 2015, *Journal of Visual Languages & Computing*, *31*, p. 57 (https://doi.org/10.1016/j.jvlc.2015.10.002). Copyright 2015 by Elsevier. Reprinted with permission.

Regarding node limits for visual clutter, I claim there is no such limit because of their threshold-based mechanisms to hide parts of the tree (scrolling).

Concerning potential node limits for usability, according to Chhetri et al. (2015) the computation time for ERELT is independent from the trees size because they only draw one sub-tree at a time. Therefore, I assume that there are no node limits regarding usability.

Chhetri et al. (2015) carried out a user study to compare their method against a list-based interface for navigating hierarchies. They recruited 38 (mostly) university students to participate in their study. Also, they explained that they used a media player application for their study for which they created a custom hierarchical music library. According to Chhetri et al. (2015) their study consisted of four parts. First, they tested the participants understanding regarding their ERELT technique by asking them five basic questions, for example, identify Recordings. The second part consisted of five questions regarding the tree structure, for instance, count albums for a given name. They designed these tasks to offer participants an opportunity to explore the music library and the application. Third, they asked the participants to carry out six timed tasks, for example, find and add tracks. They explained that they also recorded the number of touches each participant carried out to complete a task. Fourth, they gathered feedback utilising a survey. They gathered qualitative data (for instance, input on missing features) and quantitative data (for example, a difficulty rating on a scale from one to ten) with this survey. They noted that they out scoped the results from 12 participants for various reasons, for example, due to incomplete tests.

According to Chhetri et al. (2015) their results showed that ERELT required fewer touches and in most cases provided a comparable or faster performance. Therefore, they concluded that ERELT provides a better user interface compared to list-based alternatives when exploring or searching in data hierarchies. According to them 85% of their participants felt that ERELT was faster to work with. Their difficulty rating results indicated that their method was relatively easy to use for the participants.

According to Chhetri et al. (2015) a user can utilise touch-screen controls including gestures. Also, the user can tap to navigate to hidden tree levels (level threshold) while scroll gestures allow the navigation to sibling nodes (branching threshold).

Worth noting that, according to Chhetri et al. (2015) the level threshold is a set integer for the tree levels displayed at once while the branching threshold is a set integer for the maximal number of child nodes shown for each parent.

Chhetri et al. (2015) also implemented drag navigation where a user can interact with a thin scroll bar to scroll through the child nodes. They explained that their implementation shows a textual hint during this type of scrolling. Chhetri et al. (2015) also implemented gestures to set the current root's parent as the new root node (*Up* command) or to restore the previous root node as the current one (*Back* command). Additionally, they integrated a flick motion gesture for scrolling which gets triggered when a user swiftly shakes their hand while holding the mobile device.

I consider their method to be a valid option to support interaction on mobile devices. Their technique utilises all of the available screen space. Because of their layout the structural information of the data is retained. Still, I argue that the design focus for this method was user interaction, for example, by utilising thresholds. Therefore, it is not well suited to support overview orientated tasks. Also, when the text labels to visualise get too long it may be necessary to use word abbreviations instead. Regardless, because the method offers a consistent layout and utilises thresholds to ensure node labels remain readable it may be a good choice for object selection.

### 2.5.2 RECTANGULAR VIEW

Karstens et al. (2003) developed the *RECTANGULAR VIEW* to visualise and interact with large network datasets on mobile devices. According to them they utilised a hierarchical layer-based algorithm to display their graph. They stated that their implementation is based on the original method from Sugijama, Tawa and Toda 1981. They also considered improvements to the algorithm which were introduced by Battista et al. (1999, as cited in Karstens et al., 2003).

According to Karstens et al. (2003) their technique utilises a radial layout to extend the number of displayable nodes while levels are visualised as squares. They explained that they chose this square arrangement to utilise the available rectangular screen space more efficiently.

Karstens et al. (2003) limited the nodes to be displayed at once to 400.

According to Karstens et al. (2003) Figure 2.18 shows their 2D RECTANGULAR VIEW. Their solution utilises colour coding for nodes. Green is used for leaves, red for nodes with children and brown for folded nodes.

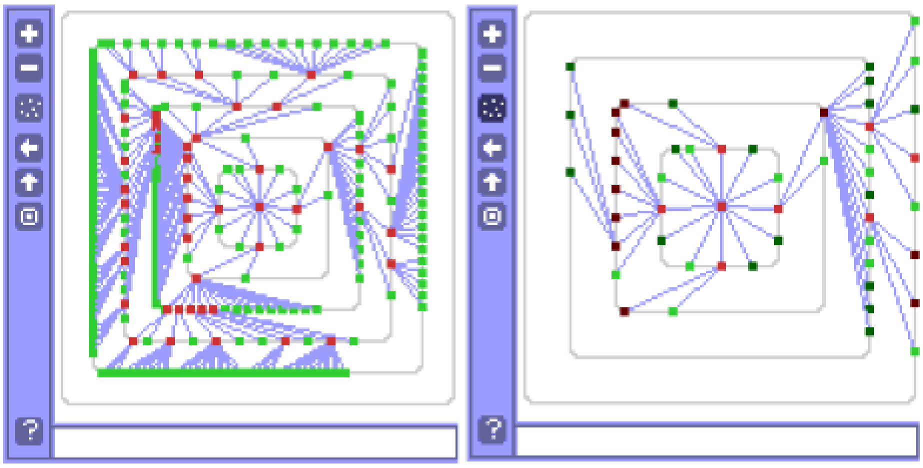


Figure 2.18: Visualization of a Graph With the RECTANGULAR VIEW Showing a Full Hierarchy With 332 Nodes on the Left and a Zoomed Version on the Right

*Note*. From "Visualization of complex structures on mobile handhelds" by B. Karstens, M. Kreuseler and H. Schumann, 2003, *Proceedings of International Workshop on Mobile Computing*, 3.3. The RECTANGLE VIEW for mobile handhelds, Reducing the complexity of graphical output section. Copyright 2003 by B. Karstens, M. Kreuseler and H. Schumann. Reprinted with permission.

For interaction Karstens et al. (2003) implemented zoom by allowing the user to adjust the number of visualised layers. Also, they featured a folding mechanism for subgraphs (collapse/expand). According to Karstens et al. (2003) individual nodes can be selected to change the visualisation. Also, they described their implementations of various toolbar buttons to change the view, for example, zoom, show a history. Moreover, they enabled a status bar for textual information at the bottom.

I consider this technique to be a well-suited option to support overview orientated tasks on mobile devices because its layout may be viewed as a form of map to support a user's orientation. Also, their layer-based node-link layout conveys structural information and also utilises the available screen space quite well. It also demonstrates the importance of considering a devices aspect ratio (for example, 1:1, 16:9) when designing such visualisation techniques. However, for object selection especially with text-based datasets it is not a good option. Hence, even on a larger screen this layout may be subject to visual clutter when adding node labels. Even for non-text-based datasets it may be challenging to select an object of interest because nodes may be too close to each other for precise selection.

### 2.5.3 MAGIC EYE VIEW for Mobile Handhelds (Context+Focus)

Karstens et al. (2003) developed a version of the *MAGIC EYE VIEW for mobile handhelds*. According to Karstens et al. (2003) the MAGIC EYE VIEW was initially designed for desktop (PC) use by Kreuseler et al. (2000). They explained that this technique utilises a 2D hierarchical layout mapped onto a hemisphere's surface. Also, they highlighted that the technique uses a customised projection technique to achieve a context+focus type of view.

Karstens et al. (2003) utilised the *event horizon user interface model* introduced by Taivalsaari (1999) to reduce the number of visualised elements. They explained that this concept utilises compression and expansion for objects which are radially moved. They described that with this concept an event horizon (sink) is displayed at the centre of the screen where objects disappear in when the visualisation is compressed. Moreover, they argued that this concept is also applicable to hierarchies where at the start the event horizon contains only root.

Karstens et al. (2003) also reduced the complexity of the visualisation by projecting the hierarchy onto a 2D region instead of utilising 3D. However, they noted that this design decision may lead to overlap challenges when using straight lines instead of curves for edges (visual clutter). Thus, they implemented curve drawing as an alternative.

To show the current elements in the sink (event horizon) Karstens et al. (2003) designed a circle in three different variants. They explained that these three variants differ in the level of detail they show, for instance, to minimise space requirements. Also, they implemented a folding mechanism to reduce the complexity, for example, nodes can be selected and their overall subtrees will be hidden (fold operation) or shown (unfold operation).

According to Karstens et al. (2003) Figure 2.19 shows their MAGIC EYE VIEW. They explained that nodes can be moved in 2D towards (focus) or away (context) from the centre.

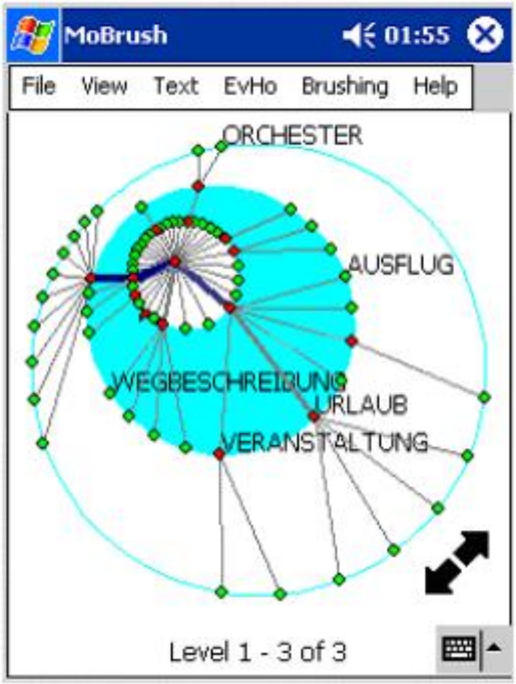



Figure 2.19: Visualization of Hierarchical Structures the MAGIC EYE VIEW on a Mobile Handheld

*Note*. From "Visualization of complex structures on mobile handhelds" by B. Karstens, M. Kreuseler and H. Schumann, 2003, *Proceedings of International Workshop on Mobile Computing*, 3.2. The MAGIC EYE VIEW for mobile handhelds, Integrating mechanisms to address the small display area section. Copyright 2003 by B. Karstens, M. Kreuseler and H. Schumann. Reprinted with permission.

Regarding node limits for usability, Karstens et al. (2003) stated that their (mobile) MAGIC EYE VIEW can visualise around 1,000 nodes in *interactive time*.

For interaction Karstens et al. (2003) implemented annotation, folding and focus features using a pen as the input device. Also, according to them their solution allows the user to select nodes (click) and to change parameter values. They stated that almost all interactions are carried out using an input pen device.

I consider this technique to be a good option to support overview orientated tasks on mobile devices. Their interpretation of the event horizon user interface model shows more information while still providing an overview of the visualised dataset. I consider their technique to be quite similar in its approach to other techniques utilising distortion, for example, the Hyperbolic browser (see Section 2.3.10). Also, the node-link layout conveys structural information. However, I argue that areas of the available screen space are not utilised because of their circular display region. Also, the addition of node text labels may introduce visual clutter due to objects overlapping. Still, for selecting objects in non-text-based datasets like an image browser it may be a suitable option because the sink may offer enough space to precisely select individual objects without them overlapping.

### 2.5.4 Space Manager

Hakala et al. (2005) introduced an interactive document manager for personal digital devices called *Space Manager*. According to Hakala et al. (2005) their method utilises a 2D tree view enhanced with a depth dimension. They described that their aim was to present the users file (folder) hierarchy in 3D to provide more structural information (overview) in comparison to a plain list view. Consequently, they had to set a spatial location for every folder in the hierarchy defining their position on the display. They explained that their target platform was the Nokia 7650 phone with a display resolution of 176x208 pixel. Also, they highlighted that their Space Manager design can be split into two basic levels. For one, the *navigation level* showing only the folder structure. Second, the *folder level* to view folder contents and to use respective (folder relevant) tools. They explained that when a folder is opened the view zooms in and the folder unfolds like a box using animation. Consequently, the user can browse the folders content, issue a return command, browse the folders content using an alternative 2D file list or activate folder specific tools.

Hakala et al. (2005) highlighted their utilisation of a flashlight metaphor for navigation. They explained this metaphor by describing a user holding a flashlight which they can point on any neighbouring folder on the ground.

According to Hakala et al. (2005) Figure 2.20 shows their Space Manager, a spatial file manager application. They stated that the viewpoint is fixed during navigation it only changes for defined actions, for example, when opening a folder.

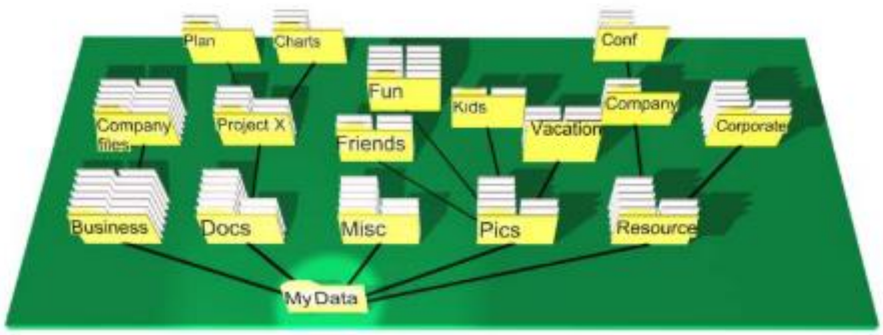



Figure 2.20: The Space Manager Concept

*Note*. From "Spatial interactive visualization on small screen" by T. Hakala, J. Lehikoinen and A. Aaltonen, 2005, *Proceedings of the 7th international conference on Human computer interaction with mobile devices & services (MobileHCI '05)*, p. 139 (https://doi.org/10.1145/1085777.1085800). Copyright 2005 by the Association for Computing Machinery. Reprinted with permission.

Hakala et al. (2005) carried out a user study with 10 participants from various backgrounds. In detail, they carried out their evaluation on a Windows workstation with a 7650 simulator. They asked the participants to carry out 10 test tasks, for example, to move to a certain folder and open a file. After the test tasks they closed the simulator and asked the participants to draw the file structure as they remembered it on a sheet of paper. They carried out semi-formal interview sessions after the drawing part. They gathered their results by observation (test tasks), interview recordings and analysing the drawings.

According to Hakala et al. (2005) all tests showed promising results. They highlighted the rapid adoption rate for their methods navigation mechanisms. However, the participants asked for a home button (return to root) to be implemented in future releases. Also, they identified challenges during navigation, for example, what folder receives the focus next when there are multiple folders above the current one. Some experienced users asked for a feature to disable animation for faster navigation. Additionally, users questioned the file browsing mechanism in regards to its applicability to other file types besides images. Moreover, the participants were able to remember the folder locations quite easily. Also, the users commented that the different levels of the hierarchy should be visually more distinguishable. Additionally, they stated that some participants commented that the general use of 3D supports navigation while others thought it is irrelevant or even visually distracting (like it looks strange).

Hakala et al. (2005) implemented a navigation mechanism utilising a phone’s four-way joystick. They enabled a user to navigate the folder structure, open folders, browse files and open files. Moreover, their design uses a zooming approach when files are opened. They explained that the file is opened in place (zoom in) instead of opening it in another application. Also, the left softkey button of the phone can be used to issue a return (back) command when files are browsed using the 2D file list view.

I do not consider their technique to be a valid option to support overview orientated tasks on mobile devices. The general design approach is similar to the Information Landscape (see Section 2.4.10). The camera is positioned at a higher position attempting to show the complete structure. However, when the tree gets too deep visualisation of the whole tree may no longer be possible. Still, the method utilises the available screen space quite well because the tree grows in width and height to fill most of the screen. Also, as a node-link type tree it conveys structural information very well as links are clearly shown. However, for datasets with a lot of children nodes some links may get occluded by other objects. Also, the visualisation of long node text labels may not be possible because this may cause objects to overlap or result in an unbalanced view with varying spacing between objects. Still, for navigation the method can be a valid option because of its easy-to-understand flashlight metaphor and the tools they developed to support interaction, for example, to browse folder contents. Especially their approach to combine file browsing and viewing in one application utilising zoom is quite interesting. This may have the potential to reduce a user's cognitive load.

### 2.5.5 PaisleyTree (Context+Focus)

Etemad et al. (2014) introduced a size invariant tree visualisation technique called *PaisleyTrees*. According to Etemad et al. (2014) they combined node-of-interest focus with tree-cut visualisations to enable navigation without using zoom or pan techniques. They explained that their layout is a combination of nested, node-link and adjacency-based layouts.

According to Etemad et al. (2014) one of their design goals was to create an aesthetically pleasing solution inspired by Paisley patterns. Also, their solution had to support hierarchies of arbitrary breadth and depth while the visualisation of the tree had to be size-invariant and scalable for different screen sizes (desktop and mobile compatible). Moreover, their solution had to support fast navigation.

Etemad et al. (2014) used a specific tree-cut for their method. They explained that this cut visualises the node of interest including its ancestor and descendant nodes. Their technique draws each node as a differently sized circle and all those circles (nodes) are arranged from top to bottom in the visualisation. They described that the current node-of-interest is always placed at the bottom of the view.

According to Etemad et al. (2014) each node circle visualises two levels of information for its descendants. They described that all of the direct children of a node are placed on a spiral as individual dots. The nodes with more children (non-leaf nodes) are put on the outer ring of the spiral while leaf nodes are placed on the inner ring. Child nodes of non-leaf nodes (grandchildren) are drawn on the perimeter of the circle.

Etemad et al. (2014) identified two scalability challenges for their design. They described that there may be too many hierarchy levels in the dataset compared to the available space on the Paisley curve. Also, there might be cases where a node has too many children or grandchildren to be visualised without introducing visual clutter or interaction problems. To address those challenges, they introduced ellipses to act as placeholders.

According to Etemad et al. (2014) Figure 2.21 shows their PaisleyTree (left) used for file browsing on a mobile device. They described that for comparison a popular file browser application is shown on the right. Also, they highlighted that the PaisleyTree is able to visualise multiple levels of the data structure.

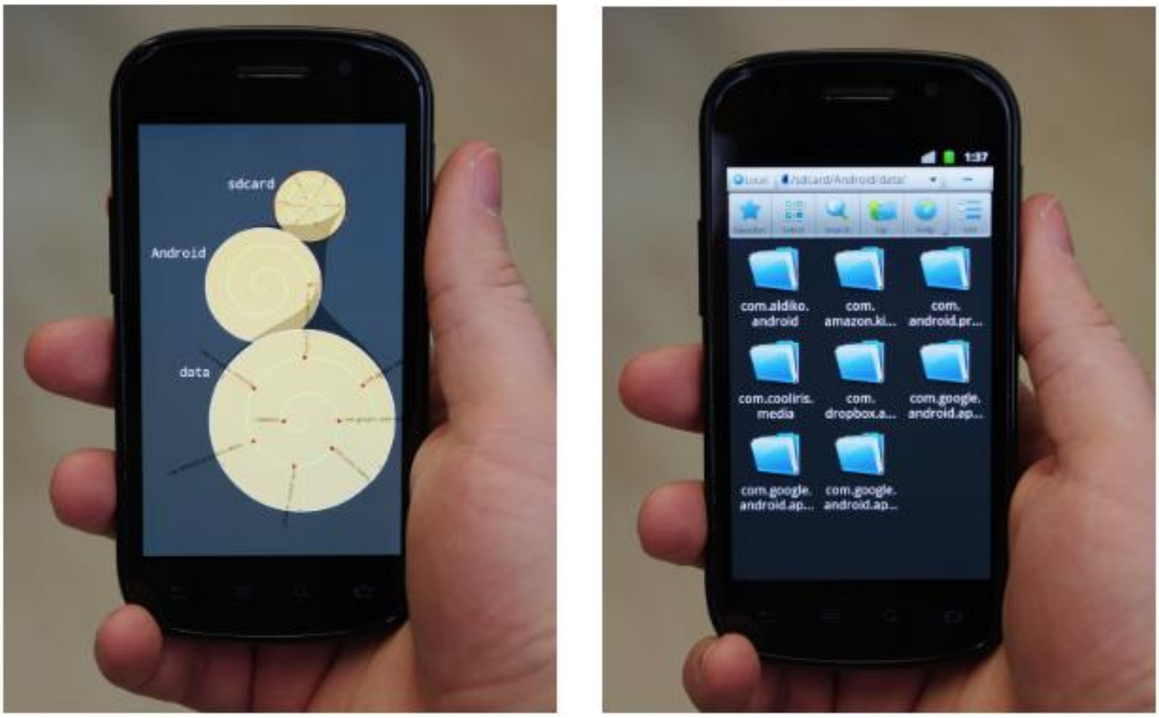



Figure 2.21: Comparison Between a File System as PaisleyTree (Left) and a Popular Mobile File Tree Browser (Right)

*Note*. From "PaisleyTrees: A Size-Invariant Tree Visualization" by K. Etemad, D. Baur, J. Brosz, S. Carpendale and F. F. Samavati, 2014, *EAI Endorsed Transactions on Creative Technologies*, *1*(1), p. 9 (https://doi.org/10.4108/ct.1.1.e2). Copyright 2014 by K. Etemad, D. Baur, J. Brosz, S. Carpendale and F. F. Samavati. CC BY 3.0.

Regarding node limits for visual clutter, I argue that there is no such limit when using this layout technique, for example, ellipses are utilised as placeholders.

Concerning potential node limits for usability, Etemad et al. (2014) used their method to visualise 334,681 city nodes.

For desktop PCs Etemad et al. (2014) explained that a user can click on a non-leaf node to expand it (node-of-interest). Also, they utilised animation for this transition. Additionally, they described that their mobile version can be operated using only a thumb (touch-based), for example, to select nodes.

When focusing on mobile devices I consider this technique to be a well-suited option to support overview orientated tasks and to some extent object selection. Their usage of the Paisley pattern and placeholders retains the structural information of the data and may even show the whole structure at once. However, I argue that the usage of the Paisley pattern results in areas of the available screen space not being utilised. Also, given the methods radial layout there may be not enough screen space available to visualise long text node labels. In general, some labels may be hard to read because of their orientation. Still, their node-of-interest concept supports object selection to a certain extent. However, when there are too many nodes in the node-of-interest selection may become challenging even with the use of placeholders. At a certain point it may no longer be possible to precisely select individual objects because they are placed too close together.

### 2.5.6 Mobile Tree Browser (Context+Focus)

Craig and Huang (2015) described a technique to browse hierarchies with labels on mobile devices. They called their technique the *mobile tree browser*. They explained that their design is based on the space filing layout introduced by Craig and Kennedy (2008). According to Craig and Huang (2015) this layout stacks adjacent nodes for improved readability. They described that a user can click on a node to navigate. Consequently, the layout changes so the (selected) nodes children receive priority spacing horizontally and vertically. Then, the spacing of potential siblings takes priority. Finally, the siblings of ancestors receive priority. Also, animation is applied between views when the user navigates the hierarchy.

Craig and Huang (2015) adjusted the layout to comply to five guidelines for mobile information visualisation design which were defined by Craig (2015).

According to Craig and Huang (2015) Figure 2.22 shows their mobile layout when a fisheye distortion gets applied for cases when there is not enough space for reading or selecting nodes.

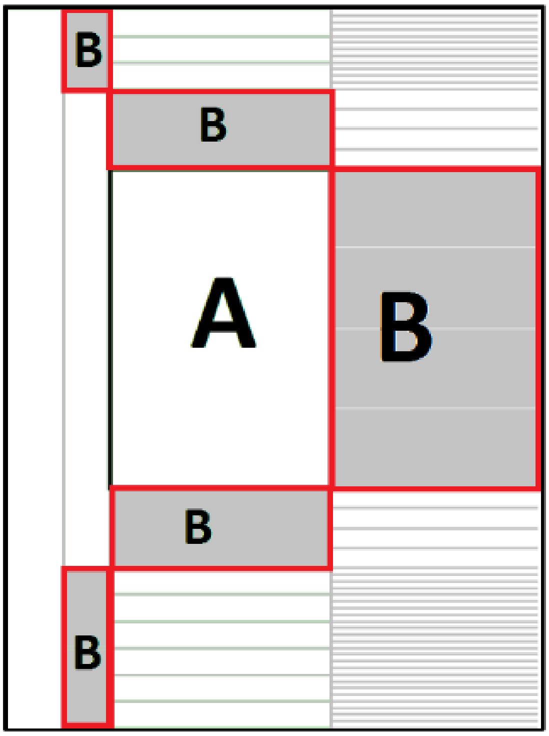



Figure 2.22: Layout With A) the Selected Focus Node and B) Nodes With Distortion Explicitly Applied

*Note*. From "The Mobile Tree Browser: A Space Filling Information Visualization for Browsing Labelled Hierarchies on Mobile Devices" by P. Craig and X. Huang, 2015, *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*, p. 2243 (https://doi.org/10.1109/cit/iucc/dasc/picom.2015.331). © 2015 IEEE. Reprinted with permission.

Regarding node limits for visual clutter, I argue that there is no such limit due to their stacked approach and appliance of a fisheye lens.

Craig and Huang (2015) carried out a user study involving 12 users rating three different interfaces. For one a traditional table view, a standard tree view (see Section 2.3.1) and their mobile tree browser. Based on their results they described that their technique shielded a relatively low score regarding learnability. Contrarily, for usability and legibility their method achieved considerably better scores than the other two techniques.

Craig and Huang (2015) described that a user can tap using their thumb. Also, they included a plus button to allow multi select for folders. Moreover, the user can control the fisheye lens effect by clicking and sliding a lens icon at the side.

I consider their method to be a solid option to support overview orientated tasks and object selection on mobile devices. The structural information of the data is maintained. However, I argue that when selecting nodes, the constant resizing of the nodes may become distracting to users. Also, given their priority spacing approach nodes far away from the focus node are visualised rather small. In general, for a large tree it is questionable if the structure can still be visualised, for example, what happens if a child node has 500 leaves? Similarly, I doubt that one can still navigate efficiently to the end of a level in the hierarchy when there are hundreds of nodes. Also, because of their scaling approach it may be challenging to visualise long text node labels.

## 2.6 Critical Discussion and Future Developments

When comparing interface schemes to move between focused and contextual views in general (cue, zoom, context+focus, overview+detail) Cockburn et al. (2009) concluded that none of the approaches is ideal. They explained that, for instance, the overview+detail approach requires the user to assimilate the link between different views of context (overview) and focus. For example, according to Hornbæk et al. (2002) the switch between an overview and detail view requires mental effort and time from the user. Similarly, Cockburn et al. (2009) claimed that when zooming the temporal separation (pre- vs. post zoom) also demands assimilation. The consequential need to reorientate oneself after each zoom action was observed by Cockburn and Savage (2004) in their evaluations. Cockburn et al. (2009) argued that because most of the context+focus methods are based on distortion mechanisms (for example, fisheye) they are likely to hinder the user’s ability to correctly assess spatial properties like, direction, distance, scale. However, according to them providing contextual and focused views can improve interaction compared to single-view applications. They explained that the efficiency of these schemes depends on various variables in particular the nature of the user's task, for example, overview.

I agree with Cockburn et al. (2009) assessments. However, I argue that a context+focus technique without distortion mechanisms can improve the user's assessment of spatial properties, see for example, the Cone Tree.

Tree data structures and representations are essential in various research areas. For instance, Pavlopoulos et al. (2010) highlighted their importance to visualise hierarchical organisations of biological data and concepts. Pavlopoulos et al. (2010) reviewed recent phylogenetic tree visualisation tools in their study. They concluded that:

- A critical objective and challenge for bioinformatics is the dynamic and efficient visualisation of data on the fly given the amount of data and their heterogeneity.
- Presented characteristics of the tools will change due to future software and hardware development.
- The tools are limited regarding the manipulation of tree data, user friendliness and interaction. Also, they are limited in regards to usability when thousands of nodes have to be processed and rendered. Therefore, they argued that more efficient visualisation tools for large datasets need to be developed.
- 3D space could be utilised for new layouts to allow a clearer structure and less visual clutter for easier navigation within the tree.
- Future tools should only load the necessary parts of the graph into memory and allow for the utilisation of multiple CPU cores or GPUs to improve overall performance.
- Tree viewers of the next generation should allow for data analysis and visualisation, for example, similar to Matlab.

- New tools should support mature import features, for instance, to import and add existing data to a tree visualisation.

When viewed in a more general context I agree with their conclusion, especially, that new visualisations shall be scalable to only load relevant parts of the graph and the use of 3D space to reduce visual clutter.

Wiss et al. (1998) compared three 3D visualisations to give guidance on when to use each design for a specific usage situation. They implemented a Cam Tree, Information Landscape and Information Cube to visualise hierarchical data. According to Wiss et al. (1998) they visualised an electronic newspaper table-of-contents and part of a file system for comparison. As a result of their study they evaluated the visualisations with regard to their suitability for these two different data sets and their support for user tasks. Wiss et al. (1998) explained that the baseline for their user task analysis were the tasks proposed by Shneiderman (1996). Based on their findings they concluded that:

- The compared visualisations behave differently when visualising various data sets. Different properties in data sets can cause serious problems when viewing the visualisation, for example, visual clutter.
- Information visualisation designs aim to support a specific task to solve a specific problem. Therefore, they proposed to integrate various designs into one information visualisation software. This includes the exploration of architectures to support such a design.
- For layout algorithms in 3D, one should be aware of the relevant criteria for the visualisation. They explained that an optimal layout is often impossible therefore one needs to know the advantages and disadvantages of the different criteria, for example, minimise unused 3D space vs. minimise overlap.

Hence, in summary it indicates that the best visualisation technique depends on the task and dataset. Their conclusions also indicate that the reviewed techniques were not designed for generic datasets, for example, when node text labels are very long.

Marriott et al. (2018) explored whether immersive analytic applications should reconsider using 3D information visualisations or not. As part of their work, they have reviewed relevant empirical studies comparing 2D with 3D visualisations. In regard to Cone Trees they concluded that these offer a poor view for hierarchical data because of occlusion and the inherent low speed of tree rotation. In summary, Marriott et al. (2018) stated that in general 3D visualisations are not better than 2D techniques and vice versa. They explained that the superior solution depends on the kind of task to support. According to them user studies suggested that 3D visualisations may be the preferable choice to show the overall structure of data and to provide orientation. In contrast 2D techniques might be advantageous for the precise manipulation, measurement or comparison of data. Therefore, they argued that it might be beneficial to support linked 2D and 3D representations in future immersive analytics software. Also, in the context of node-link diagrams they highlighted that there is evidence that diagrams laid out in 3D can benefit from improved path following (Ware & Mitchell, 2005).

I agree with all of their arguments. Particularly, I find their proposal for linked 2D and 3D representations interesting. Hence, for visualising and selecting objects in data hierarchies I argue that a combination of 2D and 3D techniques may be able to improve a user's task performance and overall experience. Such a layout may be able to show the overall data structure (3D) and still allow for precise navigation or manipulation of objects (2D).

Shen et al. (2019) reviewed state-of-the-art information visualisation techniques, discussed key visualisation design challenges and outlined a conceptual framework for future research directions in the area. They identified various challenges in visualisation design:

- New visualisations should be designed for user familiarity, for instance, to reduce time and effort for the user.
- These new systems should convert abstract information into visually meaningful and insightful representations.
- Solutions are required for the ever-increasing data to visualise including the time dimension.
- The top 10 unsolved visualisation problems described by Chen (2005), for example, usability, education and training are still challenging today.
- There is a lack of systems which enable the evaluation of developed visualisation techniques.
- Usability studies need to be defined for evaluation.

I argue that their proposal to design future solutions for user familiarity is especially noteworthy. Related, Shen et al. (2019) also highlighted the need to consider a user-centred design approach when designing new visualisations so they are perceived as meaningful and effective by the user. Hence, I argue that none of the techniques I reviewed utilised some form of a user-centred design approach. This may also be one of the reasons why none of these methods ever replaced or even challenged the tree view (see Section 2.3.1).

## 2.7 Chapter Summary

To support my conclusions I first mapped all of the described techniques to the defined tasks by Shneiderman (1996) in Table 2.2. The approach of my task analysis is similar to the analysis carried out by Wiss et al. (1998). However, for my analysis I did not define subtasks. In detail, I used the following tasks as described by Shneiderman (1996):

- Overview: Overview for the entire dataset.
- Zoom: Zoom in towards an item.
- Filter: Filter out items.

- Details-on-demand: Select item(s) and show details on demand.
- Relate: View relationships between items.
- History: Keep a history of actions to carry out, for example, replay or undo operations.
- Extract: Allow extraction of a subset of the data and query parameters.

Worth noting that, I carried out this assessment based on the available references and my personal experience. Zoom also includes techniques like fisheye views. Details-on-demand capable techniques have to provide at least one spatially separated section with additional information, for instance, to show more metadata for a file.

The method Hierarchical Edge Bundles (see Section 2.3.4) was excluded from my assessment. It aims to reduce visual clutter when displaying a large number of adjacency edges. It is not a standalone technique to visualise hierarchical data.

**Table 2.2**

*Visualisation Techniques for Hierarchical Data - Task Analysis*

| | Overview | Zoom | Filter | Details-on-demand | Relate | History | Extract |
|---|---|---|---|---|---|---|---|
| 2D - Tree view | N | N | Y | Y | Y | N | N |
| 2D - Treemap | Y | N | Y | N | N | N | N |
| 2D - Circle Packing | Y | Y | Y | N | Y | N | N |
| 2D - Hierarchical Edge Bundles | * | * | * | * | * | * | * |
| 2D - InfoSky Visual explorer | Y | Y | Y | N | Y | Y | N |
| 2D - Hierarchical Point Placement | Y | N | Y | N | N | N | N |

| | Overview | Zoom | Filter | Details-on-demand | Relate | History | Extract |
|---|---|---|---|---|---|---|---|
| 2D - STREAMIT | Y | N | Y | Y | Y | Y | N |
| 2D - Ordered and Quantum Treemaps | Y | Y | N | N | Y | N | N |
| 2D - Textual Fisheye Tree Views | Y | Y | N | N | Y | N | N |
| 2D - Hyperbolic browser | Y | Y | N | N | Y | N | N |
| 3D - Degree-of-Interest Trees | Y | Y | Y | Y | Y | N | N |
| 3D - Document Lens | Y | Y | Y | N | Y | N | N |
| 3D - Perspective Wall | Y | Y | N | N | Y | N | N |
| 3D - Cone Tree | Y | Y | Y | Y | Y | Y | Y |
| 3D - H3 Layout | Y | Y | N | N | Y | N | N |
| 3D - HotSauce | N | Y | N | Y | Y | N | N |
| 3D - Information Cube | Y | Y | N | N | Y | N | N |
| 3D - SemNet | Y | Y | Y | N | Y | Y | N |
| 3D - Cam Tree (derived from Cone Tree) | Y | Y | Y | Y | Y | Y | Y |
| 3D - Information Landscape | N | Y | N | N | Y | N | N |
| 3D - Beamtree | Y | N | N | N | Y | N | N |
| 2D - Enhanced Radial Edgeless Tree | N | N | Y | N | Y | Y | N |
| 2D - RECTANGULAR VIEW | Y | Y | N | Y | Y | Y | N |
| 2D - MAGIC EYE VIEW for mobile handhelds | Y | Y | N | N | Y | N | N |
| 3D - Space Manager | N | Y | N | Y | Y | Y | N |
| 2D - PaisleyTree | Y | Y | N | N | Y | N | N |

| | Overview | Zoom | Filter | Details-on-demand | Relate | History | Extract |
|---|---|---|---|---|---|---|---|
| 2D – Mobile Tree Browser | Y | Y | N | N | Y | N | N |

*Note*. List of the described visualisation techniques mapped to the tasks outlined by Shneiderman (1996). “Y” (yes) indicates that the method supports this task while “N” (no) indicates that this task is not supported. “*” indicates that this method was out scoped.

Based on my task analysis from Table 2.2 and my literature review (see Section 2.3, 2.4, 2.5 and 2.6) I can draw the following conclusions.

I conclude that methods which do not support tasks for overview were mainly designed to support object selection or navigation in general. Albeit the Cone Tree (Cam Tree) (see Section 2.4.4, 2.4.9) supports every task it is prone to visual clutter (occlusion). Most context+focus techniques lose screen space to show context, for example, the Document Lens (see Section 2.4.2).

Methods like InfoSky (see Section 2.3.5) also utilise separate display sections (overview+detail) to support orientation (map for overview) and precise navigation (tree view). However, I argue that this kind of spatial separation might be considered as lost screen space depending on the task.

In general, all of the described techniques have their individual strengths and weaknesses for different use cases and datasets. Again, the best one depends on the particular task and dataset. However, it is not always clear which technique is the most useful for a given scenario (Macquisten et al., 2020). Hence, there are even tools available which allow the visualisation of one hierarchy with multiple techniques simultaneously like for example *VizWick* (see Burch et al., 2020).

Also, all of the reviewed techniques followed a form of top-down design approach where the researchers derived their design based on previous research results or individual inspiration. However, they did not consider actual user needs, for example, by carrying out in depth interviews or focus groups, before starting their design or development process.

The PaisleyTree (see Section 2.5.5) is the only technique to visualise hierarchical data which was designed for mobile and desktop devices (cross platform).

Apart from the tree view only the MAGIC EYE VIEW (see Section 2.5.3) and mobile tree browser (see Section 2.5.6) were ported to mobile handhelds.

As new techniques shall be designed utilising a user-centred approach, I argue that new techniques have to be applicable to mobile devices like phones. Hence, at least one should evaluate a potential user need for such a requirement, for example, by identifying the devices users actually use to navigate their data.

I agree with Marriott et al. (2018) that 3D does not provide benefits over 2D for visualising data just because it may feel more natural.

However, for visualising and selecting objects in data hierarchies I claim a combination of 2D and 3D techniques may be a valuable approach. Hence, it may even be possible to design a new technique combining 2D and 3D in one view to offer precise, familiar navigation and overview.

Such a new technique should also be able to load data dynamically to provide real time performance for arbitrarily sized hierarchies. Ideally, the design should already be familiar to users (for example, by utilising common metaphors).

To summarise, a new technique should:

- Support overview and navigation tasks utilising the available screen space. This may be realised utilising novel approaches combining both 2D and 3D techniques.

- Be able to scale arbitrarily from a design perspective to prevent visual clutter and a technical perspective to load only the relevant parts of the tree
- Be designed utilising a user-centred approach which does include an attempt to understand the needs of potential users

# Chapter 3 Research Methodology

## 3.1 Introduction

In this chapter I introduce my research methodology detailing various methods and tools in the context to my research. Based on my research objectives and available resources I applied only methods described in the following sections. To quote Hartson and Pyla (2019) “In any design situation, you need to make choices. Even within an agile UX process, no one UX lifecycle or method is one size fits all.” (p. 49).

Section 3.2 sets the context for my methodology. Next, in Section 3.3 I describe the scope and target user group for my project. Section 3.4 outlines the overall process used for this research. Section 3.5 reflects on the ethical aspects of my research including compliance. Sections 3.6, 3.7 and 3.8 discuss the methods used for the respective chapters (Chapter 4, Chapter 5 and Chapter 6) in my thesis. Section 3.9 presents a critical discussion of the methodology and describes the challenges faced during this research project.

All software and licenses used to carry out this research project can be found in Appendix A.

## 3.2 Contextualisation

As the result of my design was not clearly defined, I first had to evaluate which methods I could utilise to identify requirements and to support the design and evaluation process. To be precise, my design is informed by user needs, which I aimed to identify. Therefore, I could not know if the result would be a 2D or a 3D design.

After reviewing literature focusing on 3D user interface design, I came to the conclusion that for my research project it is more reasonable to utilise methods from 2D interface design. The practice of 3D user interface design mainly focuses on virtual or augmented reality artifacts. For example, LaViola Jr. et al. (2017) explained that a virtual reality program can allow the user to place a virtual object at any place using any orientation. They stated that for this task a 2D mouse would be inadequate and therefore new devices, techniques and metaphors are required for such systems. Consequently, they argued that because many of these systems operate in real and/or virtual 3D space they term such interfaces 3D user interfaces.

Moreover, the heuristics I reviewed for evaluating 3D user interfaces support this impression. For instance Sutcliffe and Gault (2004) called one of their heuristics for virtual environments *Natural engagement*. They stated that the user should not be able to detect the difference between the virtual environment and reality.

By contrast my method may also be a 3D visualisation but then utilises only 2D output and input devices, for example, a computer mouse, touch input and 2D screens. Hence, I agree with Marriott et al. (2018) that a visualisation technique is 3D when it maps information to three different spatial dimensions (see Section 2.2). Consequently, I aligned my methodology to 2D interface design. Hence, I mainly selected my methods from the area of *user experience (UX)* design. User experience is defined as “the phenomenon felt by users before, during, and after usage- usually a combination of usability, usefulness, emotional impact, and meaningfulness.” (Hartson & Pyla, 2019, p. xix).

Overall, I applied a mixture of qualitative and quantitative methods investigating different data sources. In part, I compared results utilising triangulation for conversion or diversion. I applied a form of *multiple triangulation* combining different data sources and methodologies in one investigation (Denzin, 1970/2017).

All the interviews I carried out were *semi-structured*. According to Grix (2010) semi-structured interviews allow for a degree of flexibility while the results can still be compared. I chose this type of interview for its flexibility precisely to be able to follow up on unexpected events or thoughts during the interview.

In comparison a *structured* interview does not allow for this kind of flexibility (Grix, 2010). The results of *unstructured* interviews cannot be compared to each other (Grix, 2010).

According to Lazar et al. (2010) focus groups are merely interviews carried out with a group of individuals. They explained that focus groups can be disadvantageous when discussing controversial topics or when individuals monopolise the conversation. However, I also planned to carry out focus groups for my research to support a critical discussion. Unfortunately, I was not able to recruit enough participants to join a focus group.

### 3.3 Scope and Target User Group

To support a user-centred design I focused on understanding, and potentially improving, file and folder navigation across devices (mobile and PC desktop) as a practical example for visualising and interacting with hierarchical data. I argue that this topic is very familiar to a lot of users. Therefore, it is a suitable choice to receive a wide range of quality user feedback, for example in interviews, while not distracting the research participants with rather abstract terms like hierarchical data.

At the time of writing, most software applications still use list-based (tree view) visualisations for hierarchical data, such as file browsers and code editors. To challenge this predominance in general, I tried to recruit participants who are at least 18 years old. I especially focused my recruiting efforts towards university students.

University students were a seemingly good choice because they represent a broad demographic. Also, I assumed that they are not influenced by a specific work practice. For example, they did not already work with a specific accounting system for 10 years, therefore, they are less subject to bias.

Also, I was able to recruit working professionals from various industries. I assumed that their biased feedback as a result of their work practice might provide different insights regarding missing features.

Overall, I worked with a mixture of university students and working professionals. These were classified under one *work role* I called “*System User*”. The terms ”*Student*” and ”*Professional*” are *subroles*, which I identified for this work role.

According to Hartson and Pyla (2019) a work role is defined by work responsibilities of an individual not by a person. Worth noting that, because I aimed for my method to be usable by a wide range of individuals, I decided against the usage of *user class* definitions. According to Hartson and Pyla (2019) a user class for a work role is described by a set of relevant characteristics, for example, skills or demographics. Also, I did not define a *persona* because I only focused the design on one work role, namely the System User. According to Hartson and Pyla (2019) a persona is a hypothetical individual with a story and description to support designers by limiting their design focus. Similarly, I decided against the further exploration of the sub roles Student and Professional for my research. I argue that there are a lot more relevant sub roles which I cannot consider given my limited resources, for example, elderly, children, parents. Therefore, any form of differentiation between the two sub roles, for example based on characteristics, does not lead to a more rigorous result.

I argue that the mixture of university students, working professionals and different methods offer a solid baseline for an in-depth investigation. Moreover, an artificial restriction to certain professions or age groups may not lead to a more rigorous result. Hence, I argue that most individuals who are at least 18 years old and are interacting with information technology devices (mobile or PC desktop) utilise hierarchical datasets on a regular basis, for example, when navigating through their files. Also, viewing and interacting with hierarchical data is not a profession in itself.

Overall, I was able to recruit the following interview participants:

- Student00: University student.
- Professional00: Independent business consultant.
- Professional01: Marketing and finance expert.
- Professional02: SAP consultant.
- Student01: University student.

Worth noting that, I assigned a unique technical identifier to all the interview participants for a consistent, yet anonymous documentation, based on their sub role, for example, Student00. Those identifiers are used throughout my thesis. When an individual participated in multiple interviews the same identifier is used, for example, Professional02 participated in an interview to understand user needs (see Section 4.6) and also in another interview to help evaluate my conceptual design (see Section 6.2).

### 3.4 Overall Procedure of the Research

When viewed as a software development project one can differentiate between different approaches. For one, the *waterfall* (see Bell & Thayer, 1976) model is a software development approach which, according to Hartson and Pyla (2019), is a linear sequence of lifecycle activities carried out in order. I agree with Hartson and Pyla (2019) that this approach is often disadvantageous, mainly due to the lack of real user feedback until the end of the overall process.

Another software development approach is the *agile lifecycle process*. According to Hartson and Pyla (2019) it is an iterative process where all lifecycle activities are carried out for each product feature. They explained, that features to the users are regularly released in chunks.

Because I focused on user interface design and developed the method by myself the choice of a formal development approach was expendable for my project. However, I applied (in part) the iterative *UX design lifecycle* process for my technique. According to Hartson and Pyla (2019) this process represents how the various lifecycle activities are connected sequentially over time. The four fundamental iterative activities in the UX design lifecycle are "Understand Needs (of users). Design Solutions. Prototype Candidates (for promising designs). Evaluate UX." (Hartson & Pyla, 2019, p. 29).

In the context of UX design, I had to consider the following terminology for my project. The term *User work,* or *work,* describes "what needs to be done, or user goals within a given 'problem' domain." (Hartson & Pyla, 2019, p. 119).

The *User work practice* or *work practice* is a term describing how individuals carry out their work, for example, including activities or customs (Hartson & Pyla, 2019).

The *work domain* is "the complete context of the work practice, including the usage context of an associated system or product." (Hartson & Pyla, 2019, p. 119).

For my research I argue that the term work domain does not fit very well as I do not focus my design on a particular industry, for example finance. However, my work context are products for file and folder navigation. Therefore, I argue that my project adheres to a *product perspective* design point of view. According to Hartson and Pyla (2019) a product perspective is a term used to describe design targets for personal objects, for example, devices or software applications. By contrast, the *enterprise system perspective* describes a "A design viewpoint in which the design target is a large organizational information system" (Hartson & Pyla, 2019, p. 119).

To conclude, similar to the example of a mp3 music player described by Hartson and Pyla (2019), my work domain is the work milieu of my product, for example usage in a work and/or personal context.

According to Hartson and Pyla (2019) it is never the goal of a bottom-up design to change work practice at a fundamental level. They explained that, in contrast, a top-down design often leads to results which often radically change the work practice.

Worth noting that, a bottom-up design approach is subject to bias, for instance, a user's preferences are influenced by their current work practice, such as products they currently use (Hartson & Pyla, 2019). Therefore, I argue that one may utilise a mixture of qualitative and quantitative methods to reduce this form of bias when identifying user needs.

Most techniques designed in the past utilised a more top-down design approach (see Chapter 2). I aimed for a user-centred (bottom-up) design, for instance, to support actual user needs.

### 3.5 Compliance and Ethics

I initially planned to also recruit elderly (for instance, from retirement homes) as research participants. Therefore, I submitted the research ethics application form and supporting documents which were reviewed by the University's Research Ethics Committee (University of Gloucestershire). My research proposal was approved by the committee.

However, I was not able to recruit elderly users to participate in my research. Therefore, after consulting with my first supervisor I decided to solely focus on university students and individuals who are 18 or older. The University's Research Ethics Committee was not informed regarding this scope change.

All interviewees had to fill out and sign an interview consent form before the interview started. The form I created was informed by an example consent form from The University of Edinburgh School of Geosciences (2013). I also considered available resources online from the University of Gloucestershire. The final form was reviewed by my first supervisor. The form was handed out to participants in English and German language (translated by me). If the interview was carried out remotely the forms were sent by e-mail as pdf files before the meeting. Again, participants had to fill out, sign and send me back one of the forms before the interview started.

The interviews were held in German language. Before the meeting, relevant supporting documents, for example design documents in English, were translated to German by me.

I created transcripts in German for all the interviews I carried out. I followed the transcription rules as described by Dresing et al. (2015). Worth noting that, I did not utilise the additional rules described by Dresing et al. (2015). To exemplify, I did not indicate a pause's length by the number of dots in parentheses, for instance (..). Sections that I considered to be irrelevant were marked accordingly, for example "*(Out of scope: Begrüßung/Greeting)*". All quotes from the interview transcripts that I utilised in my writing were translated by me.

If requested (checkbox marked at "*Yes*" in the consent form) I sent a copy of the transcript (.rtf) to the interviewee.

To recruit participants I posted flyers online, contacted universities in my area and paid a market research institute to send out my flyer. Interviewees received a paid incentive up to 50€ per session.

With the support from the staff of the University of Gloucestershire I did a questionnaire in 2019 where I tried to learn about the student's software usage behaviour (see Section 4.5). I gave a short (remote) presentation introducing the aim of my research and sharing the password protected survey link on the 10th of December 2019. The communicated close date was the 18th of December 2019. I closed the collector a few days after the 18th of December. I used SurveyMonkey (see SurveyMonkey, n.d.) to carry out the questionnaire. I included a small disclaimer at the top of the questionnaire stating the title, my name, my university e-mail and my role as a PhD student at the University of Gloucestershire. I also briefly described my research aim and target for this survey "*I want to learn about your current software usage when navigating through data structures to identify the needs of potential users and gather requirements for the new method*".

To my knowledge, participants were gifted an Amazon voucher from the University. Therefore, I asked for the participants e-mail addresses in my survey. I forwarded these e-mail addresses to the staff of the University of Gloucestershire who supported my survey.

In 2023 I carried out another questionnaire with support from the staff of the University of Gloucestershire. This questionnaire was carried out to assess my video-based prototype (see Section 6.7). The password protected links to the video and the questionnaire were shared via e-mail with staff and students of the University of Gloucestershire. Again, I used SurveyMonkey to carry out the questionnaire. The collector for the questionnaire was open from the 20th of February 2023 until the 6th of March 2023. The questionnaire included a small disclaimer at the top stating the title, my name, my university e-mail and my role as a PhD student at the University of Gloucestershire. Also, participants had to fill out a short mandatory consent form which was derived from the consent form I used for my interviews (see above).

Again, to my knowledge participants received vouchers from the university for participating in the questionnaire. Therefore, I asked for the participants e-mail addresses in my survey. I forwarded these e-mail addresses to the staff of the University of Gloucestershire who supported my survey.

In general, all results, for example interview transcripts or questionnaire answers, were anonymised. I attempted to exclude not only direct personal information (like the name, addresses) but also related information which might hint to an individual, for example, specific projects they mentioned.

For interview transcripts, I assigned technical identifiers to the interview participants for a consistent, yet anonymous documentation, based on their sub role (see Section 3.3), for example Student00.

Worth noting that, the (sub-)sections for Chapter 4, Chapter 5 and Chapter 6 were laid out focussing on readability and may not imply a strict temporal sequence. To exemplify, in Chapter 4 triangulation was carried out informally for data elicitation. The consequential (final) formal coding, NVivo analysis and modelling was carried out based on the available data, for example, interview transcripts. Also, I evaluated different research methods based on the data available to me. These discarded methods are not mentioned or described in my thesis to maintain readability. All of my writing and argumentation in this thesis is based on scholarly resources, my personal research notes, my artifacts and elicited datasets, for example, interview transcripts, questionnaires.

Worth noting that, according to Zutlevics (2016) offering financial incentives to research participants is still a controversial topic in discussions about ethics in research. They concluded, that based on empirical findings as those from Mellström and Johannesson (2008) some areas which rely on female research participants may even have less success in their recruiting efforts when offering financial incentives. In contrast Abdelazeem et al. (2022) found that based on their meta-analysis even small monetary incentives increase the consent and response rate from individuals to participate in randomized controlled trials. From an ethical point of view McNeill (1997) argued against financial inducement because it may encourage people to expose themselves to injury, for example, when participating in medical research. However, they agreed with Wilkinson and Moore (1997) that paying individuals to participate in research may be justified when there is very little risk for the research participant, for example, to suffer from mental, emotional, physical or social injury.

I argue that, for my research project I cannot identify any major risks the participants may expose themselves to by participating in it, for example, through interviews or questionnaires. Therefore, offering a small paid incentive up to 50€ per session or gift vouchers through the university seems justified to improve the participation rate.

### 3.6 Methods for Understanding User Needs

In general, I aimed to understand how users view and navigate in hierarchical data on different platforms (mobile and PC desktop) to inform my design. I chose the example of file and folder management as the focus topic for data elicitation and analysis. Overall, I applied a two-fold strategy to understand user needs. In the first part (general context) of this strategy I aimed to understand:

- How important are mobile devices in daily routines?

- What other areas besides file and folder management may be relevant for additional case studies, for example, to apply the method for social media?
- How important are products like virtual reality devices in regular use? Does this topic motivate interviewees to speculate about new visualisation techniques relevant to my research?
- What criteria are important to a user when selecting an application? What type of data they use, for example, text documents, pictures? The answers to these questions helped to identify important requirements for my method.
- Are 3D applications being used on a regular basis or not?

The second part of the strategy was to understand user needs in the context of file and folder management. Therefore, I aimed to understand how do participants browse and maintain their filesystem(s) and what are they currently missing.

Overall, my goals were to understand needs including the target users' *work practice ecology* and *information hierarchies and work flows*. For work practice ecology one aims to understand potential connections in a work practice, for example what other systems are used (Hartson & Pyla, 2019). When understanding information hierarchies and work flows one aims to map out the central work flows for the relevant work practice (Hartson & Pyla, 2019).

To exemplify, regarding work practice ecology, I was also interested in the type of data they usually work with. For information hierarchies and work flows I asked questions regarding file navigation scenarios in a private or professional context.

Worth noting that, there is also the goal to understand *market forces and trends*. According to Hartson and Pyla (2019) market forces and trends describe the need of an UX team to understand a client's perspective on their market including trends and thoughts on the competition, for example what do they value or what works better in their system, how do the systems differ. In my research, I primarily focused on the user's current work practice, how they chose systems, what they like or dislike regarding their systems or how their systems may be improved. Hence, I did not aim to design a product for commercial use in a specific market. However, as I learned about user habits, trends were implicitly discussed as well, for example when discussing virtual reality.

For transparency, in the context of user experience design, the term *ecology* is utilised to describe “the entire set of surrounding parts of the world, including networks, other users, devices, and information structures, with which a user, product, or system interacts” (Hartson & Pyla, 2019, p. 3).

### 3.6.1 Modelling and User Stories

To support my objective to design a novel method to visualise and interact with hierarchical data I aimed to deliver the following results to inform my later design:

- Inform a *user work role model* and *flow model*
- Create *user stories* and derive *task sequence models*

According to Hartson and Pyla (2019) a user work role model represents the relevant work roles, sub roles and user classes for a project. They explained that it is necessary to identify the operational work roles as early as possible.

A flow model is a graphical visualisation of the flow of information and artifacts through a system (Hartson & Pyla, 2019). It can highlight potential points of failure in a process early on, for example, by focusing on handover processes between work roles (Hartson & Pyla, 2019).

According to Hartson and Pyla (2019) the flow model and the user work role model are crucial parts in a UX design studio. They explained that the flow model similar to the user work role model should be established as early as possible in a project’s lifecycle.

According to Hartson and Pyla (2019) a user story is a short narrative description for a feature required by a user in a work role including reasoning on why it is required. They explained that it is a form of an agile user experience design requirement. Worth noting that, I have also (partially) acted as my *own domain expert* given the broad scope for my new technique (Hartson & Pyla, 2019). Hence, for user stories I, as an analyst, have to fill in potential gaps (Hartson & Pyla, 2019).

For a transparent documentation of my research project, I chose to add various task sequence models. A task sequence model is “A step-by-step description of how a user might perform a task with a product or system, including task goals, intentions, triggers, and user actions” (Hartson & Pyla, 2019, p. 200).

I decided to deliver those models and user stories to support a transparent documentation and maintain readability, for example, by linking the developed user stories in my design stage (see Chapter 5). For transparency, the modelling was carried out in the analysis stage after data elicitation. Hence, the value proposition to include a user work role model and flow model in the context of my research project is debatable. To exemplify, an initial flow model visualising file and folder navigation is rather trivial given the narrow scope of the scenario. However, I decided to include both of them to visualise my evolving understanding of the subject matter.

In general, I differentiate user needs, for example when describing user stories, according to the following major components of the user experience: *emotional impact*, *usefulness* and *usability*.

According to Hartson and Pyla (2019) usability consists of topics like efficiency or ease of use. They described usefulness as the “Ability to use system or product to accomplish goals of work” (Hartson & Pyla, 2019, p. 9). In contrast, the emotional impact consists of topics like user satisfaction or user feelings (Hartson & Pyla, 2019).

Worth noting that, according to Hartson and Pyla (2019) a fourth component in UX is *meaningfulness*. It describes a personal relationship between a user with a product, for example companionship of a user with their mobile phone (Hartson & Pyla, 2019). I did not consider meaningfulness at any stage of my research because of resource constraints.

### 3.6.2 Interviews and Questionnaire

To construct my deliverables, for example user stories, I elicited the following data in different phases:

- Pilot (Student00): One pilot semi-structured interview in person with a university student in 2018 - around 45 minutes in length. Worth noting that, my life partner was present for this interview for note taking. Her hand written notes were handed over to me after the interview. The interview was initially planned as a focus group but only one person showed up.
- Phase 1 (Professional00, Professional01): Two semi-structured individual interviews in person with adult individuals 2019 - each around 30 minutes in length.
- Phase 2: Online questionnaire with 22 university students 2019 using SurveyMonkey.
- Phase 3 (Professional02, Student01): Two semi-structured interviews carried out remotely (digitally using screen sharing) with adult individuals also introducing a first sketch of the novel method 2021/2022 - each around 30 minutes in length.

Worth noting that, I did not carry out any observation of any kind at this stage. Even though I am aware that it “can help you see work activity with an independent eye.” (Hartson & Pyla, 2019, p. 131). However, the primary focus was file and folder navigation. I am not aware of any professional career which is mainly focused on file and folder management even though the activity is carried out regularly.

The pilot interview was carried out to ascertain that my questions are clear and unambiguous. Based on the pilot I adjusted my guiding questions to be used in phase 1 interviews. I utilised the phase 1 results to design the questionnaire for phase 2 for triangulation. Worth noting that, in an engineering context I created this questionnaire as a rather informal method to collect data. Because of the rather abstract topic, I decided to carry out another phase 3 which consisted of two semi-structured interviews to gather more task scenarios and requirements in the file and folder management context, presenting a first sketch of my novel method (in part) for triangulation. I coded and analysed the results of my interviews utilising the qualitative data analysis software NVivo (see QSR International, n.d.).

Worth noting that, in phase 1 and phase 3 I used a form of *Eclectic Coding* This is a combination of multiple first cycle coding techniques (Saldana, 2021). I chose a combination of *Provisional Coding* and *Descriptive Coding*.

Provisional Coding is based on a start list of codes (Saldana, 2021). This start list can be, for example, anticipated categories generated from a literature review. Provisional coding was deemed an appropriate choice for my research. This is because, for instance, I did indeed identify some categories such as usability, usefulness and emotional impact, before starting coding.

Descriptive Coding aims to summarise a topic with a single word or a short phrase (Saldana, 2021). According to Saldana (2021) it does not reveal much information into an interviewee's mind. However, they argued that it leads to a form of index to the data's content. Therefore, I chose it in order to be able to count code frequencies using NVivo. Consequently, I decided against alternatives like *In Vivo Coding*. According to Saldana (2021) it is a method that is particularly useful when honouring a research participant's voice, for example when working with children or adolescents. Hence, utilising In Vivo Coding would have made it more difficult for me analysing the results, for example when counting occurrences. For transparency, because my focus was on understanding user needs the topics that I identified are already interpretative to some extent, for example when identifying a suitable topic for an expressed user need.

For second cycle coding in phase 1 and phase 3 I used *Pattern Coding*. According to Saldana (2021) Pattern Codes are codes which aim to identify a theme, configuration, or provide reasoning.

I decided against alternatives like *Longitudinal Coding*. According to Saldana (2021) it is a method which utilises matrices to organise collected qualitative data into temporal categories for comparison over time. Hence, I did not focus on potential changes in a participant's life over time, for example if they changed their job.

Worth noting that, coding was carried out based on the German interview transcripts (.rtf). Again, quotes from participants used in my thesis to support my assertions were translated by me. My (interviewer) statements were coded only if I repeated participants' statements for confirmation, for example when the audio recording failed, as a part of a longer discussion or for context. Sections were coded to multiple codes if they addressed multiple topics, for example a section can describe a user task while also including references to relevant data types. Also, sections were coded to multiple codes if the source material was not specific enough, for example when a participant states that they use WhatsApp for messaging I cannot determine if they use this application for work, private messaging or both.

Also, I initially described the target dataset for my method as *related datasets*. After data elicitation to understand user needs, I decided to use the distinct term hierarchical (tree) data to prohibit potential confusion.

**3.7 Methods for the Design of a Novel Method**

The general purpose of design is the users' benefit in the context of their work practice (Hartson & Pyla, 2019). There are three essential categories of human needs that a design must meet:

- *Ecological*: Participate and thrive in the ecology of the work domain;
- *Interaction*: Ability to carry out the required tasks in the ecology of the work domain;
- *Emotional*: Emotional and cultural satisfaction, enriched while using the system (Hartson & Pyla, 2019).

According to Hartson and Pyla (2019) the category emotional also includes the need to form a possible long-term relationship with the system. They call this meaningfulness. I did not consider meaningfulness at any stage of my research because of resource constraints.

Worth noting that, the three categories build on each other in the given order, for example, meeting interaction needs is not possible without first meeting the users' ecological needs (Hartson & Pyla, 2019). Consequently, I consciously designed my method along these three design perspectives: ecological, interaction and emotional.

An *ecological perspective* aims to describe how a system communicates and interacts with other entities, such as external systems (Hartson & Pyla, 2019). The *interaction perspective* reflects how a user actually interacts with a system utilising devices and controls (Hartson & Pyla, 2019). The *emotional perspective* is a design viewpoint focussing on the emotional impact of a design, for instance by designing products which are fun to use (Hartson & Pyla, 2019).

The UX design lifecycle is an iterative process (Hartson & Pyla, 2019). Meaning, that lifecycle activities are not completed after just one iteration. Even though I carried out my research project by myself, I regularly moved from one lifecycle activity to another, for example still identifying new user stories during design.

### 3.7.1 Sketching and Conceptual Designs

As the design of my novel method was core to my research I was designing constantly from the very start. Early designs were mainly *conceptual designs* from the ecological perspective. A conceptual design is a design which aims to transport the designers' mental model to a third party, for example potential users (Hartson & Pyla, 2019).

When I aimed to understand user needs, I created a first sketch of my novel method based on my understanding from phase 1 and phase 2. In this early design phase, I chose to utilise sketching as it can help to create and explore conceptual ideas rather quickly (Greenberg et al., 2012). The sketch was presented to research participants for critiquing to identify additional requirements and to evaluate previous indications, such as evaluating a pattern from the previous phases.

To inform the design process I considered various design patterns, like the ones outlined by Tidwell et al. (2020). However, because my overall goal is to design a new, novel, method to visualise hierarchical data these patterns did not guide my design process. To exemplify, just because there is the pattern of the *Thumbnail Grid* (see Tidwell et al., 2020) to display a list of items I cannot simply copy this pattern and call it my result. Therefore, I focused more on principles than on design patterns in this thesis. However, to reach a certain amount of user acceptance I utilised, in part the, version two of the *Material Design* guidelines from Google (n.d.-k) for selected standard components like menus, fonts, icons and colours. When utilised to optimise the design of websites for small screens the Material Design guidelines can even increase the effectiveness and efficiency for delivering information (Pinandito et al., 2017). Therefore, I argue that these guidelines are a solid foundation for my project, designing across multiple devices.

Because the design of my method was at its core a structural design challenge, for example how to group objects in a meaningful way, the *Gestalt principles* were considered. According to Johnson (2021) these principles still serve as a framework for describing visual perception. I argue that I utilised these principles for a form of *heuristic evaluation* carried out by myself. Heuristic evaluation in UX is a method where an expert compares criteria of a design with a set of heuristics (Hartson & Pyla, 2019).

Also, I created conceptual designs from the ecological, interaction and emotional perspective. As a part of this, I created a *storyboard* focusing on the key (sub) tasks I identified earlier. A storyboard is a “sequence of visual ‘frames’ illustrating the interplay between a user and an envisioned ecology or device” (Hartson & Pyla, 2019, p. 365). I chose to utilise a storyboard because I claim it is a time efficient artifact to communicate one’s conceptual design as a whole.

### 3.7.2 Wireframes

Based on the conceptual design and the Material Design guidelines, I had to address the technical complexity of my node layout. Hence, I had to evaluate if the design could actually work based on the technical capabilities of computing at the time of writing. For my technical prototyping I chose to utilise the Unity engine, version 2021 LTS (see Unity Technologies, n.d.). I chose this engine as it gets used in computer science and engineering. To exemplify, Harshfield et al. (2015) described a design for an algorithm animation framework built for Unity 3D to enable algorithm visualisation and animation. They highlighted that Unity 3D supports over 20 platforms including mobile devices, that it utilises modern hardware and that development is fast because, for instance, the Unity editor offers many tools for convenience. Because I knew that, at this stage of my research, I had to consider a cross platform design for mobile and desktop devices the capability of Unity to support multiple platforms was another key argument for the usage of Unity in my project. I initially envisioned to utilise *WebGL* for my development work. According to The Khronos Group Inc. (n.d.) WebGL is an open web standard to utilise for a low-level 3D API. I used WebGL in my German diploma thesis to design and develop a form of radial menu in 3D (Reiter, 2011). However, I argue that one can easily get lost in the intricacies of low-level 3D development, for example when dealing with matrices and shaders. In this PhD thesis I wanted to keep the development focused on prototyping the actual artifact I aimed to realise. Worth noting that, I did not include the code I have written in the appendix because the amount of code is just too much. However, the code is available on request.

To continue the work on my design, mainly from the interaction perspective, I created *wireframes*. Wireframes consist of shapes to visualise interaction designs (Hartson & Pyla, 2019).

### 3.8 Methods for Development and Evaluation

To evaluate my novel method from a technical perspective, for example, rendering performance and a user-centred perspective I applied various methods throughout my research project.

#### 3.8.1 Design Walk-Through and Design Review

To evaluate my conceptual design, I carried out a single semi-structured interview as a *design walk-through* with a research participant. A design walk-through is an approach to receive initial feedback for design concepts based on, for example, storyboards and sketches (Hartson & Pyla, 2019). The interview was around 40 minutes long, the participant (Professional02) was the same person who already participated in phase 3 to understand user needs. Again, I utilised a combination of Provisional Coding and Descriptive Coding for first cycle coding (see Section 3.6.2). At this stage of my research, I aimed to identify precise changes to my design. Therefore, I did not carry out some form of second cycle coding to identify patterns or themes in the interview transcript. I coded and analysed the results utilising NVivo.

To evaluate my wireframes, I carried out a single semi-structured interview as a *design review* with a research participant presenting my wireframes to identify challenges in my design primarily from the interaction perspective. A design review is more extensive than a design walk-through and is often carried out using a click-through wireframe prototype to demonstrate navigation and workflows (Hartson & Pyla, 2019). The interview was around 60 minutes long, the participant (Student01) was the same person who already participated in phase 3 to understand user needs. For coding I utilised a combination of Provisional Coding and Descriptive Coding for first cycle coding to identify precise challenges in my design (see Section 3.6.2). I utilised NVivo to code and analyse the results.

### 3.8.2 Technical Prototyping

To evaluate my novel method from a technical perspective I measured the rendering performance with different approaches and node counts using Unity. Also, I identified an edge case for my method by rendering a very deep tree in Unity.

### 3.8.3 Video-based Prototype and Evaluation

For a final, rather user-centred evaluation of my novel method I produced a video presentation primarily visualising a scenario where a student works on an entomology project using my method. For this, I developed the whole scenario in Unity adding static image overlays for the app bars, buttons, menus, dialogs, virtual joysticks, notes, input text fields, and pointers, to highlight interaction and pop up windows. This video-based prototype covered all of my relevant user stories and key (sub) tasks. I then presented the result to research participants describing the navigation features, overall layout and aim. The whole presentation was recorded as a video presentation. I added my explanations as a voiceover to the video. The video was shared with research participants using Microsoft OneDrive. The participants received all the relevant links for their evaluation through e-mail. Participants were staff and students from the University of Gloucestershire.

At the end of the presentation the participants were asked to fill out an online questionnaire which I created using SurveyMonkey. I utilised a standard questionnaire called the *User Experience Questionnaire (UEQ)*. According to Laugwitz et al. (2008) their UEQ covers six different factors: efficiency, dependability, stimulation, novelty, perspicuity and attractiveness. My choice for this questionnaire was informed by the circumstance that other researchers who utilised a similar research design used it for their evaluation. For example, Sukamto et al. (2020) explained that they used UEQ for their final evaluation stage. Ilmberger et al. (2009) asked research participants to fill out the UEQ after showing a video visualising common tasks carried out in an online shop as a part of their user study. I, therefore, argue that it is a justifiable choice to measure certain aspects of my method. Worth noting that, my video presentation had a reduced playback speed so the participants would have a better chance to understand the individual steps of navigation in depth. However, this may have impacted the result scores of the UEQ in comparison to them actually using the method themselves.

I utilised the latest version of the UEQ and guidance in English language. The questionnaire is published by Hinderks et al. (n.d.). In total, 13 individuals participated and completed my questionnaire.

Alternatively, I could have also used the *System Usability Scale (SUS)*. According to Brooke (1996) the SUS is a scale with 10 items designed to provide a global view for subjective assessments regarding the usability of a system. SUS was already used to evaluate a lot of different systems. To exemplify, Wijaya et al. (2019) used SUS to evaluate the usability of their system, which utilises augmented reality for food advertising on mobile phones. However, I chose not to use SUS in my project as it only evaluates the usability of a system.

### 3.9 Chapter Summary

Describing my methodology while contextualising alternative approaches shows the complexity of my project. Hence, it is a combination of design and software development work. To exemplify, the task to identify user needs based on qualitative data like interview transcripts is totally different from mathematically realising a node layout in 3D.

Overall, for guidance and practicability I chose to utilise parts of the UX design lifecycle as outlined by Hartson and Pyla (2019) to design for user needs in a bottom-up approach. For a more rigorous result I analysed the qualitative data using multiple coding methods and analysis software. Similarly, I chose the UEQ as a rigorous option for the final evaluation of my video-based prototype.

Unavoidably, I had to make compromises during my research which led to changes in my methodology.

I initially planned to recruit elderlies to understand user needs. I argue, that elderlies were not exposed to modern technology to the same extent as younger individuals are. Therefore, I argue that working with elderlies may lead to a less biased result when identifying user needs. However, as I could not recruit any elderlies, for example, by contacting retirement homes, I had to change my recruitment focus.

Also, I initially aimed to fully develop my prototype for evaluation, for example, to measure task performance in a user study. Hence, I especially wanted to observe research participants when they first would have used my method. What feelings would my method evoke? However, after I finished my analysis to understand user needs it became clear that the resulting design becomes far too large to actually be developed by myself. Also, even if I could have finished development of the prototype, I would not have been able to organise a user study, for example, acquiring enough mobile phones, tablets and desktop PCs for testing. Moreover, it was also not feasible to publish a mobile application to the respective application stores and setup the required server backend for remote evaluation.

Alternatively, I could have just narrowed down my scope for development ignoring fundamental user needs that I identified, for example designing and developing my method for desktop PCs only. However, for me this was not an option as cross-device compatibility was a key requirement that I identified during this research. Hence, this challenge may be one of the main reasons why no techniques to visualise hierarchical data utilised a user-centred design approach.

# Chapter 4 Understanding User Needs

## 4.1 Introduction

In this chapter, I describe my attempt to understand how users view and navigate in hierarchical data to inform my design (see Section 2.7). I chose the example of file and folder management as the focus topic for data elicitation and analysis. Overall, I applied a two-fold strategy to understand user needs (see Section 3.6). To reiterate, in the first part (general context) of this strategy I aimed to understand:

- How important are mobile devices in daily routines?
- What other areas besides file and folder management may be relevant for additional case studies, for example, to apply the method for social media?
- How important are products like virtual reality devices in regular use? Does this topic motivate interviewees to speculate about new visualisation techniques relevant to my research?
- What criteria are important to a user when selecting an application? What type of data they use, for example, text documents, pictures? The answers to these questions helped to identify important requirements for my method.
- Are 3D applications being used on a regular basis or not?

The second part of the strategy was to understand user needs in the context of file and folder management. Hence, the design and evaluation target. Therefore, I aimed to understand how do participants browse and maintain their filesystem(s), what are they currently missing.

In Section 4.2 I describe my initial modelling to create a first work role model and flow model. Consequently, in Section 4.3 I detail the design and results of my pilot interview. Section 4.4 describes the approach and results of phase 1 interviews. Section 4.5 details the design and result of the questionnaire for triangulation. Similarly, Section 4.6 describes the approach and results for the final phase 3 interviews focusing more on the file and folder management context for triangulation. In Section 4.7 I outline my synthesis on what I have learned leading to my final models and user stories. Section 4.8 provides a short summary of the results.

Worth noting that, all my first cycle codes (Descriptive Codes) are highlighted in *italics*. My second cycle coding patterns are highlighted in **bold**.

**4.2 Initial Modelling**

I established a first draft for my user work role model based on my personal assumptions and target user group (see Section 3.3). I can identify the following roles in the context of file and folder management:

- *System User*, who interacts with a mobile or PC desktop device through 2D output (monitor) and input devices (mouse, keyboard, touch input) to store and find items of interest (documents, pictures) for viewing or editing.
    - Student, a sub-role who is a university student.
    - Professional, a sub-role who has an occupation.
- "*Developer*", who develops, or maintains, the tools to find and edit documents, for example Windows Explorer.

Worth noting that, my user work role models and flow models in this chapter are aimed to capture the current state of the work practice (ecology) at the time of writing, they are not aimed to describe the target design for my novel method.

Similarly, I created the first draft for my flow model visualising the flow of information (see Figure 4.1). For all of my models, in this chapter, I used a similar layout, similar elements (for example work roles) and symbols as in Hartson and Pyla (2019).

Still at this early-stage, modelling was very valuable. Hence, I was not thinking of the Developer work role before in this context. Although, my research focuses on the end-user of the method I should utilise standardised development tools. Otherwise, it may be difficult to evaluate my method further. Shen et al. (2019) highlighted the need for a reconfigurable information visualisation system to enable the evaluation of developed visualisation techniques.

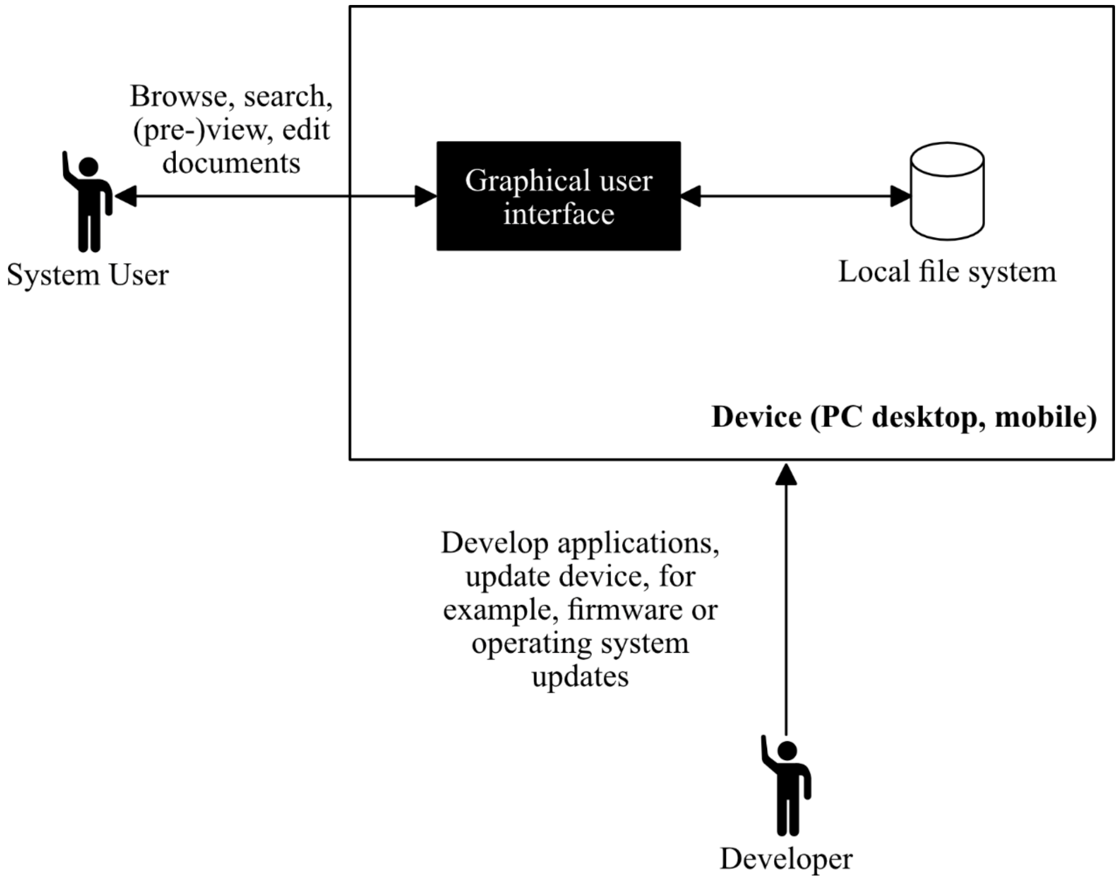



Figure 4.1: Initial Flow Model

*Note*. Initial flow model visualising file and folder navigation on PC desktop or mobile devices.

**4.3 Pilot Interview**

I carried out one semi-structured interview with a university student to ascertain that my questions are clear and unambiguous. The guiding questions were:

1. What do you do for a living (job)?
2. Describe your daily activities/tasks (focus on usage of IT devices).
3. What sort of data you use on a daily basis (for example, photos, databases, spreadsheet documents)?

4. How do you view your data (for example, mainly list based, slideshows)?
5. How do you navigate through your data (for example, find documents, browse your files)?
6. What do you miss when navigating through your data?
7. How would you like to navigate through your data (for example, personal documents)?
8. What is your primary goal when navigating through your data (for example, UI performance, design)?
9. Do you order data on your PC in a similar way as you do in real life (for example, like items on your physical desk)?

Based on the pilot I decided to keep the last question (9.) only as an expandable question. Initially, I asked this question to get an indication on the effect of the Desktop metaphor. In other words, is there a direct link supporting the analogy to real office spaces? However, I argue that based on its widespread success, for example in operating systems like Microsoft Windows, the question seemed expandable. Still, I had time to ask the question in one of the other two interviews in phase 1. The answers were analysed to potentially gain additional insights. Also, I added a new guiding question after the pilot interview:

10. Do you use virtual reality products/software? If so, why (context)?

I added this question because when I asked the participant what they want to improve when managing their files and folders (question 7) I did not get a response. Consequently, I asked if they had maybe seen some interesting or futuristic approach in a video game or movie like a holographic display, something they found inspiring. Still, I did not get any further feedback as the participant was quite satisfied with the current techniques available to them. Regardless, it seemed to me that my examples initiated a reflective thought process for the participant. Consequently, for a futuristic device I picked virtual reality devices as at the time of writing they are already in use, for example for entertainment. Also, I thought it may be interesting to learn about the actual use of those products.

In the pilot I also asked what interviewees value when selecting software in general, for example, is usability the most important component? In the subsequent interviews I also asked this question as a variant of question 8. I also concluded from the pilot that questions may only be asked if they have not already been answered by former responses.

### 4.4 Phase 1 Initial Interviews

Using the adjusted questions from the pilot I carried out two semi-structured interviews. I then imported the two interview transcripts into NVivo to start analysis.

Based on my analysis strategy and literature review I chose the following starting categories and codes as a form of Provisional Coding:

- file and folder management context (category)
    - tasks (category)
    - user stories (category)
        - emotional impact (category)
        - usability (category)
        - usefulness (category)
- general context (category)
    - *applications 2D, 3D* (code)
    - data types (category)
    - *mobile devices* (code)
    - selection criteria (category)
        - *emotional impact* (code)
        - *usability* (code)
        - *usefulness* (code)
    - tasks (category)

- *virtual reality devices* (code)

Table B1 lists my Descriptive Codes after first cycle coding. These codes were manually created and reviewed once by myself. For transparency, I could have split the codes *usefulness*, *usability* and *emotional impact* into more detailed codes. To be precise, Professional00 also mentioned *compatibility* as a relevant criterion for selecting software. However, I did not add this potential code (*compatibility*) to general context/selection criteria nor did I add it as a user story in file and folder management context/user stories/usefulness. I argue that *compatibility* is a valid example for *usefulness* but in the context of designing my method at this stage (thesis) it cannot inform my design.

Worth noting that, I also coded sections to *location independence* when participants highlighted challenges when working with others, for example, when working with shared online storage like Google Drive. One may argue that a separate code like *improve communication* would have been better to separate the requirements. However, I argue that *location independence* is a fitting term to describe the need for abstraction, overview and central control over one's data.

For transparency I introduced the code *speech recognition* to describe a potential need to allow speech recognition as an input method. When asked in the context of file and folder management what they are currently missing when managing their data Professional00 responded: "*The Alexa. But of course, that is [laughs] absolutely a dream of the future (...) that one says Alexa or some tool tell me please short (...) the registration numbers of BMW, from the first half year 2019.*". Consequently, I asked for clarification if they truly meant speech control or rather an improved, intelligent search feature. Thus, Professional00 responded "*that the device is that smart or understands it that he practically prepares this for me*". Regardless, I kept the code for transparency, as *speech recognition* in part informed their dream of the future. However, I did not use the explicit code in the following phase 2, for example, to inform the design of my questionnaire in phase 2. Given, the single coding reference and described misinterpretation I decided to not follow up on *speech recognition*.

For contextualisation, the code I called *location independence* is semantically similar to the *continuity* concept. According to Chacón-Vázquez et al. (2021) continuity is realised in systems which allow for synchronisation across devices to maintain a constant state to allow a user to continue their task. Similarly, the design for *device independence* is not a novel area of research. For example, Nebeling et al. (2014) presented an interface builder to support the development of cross-device web user interfaces.

Worth noting that, I identified indicators for activities in my source material. Norman (2005) introduced the *Activity-Centered Design* where activities consist of tasks, tasks consist of actions and actions consist of operations. These results supported my conceptual design from the ecological perspective (see Section 5.4.1). To be precise, these indicators informed my design by acknowledging that users carry out tasks to achieve their goal (activity). In my research I differentiated between tasks and their sub-tasks, such as maintaining a *customised structure* for files and folders (task) and create a new node like a folder as its sub-task.

### 4.4.1 Mobile Device a Cluttered Companion

The code *mobile devices* covered around 10 percent of my source material. In detail, for the Professional00 transcript 10,04% and for the Professional01 transcript 9,95%. The percentages indicate the importance of mobile phones in the participants' daily routines. Also, the participants' statements imply a strong feeling of companionship (meaningfulness) towards their mobile phone. When asked what they do first after getting up Professional00 responded, "*I will take the mobile phone first*". Figure 4.2 visualises a NVivo coding matrix to show the count of coding references where I utilised *mobile devices* and other codes (intersections). The results indicate that, in the context of file and folder management tasks, participants rely more on *search* features and do not use a custom file and folder structure (*no structure*) when using their mobile phone. When asked if they rely on devices' preinstalled search features and unstructured (random) browsing to find their files on their mobile phone, Professional01 responded “*Exactly, in the hope that I organised it in a way on my other computers to find it also in the cloud and it is organised.*”. In a general context the results also show an unexpected high frequency for the task *office content creation*. Again, this indicates a feeling of companionship. When asked if they deliver work results through e-mail using their mobile phone Professional00 clarified “*From the mobile phone. Because in part the data is available to me as PDFs. These are so marketing programs (...) and or also Excel tables for example.*”. Also, the results indicate that a limiting factor for using a mobile phone is the lack of available screen space. To quote Professional01 “*and my telephone it is what it is. So, I believe that it is the weakest link in the chain. Because just small and difficult to organise sometimes. Because one only sees a part, one file name or there is not enough space for a visualisation*”. Consequently, as a hint for a user story for usefulness in the context of file and folder management I called this topic *screen space utilisation*. Also, the results indicate a potential need for *device independence*, to quote Professional01: “*What would be nice, if all those desktops across my three devices would be the same.*”.

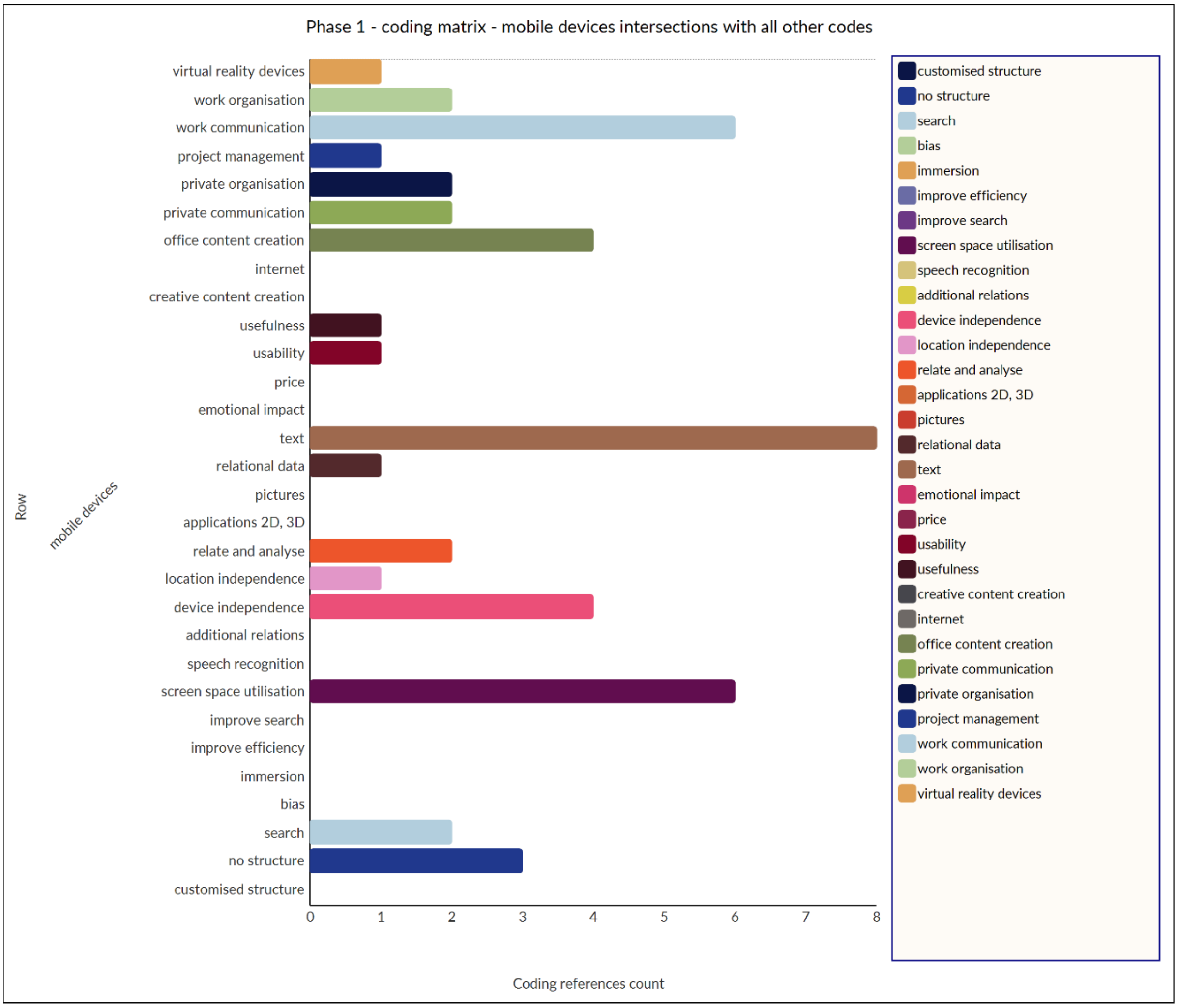



Figure 4.2: Phase 1 – Coding Matrix – Mobile Devices Intersections With all Other Codes

*Note*. NVivo coding matrix visualising which codes were used for sections also coded to *mobile devices*.

### 4.4.2 The *Be Here Now* Metaphor

In the context of file and folder management Professional01 said "*Old thoughts, old memories of the past (…), fears and wishes for the future and I believe like I clear my mind and try to be always here to see only things I want to see now. There I do not want all the old photographs or all the old stuff or always have to go through that to come to here. Also, I do not want to somehow try (…) find the connection to something. (…) Why (…), how or where I got this idea from. I always want to be here.*". I coded this section to user stories, usefulness, additional relations. When reviewing my first cycle codes in contrast to this potential pattern I questioned if this metaphor is so strong it can even inform software design more generally. Consequently, I aimed to compare all of my first cycle codes against each other by their contained words. Figure 4.3 visualises the NVivo cluster analysis result for word similarity between all first cycle codes. The result shows only one rather strong (≥0.7) relationship between *work organisation* and *private organisation* (0.918919). This is a case where I coded sections twice to two semantically similar codes because the source material did not allow for a clear distinction between work or private use. The second strongest relationship has only a Jaccard coefficient value of 0.581395, the rest falls below 0.4. Overall, these results indicate that my first cycle coding captured rather distinct topics without much redundancy. Unfortunately, these results cannot serve as an indication to the metaphor. Still, when analysing just the codes in the context of file and folder management I argue that the metaphor can be seen as a summary for the respective sections. When talking about a feature to *improve search* capabilities Professional00 also stated "*he also has to search the one 2016. The other 2017, the one at the other customer*". This quote for *location independence* aligns to the metaphor because it can also be viewed as a need to reduce the complexity of our digital life. The only code in the context of file and folder management which is not a good fit at first glance is *bias*. This code highlights a design goal for user familiarity, for example, by utilising familiar design metaphors. To quote Professional01 "*So the keyboard, the classic keyboard comes from an insane past, yes. Has actually nothing left to do with reality but we still have it.*". However, one may also view this form of *bias* as a part of the participants ecology at a certain point in time or, simply put, now.

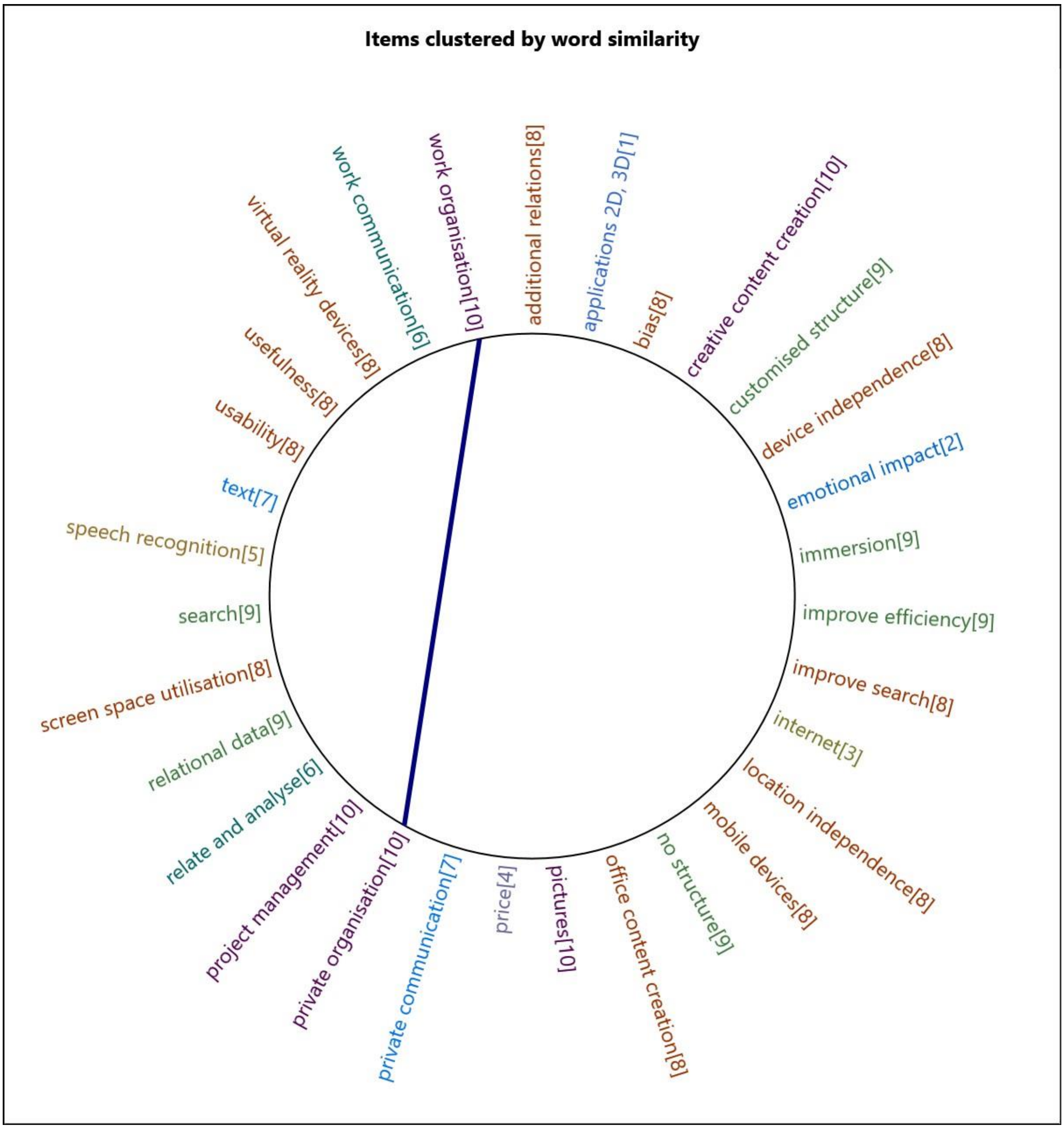



Figure 4.3: Phase 1 – Cluster Analysis – Word Similarity Between all First Cycle Codes

*Note*. NVivo (code) cluster analysis result as a NVivo circle graph visualising all first cycle codes and their similarities (distance) clustered by word similarity utilising the Jaccard coefficient as a similarity metric. Similar items are connected by a blue line, a thicker line indicates a stronger similarity. Lines are only drawn for items with a Jaccard coefficient of 0.7 or higher (≥0.7). Cluster IDs are listed in brackets after the labels, for example, [1]. Results were calculated for 10 target clusters.

### 4.4.3 Structure is Control

When I summarised my understanding of how they use their laptop that they keep their files quite unorganised, Professional01 responded "*Exactly … These are the ideas I need to deliver quickly. Telephone is too small, home computer does not work*". In the context of file and folder management this whole discussion informed multiple codes. For one, it indicates a scenario when *no structure* is used. Also, it indicates a need to *improve efficiency* for file and folder management and *screen space utilisation* for *mobile devices*. Moreover, it is pointing to the need for *device independence*. In short, the results imply that a *customised structure* is the preferred approach when the situation allows for it, for example when there is time or efficient tools are available to do it. Consequently, I argue that this pattern also fits nicely into the **be here now** metaphor. Figure 4.4 shows a NVivo hierarchy chart (Treemap) where the first cycle codes in file and folder management context/tasks are compared by their number of coding references. The chart shows that *no structure* has the most coding references. However, it has only a small lead with seven references over *search* and *customised structure* with five references each.

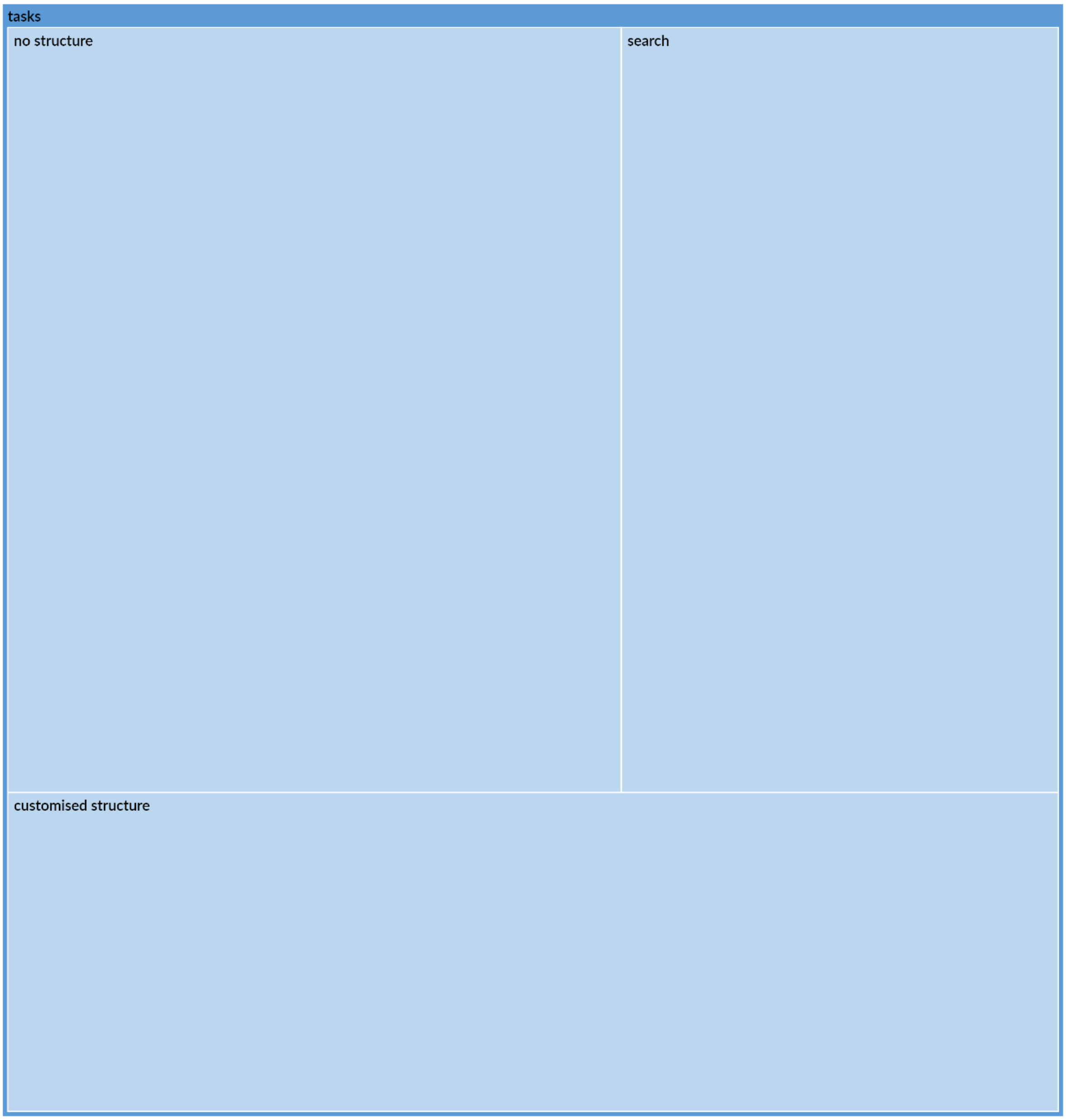



Figure 4.4: Phase 1 – Treemap – File and Folder Management Context/Tasks – Number of Coding References

*Note*. NVivo hierarchy chart (Treemap) visualising (compared by) the number of coding references for file and folder management context/tasks.

### 4.4.4 Faith in Virtual Reality

The results indicate that *virtual reality devices* are rarely used but are still linked to the possibilities of the future. For example, Professional01 stated "*I developed virtual reality software myself*". However, later in the interview they said "*I have two virtual reality devices* *at home. But I think there is little content. Momentarily, what is really interesting to me but I believe it is an area where much could come.*". However, this mental link to the possibilities of the future was a fruitful path to discuss possible solutions for viewing data. Figure 4.5 visualises a NVivo coding matrix to show the count of coding references where I utilised *virtual reality devices* and other codes (intersections). The results imply a user need for *device independence*, *immersion* and improved *screen space utilisation*. To exemplify, Professional01 described a future scenario to address *device independence* and improved *screen space utilisation* in the context of *mobile devices* "*And then that VR would, so the glasses and the glasses would show the same from all three devices. Because then then we do* not *have that (…), this interface (…) (shorthand?) display area. Because I believe that is always the bottleneck, yes. The display area, size, style. That would be interesting, then one could with Bluetooth have a way to work with a smart phone, with a laptop and with other devices at the same time*". Again, the results indicate a nice fit with the **be here now** metaphor. For instance, Professional01 stated "*Of course, yes. And therefore, these 3D rooms and 3D mapping therefore the two topics are really strongly connected by me, I believe. If one could look around and be in the middle of the mind map*". This quote indicates a need to be at the centre of one's cleaned up digital world.

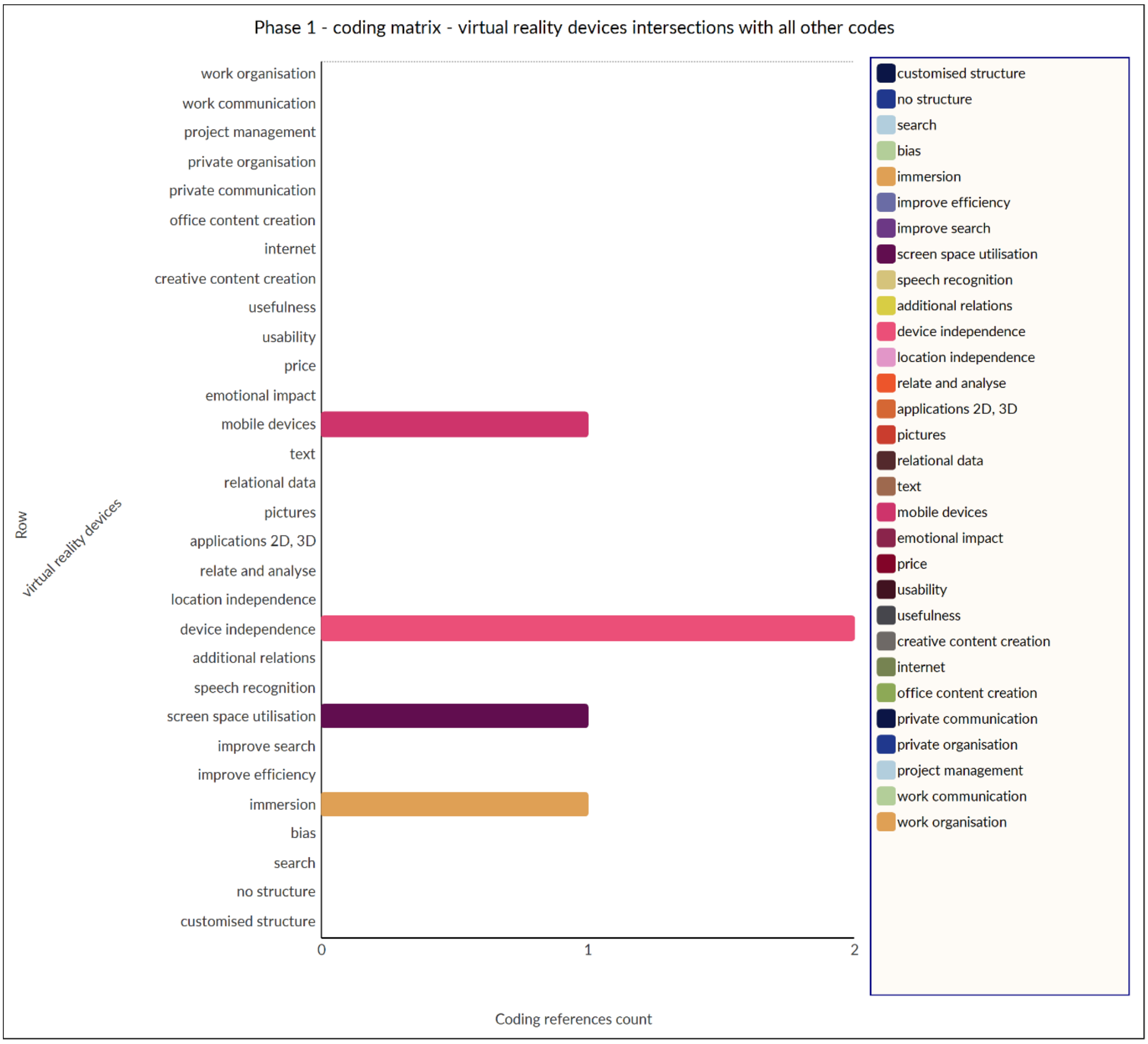



Figure 4.5: Phase 1 – Coding Matrix – Virtual Reality Devices Intersections With all Other Codes

*Note*. NVivo coding matrix visualising which codes were used for sections also coded to *virtual reality devices*.

### 4.4.5 Office, Communication and Organisation

Tasks are mainly **office, communication and organisation** related. The results imply that tasks mostly consist of *office content creation*, *private communication*, *work communication*, *private organisation* and *work organisation*. Figure 4.6 shows a NVivo hierarchy chart (Treemap) where the first cycle codes in general context/tasks are compared by their number of coding references. The most coding references has *office content creation* with a total of 12 references. In second place is *work communication* with nine references. Third is *private organisation* with four references. Fourth is shared between *work organisation* and *creative content creation* with three references each. Fifth is *private communication* with two references. Last place is shared between *internet* and *project management* with only one reference.

Worth noting that, there are cases where I coded sections twice to two semantically similar codes because the source material did not allow for a clear distinction between work or private use, for example, for *work communication* and *private communication*. After reviewing the redundancies, I counted four distinct code references in total for organisation, coded to *work organisation* and/or *private organisation*. For communication I counted nine distinct code references coded to *work communication* and/or *private communication*. However, this does not change the order for the top three tasks with the most coding references. To summarise, *office content creation* has the most references followed by communication and then organisation.

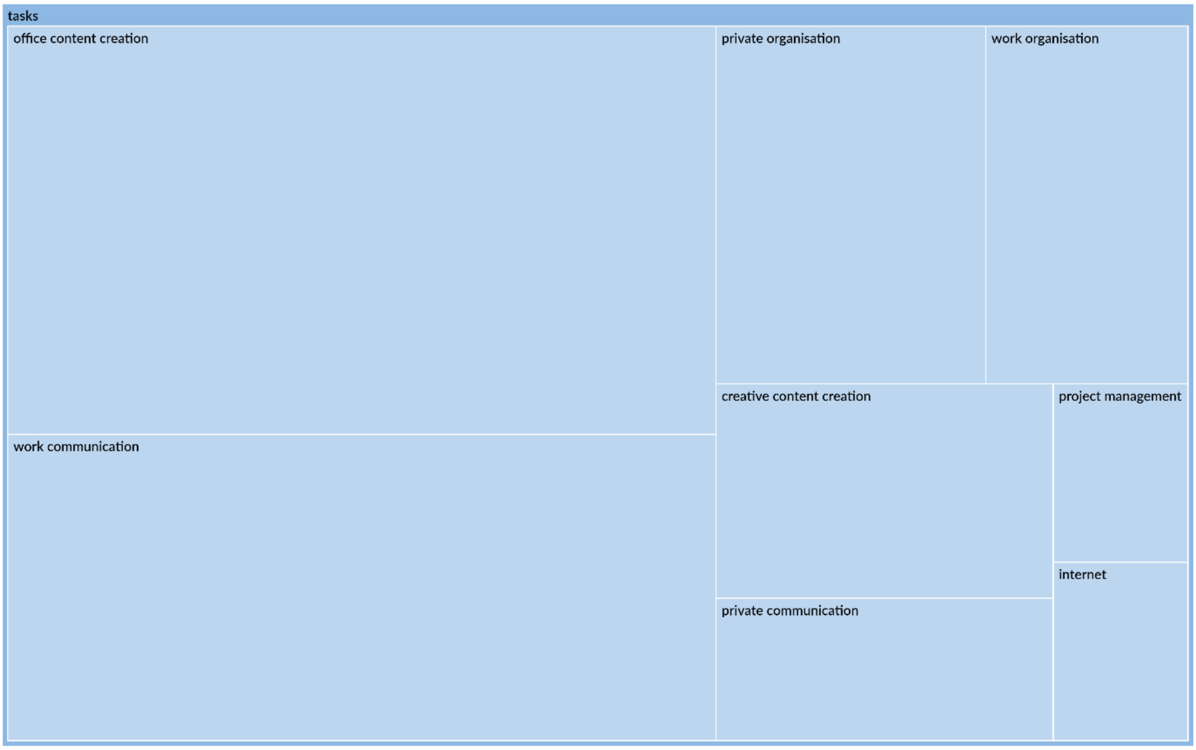



Figure 4.6: Phase 1 – Treemap – General Context/Tasks – Number of Coding References

*Note*. NVivo hierarchy chart (Treemap) visualising (compared by) the number of coding references for general context/tasks.

### 4.4.6 Usable Software With Perceived Value

In the general context, results indicate that software is primarily chosen based on the application's perceived *usefulness* and *usability*. *Usefulness* is perceived as the most important component, to quote Professional00 "*So in the end, the first point has been that the different countries and organisations work with the same tool*". The *usability* is perceived as the second most important component. When I summarised my understanding on how they choose software primarily by its *usefulness*, business case Professional01 responded: "*Exactly. Then the usability*". The results indicate that *emotional impact* does not consciously inform an individual's choice towards an application. When asked if an appealing design is rather secondary to them Professional00 answered "*Yes, totally irrelevant. Zero point zero.*". For transparency, Professional00 also talked about how the *price* of an application was the least relevant decision criteria, for example, “*But cost point was at the back*”. Figure 4.7 shows a NVivo hierarchy chart (Treemap) where the first cycle codes in general context/selection criteria are compared by their number of coding references. The figure supports the pattern as the most coding references has *usefulness* with a total of 10 references followed by *usability* with eight references. In third place is *emotional impact* with three references. In last place is *price* with just two references.

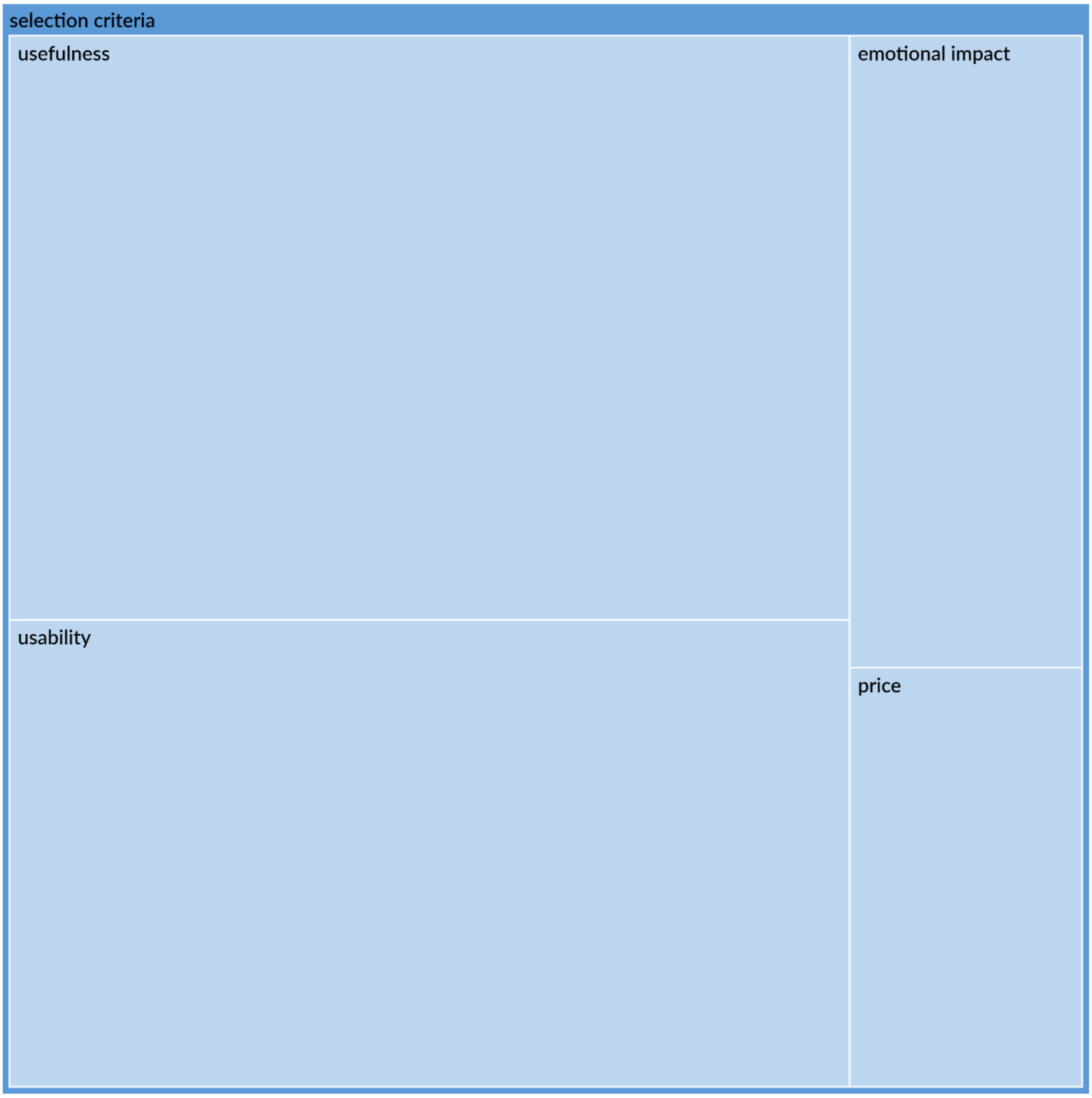



Figure 4.7: Phase 1 – Treemap – General Context/Selection Criteria – Number of Coding References

*Note*. NVivo hierarchy chart (Treemap) visualising (compared by) the number of coding references for general context/selection criteria.

### 4.4.7 Work Practice Ecology is Text Based and 2D

Overall, results indicate that the most used data type is *text* with twenty-eight coding references. The second most used data type is *relational data* with eight references followed by *pictures* in last place with four references. It indicates a strong focus on text-based content, for example in the context of working with Microsoft Office applications. For instance, Professional01 said “*But essentially, a lot of Excel spreadsheets.*”. Figure 4.8 visualises the result of a NVivo word frequency query as a word cloud for sections coded to general context/data types. This result also indicates that the most used data type is indeed *text* followed by *relational data*. Moreover, it indicates that most of the utilised applications are 2D based, for example, databases, Microsoft Excel, e-mails. Hence, when I summarised my understanding that Professional00 still utilises more list based 2D applications in their daily routine they responded “*Yes, still.*”.


Figure 4.8: Phase 1 – Word Frequency Query – General Context/Data Types – Word Cloud

*Note*. NVivo word frequency query result visualised as a world cloud for sections coded to general context/data types. The display words were limited to 1,000 with a minimum length of five characters. For grouping the setting with stemmed words was chosen. The top nine words with each more than two counts were Datenbanken (databases), Excel, genau (exactly), Mails, Access, PowerPoint, reden (to talk), Spreadsheets, Zeilen (lines).

## 4.5 Phase 2 Online Questionnaire

For triangulation I designed an online questionnaire to either support or reject my findings from phase 1. To reiterate, 22 university students from the University of Gloucestershire participated in this questionnaire. Table B2 shows the questions which were asked in the context of my results from phase 1.

### 4.5.1 Mobile Device a Cluttered Companion

I argue that the result supports the pattern **mobile device a cluttered companion** regarding the importance of *mobile devices* in daily routines. Table 4.1 shows the responses for question Q9. On average it is close to a 50:50 split between mobile and desktop PC devices. A total of 10 respondents entered a percentage of more than 50 (>50) for "*Mobile (Smart phone, Tablet)*". Of those 10 respondents all entered a number equal or greater than 60 (≥60) for "*Mobile (Smart phone, Tablet)*", on average 69.

**Table 4.1**

*Q9: Which IT Device do you Mostly use in General (Work and/or Home)? Please Insert Numbers to Reflect Percentage (for Example, 60 to 40)*

| ANSWER CHOICES | AVERAGE NUMBER | TOTAL NUMBER | RESPONSES |
|---|---|---|---|
| Mobile (Smart phone, Tablet) | 51 | 1,120 | 22 |
| PC (Laptop, Desktop) | 49 | 1,080 | 22 |
| Total Respondents: 22 | | | |

*Note*. SurveyMonkey result data table. Exported survey data, SurveyMonkey export settings: summary data, PPT, original view, charts and data tables. Grammatical and spelling errors (if any) in the questions (see figure title) were corrected for this view. Also, the readability was improved by replacing "*e.g.*" with "*for example*,".

The results for question Q16 also support in part the pattern regarding the reliance on *search* and *no structure* when using *mobile devices*. Table 4.2 shows the responses for question Q16. Most, to be precise eight participants mainly utilise *search* features when looking for their files on their mobile device. Surprisingly, at second place with seven responses is the utilisation of a *customised structure*. Only four respondents use *no structure* at all and three answered that they use a combination of a *customised structure* and *search*. Hence, the mere three responses for the combination to use a *customised structure* and *search* supports the impression that these tasks contradict each other at least for *mobile devices*. When viewed in conjunction with the seven responses for using mainly a *customised structure* it supports the pattern that I called **structure is control**. In other words, individuals seem to prefer a *customised structure* when the situation allows for it. Also, I argue that the seven responses for mainly using a *customised structure* in part support the feeling of companionship towards *mobile devices*. Hence, building and maintaining a *customised structure* takes effort. One may argue that it is also possible that the participants build and maintain those structures with other devices and they only mirror the result to their mobile devices, for example, by utilising a cloud service. However, when asked which data locations they regularly use for question Q11 (see Table 4.7) in total only two respondents answered that they solely use cloud services or personal network storage devices. Of those two none of them also answered that they mainly use a *customised structure* for question Q16.

**Table 4.2**

*Q16: When you are Looking for Your Relevant Files (for Example, Photos, Spreadsheets, Mails, Messages) on Your Mobile Device (for Example, Smart Phone) how do you Find Them?*

| ANSWER CHOICES | RESPONSES | |
|---|---|---|
| I mainly use a custom folder/file structure (e.g. folders by subject->date, mail by categories) | 31.82% | 7 |
| I mainly rely on search tools (e.g. iPhone swipe to search) | 36.36% | 8 |
| I use a combination of both (structure and search) | 13.64% | 3 |
| I mainly just browse folder by folder until I find what I need (no structure, no search) | 18.18% | 4 |
| TOTAL | | 22 |

*Note*. SurveyMonkey result data table. Exported survey data, SurveyMonkey export settings: summary data, PPT, original view, charts and data tables. Grammatical and spelling errors (if any) in the questions (see figure title) were corrected for this view. Also, the readability was improved by replacing "*e.g.*" with "*for example,*".

### 4.5.2 Be Here Now

When asking for what the participants are currently missing when managing their files and folders, I aimed to evaluate the core of the **be here now** design metaphor. Table 4.3 shows the responses for this question Q19. The options for improvement were informed by the following codes (in order): *location independence*; *improve search*; *immersion* and *screen space utilisation*; *device independence*; *additional relations* and *relate and analyse*; *bias*. The results support the design metaphor, the average across the six options (weighted average) is $(3.41 + 3.77 + 3.77 + 3.95 + 3.27 + 3.86)\ /\ 6 = 3.67$. The value indicates that the participants overall somewhat agree that the offered options for improvement would be welcome additions. Interestingly, the option for *device independence* has a relatively high weighted average of 3.95. This result supports the interest for a coherent interface across devices. Similarly, the option for more customising options (bias) has a weighted average of 3.86 which supports an interest in owning the experience, for example, by utilising familiar or pleasant icons. On the other side of the spectrum the lowest weighted average has the option for new ways to structure data (*additional relations* and *relate and analyse*) with a value of 3.27. Hence, I assume that this option was difficult to interpret or better imagine for the participants. Therefore, the response was rather neutral. Worth noting that, participants also had the option to enter other features they want as free text. There was only one entry "*the option to use regular expression on multiple platforms i.e. Google Drive*". I argue that this quote fits to the code *improve search*.

The results for question Q22 (see Table 4.4) also support the be here now metaphor regarding utilising bias for a potential design. Hence, 54.55% already tried at least once to order their data on a PC in a similar way as they do in real life. Again, utilising user habits, customs to inform their digital reality.

Results for question Q15 (see Table 4.5) and 16 (see Table 4.2) were non-conclusive. If the participants would have used only a single application to manage their files and folders on PC and *mobile devices,* I argue that the result would have supported a need for *device independence*. Meaning, a coherent user experience would have been more important to them than an activity dependant, optimised one. On the PC desktop side (question Q15) 90.91% of the participants stated that they solely use a single application to manage their files and folders. In contrast for mobile devices only 50.00% answered that they use a single application. I conclude that the results in part support a need for *device independence*. I can only speculate on why for *mobile devices* 50.00% of the participants use multiple applications.

The results for question Q11 are shown in Table 4.7. Most of the participants (45.45% or 10 responses) only use local device storage in combination with one cloud service to store their regularly used files. When viewed in conjunction with the 13.64% (or three responses) who stated that they only use local device storage one may initially assume that there is no argument for a need for *location independence*.

However, when filtering for all respondents in question Q19 who stated that they either somewhat agree or agree that they miss features for *location independence* and reviewing their answers for question Q11 the picture changes or becomes rather interesting. Out of those 10 filtered participants five just use a combination of local device storage and one, single cloud service. Another, single respondent just uses local device storage. Two participants use a combination of one cloud service, local device storage and personal network storage. Another, single respondent just uses personal network storage while the last participant selected that they only use two or more cloud services. Initially I assumed that with an increasing number of data locations the need for centrally managing those data sources would increase. Hence, the need for *location independence* would increase. However, six participants (60%) out of the 10 either just use a combination of local device storage and one single cloud service or local device storage only. I conclude that the amount of data locations is not linked to a need for *location independence*.

The results for question Q10 support a need for *device independence*. 36.36% (or eight respondents) stated that they mostly use cloud-based applications to complete their regular tasks. When filtering for these eight respondents to review their answers for question Q19 the impression increases. Out of those eight respondents three of them somewhat agree that they are missing features for *device independence*, another three agree and two chose neutral. The weighted average is $(2 \times 3 + 3 \times 4 + 3 \times 5) / 8 = 4.13$, in comparison the weighted average for all 22 respondents is 3.95 (see Table 4.3). Therefore, I argue that the usage of cloud-based application is related to the need for *device independence*. Consequently, because the number of participants who stated that they mostly use cloud-based applications was unexpectedly high I argue that the results for question Q10 support the need for *device independence*. One may also claim that there is a potential relationship between the need for *location independence* and the usage of cloud-based applications. When filtering for the same eight respondents to review their answers for question Q19 this claim can be supported. Out of the eight respondents five stated that they view a potentially missing feature for location independence rather neutral while one participant somewhat agrees and two agree. This shields a weighted average of $(5 \times 3 + 1 \times 4 + 2 \times 5) \,/\, 8 = 3.63$, the weighted average for all 22 respondents is 3.41 (see Table 4.3). Again, participants who mainly use cloud-based applications also somewhat agree for a need towards features for *location independence*. Hence, Table 4.9 shows a comparison for all of the six potential features and their weighted averages from question Q19. I argue that because in general the averages for almost all features are higher the support for the **be here now** metaphor is larger when individuals rely more on cloud-based applications.

**Table 4.3**

*Q19: What are you Currently Missing When Navigating / Searching for Your Files (for Example, Photos, Spreadsheets, Mails, Messages) on Your IT Device (Mobile and/or PC)?*

| | DISAGREE | SOMEWHAT DISAGREE | NEUTRAL | SOMEWHAT AGREE | AGREE | TOTAL | WEIGHTED AVERAGE |
|---|---|---|---|---|---|---|---|
| Location independence (navigate and search various locations like Dropbox and your local stored files) | 4.55%<br>1 | 4.55%<br>1 | 45.45%<br>10 | 36.36%<br>8 | 9.09%<br>2 | 22 | 3.41 |
| More efficient search engines for quicker and better results | 0.00%<br>0 | 9.09%<br>2 | 22.73%<br>5 | 50.00%<br>11 | 18.18%<br>4 | 22 | 3.77 |
| Enhanced field of view to see more information (e.g. see more than just one or two folders at a time) | 4.55%<br>1 | 13.64%<br>3 | 13.64%<br>3 | 36.36%<br>8 | 31.82%<br>7 | 22 | 3.77 |
| Device independence (have one coherent GUI for navigation across mobile and desktop devices) | 0.00%<br>0 | 4.55%<br>1 | 18.18%<br>4 | 54.55%<br>12 | 22.73%<br>5 | 22 | 3.95 |
| Offer new ways to structure your data (e.g. order all your data around subjects e.g. projects) | 0.00%<br>0 | 27.27%<br>6 | 31.82%<br>7 | 27.27%<br>6 | 13.64%<br>3 | 22 | 3.27 |
| More customizing options regarding the design (e.g. change the colour scheme or icons) | 0.00%<br>0 | 9.09%<br>2 | 13.64%<br>3 | 59.09%<br>13 | 18.18%<br>4 | 22 | 3.86 |

*Note*. SurveyMonkey result data table. Exported survey data, SurveyMonkey export settings: summary data, PPT, original view, charts and data tables. Grammatical and spelling errors (if any) in the questions (see figure title) were corrected for this view. Also, the readability was improved by replacing “*e.g.*” with “*for example*,”.

**Table 4.4**

*Q22: Do you (or did you Once try to) Order Your Data on Your PC in a Similar way as you do in Real Life (for Example, Like Items on Your Desk)?*

| ANSWER CHOICES | RESPONSES | |
|---|---|---|
| Yes | 54.55% | 12 |
| No | 45.45% | 10 |
| TOTAL | | 22 |

*Note*. SurveyMonkey result data table. Exported survey data, SurveyMonkey export settings: summary data, PPT, original view, charts and data tables. Grammatical and spelling errors (if any) in the questions (see figure title) were corrected for this view. Also, the readability was improved by replacing "*e.g.*" with "*for example*,".

**Table 4.5**

*Q15: Do you use Only one Application to Find/Navigate all Your Relevant Files on Your PC (for Example, Windows-Explorer)?*

| ANSWER CHOICES | RESPONSES | |
|---|---|---|
| Yes | 90.91% | 20 |
| No | 9.09% | 2 |
| TOTAL | | 22 |

*Note*. SurveyMonkey result data table. Exported survey data, SurveyMonkey export settings: summary data, PPT, original view, charts and data tables. Grammatical and spelling errors (if any) in the questions (see figure title) were corrected for this view. Also, the readability was improved by replacing "*e.g.*" with "*for example*,".

**Table 4.6**

*Q17: Do you use Only one Application to Find/Navigate all Your Relevant Files on Your Mobile Device (for Example, iPhone Documents)?*

| ANSWER CHOICES | RESPONSES | |
|---|---|---|
| Yes | 50.00% | 11 |
| No | 50.00% | 11 |
| TOTAL | | 22 |

*Note*. SurveyMonkey result data table. Exported survey data, SurveyMonkey export settings: summary data, PPT, original view, charts and data tables. Grammatical and spelling errors (if any) in the questions (see figure title) were corrected for this view. Also, the readability was improved by replacing "*e.g.*" with "*for example*,".

**Table 4.7**

*Q11: Which Data Locations do you use in a Professional and/or Personal Context for Your Regular (Used More Than Once per Week) Digital Documents (for Example, .pdf, .doc, .jpg etc.)?*

| ANSWER CHOICES | RESPONSES | |
|---|---|---|
| Local device storage only | 13.64% | 3 |
| Local device storage and one cloud service (e.g. Dropbox) | 45.45% | 10 |
| Local device storage and two or more cloud services (e.g. Dropbox) | 4.55% | 1 |
| One cloud service only (e.g. Dropbox) | 0.00% | 0 |
| Two or more cloud services only (e.g. Dropbox) | 4.55% | 1 |
| Local device storage and personal network storage (e.g. NAS) | 0.00% | 0 |
| Personal network storage only (e.g. NAS) | 4.55% | 1 |
| A combination of one cloud service, local device storage and personal network storage (e.g. NAS) | 9.09% | 2 |
| A combination of two or more cloud services, local device storage and personal network storage (e.g. NAS) | 18.18% | 4 |
| TOTAL | | 22 |

*Note*. SurveyMonkey result data table. Exported survey data, SurveyMonkey export settings: summary data, PPT, original view, charts and data tables. Grammatical and spelling errors (if any) in the questions (see figure title) were corrected for this view. Also, the readability was improved by replacing "*e.g.*" with "*for example,*".

**Table 4.8**

*Q10: Do you Mostly use Cloud Based (for Example, Google Docs) or Locally Installed (for Example, Microsoft Office) Applications to Complete Your Regular Tasks (Work and/or Home Usage Related)?*

| ANSWER CHOICES | RESPONSES | |
|---|---|---|
| Cloud based | 36.36% | 8 |
| Local applications | 63.64% | 14 |
| TOTAL | | 22 |

*Note*. SurveyMonkey result data table. Exported survey data, SurveyMonkey export settings: summary data, PPT, original view, charts and data tables. Grammatical and spelling errors (if any) in the questions (see figure title) were corrected for this view. Also, the readability was improved by replacing "*e.g.*" with "*for example*,".

**Table 4.9**

*Weighted Averages for Question Q19 Filtered for Participants who Mainly use Cloud-Based Applications*

| Feature | Weighted average (eight respondents, filtered, mainly use cloud-based applications) | Weighted average (22 respondents, total) |
|---|---|---|
| Location independence (navigate and search various locations like Dropbox and your local stored files) | 3.63 | 3.41 |
| More efficient search engines for quicker and better results | 4.13 | 3.77 |

| Feature | Weighted average (eight respondents, filtered, mainly use cloud-based applications) | Weighted average (22 respondents, total) |
|---|---|---|
| Enhanced field of view to see more information (e.g. see more than just one or two folders at a time) | 3.75 | 3.77 |
| Device independence (have one coherent GUI for navigation across mobile and desktop devices) | 4.13 | 3.95 |
| Offer new ways to structure your data (e.g. order all your data around subjects e.g. projects) | 3.38 | 3.27 |
| More customizing options regarding the design (e.g. change the colour scheme or icons) | 4.13 | 3.86 |

*Note*. Table visualising the weighted averages of answers from question Q19 restricted to the eight participants who stated that they mainly use cloud-based application in question Q10 (second column). Input data for calculation from SurveyMonkey response data. SurveyMonkey export settings: all response data, XLS, original view, columns expanded, cells actual answer text. For comparison, the third column shows the weighted averages for all respondents, values taken from SurveyMonkey result data table (see Table 4.3).

### 4.5.3 Structure is Control

The results for question Q14 support the pattern regarding the reliance on a *customised structure* when managing files and folders on a PC desktop device. Table 4.10 shows the responses for question Q14. Most, to be precise 12 participants use a combination of a *customised structure* and *search* when looking for their files. However, at second place are nine respondents who mainly use a *customised structure*. In both cases they utilise a *customised structure* while none of them mainly relies on *search* features to find their files. Again, I argue that these results fit the pattern I called structure is control very well. Meaning, individuals seem to prefer a customised structure when the situation allows it. Features like *search* or utilising *no structure* are used more as a last way out.

**Table 4.10**

*Q14: When you are Looking for Your Relevant Files (for Example, Photos, Spreadsheets, Mails, Messages) on Your PC how do you Find Them?*

| ANSWER CHOICES | RESPONSES | |
|---|---|---|
| I mainly use a custom folder/file structure (e.g. folders by subject->date, mail by categories) | 40.91% | 9 |
| I mainly rely on search tools (e.g. Spotlight, Windows Search) | 0.00% | 0 |
| I use a combination of both (structure and search) | 54.55% | 12 |
| I mainly just browse folder by folder until I find what I need (no structure, no search) | 4.55% | 1 |
| TOTAL | | 22 |

*Note*. SurveyMonkey result data table. Exported survey data, SurveyMonkey export settings: summary data, PPT, original view, charts and data tables. Grammatical and spelling errors (if any) in the questions (see figure title) were corrected for this view. Also, the readability was improved by replacing “*e.g.*” with “*for example,*”.

### 4.5.4 Faith in Virtual Reality

The results for question Q20 and Q21 support the pattern I called **faith in virtual reality**. Table 4.11 shows the responses for question Q20. Almost all of the respondents, to be precise 95.45% stated that they do not use virtual reality or augmented reality appliances regularly (at least once a week). At the same time 68.18% of them responded that they can imagine that some kind of virtual or augmented reality device will replace all of our existing devices in the coming years (see Table 4.12).

It is interesting to see that various applications of technologies like virtual reality are being researched on. For instance, Belter and Lukosch (2022) created and evaluated the use of an immersive virtual reality math game. At the same time, I cannot find references critically evaluating the need for technologies like virtual reality in the first place. Consequently, albeit not in scope of my research I ask where this interesting link is coming from, for example, are these devices not being used today because of a lack of technology advancements, are there other reasons, is this link to the future just a form of bias? Hence, how valid is the criticism of individuals like Ken Kutaragi who stated "'Headsets would isolate you from the real world, and I can't agree with that,'" (Mochizuki & Furukawa, 2022, para. 6)?

**Table 4.11**

*Q20: Do you Regularly (at Least Once a Week) use Virtual Reality and/or Augmented Reality Software/Products?*

| ANSWER CHOICES | RESPONSES | |
|---|---|---|
| Yes | 4.55% | 1 |
| No | 95.45% | 21 |
| TOTAL | | 22 |

*Note*. SurveyMonkey result data table. Exported survey data, SurveyMonkey export settings: summary data, PPT, original view, charts and data tables. Grammatical and spelling errors (if any) in the questions (see figure title) were corrected for this view. Also, the readability was improved by replacing "*e.g.*" with "*for example*,".

**Table 4.12**

*Q21: Can you Imagine a Near Future (max. 10 Years From now) Were Some Kind of Virtual and/or Augmented Reality Device (for Example, AR Glasses) Will Replace all of our Current Devices (Smart Phones, Laptops, Desktops etc.) Completely (in all Areas)?*

| ANSWER CHOICES | RESPONSES | |
|---|---|---|
| Yes | 68.18% | 15 |
| No | 31.82% | 7 |
| TOTAL | | 22 |

*Note*. SurveyMonkey result data table. Exported survey data, SurveyMonkey export settings: summary data, PPT, original view, charts and data tables. Grammatical and spelling errors (if any) in the questions (see figure title) were corrected for this view. Also, the readability was improved by replacing "*e.g.*" with "*for example*,".

### 4.5.5 Office, Communication and Organisation

The results for question Q6 support in part the pattern. Table 4.13 visualises the responses for question Q6. Worth noting that, because in the phase 1 interviews *internet* usage was only mentioned once (one coding reference) in a very general way, to quote Professional00 "*Yes, sure. I also look into the internet, sure.*", I decided to fan out *internet* use to multiple options: "*Social media*"; "*Media consumption*"; "*Online Shopping*"; "*Offline Shopping*". When one just reviews everything but the internet related options the top three tasks are: communication (average responses for work and private $(21 + 14) / 2 = 17.50$), *office content creation* with 17 responses, organisation (average responses for work and private $(10 + 10) / 2 = 10.00$). Consequently, the result is very similar to phase 1 with the only difference being that this time communication was the most prominent choice instead of *office content creation*. However, while in the phase 1 interviews *internet* usage was barely mentioned it was a very prominent choice in the questionnaire. Hence, all of the following three options had more than 20 responses each: "*Social media*"; "*Media consumption*"; "*Online Shopping*". Consequently, all of the evaluated areas are valid candidates for potential case studies except for *project management* and *creative content creation* given their consistently (in phase 1 and phase 2) and comparatively low usage rates. To conclude, tasks are mainly **internet, office, communication and organisation** related.

**Table 4.13**

*Q6: For Which of the Following Regular (More Than Once per Week) Tasks (Work and/or Home Usage Related) do you use IT Devices (for Example, a Smart Phone, a Laptop etc.)?*

| ANSWER CHOICES | RESPONSES | |
|---|---|---|
| Work organization (e.g. appointments/office meetings) | 45.45% | 10 |
| Private organization (e.g. appointments) | 45.45% | 10 |
| Office content creation (e.g. text documents, spreadsheets) | 77.27% | 17 |
| Creative content creation (e.g. video/photo recording and editing) | 31.82% | 7 |
| Project management (e.g. time management, mind maps) | 27.27% | 6 |
| Private communication (e.g. messaging, e-mail) | 95.45% | 21 |
| Work communication (e.g. e-mail) | 63.64% | 14 |
| Social media (e.g. social networks like Facebook) | 95.45% | 21 |
| Media consumption (e.g. music, video streaming) | 100.00% | 22 |
| Online Shopping (e.g. electronics, office supplies, clothes) | 95.45% | 21 |
| Offline Shopping (e.g. for payment like Apple Pay) | 59.09% | 13 |
| Total Respondents: 22 | | |

*Note*. SurveyMonkey result data table. Exported survey data, SurveyMonkey export settings: summary data, PPT, original view, charts and data tables. Grammatical and spelling errors (if any) in the questions (see figure title) were corrected for this view. Also, the readability was improved by replacing "*e.g.*" with "*for example*,".

### 4.5.6 Usable Software with Perceived Value

The results for question Q13 support in part the pattern. Table 4.14 visualises the responses for question Q13. Worth noting that, the offered criteria addressed the following components of user experience (in order, top to bottom): *usefulness*, *usability*, *emotional impact*, *emotional impact*, *usefulness*, *usefulness*, *usability*. Calculating the averages of confirmation for the three components results in:

- Average for usefulness: $(17 + 17 + 12) / 3 = 15.33$
- Average for usability: $(18 + 18) / 2 = 18.00$

- Average for emotional impact: $(17 + 17)\ /\ 2 = 17.00$

When ranked usability is the most important component followed by emotional impact and usefulness. Hence, all of the criteria had a relatively high agreement rate with the exception of technical compatibility. Still, the importance of the emotional impact is a surprise. Meaning, that for example, the colour scheme of an application consciously informed their decision. Also, the results for question Q18 (see Table 4.15) support this assessment. A surprising 27.27% (six respondents) stated that they value a pleasant design more than anything else when managing their files and folders. Hence, these six respondents value a pleasant design more than finding their data in the most efficient way. Consequently, individuals choose **beautiful and usable software with perceived value**.

**Table 4.14**

*Q13: Were the Following Criteria Relevant for Your Decision Regarding Your Favourite Application?*

| | YES | NO | TOTAL |
|---|---|---|---|
| The application offered features you needed as a user which were not available with competing solutions | 77.27%<br>17 | 22.73%<br>5 | 22 |
| You were able to finish your task faster than with competing solutions | 81.82%<br>18 | 18.18%<br>4 | 22 |
| You liked the design (e.g. colour scheme) of the application more than others | 77.27%<br>17 | 22.73%<br>5 | 22 |
| You trust the brand/manufacturer of the application more than others | 77.27%<br>17 | 22.73%<br>5 | 22 |
| Social compatibility (e.g. colleagues and/or friends used it) | 77.27%<br>17 | 22.73%<br>5 | 22 |
| Technical compatibility (e.g. was the only available option for your platform(s)/device(s)) | 54.55%<br>12 | 45.45%<br>10 | 22 |
| The application offered superior integration with your platform(s)/device(s) (e.g. performance, additional features) | 81.82%<br>18 | 18.18%<br>4 | 22 |

*Note*. SurveyMonkey result data table. Exported survey data, SurveyMonkey export settings: summary data, PPT, original view, charts and data tables. Grammatical and spelling errors (if any) in the questions (see figure title) were corrected for this view. Also, the readability was improved by replacing "*e.g.*" with "*for example*,".

**Table 4.15**

*Q18: What do you Value the Most in an Application to Navigate Through Your Data (for Example, Photos, Spreadsheets) on Your IT Device (Mobile or PC)?*

| ANSWER CHOICES | RESPONSES | |
|---|---|---|
| I want to find the relevant information as fast as possible | 72.73% | 16 |
| I want a pleasant design (e.g. colour scheme, integration) when navigating | 27.27% | 6 |
| Other, please specify | 0.00% | 0 |
| TOTAL | | 22 |

*Note*. SurveyMonkey result data table. Exported survey data, SurveyMonkey export settings: summary data, PPT, original view, charts and data tables. Grammatical and spelling errors (if any) in the questions (see figure title) were corrected for this view. Also, the readability was improved by replacing "*e.g.*" with "*for example*,".

### 4.5.7 Work Practice Ecology is Text Based and 2D

The results for question Q7 and Q8 support in part the pattern. Table 4.16 shows the results for question Q7. Still, the interfaces of regularly used applications are mostly in 2D. Table 4.17 visualises the results for question Q8. Worth noting that, for this question I decided to skip *relational data* as I wanted to understand the participants ecology in a more general context working with computing devices (digital) or not (non-digital). Also, I included video and audio as potentially relevant options. Interestingly, the most used data type chosen is video followed by text and audio. To conclude, the **work practice ecology is 2D videos and text**.

**Table 4.16**

*Q7: How are Most of the Interfaces of Your Regularly Used (Work and/or Home Usage) Applications Designed?*

| ANSWER CHOICES | RESPONSES | |
|---|---|---|
| As 2D interfaces (e.g. lists, buttons, grids, tabs) | 100.00% | 22 |
| As 3D interfaces (e.g. 3d carousel, VR environment) | 0.00% | 0 |
| TOTAL | | 22 |

*Note*. SurveyMonkey result data table. Exported survey data, SurveyMonkey export settings: summary data, PPT, original view, charts and data tables. Grammatical and spelling errors (if any) in the questions (see figure title) were corrected for this view. Also, the readability was improved by replacing "*e.g.*" with "*for example*,".

**Table 4.17**

*Q8: Which Type of Data do you use the Most (in Regards to Time Spent) – Digital and/or Non-Digital at Work and/or Home?*

| ANSWER CHOICES | RESPONSES | |
|---|---|---|
| Textual data (e.g. books, magazines, web surfing) | 36.36% | 8 |
| Video (e.g. movies, streaming, games) | 54.55% | 12 |
| Pictures (e.g. art, photos, still shots) | 0.00% | 0 |
| Audio Recordings (e.g. music, personal recordings) | 9.09% | 2 |
| TOTAL | | 22 |

*Note*. SurveyMonkey result data table. Exported survey data, SurveyMonkey export settings: summary data, PPT, original view, charts and data tables. Grammatical and spelling errors (if any) in the questions (see figure title) were corrected for this view. Also, the readability was improved by replacing "*e.g.*" with "*for example*,".

## 4.6 Phase 3 Final Interviews

Because of the rather abstract topic I decided to carry out another phase which consisted of two semi-structured interviews to gather more task scenarios and requirements in the file and folder management context. After a personal introduction, I asked the participants the following guiding questions:

1. How do you store files on your PC desktop?
2. How do you find files on your PC desktop?
3. How do you store files on your mobile device?
4. How do you find files on your mobile device?
5. How do you store files on shared (network) drives?
6. How do you find files on share (network) drives?

For transparency, I also asked Student01 what they are currently missing when navigating their files and folders as a follow up question because they described in depth scenarios for all six questions.

After these questions I presented a first sketch (see Figure 4.9) as a potential prototype of my novel method for PC desktop use. I clarified open questions with the participants concerning the sketch and asked for their initial feedback. Worth noting that, the design of this sketch was informed by what I have learned in phase 1 and phase 2. The design is described in Section 5.3.

**Symbols**

Symbol for the end-user position in 3D space - note user/player views artifact from first person perspective - user character movement with arrow keys (desktop pc keyboard) and (pc) mouse input (x-/y-axis) to change view direction

A folder links to another "space" and other content e.g. more files/links

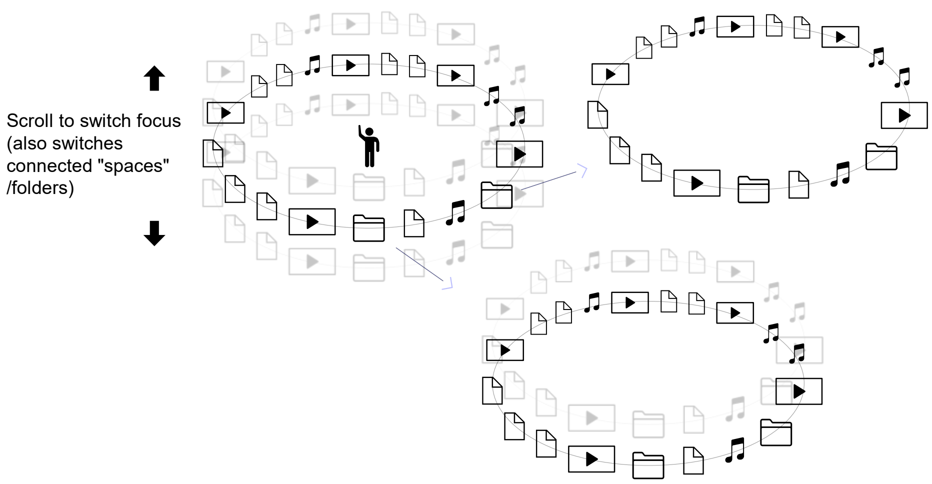
A video file - on user focus (e.g. more than 3sec.) starts autoplay

A document - shows preview content (if available) e.g. first text paragraph

A music file - on user focus (e.g. more than 3sec.) starts autoplay

A symbolic link to another "space" (folder link) - just for illustration

**Design - 2D Abstraction - Navigate Current State**




**Design - 2D Abstraction - Navigate History and "Things of interest"**

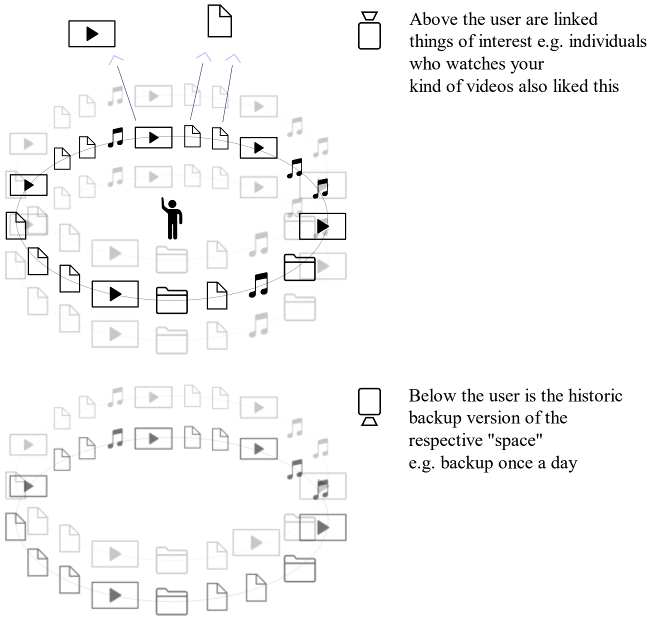



Figure 4.9: Draft Design of my novel Method as a Sketch to Gather Feedback

*Note*. English version (0.3) sketch of my method (PC desktop scope). German version (translated by myself) presented to interview participants. Document title removed.

Worth noting that, I reused my categories and one code from phase 1 as a form of Provisional Coding. I reused the code *mobile devices* just as a kind of filter criterion for later analysis:

- file and folder management context (category)
    - tasks (category)
    - user stories (category)
        - emotional impact (category)
        - usability (category)
        - usefulness (category)
- general context (category)
    - *mobile devices* (code)

Table B3 lists my Descriptive Codes after first cycle coding. Worth noting that, these codes were manually created and reviewed once by myself.

**4.6.1 Mobile Device a Cluttered Companion**

The results support a feeling of companionship towards mobile devices. To quote Student01 "*I have exactly the same access to the cloud via my mobile device and yes I would also say I use it almost every day, yes.*".

In phase 1 I argued that in the context of file and folder management tasks participants rely more on *search* features and do not utilise a custom file and folder structure (*no structure*) when using their mobile phone (see Section 4.4.1). However, in phase 2 the results did not fully support this claim (see Section 4.5.1). Hence, a lot (seven) of the respondents stated that they mainly utilise a *customised structure* to look for their files and folders when using *mobile devices* (see Table 4.2).

Similarly, in phase 3 the results also do not support the initial indication from phase 1. Figure 4.10 visualises a NVivo coding matrix to show the count of coding references where I utilised *mobile devices* and other codes (intersections). The results indicate that in the context of file and folder management tasks participants prefer a *customised structure* and use *search* as a kind of last resort option on *mobile devices*. To quote Professional02 "*And if I cannot find something then I just try to search for it.*". When I asked Student01 if they also try to use their central cloud solution on their mobile device to store or sort their files they responded "*Exactly, so (basically I would say?) I do file storage and file management significantly more with the laptop than the mobile device.*". Consequently, I argue that tasks on mobile devices related to file and folder management also fit the **structure is control** pattern. Indeed, the usage may differ because of certain restrictions, for example, a lack of available screen space. When I asked Student01 if they use the search function more on their mobile device than on the desktop because of the display size they said "*Yes, I have never really paid attention to that. But you could be right, yes.*". If possible, participants aim for a centralised approach, *device independence*, their *customised structure*. When I asked Professional02 to describe how they find their files on their mobile device they described their approach for file storage using a *customised structure*, just relying on *search* when they cannot find what they are looking for and closed with the sentence "*Exactly the same as on Windows.*". To conclude, regarding file and folder management tasks on *mobile devices* the only consistent result across all three phases was the preference for a *customised structure*.

Consequently, I change my initial pattern **mobile device a cluttered companion** to **mobile device a restricted companion** to highlight that based on my results in the context of file and folder management individuals prefer a centralised, device independent approach with their own *customised structure* across devices. However, some usage scenarios do not support this approach, for example, because of the available screen space or other restrictions.

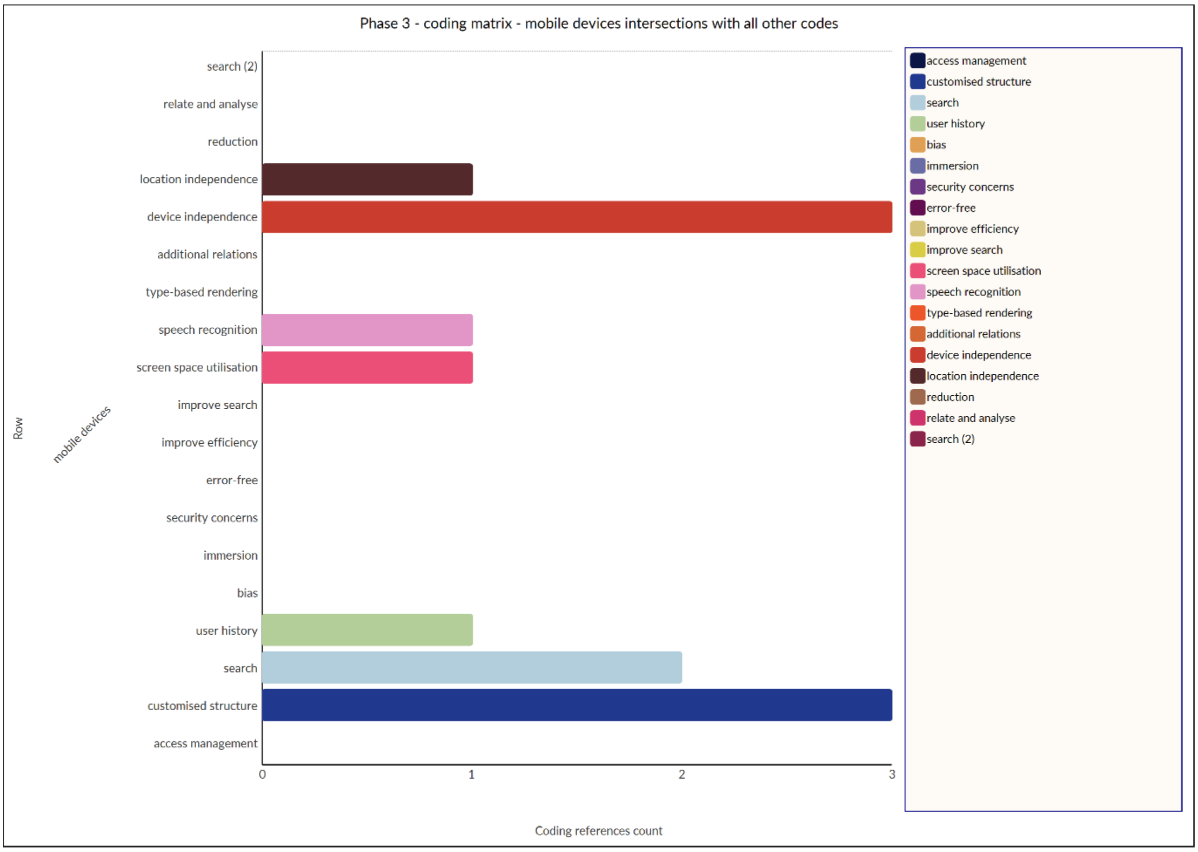



Figure 4.10: Phase 3 – Coding Matrix – Mobile Devices Intersections With all Other Codes

*Note*. NVivo coding matrix visualising which codes were used for sections also coded to *mobile devices*. For transparency, the item in the chart labelled as "search (2)" denotes my code at "file and folder management context/user stories/usefulness/search". The item labelled as "search" denotes my code at "file and folder management context/tasks/search".

### 4.6.2 Be Here Now

The codes for user stories are very similar to the ones I have identified in phase 1 (see Section 4.4). Figure 4.11 shows a NVivo hierarchy chart (Treemap) where the first cycle codes in file and folder management context/user stories are compared by their number of coding references. The most coding references has *location independence* with nine references next is *device independence* and *relate and analyse* with seven references each followed by *improve efficiency*, *additional relations* and *security concerns* with five references each. For *location independence* I identified similar statements as in phase 1 regarding working with others or the reliance on remote storage. For example, to quote Student01 "*Yes, there have been problems before, of course. Because of course it's somehow often a structure which you have not specified yourself.*". Similar to phase 1, *location independence* viewed as a topic for a need to reduce the complexity of our digital life, centrally controlling data. Moreover, *device independence* a topic about a need for coherency, to quote Student01: "*But I also use the device independence is also very important to me and exactly.*". But even more importantly *device independence* describing a coherent thought process, Professional02 said "*I also have a personal drive, of course and that's where I put my things just like I do privately.*".

In the presented sketch (see Figure 4.9) I described the section "*Things of interest*" as an attempt to support activities in my design or as I called this topic *relate and analyse*. Both interviewees supported this feature to visualise related content above the viewer, to quote Professional02 "*I definitely see a very, very large added value in it.*".

I noticed that with my initial sketch I already violated the design metaphor. Hence, to quote Student01 regarding *improve efficiency* "*Yes, although there is also a bit of the question, whether I now have to scroll down in a 2D list or have to turn with the little man by 180 degrees to also see the files behind me. Still equal for me.*". Consequently, I had to acknowledge that for my next design attempts I need to address this crucial usability related concern.

Similarly, I presented a "*History*" (see Figure 4.9) feature in my sketch to support the potential need for *additional relations*, additional tasks. Again, both interviewees supported this feature, to quote Student01 "*Exactly, that's definitely a very, very cool thing which definitely also has an added value (inc.) compared to the 2D lists. That you have so to say these versions right underneath and above these things of interest.*".

Interestingly, *security concerns* describing concerns regarding potential data loss or data privacy was a new topic in phase 3. It is noteworthy that the example of related video content I chose for the section Things of interest (see Figure 4.9) resulted in Professional02 expressing their concerns for data privacy. To quote Professional02 "*Exactly, I find it almost scary what is recommended to me on YouTube. Things that really interest me but there I also know ok they also collect data and create profiles about me and know what could really interest me now and that would now be almost similar in context, I think.*". Only after clarification to imagine a local only engine and providing another work-related example their opinion changed (see above).

Overall, the results supported the **be here now** metaphor. For instance, in general the initial design attempt (see Figure 4.9) was received very well by the participants. Also, I argue that even a code like *security concerns* as a need for control fits the metaphor quite nicely.

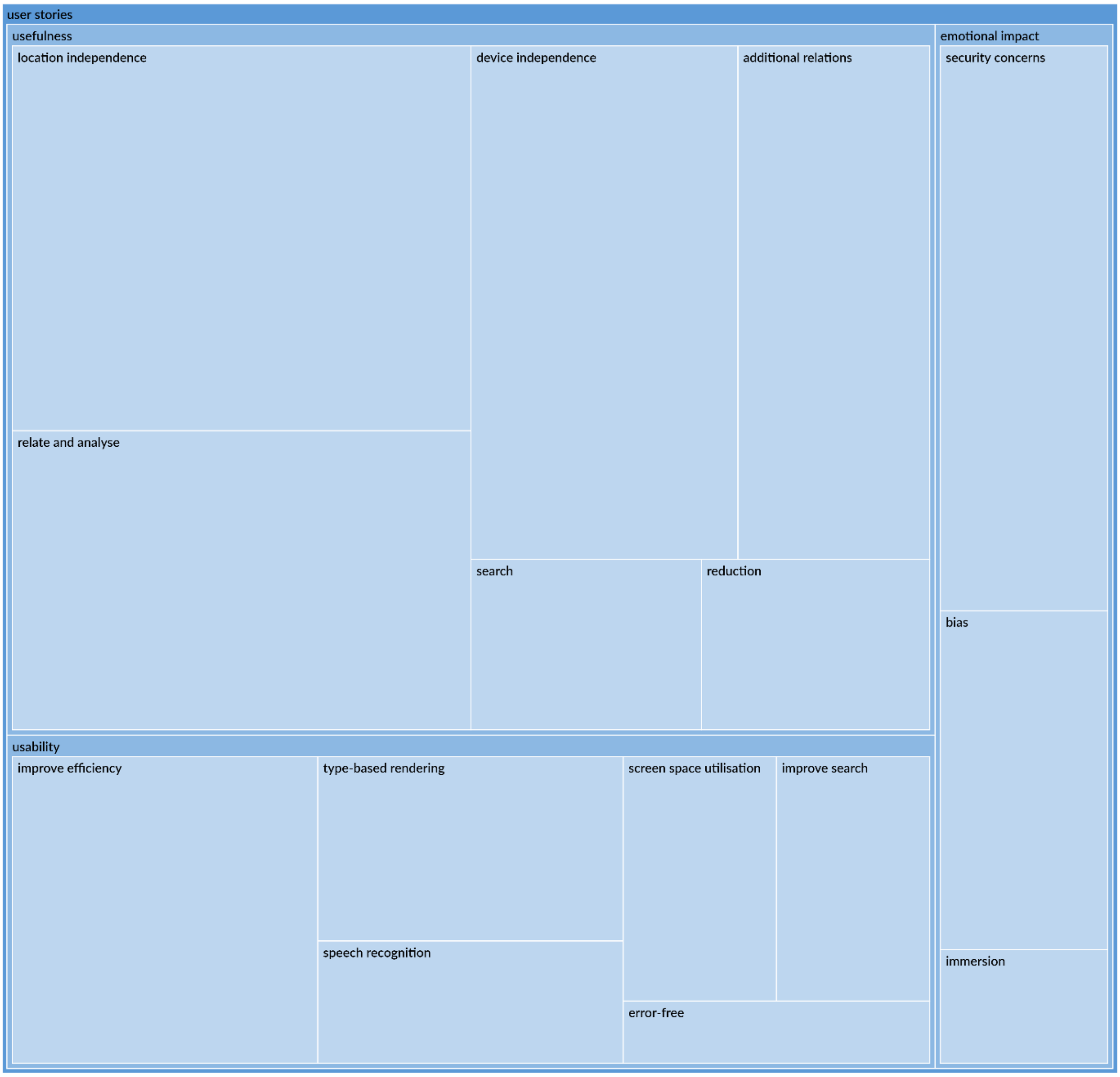



Figure 4.11: Phase 3 – Treemap – File and Folder Management Context/User Stories – Number of Coding References

*Note*. NVivo hierarchy chart (Treemap) visualising (compared by) the number of coding references for file and folder management context/user stories.

### 4.6.3 Structure is Control

The results support the pattern I called **structure is control**. Figure 4.12 shows a NVivo hierarchy chart (Treemap) where the first cycle codes in file and folder management context/tasks are compared by their number of coding references. The chart shows that *customised structure* has the most coding references with 12 in total. At second place is *search* with five references followed by *user history* with three and *access management* with one reference. Hence, the results support a preference for a *customised structure* when the situation allows for it. Other tasks are utilised as a supplementary approach, for example, when a file was put in the wrong place. When asked how they find a file on their PC desktop where they cannot remember its location Professional02 answered "*I try to click through everything first and see if I might not have put it somewhere else and try to remember. … but just to make it go fast I actually use the Windows search because I actually already know what the name of the file is and then I just have to type it in and then that will come so to say. If it does not come up then I will just have to click through again.*".

Noteworthy, is the code I named *access management*. Actually, I interpret this code as a kind of sub code for *customised structure*. Albeit, just a single coding reference I argue that this task may become more important in the near future. Hence, the work practice of file and folder management is also in part working with others.

Similarly, *user history* was mentioned in the context of searching for files. When asked how they find their files on their PC desktop also for older data Student01 explained "*And what also happens is that one so to say last opened files are simply opened (again?) via quick access.*". In contrast Professional02 mentioned that they utilise a user history feature on their mobile phone to maintain and utilise their own *customised structure*, to quote "*And I did not know, that one, down there there's another history. … but on the right side is another icon and there it says search and I did not know that (I?) can create another folder there where I then can store my files explicitly.*".


Figure 4.12: Phase 3 – Treemap – File and Folder Management Context/Tasks – Number of Coding References

*Note*. NVivo hierarchy chart (Treemap) visualising (compared by) the number of coding references for file and folder management context/tasks.

### 4.7 Final Modelling and User Stories

At this stage I utilised my patterns and codes to update my models and inform my user stories.

I can identify the following final roles for my user work role model in the context of file and folder management:

- System User, who interacts with a mobile or PC desktop device through 2D output (monitor) and input devices (mouse, keyboard, touch input) to store and find items of interest (documents, pictures) for viewing or editing. They interact mainly with videos and text-based content using 2D interfaces (**work practice ecology is 2D videos and text**). If possible, they construct and apply their personal *customised structure* across their devices to store and find files (**structure is control**, **mobile device a restricted companion**). They work with versioning and sharing files (*additional relations, access management*). Moreover, they utilise remote storage locations like cloud storage (*location independence*). Also, they carry out activities when navigating their data, for example, find two files for data analysis (*relate and analyse*). In general, their tasks are mainly related to the areas of **internet, office, communication and organisation**.
    - Student, a subrole who is a university student.
    - Professional, a subrole who has an occupation.
- Developer, who develops or maintains the tools to find and edit documents, for example, Windows Explorer.
- (Temporary,) Third-party User, who has access and uses shared files and folders, for example, files shared by the System User.

My final flow model to visualise the flow of information is shown in Figure 4.13. Based on my research results I conclude that the **work practice ecology is 2D videos and text** therefore I inserted a document symbol for videos and text in the model. Similarly, I highlighted that graphical user interfaces are different, not coherent (*device independence*) and in 2D. As another example, I visualised the pattern of **structure is control** by highlighting that a System User aims to create a *customised structure,* if possible, for a centralised approach (*device independence*). However, I also highlighted as a challenge that this is not always possible, for example, because of a lack of available screen space on *mobile* *devices*. Also, I highlighted that user activities are not supported (*relate and analyse*) and that tasks are mainly related to **internet, office, communication and organisation**. Moreover, I inserted a section for remote storage (*location independence*) and working with others describing the challenges to manage *additional relations* (for example, versions) and increased complexity because of *access management*.

Overall, the model visualises the complexity of file and folder management. In comparison to my initial flow model (see Figure 4.1) it shows a potential for compression, meaning, questioning what tasks can be optimised or centralised, for example, by optimising the visualisation and interaction.

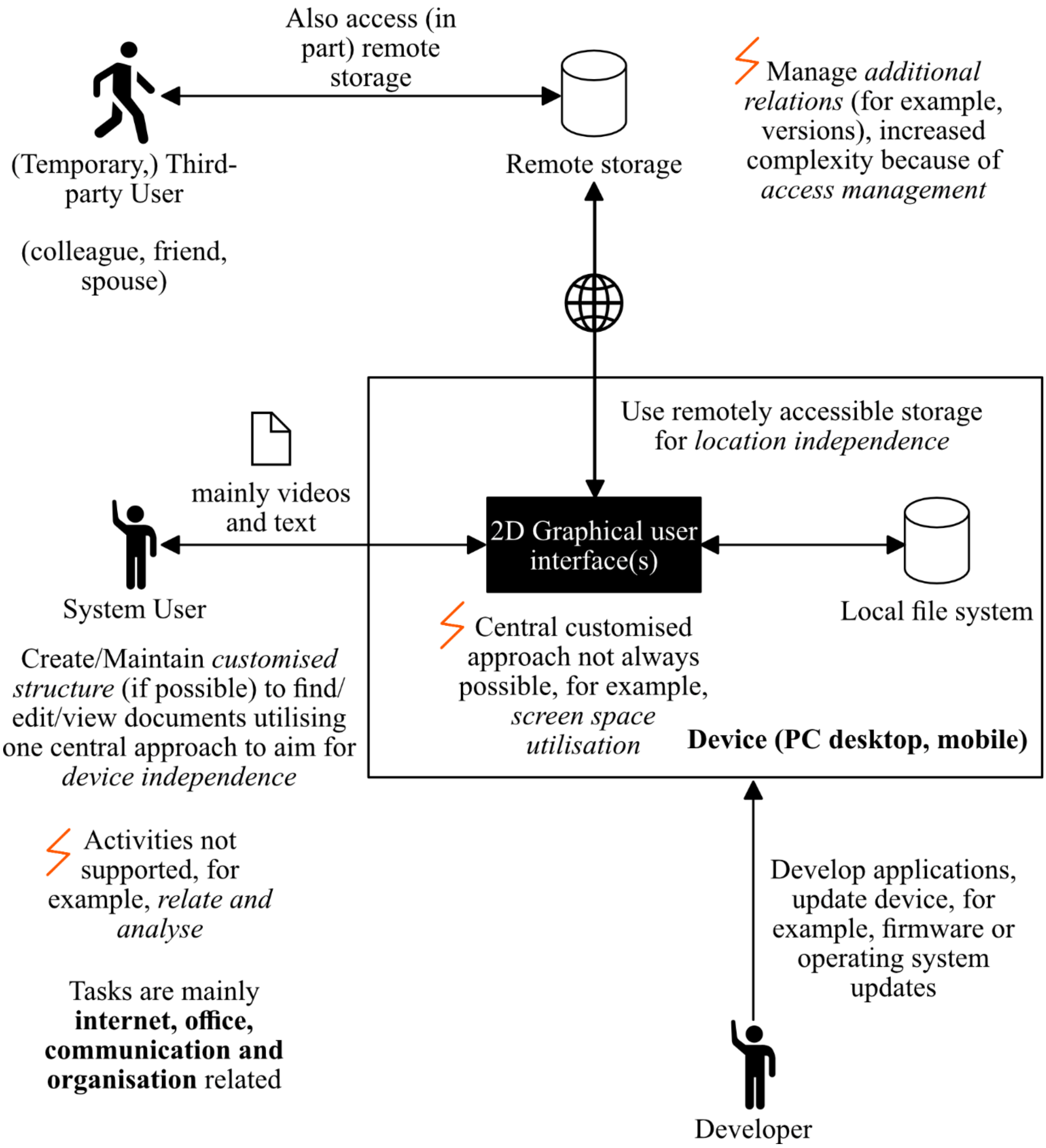



Figure 4.13: Final Flow Model

*Note*. Final flow model visualising file and folder navigation on PC desktop or mobile devices. Bolts in orange colour signal challenges.

Table B4 lists my user stories and acceptance criteria derived from my codes and patterns to describe the requirements for my novel method. I applied a similar heuristic as described by Patton and Economy (2014) when prioritising my user stories into good, better and best. Worth noting that, for the scope of my thesis I focused (design, evaluation stages) on user stories which are just “GOOD ENOUGH to get things working” (Patton & Economy, 2014, p. 286).

Derived from my user stories (see Table B4) I created various task sequence models (see Figure 4.14, Figure 4.15, Figure 4.16, Figure 4.17, Figure 4.18) to detail the sub tasks associated to using a *customised structure* in the context of my novel method. Worth noting that, at this stage going forward I only focus on user stories with a priority rating set to “*good*”. The models are used in later stages, for example, to restrict prototyping for evaluation.

Task name: Create a new node (folder, file video/text)
Task goal: Node created
Task trigger: System User need to create a new node (for example, a new folder)

Task (general) barriers: Are the symbols recognisable ( *bias*)?; Feel immersed during navigation (*immersion*)?; Can I carry out precise and fast navigation ( *error-free*, *improve efficiency*)?; Can I identify the data I am looking for ( *screen space utilisation*, *type-based rendering*)?; Is navigation coherent (*device independence*)?; Where is the target location (*location independence*)?
Response to (general) barriers: Have to be addressed by the design

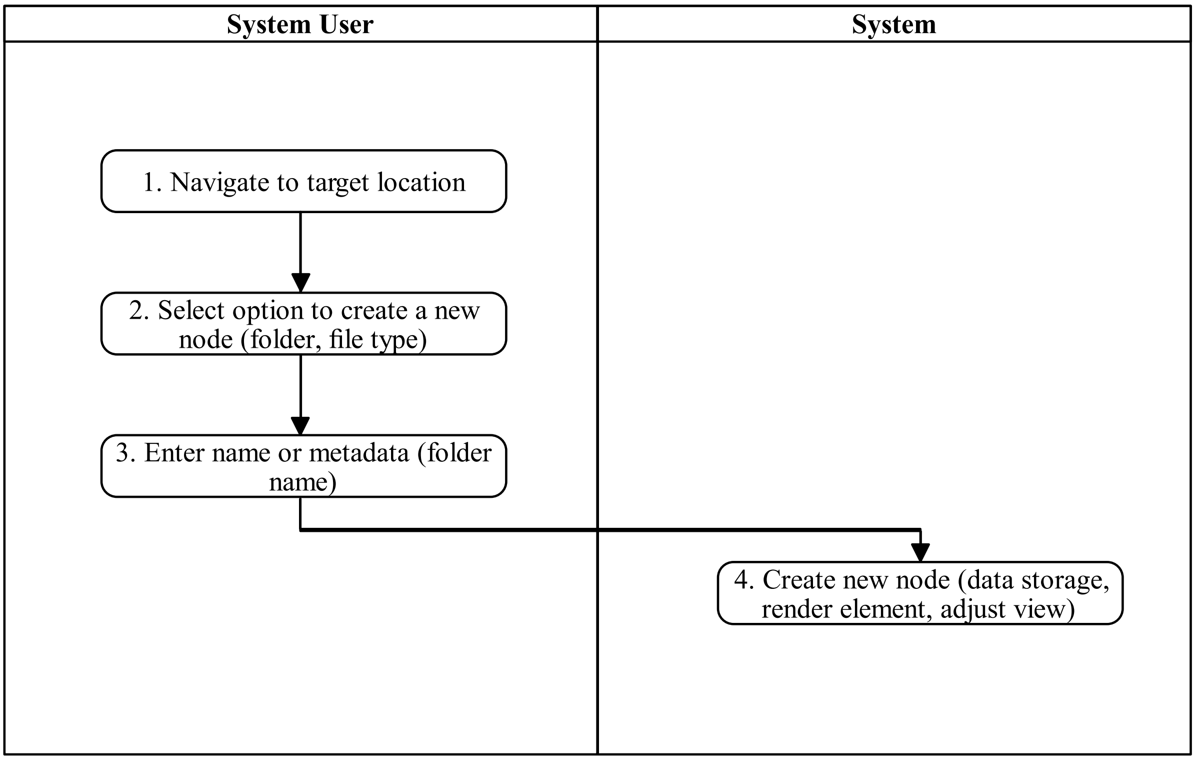



Figure 4.14: Task Sequence Model – Create a new Node

*Note*. Task sequence model visualising the creation of a new node (file or folder) derived from my user stories (see Table B4). Bolts in orange colour signal challenges. Potential other needs are highlighted with an orange N symbol.

Task name: Delete an existing node (folder , file like video/text) and (if available) its children
Task goal: Node and (if available) its child nodes are deleted
Task trigger: System User need to delete a node (for example, data not required any more)

Task (general) barriers: Are the symbols recognisable ( *bias*)?; Feel immersed during navigation (*immersion*)?; Can I carry out precise and fast navigation ( *error-free*, *improve efficiency*)?; Can I identify the data I am looking for ( *screen space utilisation*, *type-based rendering*)?; Is navigation coherent (*device independence*)?; Where is the target location (*location independence*)?
Response to (general) barriers: Have to be addressed by the design

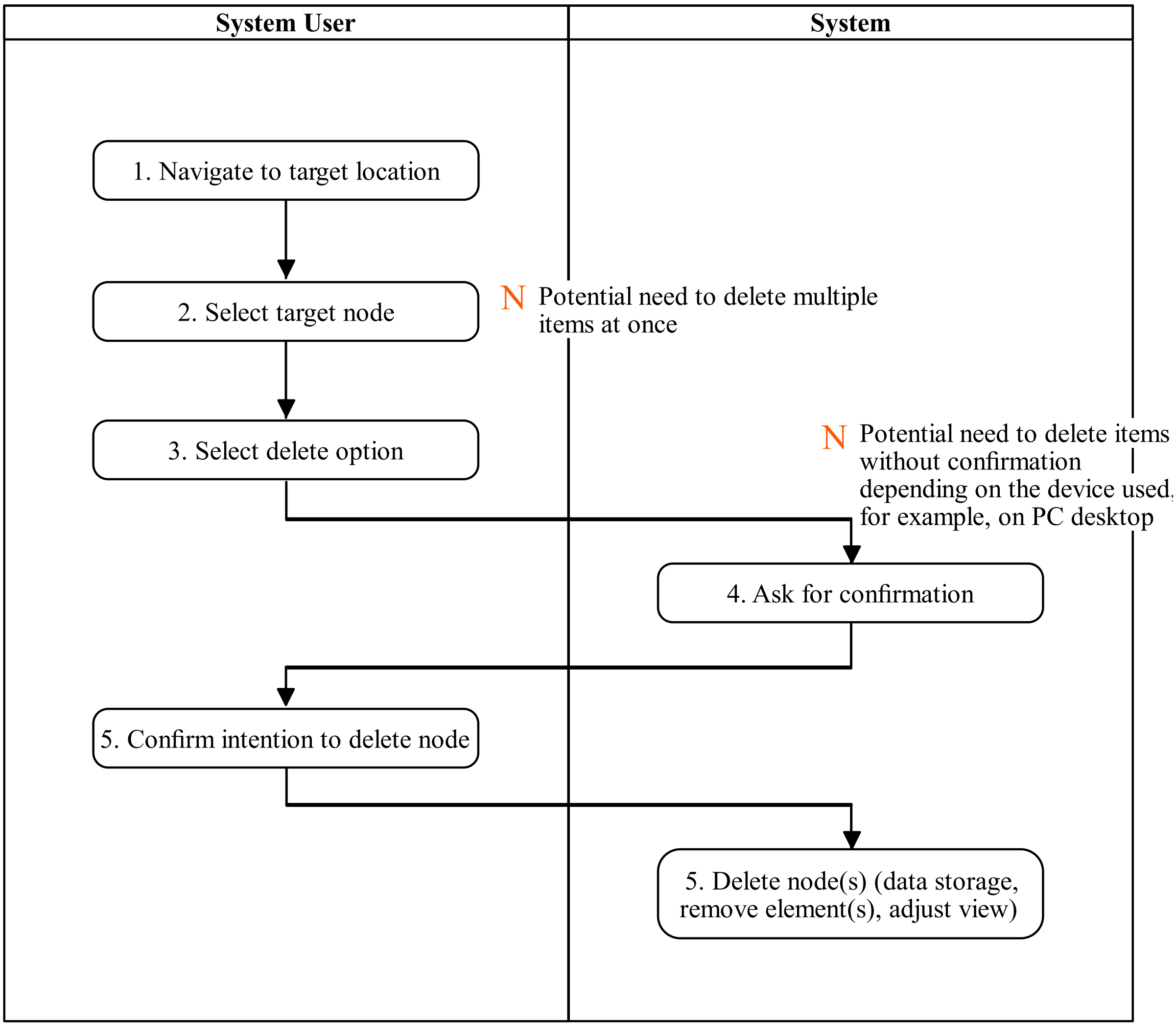



Figure 4.15: Task Sequence Model – Delete a Node and its Children

*Note*. Task sequence model visualising the deletion of a node (file or folder) derived from my user stories (see Table B4). Bolts in orange colour signal challenges. Potential other needs are highlighted with an orange N symbol.

Task name: Edit text label for existing node (folder , file like video/text)
Task goal: Text label changed
Task trigger: System User need to rename a node (for example, a file)

Task (general) barriers: Are the symbols recognisable ( *bias*)?; Feel immersed during navigation (*immersion*)?; Can I carry out precise and fast navigation (*error-free*, *improve efficiency*)?; Can I identify the data I am looking for (*screen space utilisation*, *type-based rendering*)?; Is navigation coherent (*device independence*)?; Where is the target location (*location independence*)?
Response to (general) barriers: Have to be addressed by the design

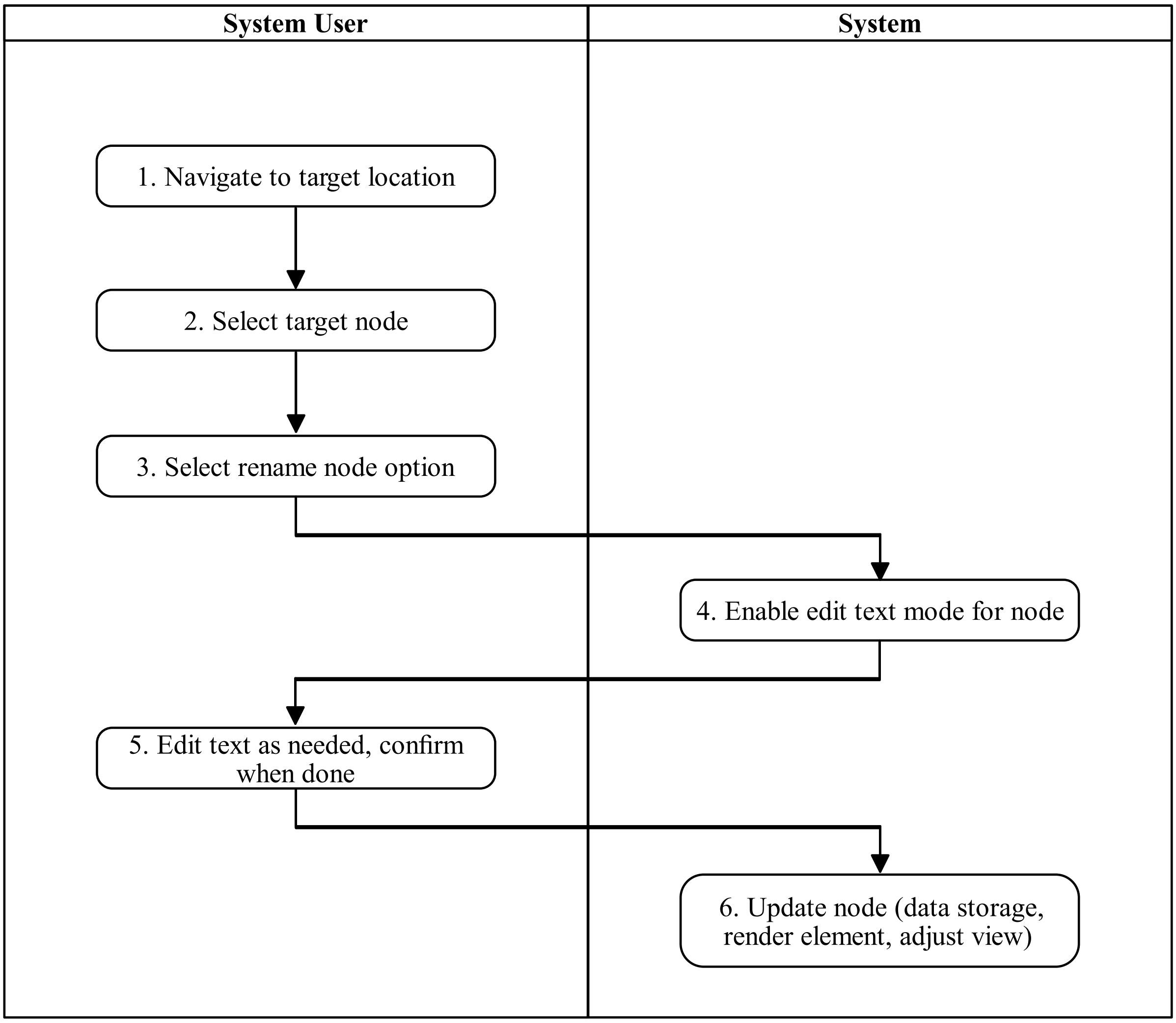



Figure 4.16: Task Sequence Model – Edit a Node

*Note*. Task sequence model visualising the change of a node text label (file or folder). The model is derived from my user stories (see Table B4). Bolts in orange colour signal challenges. Potential other needs are highlighted with an orange N symbol.

Task name: Move (or cut, copy and paste) an existing node (folder, file like video/text) to another position
Task goal: Node moved to new position
Task trigger: System User need to move a node (for example, misplaced a file)

Task (general) barriers: Are the symbols recognisable (*bias*)?; Feel immersed during navigation (*immersion*)?; Can I carry out precise and fast navigation (*error-free*, *improve efficiency*)?; Can I identify the data I am looking for (*screen space utilisation*, *type-based rendering*)?; Is navigation coherent (*device independence*)?; Where is the target location (*location independence*)?
Response to (general) barriers: Have to be addressed by the design

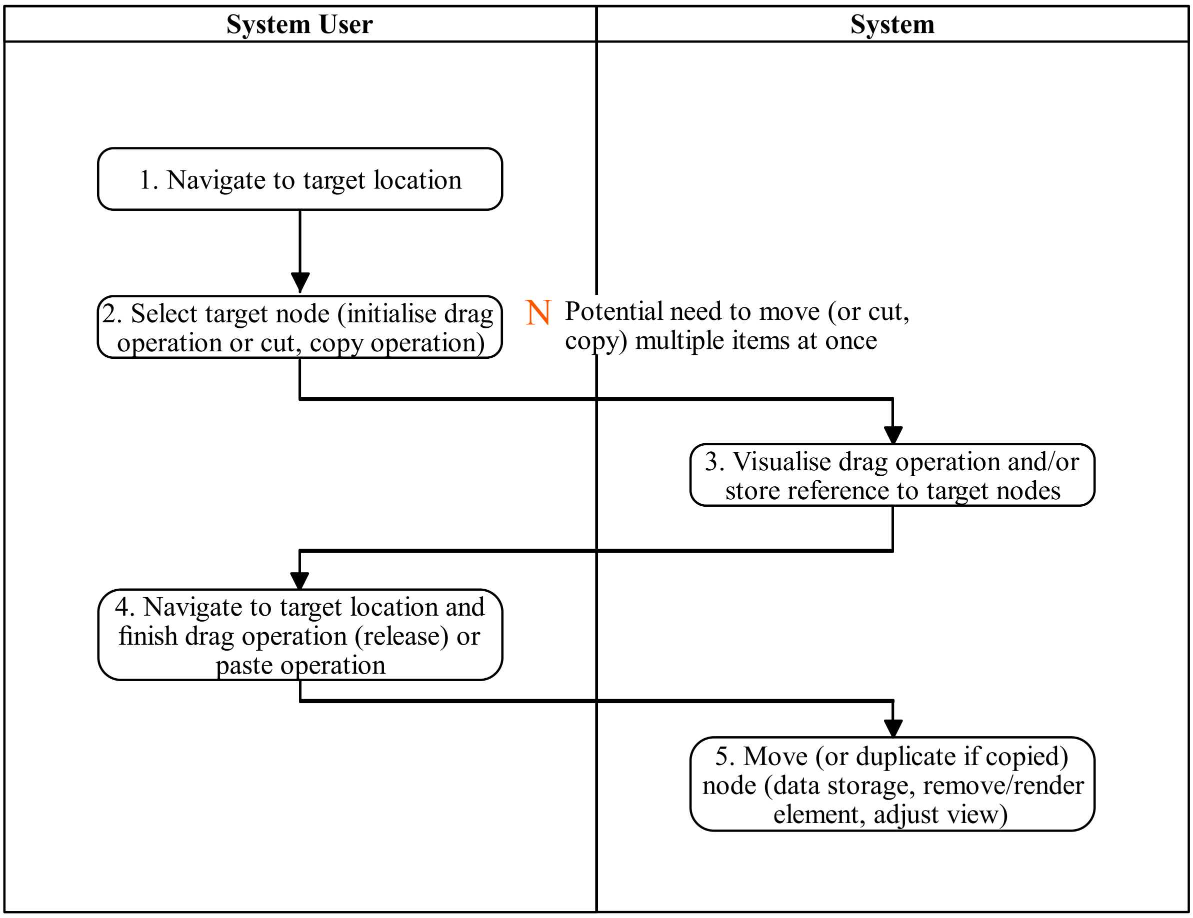



Figure 4.17: Task Sequence Model – Move a Node

*Note*. Task sequence model visualising moving a node (file or folder) to another position in the hierarchy. The model is derived from my user stories (see Table B4). Bolts in orange colour signal challenges. Potential other needs are highlighted with an orange N symbol.

Task name: Select a node of interest (folder, file video/text)
Task goal: Node selected
Task trigger: System User need to find/identify a node of interest (for example, a project file)

Task (general) barriers: Are the symbols recognisable ( *bias*)?; Feel immersed during navigation (*immersion*)?; Can I carry out precise and fast navigation (*error-free*, *improve efficiency*)?; Can I identify the data I am looking for (*screen space utilisation*, *type-based rendering*)?; Is navigation coherent (*device independence*)?; Where is the target location (*location independence*)?
Response to (general) barriers: Have to be addressed by the design

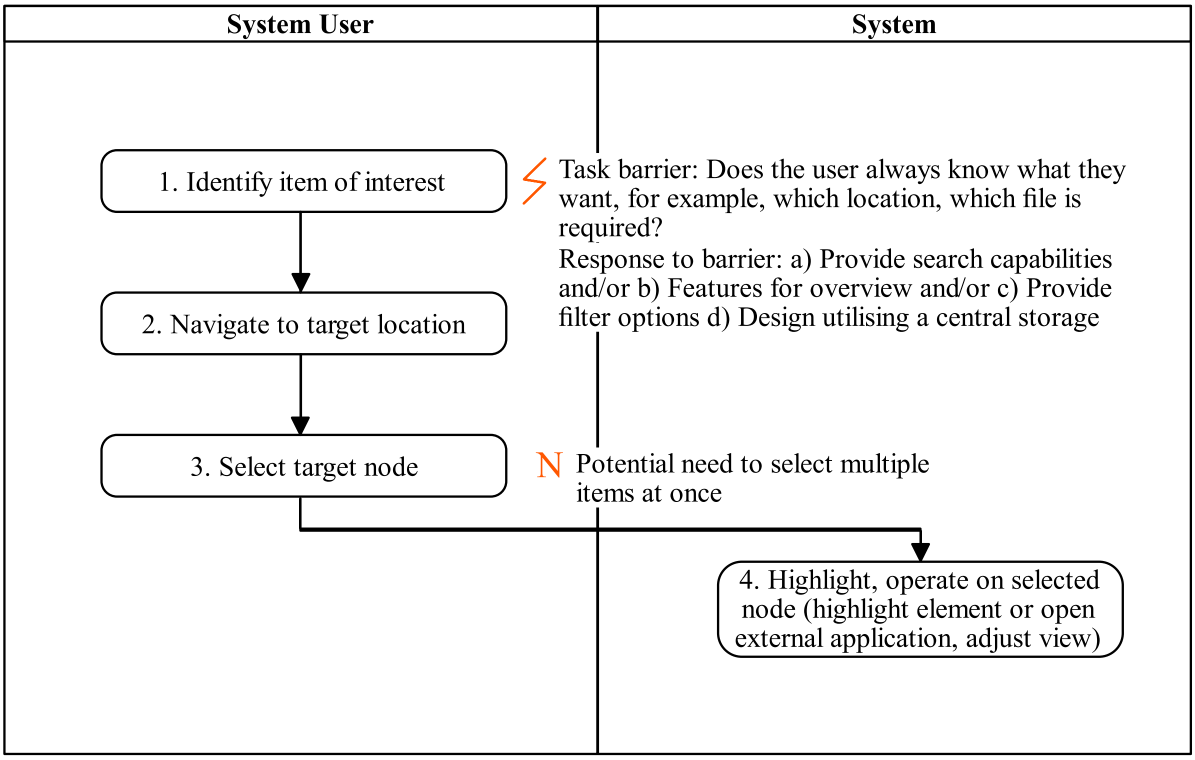



Figure 4.18: Task Sequence Model – Select a Node

*Note*. Task sequence model visualising the selection of a node (file or folder). The model is derived from my user stories (see Table B4). Bolts in orange colour signal challenges. Potential other needs are highlighted with an orange N symbol.

## 4.8 Chapter Summary

In the context of file and folder management the **be here now** metaphor to me is the most interesting result from my attempt to understand user needs. At its core it describes a need for compressing information (*improve search, screen space utilisation*), to support user goals (*improve efficiency, additional relations, relate and analyse, error-free, type-based rendering*), to acknowledge the user’s ecology (*bias, immersion*), for coherency and control or simply put overview (*location independence*, *device independence*). Consequently, I mainly designed my first sketch (see Section 4.6) and user stories (see Section 4.7) based on this metaphor. At the time of writing, I have to acknowledge that the visualisation of hierarchical data cannot be limited to the mere design of some new frontend. Hence, I need to at least provide some form of architectural design for back-end operations, for example, to even prototype an artifact supporting *device independence*. Contrary, I have to out scope certain aspects like *improve search* because they are not informing the end result of my research which is visualisation.

Closely linked to the **be here now** metaphor is the pattern I called **structure is control**. Meaning, in the context of file and folder management the results support a user preference to utilise a *customised structure* when the situation allows for it, for example, when efficient tools are available. Consequently, my method has to (initially) support the maintenance of a *customised structure*. Hence, other tasks like *search* are just utilised as a supplementary approach.

Based on my results this is also true for *mobile devices*. If possible, these devices are used like PC desktop devices when managing files and folders (*device independence*). A limiting factor for their use is the available screen space (*screen space utilisation*). In a general context, *mobile devices* are used very regularly, almost as a form of companion. Hence, I summarised my findings as the **mobile device a restricted companion**.

In a general context the main tasks carried out by users are **internet, office, communication and organisation** related. Consequently, besides file and folder management these areas are valid candidates for further case studies, for example, utilise the novel method for online shopping.

Also, individuals choose **beautiful and usable software with perceived value**. Meaning, *usefulness*, *usability* and *emotional impact* are equally important for user acceptance. Consequently, a design has to consider all three components. For *emotional* *impact* this result was rather surprising to me as I did not expect it to be a conscious component for user preference.

My results also show that the **work practice ecology is 2D videos and text**. Meaning, most applications are still designed as 2D interfaces and the most used data types are video and *text*. Consequently, my method shall (initially) at least support *text* and video files.

Moreover, I conclude that virtual or augmented reality devices are closely linked to the possibilities of the future while at the same time are rarely used. Albeit not part of my research I wonder where this **faith in virtual reality** is coming from (see Section 4.5.4). Regardless, devices to support a form of virtual reality are not part of the work practice ecology.

# Chapter 5 Design of a Novel Method

## 5.1 Introduction

In this chapter I describe the design of my novel method. In general, I aimed to create a design considering my findings identified in my literature review (see Section 2.7):

- Support overview and navigation tasks utilising the available screen space. This may be realised utilising novel approaches combining both 2D and 3D techniques.
- Be able to scale arbitrarily from a design perspective to prevent visual clutter and a technical perspective to load only the relevant parts of the tree
- Be designed utilising a user-centred approach which does include an attempt to understand the needs of potential users

The user needs I identified and have to consider in my design, precisely, the final user stories (priority set to “*good*”) and models are outlined in Section 4.7.

To reiterate, my method was designed to provide a visualisation of hierarchical (tree) data in general. File and folder management was chosen as a focus area or topic to support user familiarity. Also, as the UX design lifecycle is an iterative process (see Section 3.7) I regularly moved from one lifecycle activity to another, for example, still identifying new user stories during design.

In Section 5.2 I describe my initial thoughts and design ideas based on my initial literature review. In Section 5.3 I explain the first sketch presented to research participants after phase 2 of understanding user needs (see Section 4.6). Consequently, in Section 5.4 I describe my conceptual design. Section 5.5 describes the transition to my wireframes. Section 5.6 provides a short summary of the design process.

## 5.2 Initial Design

My initial literature review informed my conceptual design phase. At this stage my design was inspired by techniques such as the Cone Tree (see Section 2.4.4). Intrigued by the potential of a *perspective projection* I worked on an early conceptual design. Worth noting that, a perspective projection is a type of projection which relates to our view in the real world where objects which are farther away appear smaller (Van Verth & Bishop, 2016). My early conceptual design was devised from an ecological perspective.

Figure 5.1 shows this early conceptual design from the ecological perspective. At this stage I envisioned a system where one describes a data layout for visualisation and interaction, for example, for a relational database. My system would interpret this description to render the objects for view and interaction. For visualisation my design metaphor was a sort of solar system where each planet visualises the children of a node, for example, files in a folder.

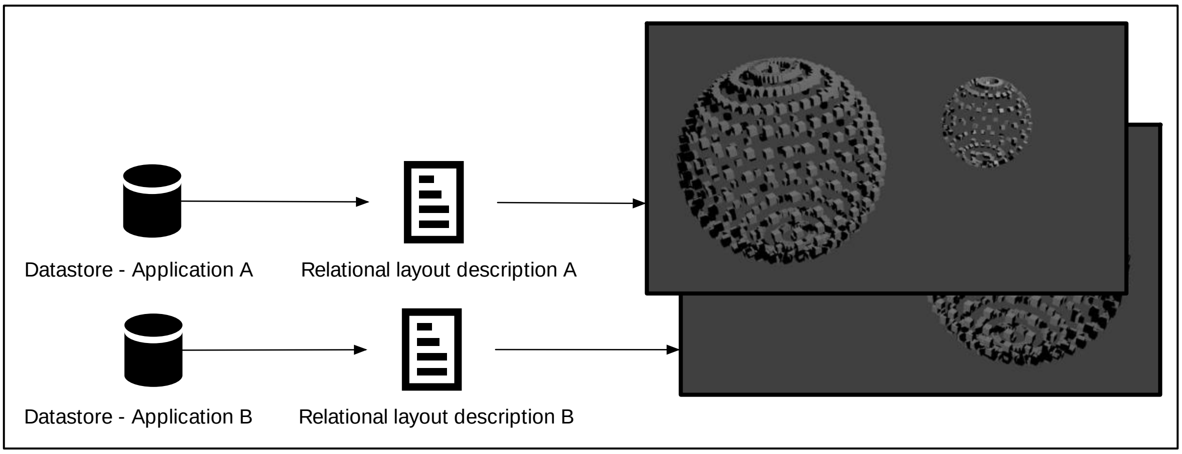



Figure 5.1: Initial Conceptual Design – Ecological Perspective

*Note*. Initial conceptual design from the ecological perspective. The spherical rendering on the right was created using Blender (see Blender Foundation, n.d.).

## 5.3 Intermediate Sketch to Understand User Needs

I noticed after carrying out my phase 1 interviews and phase 2 online questionnaire that it may be valuable to carry out additional interviews including a first sketch to visualise my understanding of user needs at this point in time (see Section 4.6). The sketch was presented for critiquing to identify additional requirements and primarily to evaluate previous indications, for example to evaluate a pattern from the previous phases.

Worth noting that, the critiquing of the sketch can also be interpreted as a form of a design walk-through.

The sketch (see Figure 4.9) visualises an early concept mainly from the interaction perspective for PC desktop use. It shows how a System User is placed in the middle of their data from a first-person perspective. They can move using the arrow keys on their keyboard and change the view direction with their mouse. The contents of every folder are arranged as circles. If the items of a folder do not fit the circumference of one circle they are shifted to new circles. The user can switch between these circles using the scroll wheel of their mouse. To switch folders the user can move towards the target folder. I called the contents of each folder a "*Space*" in my sketch. If the user looks at a file long enough (3 seconds) a form of preview would start, for example, a video would start playing. Additionally, the user can look up to explore related content to files (Things of interest), for example, files from similar projects or videos which are similar to their content. If the user looks down, they would see older versions of their files, for example, in the sense of a daily backup (History).

The described sketch was my first attempt to realise the **be here now** metaphor. In detail, I chose a first-person perspective to create a sense of *immersion*. Also, I aimed to optimise *screen space utilisation* by not only filling the horizontal space with one circle but also utilising the vertical screen space for more items accessible through scrolling visualised as translucent circles. The symbols for the files and folders were inspired by existing icons to support user familiarity (*bias*). The feature to show related content when the user looks up was designed to support activities (*relate and analyse*). Similarly, I designed the feature to view older versions of files when the user looks down to support the need for *additional relations*.

As **structure is control** the sketch also visualises a user navigating their own *customised structure* to manage files and folders. The data is structured as multiple interconnected circles. Also, I put emphasis on showing videos and text files as the **work practice ecology is 2D videos and text** based.

Also, I aimed to address my findings from my literature review by supporting a form of overview and precise navigation utilising the available screen space. Additionally, I designed my method to be scalable without introducing visual clutter and user familiarity, for example through the use of familiar icons.

For contextualisation, my method in this sketch is an adjacency layout primarily utilising continuous zooming for context+focus. At the time of writing, HotSauce (see Section 2.4.6) is quite similar to my technique as shown in my sketch (see Figure 4.9). However, I argue that my technique can convey structural information better because my layout using circles clearly separates the content of folders from each other. Also, I argue that my design offers better results to reduce visual clutter because active objects, like files, are all placed at the same height for one level. Consequently, object occlusion can still occur but only to a lesser degree.

The idea to utilise a circular shape for the layout of nodes was initially informed by it feeling natural to me and my previous work designing and developing a form of radial menu in 3D (Reiter, 2011). To address this form of a potential bias early, I evaluated my design (sketch) utilising the Gestalt principles of visual perception. According to Johnson (2021) the most relevant Gestalt principles to user interface design are: Proximity, Similarity, Continuity, Closure, Symmetry, Figure/Ground and Common Fate. Overall, I came to the conclusion that my design (sketch) had a lot of potential to efficiently convey structural information. In particular:

- Proximity: The layout for child nodes (files and folders) in one parent node (folder) appear grouped to each other because they are placed close together. The child nodes of other parent nodes are placed farther away to signal another, separate group of nodes.

- Similarity: The icons in the design support grouping of similar node types, for example, to identify files of the same type.
- Continuity: The layout to have multiple circles full of child nodes (files and folders) for one parent node (folder) may transport a continuous view. Meaning, the separate circles which are stacked on each other may be viewed as one entity, for example, a stack of images.
- Closure: A clearly defined shape like a circle may always be perceived as a whole. To be precise, even if a parent node (folder) only has a small amount of child nodes (files and folders) the user's vision may be biased to view it as a complete object (circle). This may support a consistence user experience.
- Symmetry: The complete layout can be viewed as a form of a 3D table with a set place for every node. This setup may support the user's ability to scan for information.
- Figure/Ground: Because of the perspective projection the nodes closer to the user appear larger than the nodes farther away (context). However, there is a risk that the background contextual information gets distracting to the user.
- Common Fate: The scrolling operation moves all of the concerned nodes at once also highlighting that they are grouped.

Worth noting that, I planned to create and evaluate multiple, distinct concepts and approaches for my design phase. However, I did only come up with one concept. To mitigate this, I aimed to offer a rather open discussion whenever I presented design related documents to research participants. To exemplify, in the semi-structured interview with Professional02 in phase 3 for understanding user needs (see Section 4.6) I clearly stated that I am also open for opposing design concepts. In a sense, I was hoping to start a discussion for a *participatory* *design*. In short, participatory design is a field where stakeholders (especially users) are directly involved in the design of a system (Muller & Kuhn, 1993). Unfortunately, the participants also did not come up with other opposing concepts.

**5.4 Conceptual Design**

After finalising my analysis to understand user needs (see Chapter 4) I started working on my conceptual design. The overall positive feedback I received (see Section 4.6.2) for my intermediate sketch (see Section 5.3) motivated me to follow up on this concept.

Conceptually, I changed the design from the sketch to offer a split “*2D Navigation*” and “*3D Overview*” mode to *improve efficiency* and to support *error-free* navigation. This decision was mainly informed by the feedback I received from Student01 in phase 3 of understanding user needs after presenting the sketch (see Section 4.6). To exemplify, when talking about the control behaviour of the method Student01 emphasised the importance of a balance between navigation speed (*improve efficiency*, see user story ID 10) and precision (*error-free*, see user story ID 9). To quote Student01: “*And then this split between moving fast to the files and folders but not so fast that one somehow runs the risk of running past somewhere and so to say overlooks that*”. Consequently, I envisioned a 2D Navigation view for fast and precise navigation while the 3D Overview mode may offer structural information.

Similarly, I removed the preview option from my design as it was initially meant as a form to support details-on-demand in a 3D only view. Also, I noticed that the interviewees in phase 3 (see Section 4.6) focused quite a lot on these features. Consequently, with this new split design approach I considered it a feature to be potentially distracting and only relevant for a commercial product or later iterations.

Similarly, I changed the design from the sketch (see Section 5.3) so the nodes (files, folders) are laid out in a vertically orientated circle instead of a horizontal one. I changed the orientation so more active items can be shown on the screen at the same time (*screen space utilisation*, see user story ID 12).

Worth noting that, all of the shown conceptual designs in this section utilise a similar layout, elements (for example, system boundary) and symbols as Hartson and Pyla (2019).

### 5.4.1 Conceptual Design From the Ecological Perspective

Figure 5.2 shows my conceptual design from the ecological perspective. It visualises the **be here now** metaphor showing a remote server and database at the centre. The System User connects to this server using a mobile or PC desktop device to maintain their *customised structure* (see user story ID 2), for example, to create files. They utilise an identical view (user interface) across devices for *device independence* (see user story ID 17). Their current navigation position and actions, for example, the sub folder they are currently in gets synchronised with the be here now server so that when they switch devices, they are always were they left off to strengthen their feeling for central control (*location independence,* see user story ID 18). Consequently, the System User may have the option to carry out connected tasks to support activities, for example, search for images on their smart phone to later switch to their laptop to utilise these images for writing an e-mail or to create a Microsoft PowerPoint presentation, in short to *relate and analyse* (see user story ID 20). In this conceptual design remote storage locations of a user are degraded to mere available storage space. In detail, when a user creates files, the remote server decides where to put these files, for example, if the user connected a Google Drive instance to the server the files may be put there if enough storage space is still available. The System User can connect multiple storage locations to the server while maintaining only one *customised structure* (see user story ID 2) for *location independence* (see user story ID 18).

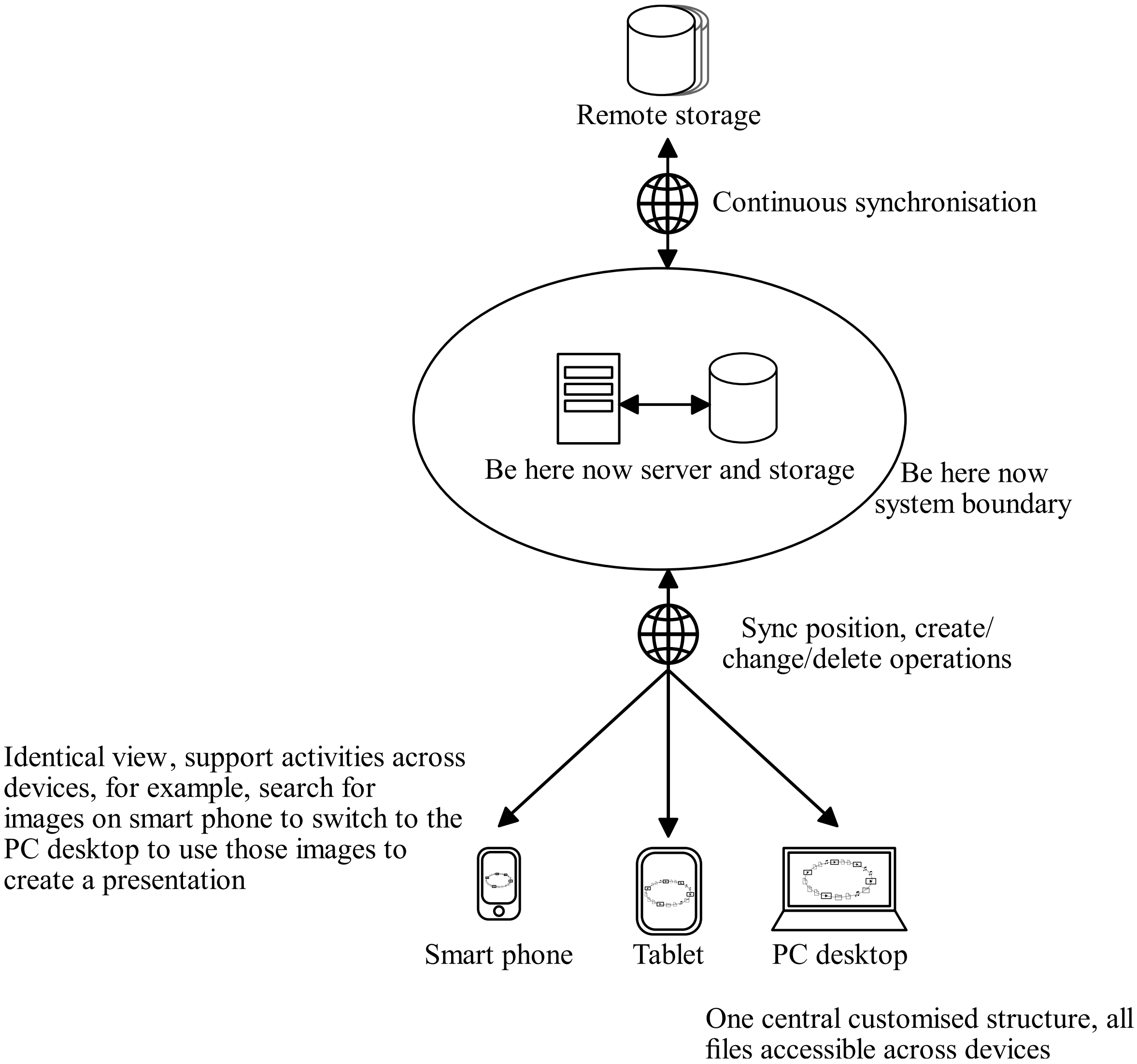



Figure 5.2: Conceptual Design From the Ecological Perspective

*Note*. English version of my conceptual design from the ecological perspective for my novel method. A German version of this design was presented to interview participants. I translated the German version myself based on the English original.

### 5.4.2 Storyboard

Figure 5.3 shows page 1 of my storyboard from the ecological, interaction and emotional perspective. I chose to focus on my task sequence model to select a node (see Figure 4.18) for this storyboard as I claim it represents one of the most common and therefore key (sub) tasks (see Section 4.7). The scenario for my storyboard tells the story of a working professional who has to start working on another project after leaving work. On their way home they take their mobile phone to start browsing their content (frame at the centre). They notice that they are still in the same position (folder path) even after shutting down their work computer (*location independence*, see user story ID 18). Next, in the interaction perspective (bottom left frame) they are viewing their data (**work practice ecology is 2D videos and text**, see user story ID 21) in a 2D Navigation view with the option to swipe for scrolling. The symbols for folders and documents are familiar to the user (*bias*, see user story ID 5). The professional notices the blue ribbon on the right side of their notes file. They tap on it to reveal related content. In the next frame (bottom centre) they review the related content (*relate and analyse*, see user story ID 20) from a previous, similar project (Things of interest). Consequently, they switch back to the main screen to review older versions of their notes file (*additional relations*, see user story ID 16) by tapping on the green ribbon (bottom right frame, History feature).

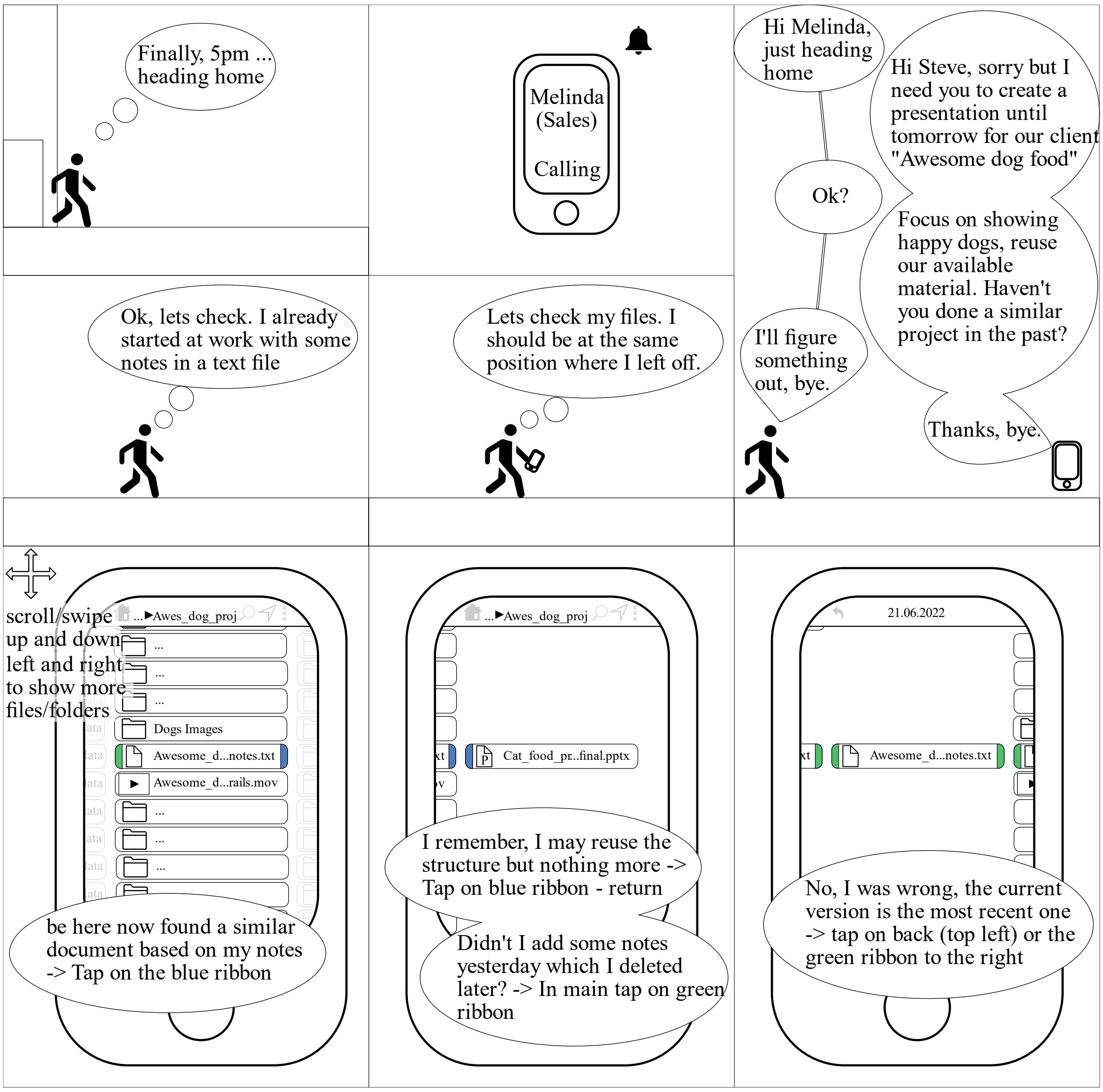



Figure 5.3: Storyboard From the Ecological, Interaction and Emotional Perspective – Page 1

*Note*. English version of my storyboard as part of my conceptual design from the ecological, interaction and emotional perspective. A German version of this design was presented to interview participants. I translated the German version myself based on the English original.

Figure 5.4 shows page 2 of my storyboard from the ecological, interaction and emotional perspective. Continuing from page 1, the working professional in my scenario arrives at home (top left frame). In the next frame (top right) they open their laptop and notice that they are still in the same position (*location independence*, see user story ID 18). Next, they open a folder with images for the project. The user interface has the same layout as the mobile version they can only see more items left and right (*device independence*, *screen space utilisation*, see user story ID 17 and ID 12). To see more images at once and to see potentially more relevant images the user decides to enter in the 3D Overview mode (*location independence*, see user story ID 18) by clicking on the arrow symbol (frame bottom left). In the last frame (bottom right) the user views the folder content in the 3D Overview mode. They only see the icons of the folder content as icons, or thumbnails for images or videos (*improve efficiency*, see user story ID 10). To control their position and view direction (continuous zoom) they can utilise their keyboard and mouse or the touch controls. A crosshair at the centre of the screen highlights their current focus point. They joyfully notice that they are now in the middle of their images (*immersion*, see user story 7) and can identify potentially more relevant content in the distance thanks to the perspective projection of the view (*location independence*, see user story ID 18).

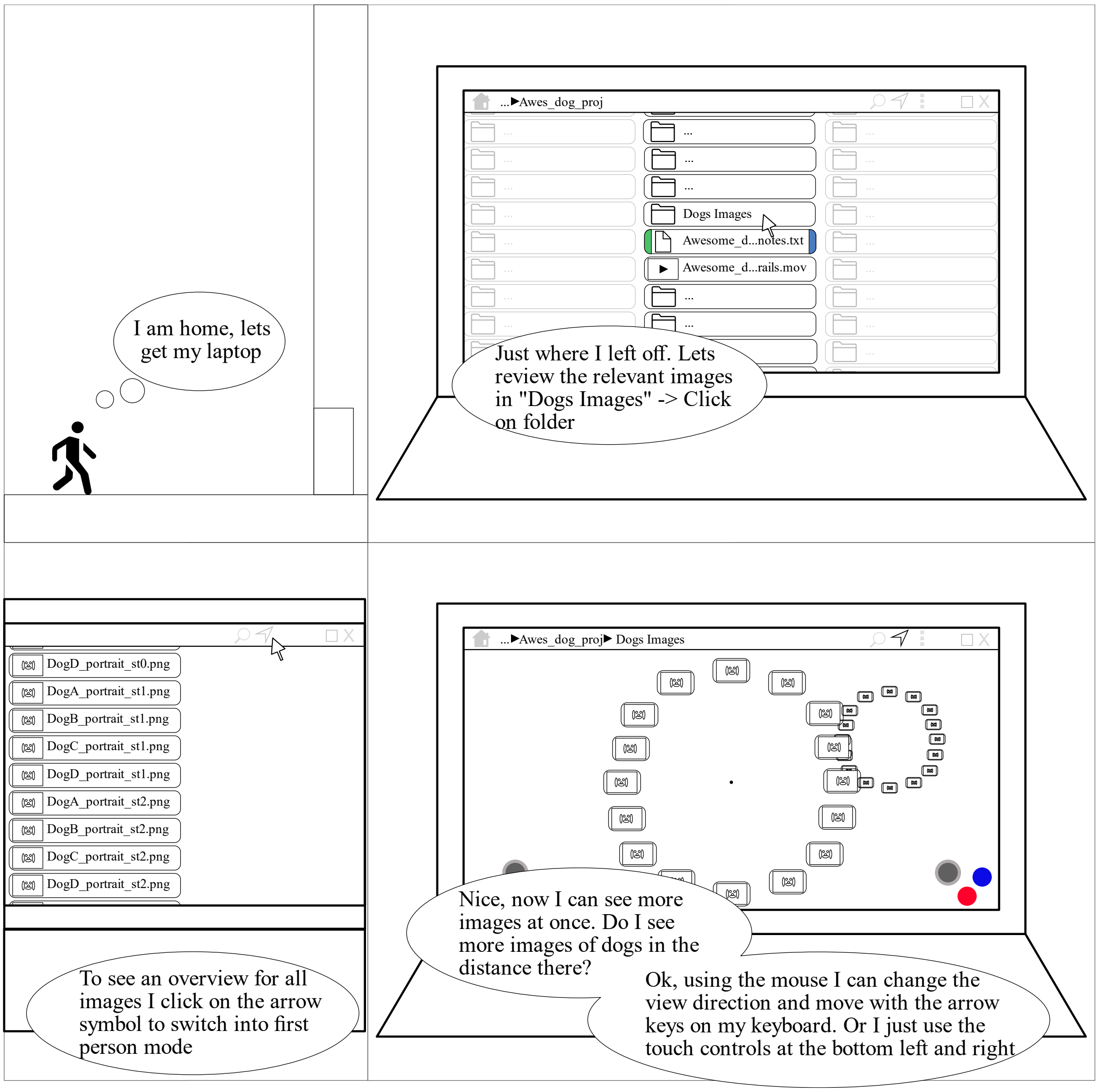



Figure 5.4: Storyboard From the Ecological, Interaction and Emotional Perspective – Page 2

*Note*. English version of my storyboard as part of my conceptual design from the ecological, interaction and emotional perspective. A German version of this design was presented to interview participants. I translated the German version myself based on the English original.

## 5.5 Wireframes

My conceptual designs were received rather positively (see Section 6.2). Primarily, the features for *location independence* (see user story ID 18), for example, utilising the 3D

Overview mode were highlighted by the participant. I did not identify any significant change requests to my conceptual design from the ecological perspective (see Figure 5.2) or my storyboard (see Figure 5.3 and Figure 5.4).

Based on my user stories, task sequence models (see Section 4.7) and storyboard (see Section 5.4.2) I started to design my wireframe prototype.

### 5.5.1 Design Challenges and Guidelines

At this stage I had to face a two-fold challenge. One the one hand I had to design my novel node layout while at the same time utilising rather standardised components like menus for general interaction to not distract potential users. Initially one may assume that the utilisation of standardised components is a trivial task. However, when viewed in detail I identified the following key challenges:

- How large should these components be to still be usable, for example, that touch targets are large enough?
- What icons should one use which are pleasing to research participants but not distracting?
- What fonts and more importantly font sizes may be used?
- Are these components overall usable? Meaning, presenting a kind of non-standardised menu may distract the participant and negatively impact their assessment.
- What colour scheme may be utilised to be pleasing to the research participant while not being distracting?

Consequently, I decided to inform my design utilising parts of the *Material Design* guidelines, version two from Google (n.d.-k). Hence, when utilised to optimise the design of websites for small screens the Material Design guidelines can even increase the effectivity and efficiency for delivering information (Pinandito et al., 2017). I argue that these guidelines are a solid foundation for my type of responsive design. Worth noting that, I did not utilise these guidelines as a kind of legal code. To exemplify, I used the specifications of the bottom app bar (Google, n.d.-a) as a baseline for the bottom navigation bar in my design.

Also, at the time of writing, it is quite challenging to identify the suitable screen sizes to design the interaction across devices. For mobile phones I chose to design for a screen resolution of 360 pixel width and 640 pixel height. I chose this resolution as a majority of individuals is still using this screen size (Bose, 2021). Similarly, for PC desktop devices I chose a resolution of 1920 pixel width and 1080 pixel height as this resolution is one of the most commonly used resolutions across devices (Bose, 2021). For tablet devices I chose the Apple iPad with a “10.2-inch (diagonal) … display” (Apple Inc., 2022, Display section) to be the design target. The screen has a “2160x1620-pixel resolution at 264 pixels per inch” (Apple Inc., 2022, Display section). I chose to use this device as the category of tablet devices is still linked to the success of the iPad. Hence, when looking at tablet shipments Apple is still the leading vendor (Laricchia, 2022).

A consequential challenge was to identify a suitable screen density independent measurement unit to size the respective components, for example, the width of a button. I utilised the measurement unit of *density-independent pixels (dp)*. One dp is a physical pixel on a device with a set screen density of 160 (Google, n.d.-f). Equation 1 shows the formula on how to calculate dp (Google, n.d.-f, Dps and screen density section).

$$\boldsymbol{dp = \frac{(width\ in\ pixels \times 160)}{screen\ density}} \quad (1)$$

Consequently, I designed my method for the following screen dimensions and interaction channels:

- Mobile phone: 360x640 pixels at 160 pixels per inch (ppi) using touch controls
- Tablet: 982x1309 pixels at 160 ppi using touch controls

- PC desktop: 1920x1080 pixels at 110 ppi approximating a 20-inch monitor using a PC mouse, keyboard and (alternatively) touch controls

Worth noting that, regarding PC desktop a target ppi of 110 seemed reasonable because a lot of commonly used monitors have a similar ppi, for example, 27-inch monitors with a resolution of 2560x1440 pixels have a ppi of around 109.

For typography I used a reduced *type scale* derived from Google (n.d.-i). A type scale in this context means a set of exactly 13 styles where one style describes a category of text, for example, a specific headline level with its font details (Google, n.d.-i). I did not have a need for some styles, for example, large headline sections. Therefore, I used less than 13 styles for my design. Regarding fonts, I applied just one family of fonts to my design namely *Roboto* by Robertson (n.d.).

For a rather simple colour scheme I utilised a primary colour in combination with a light and dark variant as described by Google (n.d.-b). I chose the following colours (hex values) utilising the COLOR TOOL:

- Primary colour: #3f50b5
- Dark variant: #002884
- Light variant: #757ce8
- Text on primary colour: #ffffff

Text on light background was set to black (hex value #000000) with different opacity values as described by Google (n.d.-h).

I utilised various components for my design. A component describes an interactive group of elements to construct a user interface (Google, n.d.-d). Primarily, I utilised the specifications for app bars (bottom and top), buttons, lists, menus, text fields and dialogs (Google, n.d.-d). Also, I used icons from Google (2022) in my design.

Figure 5.5 shows an example wireframe for the mobile phone screen. Worth noting that, in general the design was also influenced by existing applications to manage files and folders, for example, the Windows File Explorer on PC and mobile applications from Samsung and Google.

All of my wireframes were created using the program Affinity Designer (see Serif (Europe) Ltd, n.d.) version one on a Windows PC.

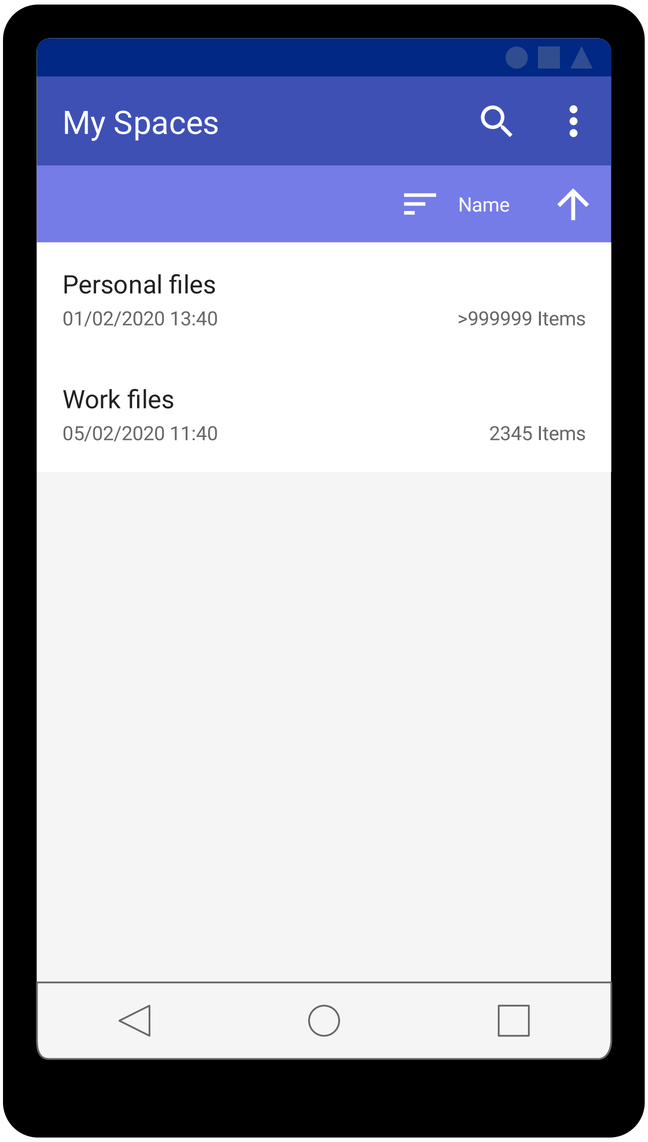



Figure 5.5: Wireframe Design Example

*Note*. Wireframe example outlining the design approach from the interaction perspective. Designed for a mobile phone with a screen resolution of 360x640 pixels at 160 ppi. Wireframe created using Affinity Designer on Windows.

### 5.5.2 Node Layout

Given the preference towards a vertical circular *node layout* (see Section 6.2) I had to address various consequential challenges. In detail, the key challenge for my design was the transition from the conceptual node layout to an actual, precise design from the interaction perspective. Also, I had to consider my findings from my literature review, for example, regarding the scaling of the tree (see Section 2.7). Consequently, I had to address the following questions:

- How many child nodes may fit in one vertical circle for an optimised *screen space utilisation* (see user story ID 12)?
- How can this structure grow, where does it begin?
- How does the method scale when the number of child nodes does not fit into one circle?
- How can the overall structure grow indefinitely?

At this stage I knew that according to the Material Design guidelines a child node (two-line item with icon) has a height of 72dp (Google, n.d.-e). Also, to fill the screen I have to take into account the PC desktop screen size as a worst case with a height of 1080 pixels at 110 ppi. Converting 72dp to pixels results in a value of (rounded) 50 pixels (see Equation 1). Meaning, the node in focus of the user at the centre of the screen has a height of 50 pixels.

To carry out the rest of the calculation I utilised the capabilities of the Unity engine (version 2021.3.6f1 LTS). I used a Unity preset to create a 3D *universal render pipeline (URP)* project. Borromeo (2021) explained that this Unity render pipeline (URP) can run on almost any target device. Therefore, I chose this pipeline to be able to test my design for PC desktop and mobile devices.

When creating such a Unity project the camera is placed at the default coordinates of 0 (x),1 (y),-10 (z). Worth noting that, these coordinates are in *World* space. World space utilises a static reference for all objects in the scene, this reference point in Unity is at 0, 0, 0

(Ferrone, 2020). In contrast, the *Local* space utilises the respective parent transform of the object as the reference point (Ferrone, 2020).

For the 2D Navigation mode I aimed to identify a suitable balance between the camera position and the size of the first vertical circle. I utilised a rotated (90 degrees around the z-axis), default Unity cylinder mesh to identify these variables. Consequently, I decided to move the camera back two units along the z-axis resulting at the coordinates 0, 1, -12. The centre of the first circle (cylinder) was at 0, 1, -10 with a radius of 1. Figure 5.6 shows the output result from Unity.

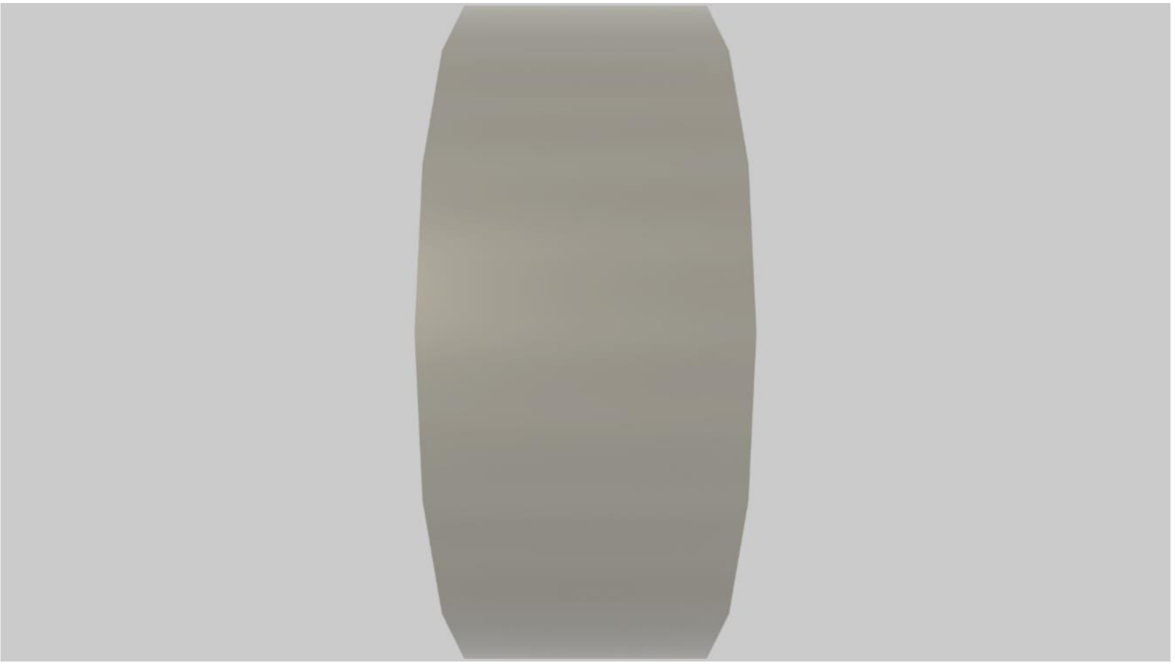


Figure 5.6: Node Layout Measurements Utilising a Cylinder

*Note*. Unity render result at 1920x1080 pixels simulating 2D Navigation mode with a fixed camera and one vertical circle visualised by a placeholder (cylinder mesh).

Next, I rendered a first child node as a Unity quad mesh in the centre of the screen. I positioned the mesh at the circumference of the cylinder at 0, 1, -11. I sized the height so the output is at 50 pixels resulting at a y-axis scale value of 0.054 (in World space). I restricted the node width to 440 pixels (640 dp) on PC desktop to fit a maximum of 60 uppercase characters. Hence, I adapted the preference of Professional02 to utilise the whole width of the available screen space for *node scaling* (see Section 6.2). However, I only adjust the width of a child node depending on the device or orientation to fit a maximum of 60 uppercase characters on the first line. Hence, the ideal length of a line for text is between 40 and 60 characters to maintain readability (Google, n.d.-j). Figure 5.7 shows the visualisation of the child node at the centre. The child node adheres to the specifications of a two-line item with an icon (Google, n.d.-e). Worth noting that, the child node closest to the camera is always at the centre of the screen (detail) while the other child nodes follow the circumference of the circle and are farther away (context+focus). Hence, I also tried to move the camera into the circle so the child node at the centre would have been the smallest one. Albeit an interesting experiment, I rejected this approach as the contextual content would be more prominent than the child node at the centre of the screen.


Figure 5.7: Node Layout Measurements Centre Node

*Note*. Unity render result at 1920x1080 pixels simulating 2D Navigation mode with a fixed camera and one vertical circle visualised by a placeholder (cylinder mesh) with the first child node at the centre.

To calculate the maximum amount of child nodes which fit on one circle I needed to consider the equation to calculate a circles circumference ($c$) (see Equation 2). For the radius of the cylinder ($r = 1$) this results in a circumference of $2 \times \pi$. Dividing the circumference by the height of one child node (0.054) results in a maximum of 116 child nodes (rounded).

$$\boldsymbol{c = 2 \times \pi \times r} \tag{2}$$

The next challenge was to place those 116 child nodes at the circumference. To calculate the individual positions from the centre of the circle at 0, 1, -10 I utilised Equation 3 and Equation 4. To get the coordinates for $z_i$ and $y_i$ one just needs to calculate the cosine and sine for the individual position of the node in a circle with $360°$ or $2 \times \pi$, for example, the first node $i = 0$ at $2 \times \pi \times (0 \div 116) = 0$. Consequently, one just needs to multiply the result with the radius $r$ to get the individual coordinates. Worth noting that, I also added a multiplier of $-1$ so the first child node is placed at the centre of the screen.

Next, the individual results ($z$ and $y$) for each child node $i$ of the 116 nodes ($N$) were added to the centre position of the circle. Figure 5.8 shows the result rendering these 116 child nodes. Worth noting that, the child nodes were also rotated to face the camera straight on.

$$\boldsymbol{z_i = -1 \times r \times \cos\left(2 \times \pi \times \left(\frac{i}{N}\right)\right) \ (0 \leq i < N)} \tag{3}$$

$$\boldsymbol{y_i = -1 \times r \times \sin\left(2 \times \pi \times \left(\frac{i}{N}\right)\right) \ (0 \leq i < N)} \tag{4}$$

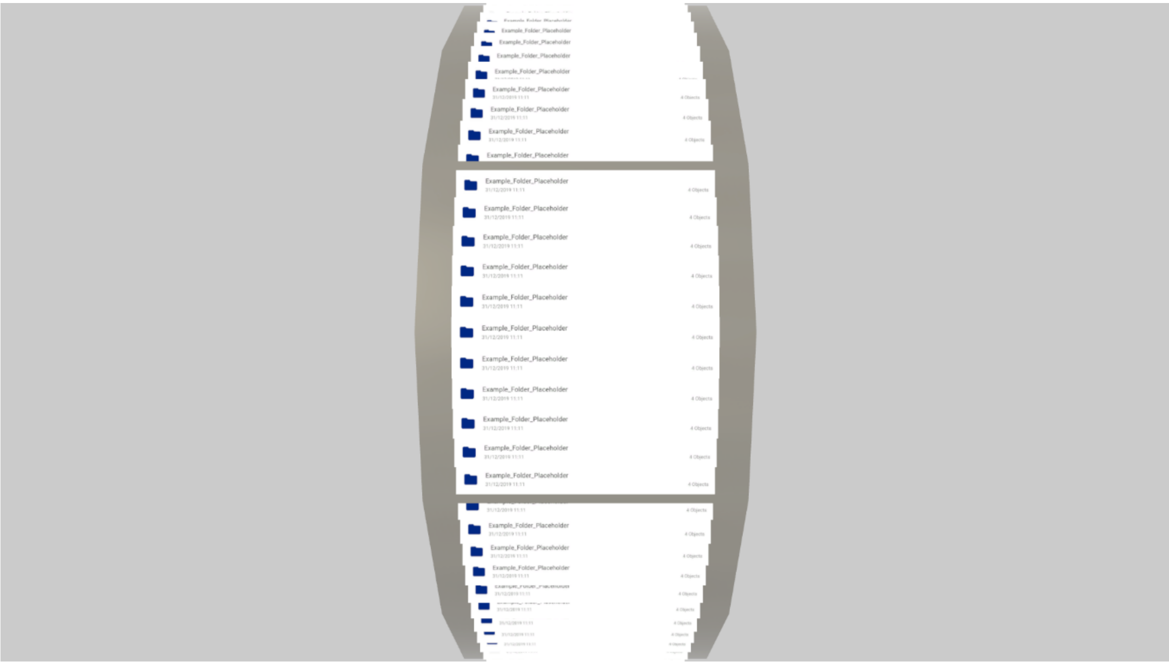


Figure 5.8: Node Layout Measurements 116 Child Nodes

*Note*. Unity render result at 1920x1080 pixels simulating 2D Navigation mode with a fixed camera and one vertical circle visualised by a placeholder (cylinder mesh) with the 116 child nodes. The child nodes are rotated to face towards the camera straight on.

Albeit working, I decided to reduce the number of child nodes to achieve a better separation between the nodes. In the end I settled for 110 child nodes to be the maximum of child nodes in one circle. Figure 5.9 shows the result for 110 child nodes. It highlights the improved spatial separation between the individual nodes.

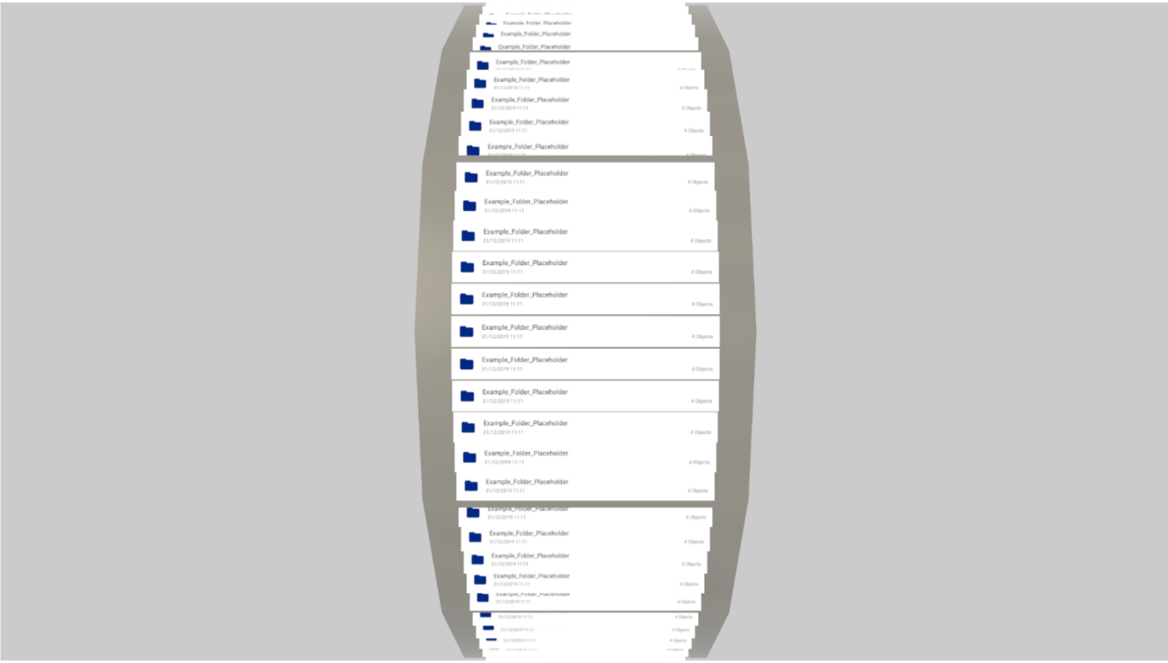


Figure 5.9: Node Layout Measurements 110 Child Nodes

*Note*. Unity render result at 1920x1080 pixels simulating 2D Navigation mode with a fixed camera and one vertical circle visualised by a placeholder (cylinder mesh) with 110 child nodes. The child nodes are rotated to face towards the camera straight on.

Equation 3 and Equation 4 already answer the question regarding how the structure grows. In detail, every new child node is added below the last item until the circle is complete at 110 child nodes. Figure 5.10 visualises this approach showing three child nodes as an example. Worth noting that, whenever a new child node is created the camera is moved accordingly so the new node is at the centre of the screen (see Figure 5.11). This scrolling operation like manual scrolling or swiping may be carried out by moving the view of the System User and not the actual data.

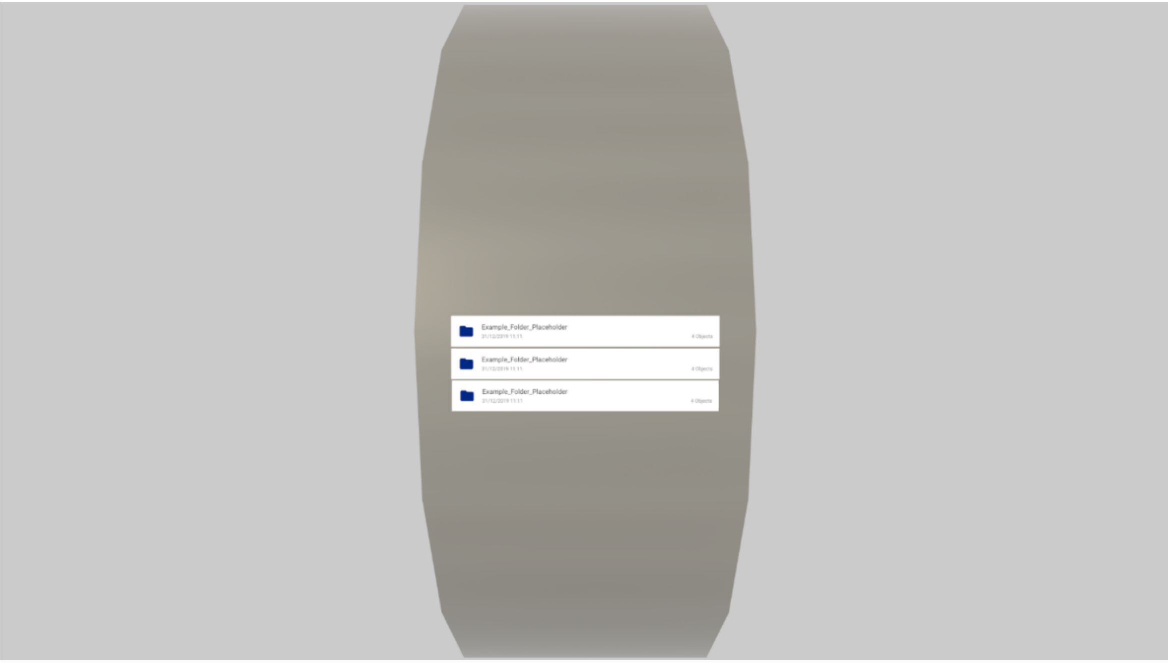



Figure 5.10: Node Layout Measurements Three Child Nodes

*Note*. Unity render result at 1920x1080 pixels simulating 2D Navigation mode with a fixed camera and one vertical circle visualised by a placeholder (cylinder mesh) with three child nodes. The child nodes are rotated to face towards the camera straight on.

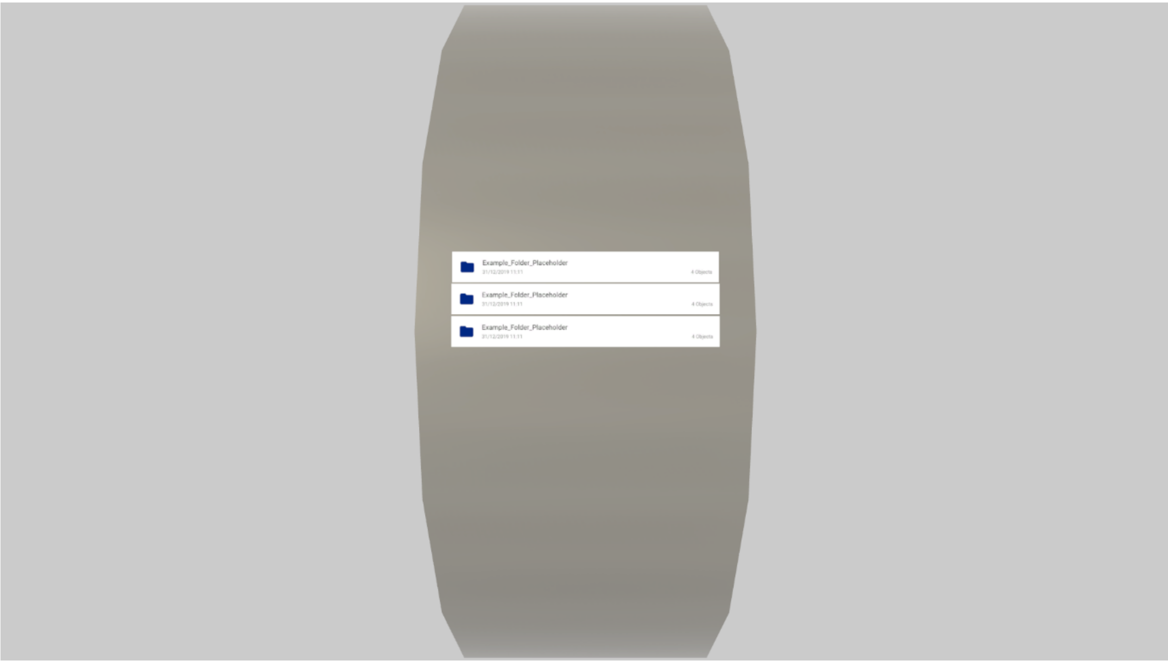


Figure 5.11: Node Layout Measurements Three Child Nodes Scrolled View

*Note*. Unity render result at 1920x1080 pixels simulating 2D Navigation mode with a fixed camera and one vertical circle visualised by a placeholder (cylinder mesh) with three child nodes. The child nodes are rotated to face towards the camera straight on. This view shows the scrolled view after the third node has been created.

When the amount of child nodes surpasses 110 items new circles are created to the right utilising transparency to highlight their activity status. Worth noting that, I call all circles in my method a “*Node Layer*”. Each Node Layer adheres to the same principles as described earlier, for example, how the circle gets filled with additional child nodes. When a new child node gets created the focus switches to the respective Node Layer where the node resides. Hence, each Node Layer can be viewed as a page to switch to, for example, similar to pages of Internet search results. Figure 5.12 visualises an example with a total of 190 child nodes where the Node Layer in the middle of the screen shows 110 nodes while the inactive Node Layers visualises 80 child nodes. Worth noting that, the placeholder cylinder mesh has been removed for this visualisation.

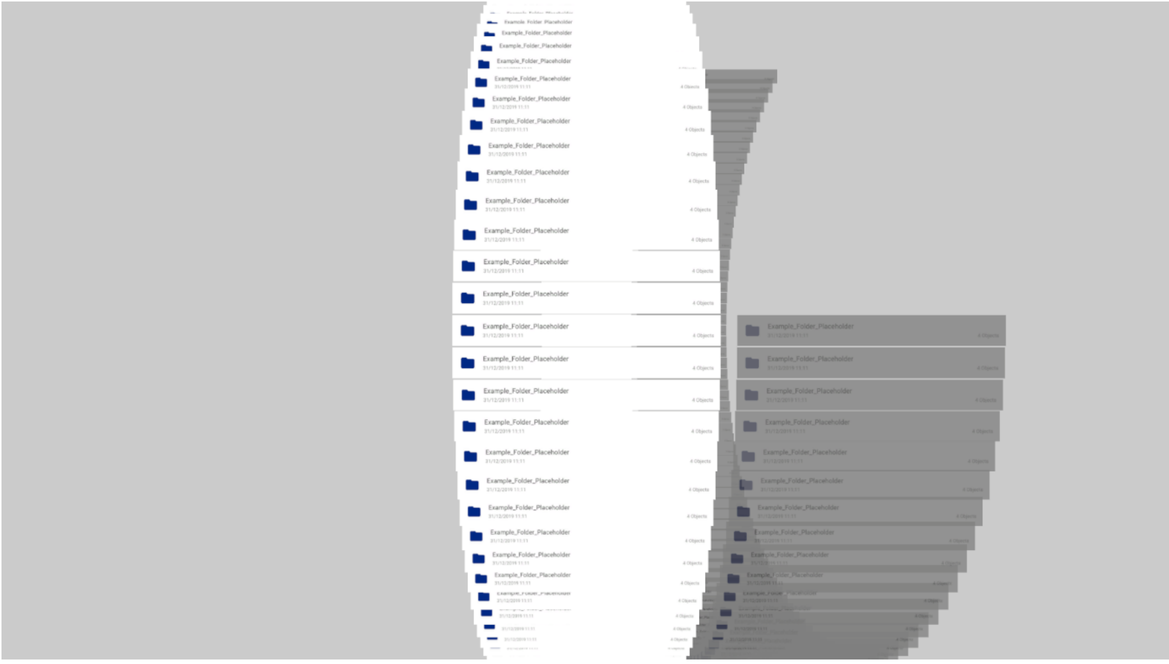


Figure 5.12: Node Layout for 190 Child Nodes

*Note*. Unity render result at 1920x1080 pixels simulating 2D Navigation mode with a fixed camera and two vertical circles (Node Layers) visualising a total of 190 child nodes. The child nodes are rotated to face towards the camera straight on.

The next challenge was to calculate how the structure can grow indefinitely. To be precise, how may the tree grow in depth. With the prerequisite that one Node Layer can contain a maximum of 110 child nodes and at the worst case all of these 110 nodes can be parent nodes (folders) I had to address another mathematical challenge. This challenge was to calculate the radius $R$ required to place the respective child Node Layers around root in a circular layout. Equation 5 describes the required calculation for a set number of child nodes $E$ and the diameter of one Node Layer $2 \times r$. However, because $E$ is not constant the required radius $R$ grows exponentially with the depth of the tree. To exemplify, the first circle around root has to be able to visualise 110 Node Layers where each one of these Node Layers can consist of 110 parent nodes (folders) themselves. Meaning, the second circle around root can already consist of $E = 110^2 = 12100$ Node Layers. This would result in very long distances the System User had to move between these circles in 3D Overview mode.

$$\boldsymbol{R = E \times \frac{2 \times r}{2 \times \pi} \; yielding \; R = E \times \frac{r}{\pi}} \tag{5}$$

Consequently, I decided to keep $E$ static at 110 Node Layers for each outer circle. Equation 6 describes the final calculation for $R_j$ where $j$ is the level of the outer ring and $L$ is the number of outer rings to visualise. Worth noting that, $j$ starts at 1 to keep enough distance between the Node Layers to minimise overlap in 3D Overview mode (see Figure 5.13).

$$\boldsymbol{R_j = (j \times 3 \times r) + (110 \times \frac{r}{\pi}) \quad (0 < j < L)} \tag{6}$$

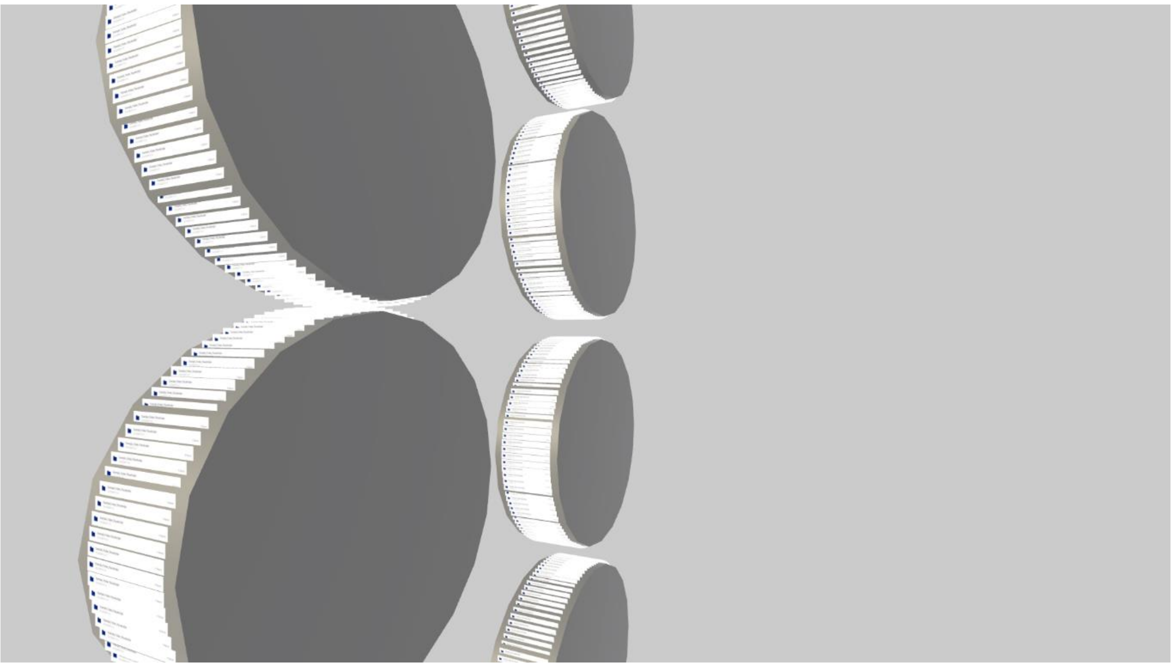


Figure 5.13: Node Layout Measurements for $\mathbf{R_j}$

*Note*. Unity render result at 1920x1080 pixels simulating early 3D Overview mode with placeholder cylinder meshes and 110 child nodes for each Node Layer. On the left, Node Layer child nodes slightly overlap at $j = 0$ while on the centre for $j = 1$ Node Layers are not overlapping.

The final layout visualises the root node and its children at the centre Node Layer. These children can all be parent nodes (folders). The first circle around root visualises all of their content in Node Layers. However, the following outer circle only visualises the contents of the Node Layer currently in focus. The same principle applies for each outer ring after that. For reference, Figure 5.14 describes a tree structure as an example while Figure 5.15 visualises how this structure is rendered using my layout.

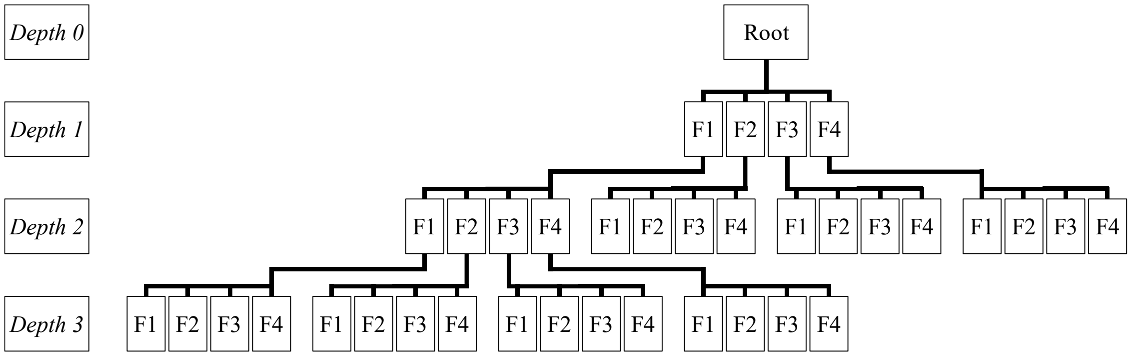



Figure 5.14: Tree Example With Depth Three

*Note*. Example tree with a depth of three with four child nodes (folders F1 to F4) for each parent.

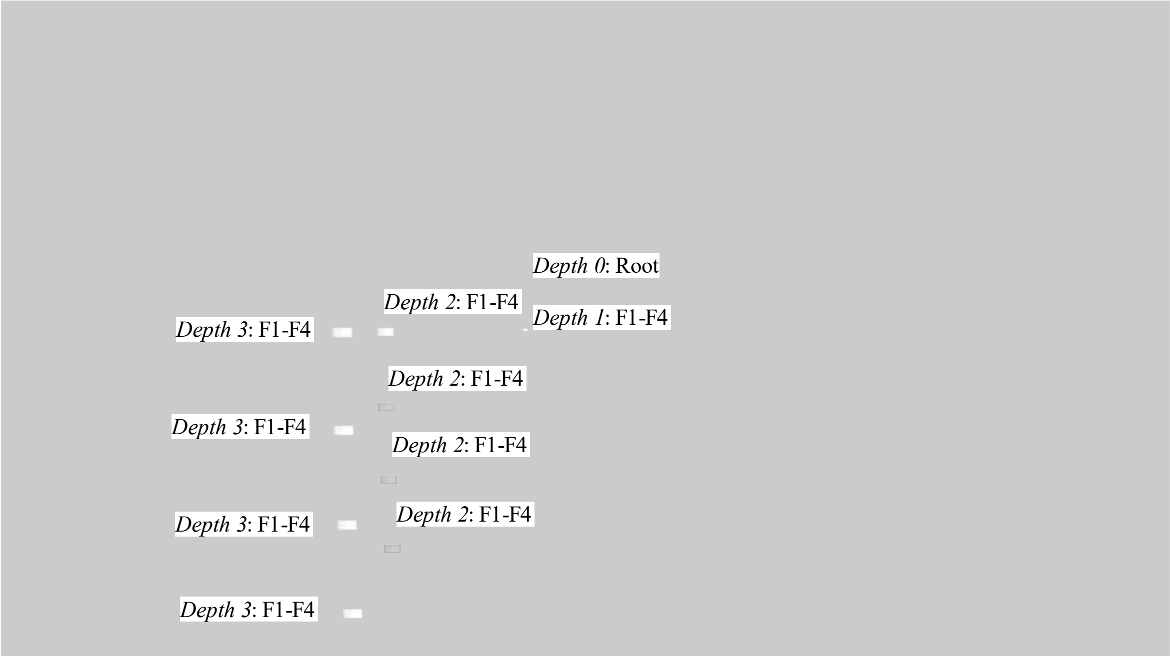



Figure 5.15: Node Layout Example Tree Depth Three

*Note*. Unity render result at 1920x1080 pixels simulating early 3D Overview mode with 4 child nodes (folders) for each Node Layer. At depth two the first Node Layer is active.

### 5.5.3 Wireframes

Based on my node layout I consequently started my design process from the interaction perspective. I focused my design on the worst-case scenario regarding available screen space, precisely, the mobile phone.

To provide System Users the option to separate their work environments I introduced the concept of multiple “*Spaces*”. A Space describes one node layout (see Section 5.5.2) with an unlimited number of nodes (files and folders). This concept was derived from user story ID 22 as an additional approach to spatially separate data as a form of favourites mechanism. However, the purpose of this approach is to be able to split data between different contexts, for example, a split between one’s work and private files for legal reasons. Worth noting that, this definition of a Space is semantically different from a Space in Section 5.3 where I called the contents of a folder a Space in my sketch.

Consequently, the main window of the application is a (standard) list offering the selection of a Space to go to. Worth noting that, even though I call this the main window in my concept it is just another setup screen. Meaning, when the System User for instance switches devices while in a Space they are still at this position when they start using the other device (*location independence*, see user story ID 18). Only when they exit a Space and (re-)open the application they would start at the main window.

Figure 5.16 shows this main window for the mobile phone. At the top right showing the option for a search function and to sort, for example, according to Space names. Albeit not a focus of my design it is still crucial to consider those features for a potential future integration (see *reduction*, user story ID 19 and *improve search*, user story ID 11). The (standard) list shows the available Spaces to go to and according meta data, for example, the last change date and the number of items in this Space.

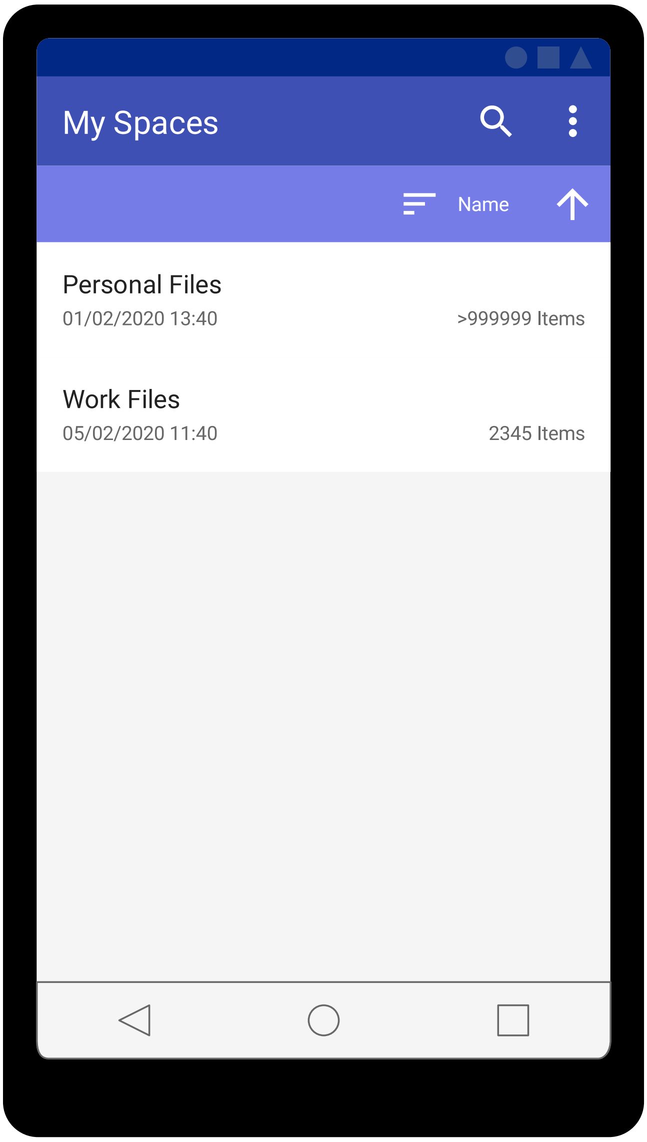



Figure 5.16: Wireframe Mobile Main

*Note*. Wireframe outlining the main window of the application. Designed for a mobile phone with a screen resolution of 360x640 pixels at 160 ppi. Wireframe created using Affinity Designer on Windows. Changed "*files*" to "*Files*" in this version of the wireframe for consistency.

The next wireframe Figure 5.17 shows a contextual overflow menu when the icon at the top right (three vertical dots) gets selected. It offers the following options:

- "*Edit*": Change the setup for an existing Space (see Figure 5.18).
- "*Create Space*": Create a new Space (s) (see Figure 5.21).

- "*Recycle Bin*": Go to a recycle bin which lists recently deleted Spaces with options to restore them or to empty the recycle bin. Worth noting that, the concept of a recycle bin was not detailed further as I claim it is a standard functionality.
- "*Settings*": Go to the settings menu (see Figure 5.22).

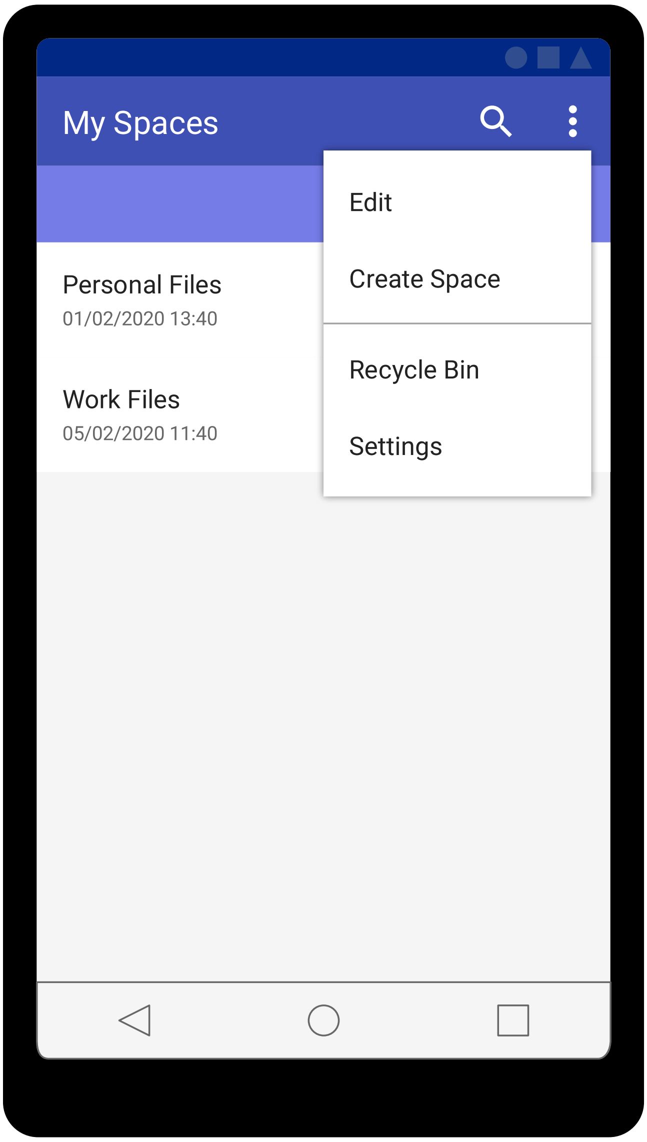


Figure 5.17: Wireframe Mobile Main With Overflow Menu

*Note*. Wireframe outlining the main window of the application and activated overflow menu. Designed for a mobile phone with a screen resolution of 360x640 pixels at 160 ppi. Wireframe created using Affinity Designer on Windows. Changed "*files*" to "*Files*" in this version of the wireframe for consistency.

Figure 5.18 visualises the edit mode for Spaces. Sort features get disabled, at the top left all items can be selected using the checkbox. If only one Space is selected the System User can tap on the pencil icon at the bottom left corner to edit the Space settings (see Figure

5.19). The information icon right beside the pencil shows metadata (dialog) for the selected Spaces while the recycle bin icon shows a dialog to move the selected items to the recycle bin (see Figure 5.20). The back arrow at the bottom right exits the edit mode.

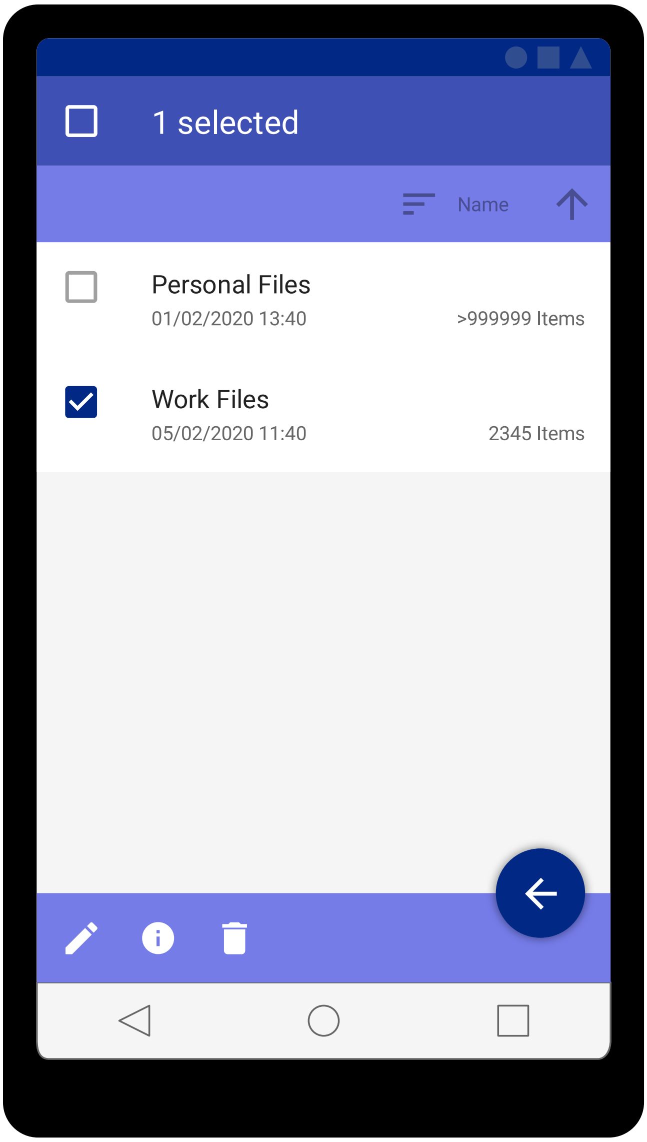



Figure 5.18: Wireframe Mobile Main – Edit

*Note*. Wireframe visualising the Edit mode. Designed for a mobile phone with a screen resolution of 360x640 pixels at 160 ppi. Wireframe created using Affinity Designer on Windows. Changed "*files*" to "*Files*" in this version of the wireframe for consistency.

Figure 5.19 visualises the editing of an existing Space. The System User can change the name for the Space, for example, using the touch keyboard (not shown). They can also change the target **be here now** application server, for example, when the application server moved. Worth noting that, they can only change the application account (disabled) if more than account has been setup. Lastly, they can change where to store the actual files to, for example, Google Drive.

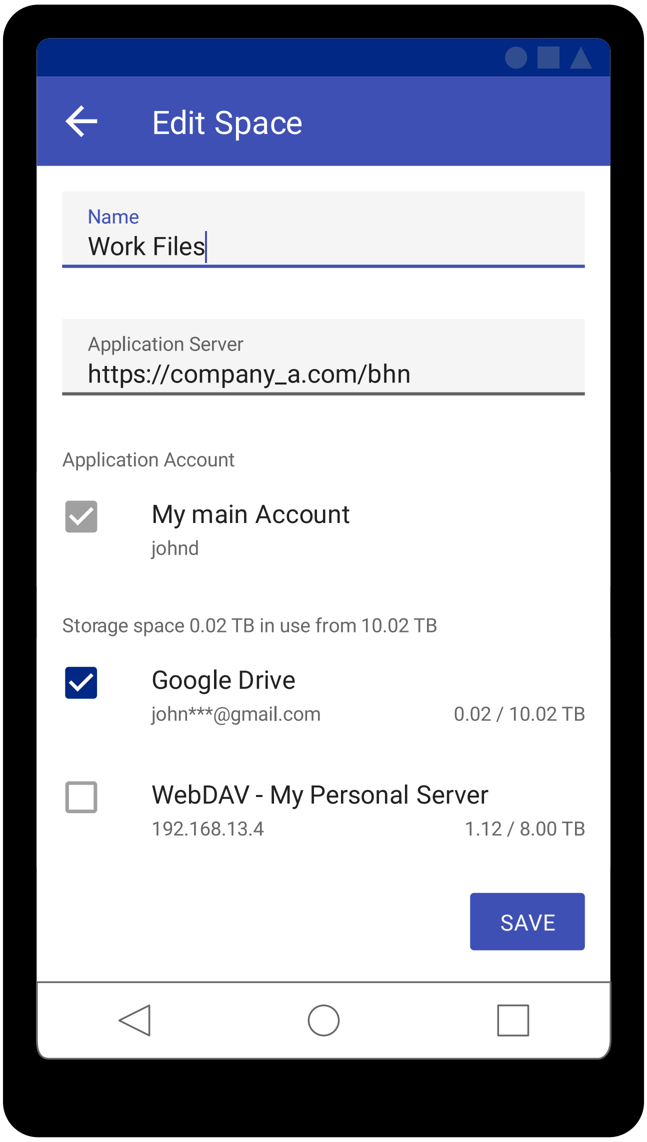



Figure 5.19: Wireframe Mobile Main - Edit - Edit Space

*Note*. Wireframe visualising the form to edit an existing Space. Designed for a mobile phone with a screen resolution of 360x640 pixels at 160 ppi. Wireframe created using Affinity Designer on Windows. Changed "*files*" to "*Files*" in this version of the wireframe for consistency.

Figure 5.20 visualises the dialogs to show the metadata for the selected Spaces (left) and the request to move the selected Spaces to the recycle bin (right).

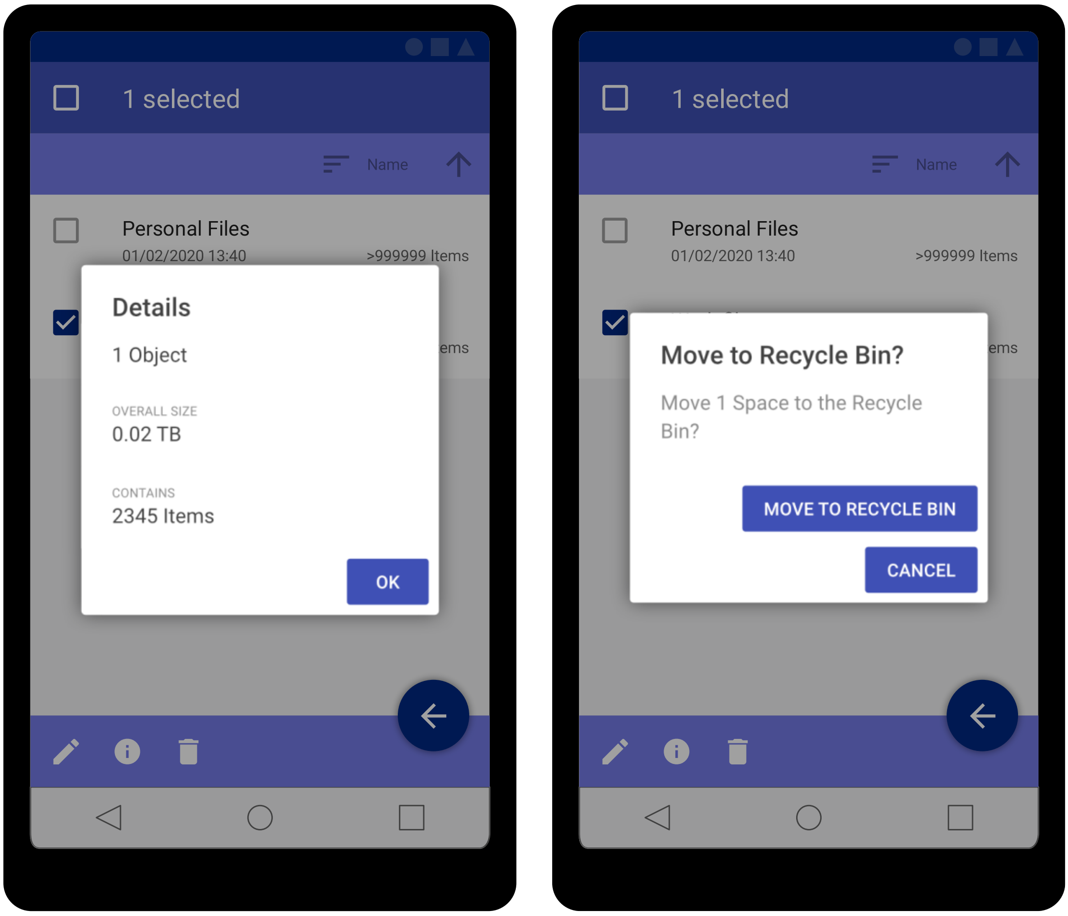



Figure 5.20: Wireframes Mobile Main - Edit - Dialogs

*Note*. Wireframes visualising dialogs to show Space(s) metadata and to move them to the recycle bin. Designed for a mobile phone with a screen resolution of 360x640 pixels at 160 ppi. Wireframes created using Affinity Designer on Windows. Changed "*files*" to "*Files*" in this version of the wireframe for consistency.

Figure 5.21 demonstrates the form how to create a new Space. The usage is identical to the editing form for an existing Space (see Figure 5.19).

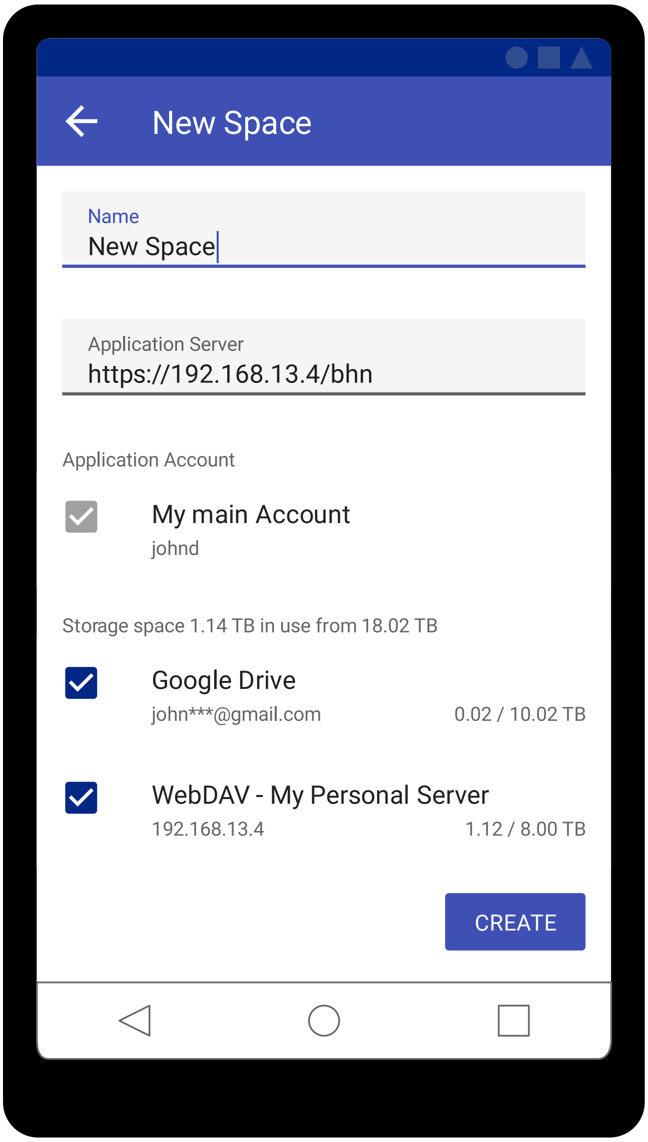



Figure 5.21: Wireframe Mobile Main – Create Space

*Note*. Wireframe visualising the form to create a new Space. Designed for a mobile phone with a screen resolution of 360x640 pixels at 160 ppi. Wireframe created using Affinity Designer on Windows.

Figure 5.22 visualises a subset of the available settings. At the top are sections to add or edit either cloud or individual storage locations, for example, a Google Drive account. Below this is a section to add or edit the application accounts used to authenticate against a **be here now** application server. At the bottom the System User can change the icons and colour scheme the application uses (*bias*, see user story ID 6). Worth noting that, that some settings are not visualised, for example, concerning the "*Related Content*" (Things of interest) feature. In detail, these settings are (scroll or swipe down):

- "*Delete Spaces in Recycle Bin after 30 days*" (toggle switch).
- "*Delete Files, Folders (in Spaces) in Recycle Bin after 30 days*" (toggle switch).
- "*Show all Related Content and History in 3D Overview mode*" (toggle switch).
- "*Show preview of Node Layers for 2D Navigation*" (toggle switch).
- "*Show preview of Node Layers for 3D Overview*" (toggle switch).
- "*Set number of preview Node Layers to show*" (Slider - Integer - 1 to N).
- "*Required application authorisations*" (on select show dialog with a view only).
- "*About the application*" (on select show dialog with a view only). I want to highlight that, the application always has to use mobile data if a local network is not available.

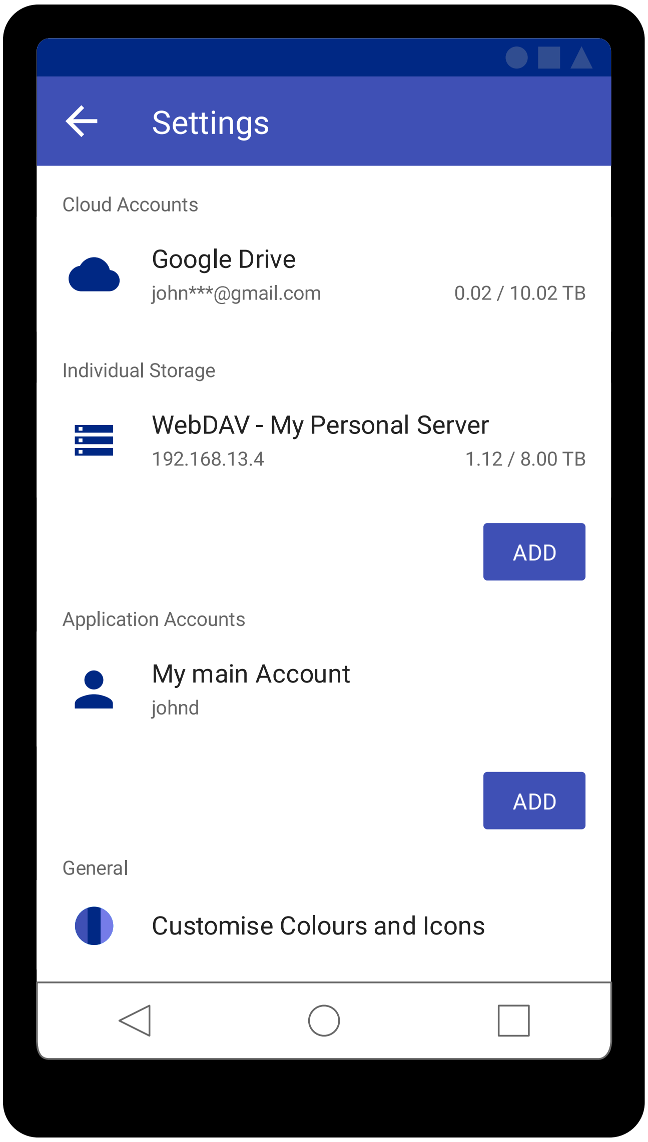



Figure 5.22: Wireframe Mobile Main – Settings

*Note*. Wireframe visualising a subset of possible application settings. Designed for a mobile phone with a screen resolution of 360x640 pixels at 160 ppi. Wireframe created using Affinity Designer on Windows.

Figure 5.23 shows that the System User can tap on a Space to enter 2D Navigation mode for this set of data. Worth noting that, such a select always opens the Space in 2D Navigation mode regardless of the mode last used.

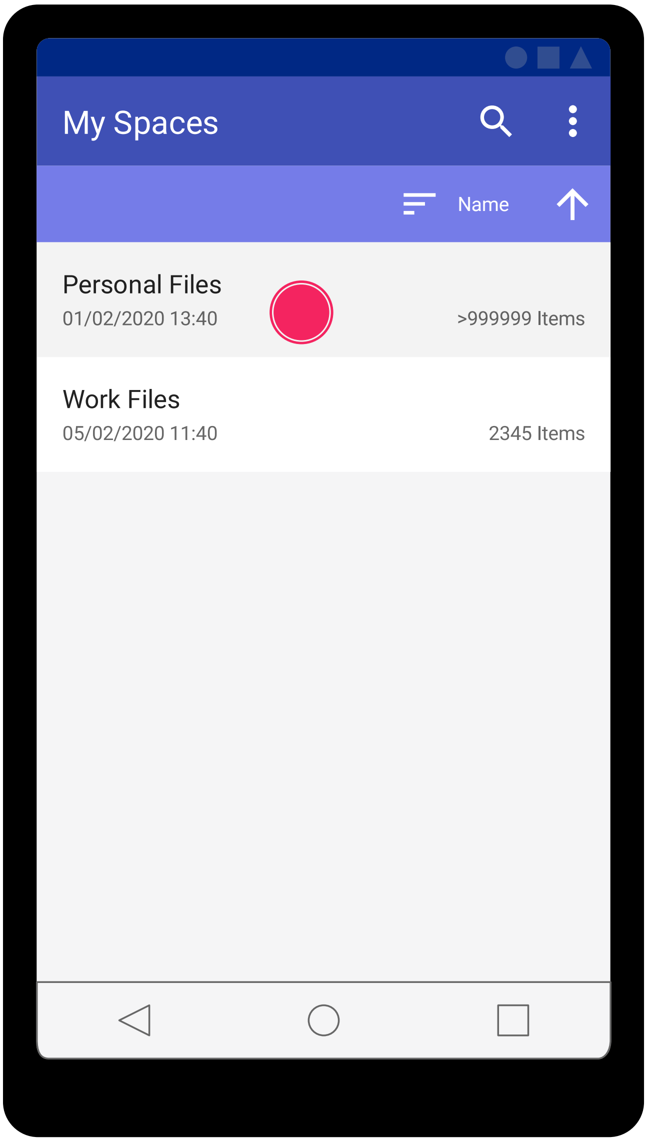



Figure 5.23: Wireframe Mobile Main - Select Space

*Note*. Wireframe outlining the selection of a Space in the main window. Designed for a mobile phone with a screen resolution of 360x640 pixels at 160 ppi. Wireframe created using Affinity Designer on Windows. Changed "*files*" to "*Files*" in this version of the wireframe for consistency.

Figure 5.24 visualises a Space after selecting it (see Figure 5.23). Although it may appear as a standard list this 2D Navigation mode is actually just utilising a fixed camera view on my node layout (see Section 5.5.2). At the top of the screen the name of the Space is shown, for example, "*Personal Files*". The System User can exit the Space by tapping on the back arrow at the top left corner. When moving deeper in the hierarchy the back arrow jumps back to the last folder as a form of *user history* (see user story ID 4). When the home symbol is selected (top left) the view is reset to the root node showing the linked nodes at depth one. Hence, the System User is brought back to this starting screen for the current Space. At the top right the user can start searching within the data of the current Space (see *improve search*, user story ID 11). Below the search icon they also have the option to sort the data, for example, by name (see *reduction*, user story ID 19). To the left from the sort feature the System User can set or unset a folder (node) as a favourite by selecting the star icon (see *favourites and multiple instances*, user story ID 22). At the top right corner an overflow menu can be shown offering various options (see Figure 5.27). The bottom app bar can be used to move to another Node Layer by tapping on the respective number or the arrow symbols (forward or back). The System User can also switch Node Layers by swiping left or right using one finger. A node (file or folder) gets selected by tapping on it once (on a mobile device).

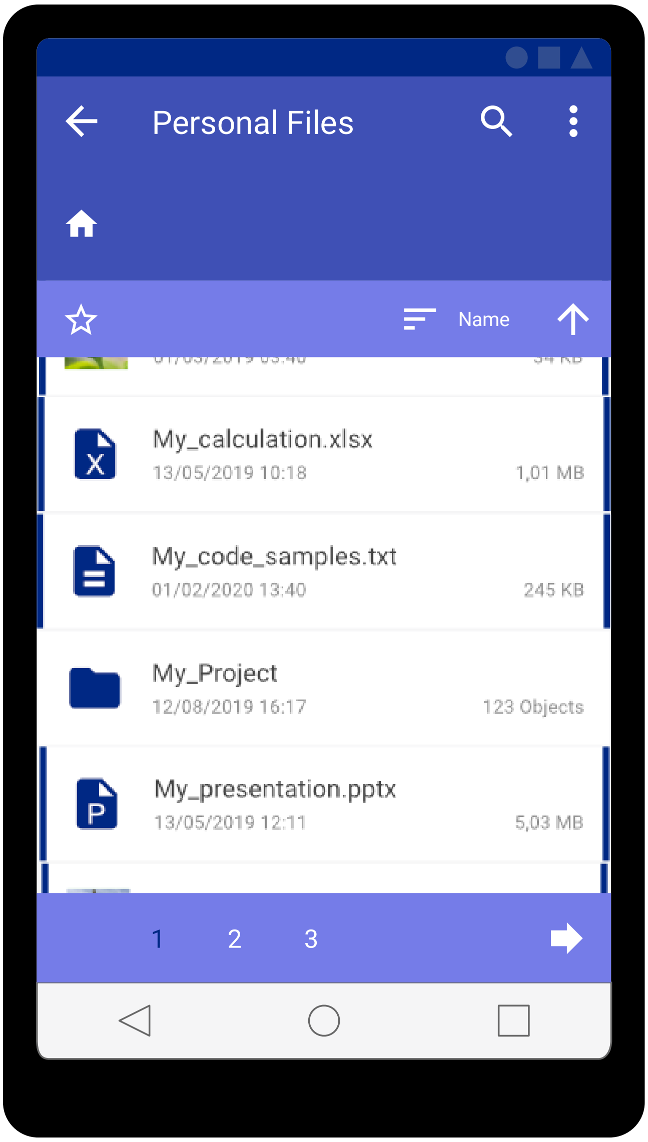



Figure 5.24: Wireframe 2D Navigation (Space)

*Note*. Wireframe outlining the 2D navigation mode for one Space. Designed for a mobile phone with a screen resolution of 360x640 pixels at 160 ppi. Wireframe created using Affinity Designer on Windows.

Next, Figure 5.25 shows the History (see Section 4.6.2) feature to support *additional relations* (see user story ID 16). When there are older versions available for a leaf node (file) a blue ribbon on the left side of this node is shown. Worth noting that, I considered the importance of the *size of touch targets* by adjusting the dimensions of the actual touch targets for those ribbons. Meaning, the width of the touch targets for those ribbons is actually larger than visualised. Precisely, the touch target is 48dp wide to allow for touch functionality. The actual ribbon is placed on the left or right side within this touch target. This decision was informed by the recommendations from Google (n.d.-g). Continuing, when the System User selects this ribbon on the left, the last version of this file gets displayed. This new view is actually just a camera shift to the left along the x-axis. When the setting to show preview Node Layers in 2D Navigation mode is enabled those layers are still shown behind the version of the file in focus. However, all interaction to either switch to Node Layers is disabled when using this History feature. The same applies to the Related Content (see Figure 5.26) feature.

In History mode (right side) a file (version) can be opened by tapping on it. At the top of the screen the title of file is displayed. Tapping on the blue ribbon on the right or the back button (upper left) exits the History mode or allows browsing through the versions (move back to the last version). Below the title of the file the date of the version is displayed. To the left of this date the System User can tap on the close icon to exit the History mode at any time. Tapping on the clock symbol below (upper left) shows a confirmation dialog to restore the version (dialog not detailed). When the recycle bin icon gets selected a confirmation dialog is displayed with a request to delete this version of the file. If confirmed the version will be deleted immediately (no recycle bin). While in History mode the sort function is disabled.

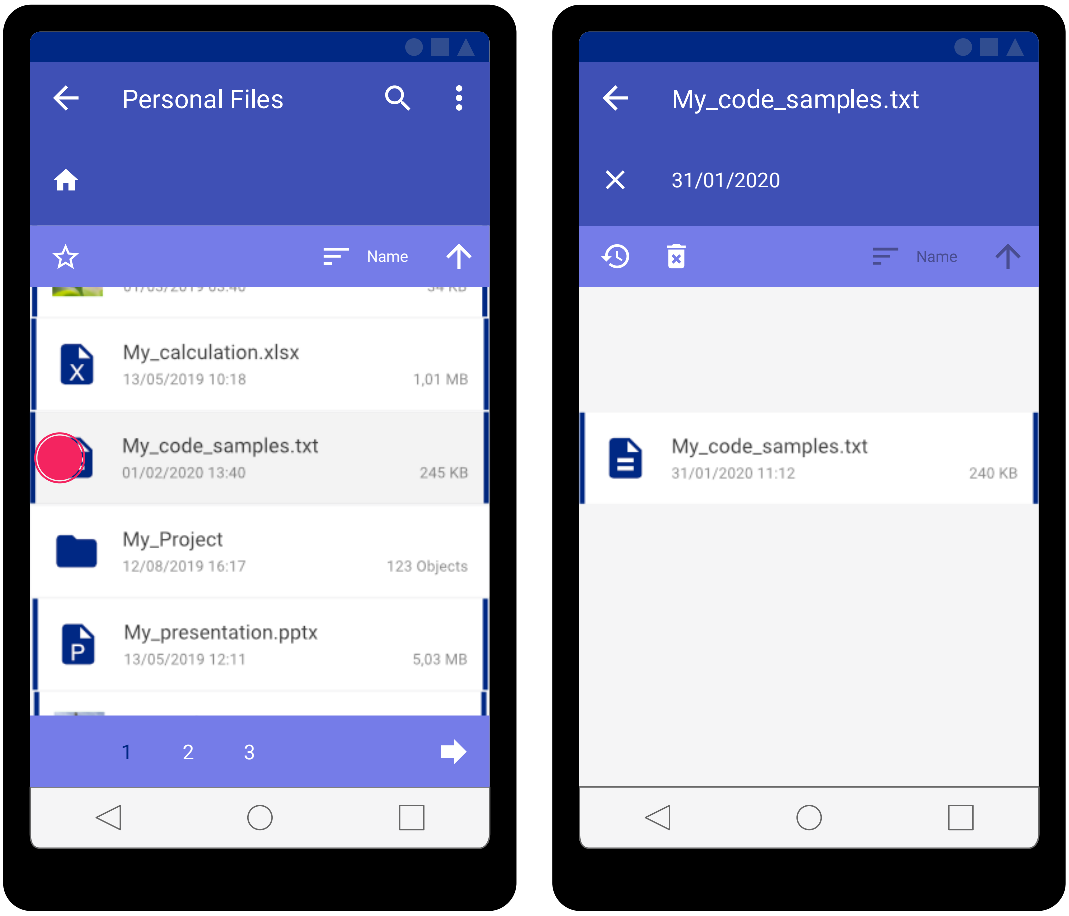



Figure 5.25: Wireframes 2D Navigation (Space) – History

*Note*. Wireframes outlining the History mode feature in 2D Navigation mode. Designed for a mobile phone with a screen resolution of 360x640 pixels at 160 ppi. Wireframes created using Affinity Designer on Windows.

Figure 5.26 visualises the Related Content feature (*relate and analyse*, see user story ID 20). When Related Content is available for a file, for example, a (word) similar document a blue ribbon to the right of the file gets shown. The System User can tap on this ribbon to enter Related Content mode.

In this mode (right side) a related file can be opened by tapping on it. Tapping on the blue ribbon on the left or the back arrow symbol (upper left) exits the Related Content mode. Below the back arrow symbol the home icon and path of the related file is visualised. The System User can interact with the path to exit the mode and jump directly to a subfolder. Also, they can select the home symbol (top left) to exit the mode and move back to the root node showing the linked nodes at depth one. A tap on the folder icon (upper left) moves to the folder location of the Related Content and centres the view on the file, for example, Final_project_calculation.xlsx. The sort feature is disabled while using this mode.

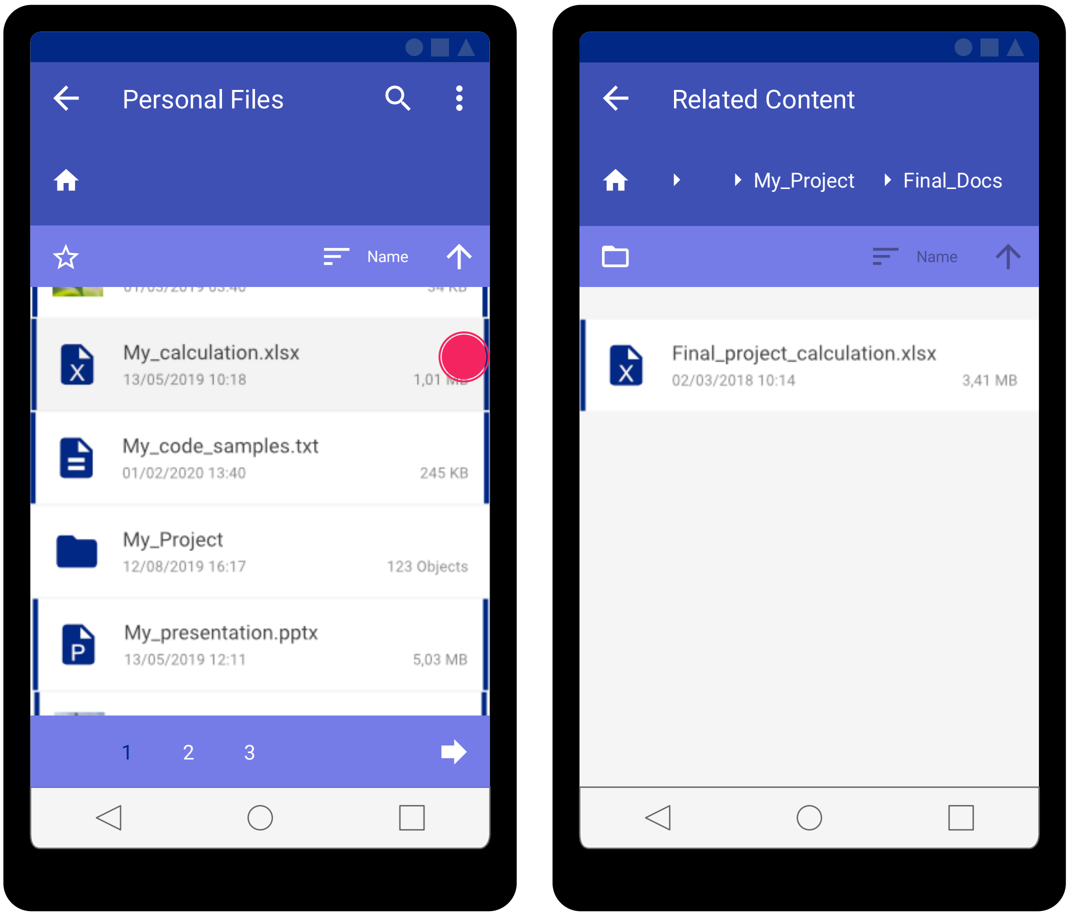



Figure 5.26: Wireframes 2D Navigation (Space) – Related Content

*Note*. Wireframes outlining the Related Content mode feature in 2D Navigation mode. Designed for a mobile phone with a screen resolution of 360x640 pixels at 160 ppi. Wireframes created using Affinity Designer on Windows.

The next wireframe Figure 5.27 shows a contextual overflow menu when the icon at the top right (three vertical dots) gets selected. It offers the following options:

- "*Edit*": Editing files and folders in a Space (not detailed for 2D Navigation mode, similar usage to Figure 5.18, for 3D Overview mode see Figure 5.34).

- "*Create Folder*": Create a new folder in this Space (see Figure 5.28).
- "*Enter Overview*": Switch to 3D Overview mode (see Figure 5.29).
- "*Favorites*": Go to a (standard) list of favourites to maintain them, for example, delete favourites (not detailed, standard functionality and not a core user story, see user story ID 22).
- "*Recycle Bin*": Go to a recycle bin which lists recently deleted files and folders with options to restore them or to empty the recycle bin. Worth noting that, the concept of a recycle bin was not detailed further as I claim it is a standard functionality.
- "*Settings*": Go to the settings menu (see Figure 5.22).

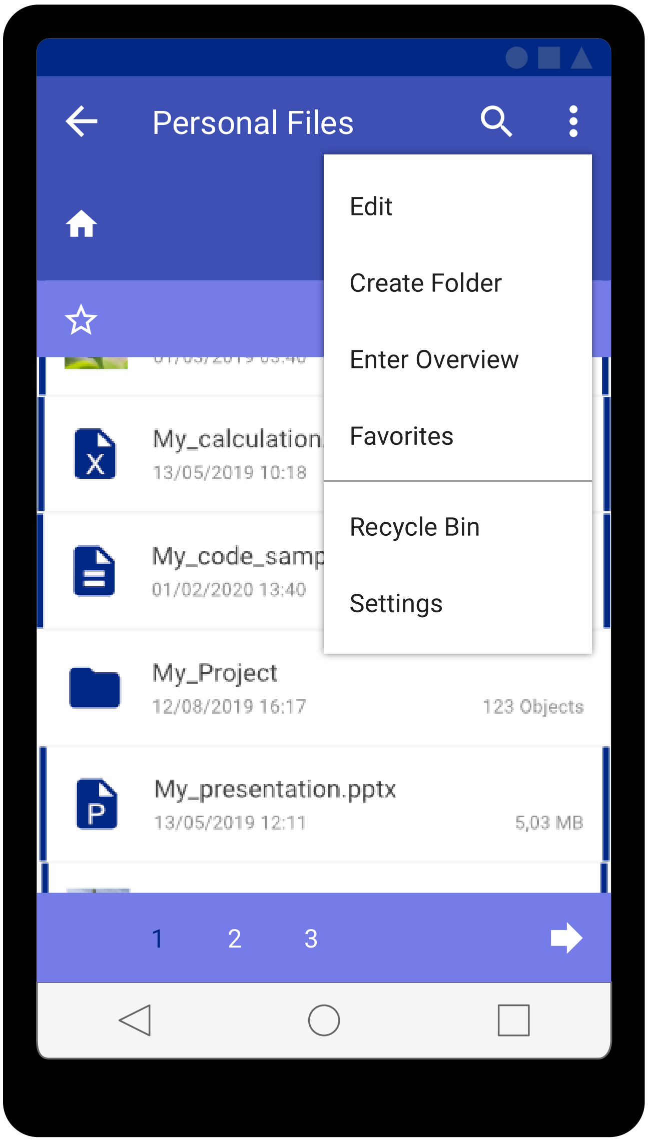


Figure 5.27: Wireframe 2D Navigation (Space) with Overflow Menu

*Note*. Wireframe outlining the 2D Navigation mode with an activated overflow menu. Designed for a mobile phone with a screen resolution of 360x640 pixels at 160 ppi. Wireframe created using Affinity Designer on Windows.

Next, Figure 5.28 shows how to create a new folder. The System User can tap on the option to create a folder (left side). Consequently, has to provide a folder name using, for example, a touch keyboard (middle). The folder gets created and in this example is put to the end of the list (right side). Because all of the other three Node Layers are already full a new fourth Node Layer gets created with the new node (folder) at its centre (see Section 5.5.2).

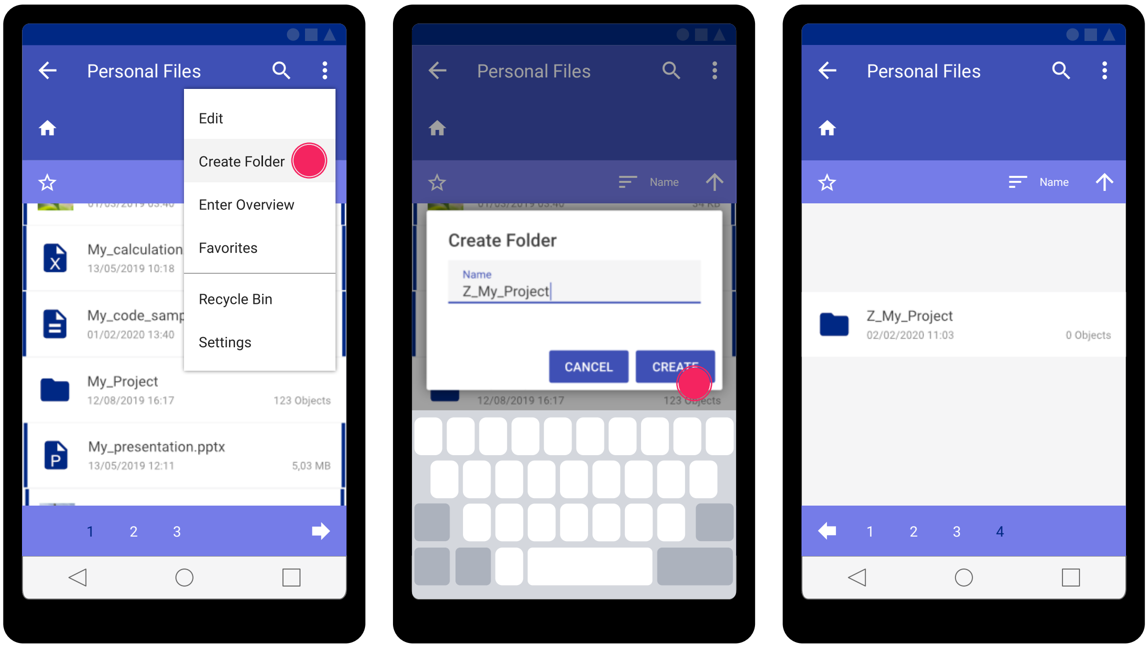



Figure 5.28: Wireframes 2D Navigation (Space) - Create Folder

*Note*. Wireframe visualising the creation of a new folder in 2D Navigation mode. Designed for a mobile phone with a screen resolution of 360x640 pixels at 160 ppi. Wireframes created using Affinity Designer on Windows.

Figure 5.29 shows the 3D Overview mode for a Space. The camera automatically moves away from the current dataset to provide an instant overview (see *automatic zoom in 3D Overview* in 6.2). To improve visibility and therefore overview (*location independence*, see user story ID 18) the nodes are reduced to their icons which are also rendered larger than in 2D Navigation mode. Precisely, each node icon or thumbnail was scaled from 40dp (40x40) in 2D Navigation mode to 56dp (56x56) based on their bounding box. Continuing, the System User navigates in a first-person mode using touch controls. The virtual joy stick at the bottom left control's camera movement while the bottom right joy stick can be used to change the view direction. The primary action button at the bottom right triggers object selection (open file, go to folder) while the smaller button is a shortcut to the overflow menu (top right, see Figure 5.27). In this mode all nodes (files, folders) are rotated towards the System User for every frame. Therefore, it cannot happen that the System User moves behind the data and loses orientation.

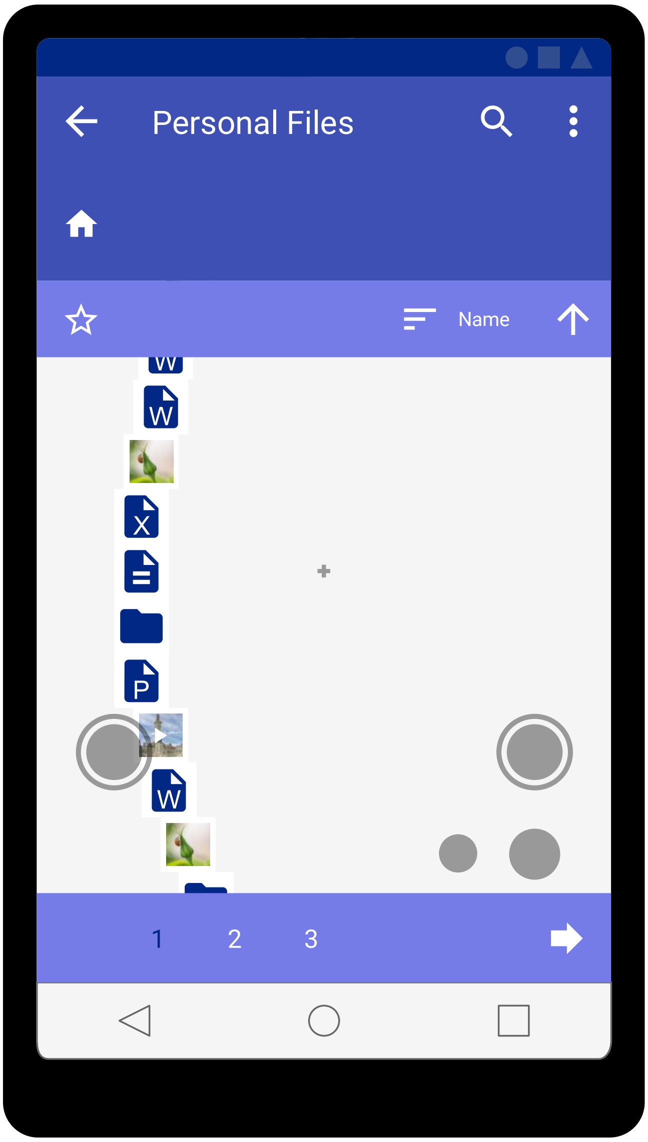



Figure 5.29: Wireframe 3D Overview (Space) - Enter Overview

*Note*. Wireframe outlining the 3D Overview mode right after switching to it from 2D Navigation mode. Designed for a mobile phone with a screen resolution of 360x640 pixels at 160 ppi. Wireframe created using Affinity Designer on Windows.

Again, Figure 5.30 shows the 3D Overview mode but this time with the option enabled to show a preview of Node Layers for 3D Overview (see descriptions for Figure 5.22). When the crosshair is moved over an inactive Node Layer the System User can switch to this Layer by tapping on the primary action button (bottom right). Alternatively, one can still utilise the controls at the bottom of the screen to switch layers or swipe left and right with one finger.

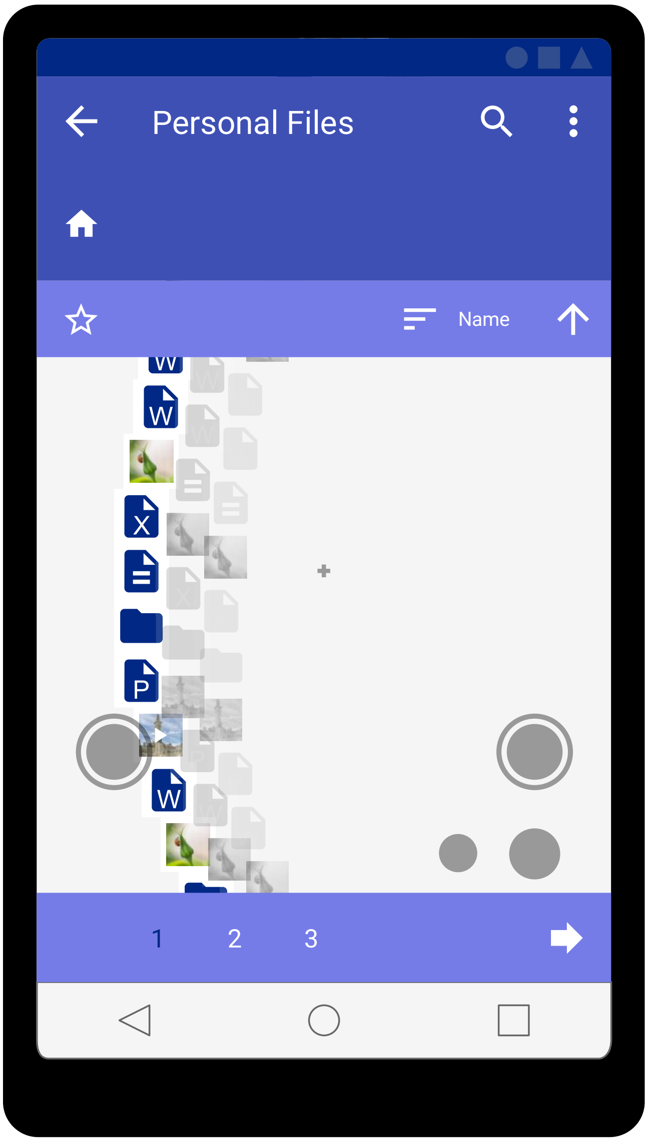



Figure 5.30: Wireframe 3D Overview (Space) - Preview of Node Layers for 3D Overview Enabled

*Note*. Wireframe outlining the 3D Overview mode right after switching to it from 2D Navigation mode. In this wireframe the option to show preview Node Layers for 3D Overview is enabled. Designed for a mobile phone with a screen resolution of 360x640 pixels at 160 ppi. Wireframe created using Affinity Designer on Windows.

Figure 5.31 highlights the navigation mechanism in 3D Overview mode. When the System User navigates so the crosshair is above a node (file or folder) a pop up window gets displayed (details-on-demand, *dynamic scaling in 3D Overview* see Section 6.2). A node (file or folder) gets selected by tapping on the centre area of the pop up window once or by tapping on the primary action button (bottom right).

A tap on the blue ribbons shows the corresponding content, for example, an older version of a file. Also, the camera is automatically moved to the location of this node. When the crosshair is moved away from this row of content (x-axis), for example, so another node (folder, file) is in focus the content is hidden again. When the System User exits out of the respective mode using the 2D controls, for example, using the back arrow at the top left when in Related Content mode (see Figure 5.26) the crosshair automatically moves back to the origin file. However, when the crosshair as described is moved out of bounds the nodes (Related Content or History) are just hidden without automatically moving the camera.

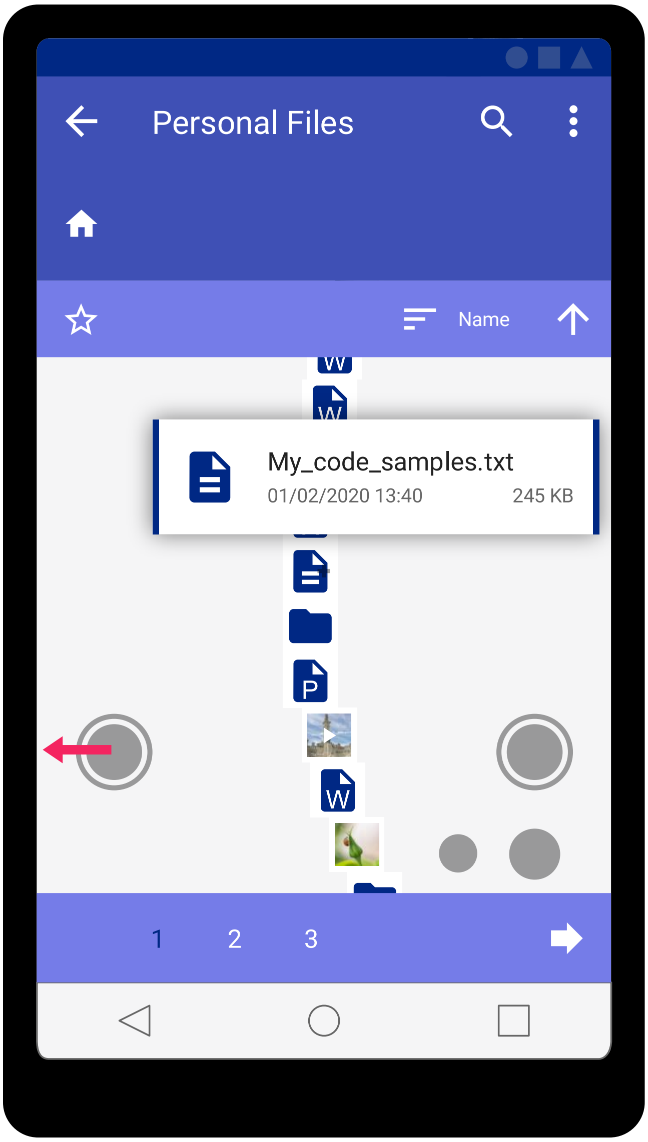



Figure 5.31: Wireframe 3D Overview (Space) – Navigation

*Note*. Wireframe outlining navigation in 3D Overview mode. In this wireframe the option to show all Related Content and History in 3D Overview mode is disabled. Designed for a mobile phone with a screen resolution of 360x640 pixels at 160 ppi. Wireframe created using Affinity Designer on Windows.

Figure 5.32 visualises the continuous zoom in 3D Overview mode. The System User moves away from the data to see more nodes (files and folders) at once (*location independence*, see user story ID 18).

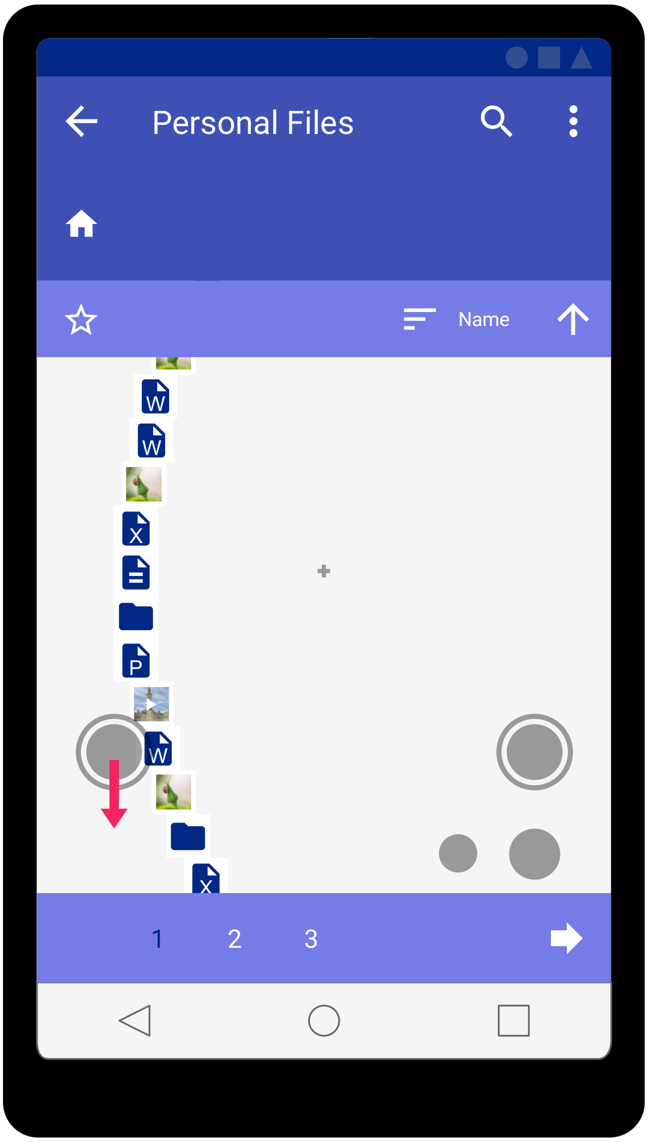



Figure 5.32: Wireframe 3D Overview (Space) - Navigation - Zoom

*Note*. Wireframe outlining navigation in 3D Overview mode highlighting a zoom out operation. Designed for a mobile phone with a screen resolution of 360x640 pixels at 160 ppi. Wireframe created using Affinity Designer on Windows.

Figure 5.33 shows the Related Content mode in 3D Overview. In this example, the setting to show all Related Content and History in 3D Overview mode is enabled (see descriptions for Figure 5.22). When the controls for browsing are utilised, for example, the back arrow at the top left corner is tapped on the camera moves to the respective parent node along the x-axis. In the shown example the camera would move back to the file

"*My_calculation.xlsx*" (see Figure 5.26). For the History mode (see Figure 5.25) the camera would move back or forward from version to version along the x-axis.

Worth noting that, one can imagine scenarios for development where this can pose a challenge. Meaning, I would utilise a *linked list* to develop a feature to allow the traversal for the last user actions or positions, for example, to move back to the last visited folder. However, for the visualised scenario this could lead to missing links, for example, when the System User moves from one Related Content file to another. Meaning, one may have to evaluate utilising either multiple linked lists or more complex data structures for developing the described forms of *user history* (see user story ID 4) features.

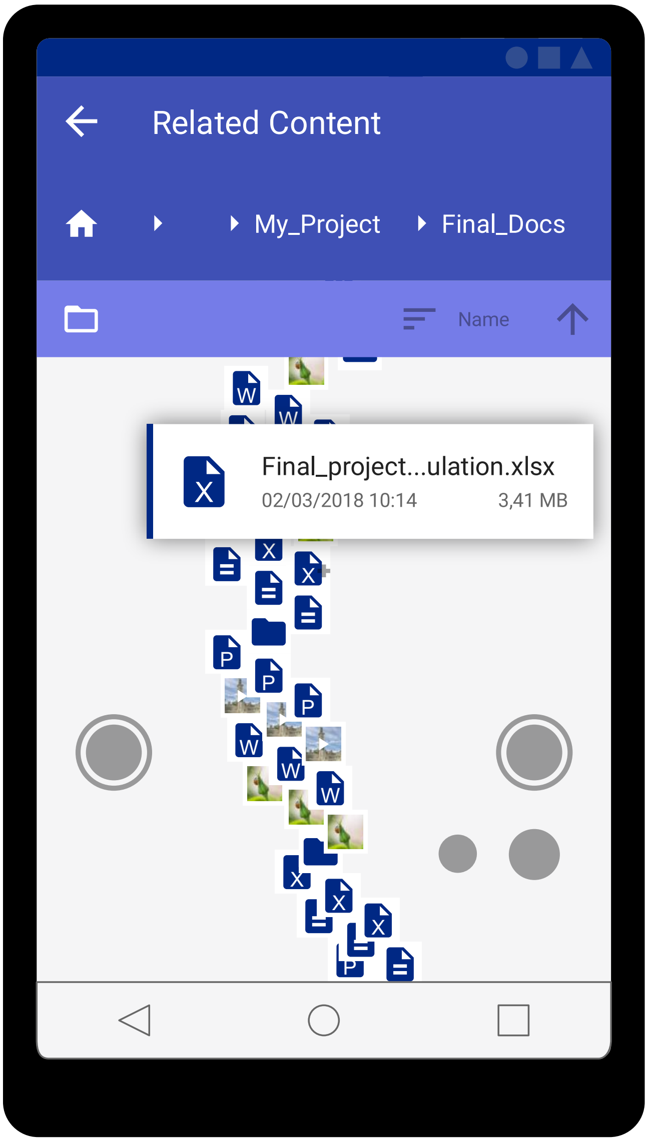



Figure 5.33: Wireframe 3D Overview (Space) - Navigation - Zoom - Show all Related Content and History

*Note*. Wireframe outlining navigation in 3D Overview mode (zoomed out view). In this wireframe the option to show all Related Content and History in 3D Overview mode is enabled. Designed for a mobile phone with a screen resolution of 360x640 pixels at 160 ppi. Wireframe created using Affinity Designer on Windows.

Figure 5.34 visualises the overflow menu (left side) and edit options (right side) in 3D Overview mode. The changed overflow menu options in comparison to 2D Navigation mode (see Figure 5.27) are:

- "*Exit Overview*": Switch to 2D Navigation mode for the current folder location and Node Layer.
- "*Lock Folder Path*": Render an arc between folder locations to support orientation (see Figure 5.35). Changes to "*Unlock Folder Path*" after selection to hide path again.

Regarding the editing options, the System User can select all nodes in one folder using the checkbox at the top left corner. They can exit the edit mode by tapping on the X icon below the checkbox (top left). The options to set or unset a favourite and sort features are disabled in edit mode. The System User can utilise the touch controls (virtual joysticks) and primary action button to select nodes in scope for editing. They can also switch Node Layers using the one finger touch gesture. When the setting to show a preview of Node Layers for 3D Overview (see details for Figure 5.22) is enabled they can also move the crosshair over an inactive Node Layer and tap on the primary action button (bottom right) to switch layers. However, the System User cannot change folders in edit mode. The icons at the bottom offer the following edit operations (left to right):

- "*Pencil*": If only a single folder is selected one can change the name of this folder (similar to Figure 5.28).
- "*Copy*": Copies the selected items to the clipboard. The icon at this location changes to a paste symbol afterwards.
- "*Scissors*": Cuts the selected items to the clipboard. Consequently, the icon changes to a paste symbol.
- "*Info*": Shows metadata for the selected items in a dialog (similar to Figure 5.20): number of items, overall size including items in folders.

- “*Recycle bin*”: Shows a dialog to move selected items to the Recycle Bin (similar to Figure 5.20).

When the System User selected either the copy or the scissor (cut) icon the other one gets temporarily disabled. They have the option to tap on the paste icon right away which shows a dialog asking them to confirm their intention, for example, “*Do you want to move 5 items to this location?*”. Alternatively, they can also exit the edit mode and move to a different target location folder to insert the content there by re-entering edit mode. If they want to clear the clipboard, they can long press on the paste symbol or the deactivated option (copy or scissors symbol). This results in a new dialog asking for confirmation if they want to empty the clipboard.

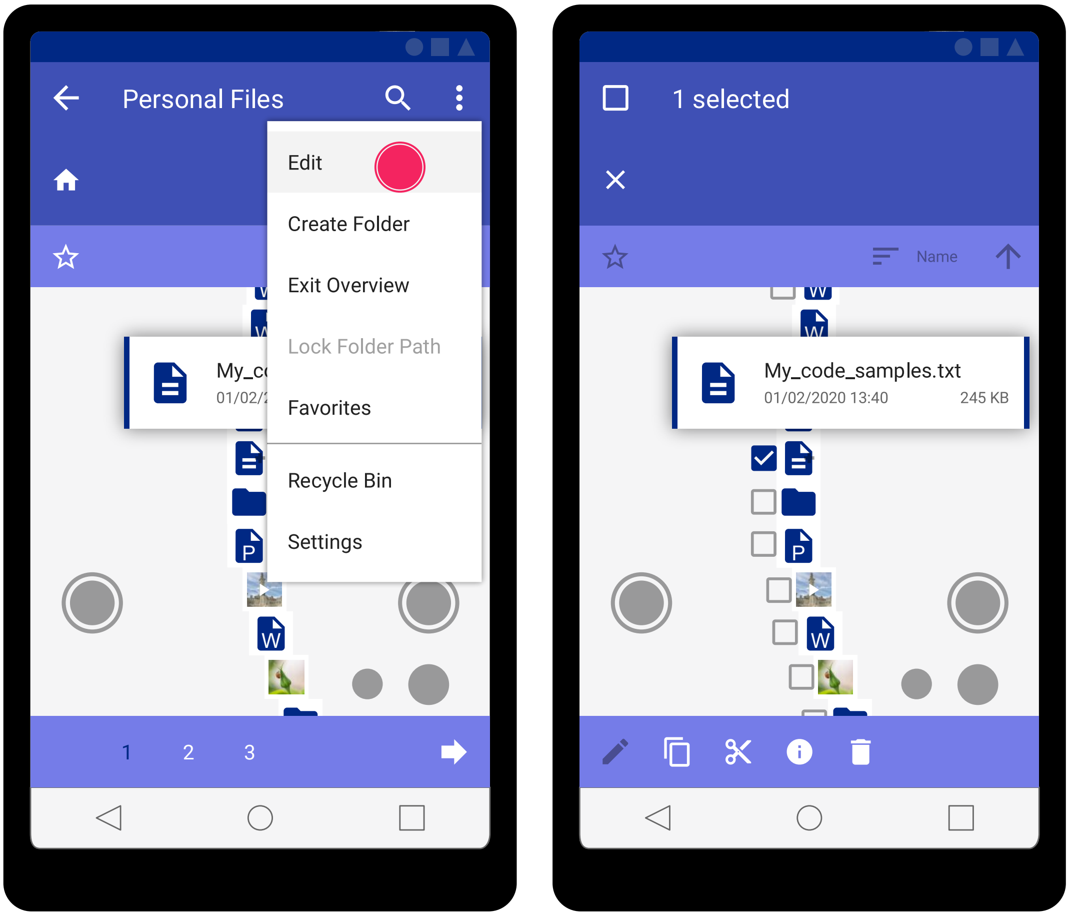



Figure 5.34: Wireframes 3D Overview (Space) - Navigation – Overflow Menu - Edit

*Note*. Wireframes outlining the overflow menu in 3D Overview mode and possible file and folder operations. Designed for a mobile phone with a screen resolution of 360x640 pixels at 160 ppi. Wireframes created using Affinity Designer on Windows.

Figure 5.35 visualises the movement between folders in 3D Overview mode. When a System User moves the crosshair over a folder and selects it, for example, by tapping on the primary action button (bottom right) the camera automatically moves to the target location along the z and y-axis (left side). Alternatively, they can open the overflow menu and select to show a link connecting the folder to its respective content (Node Layer) (middle). Consequently, they can follow this path manually using the virtual joysticks (right side). This link or path is shown until they decide to hide it again by opening the overflow menu and selecting the option Unlock Folder Path.

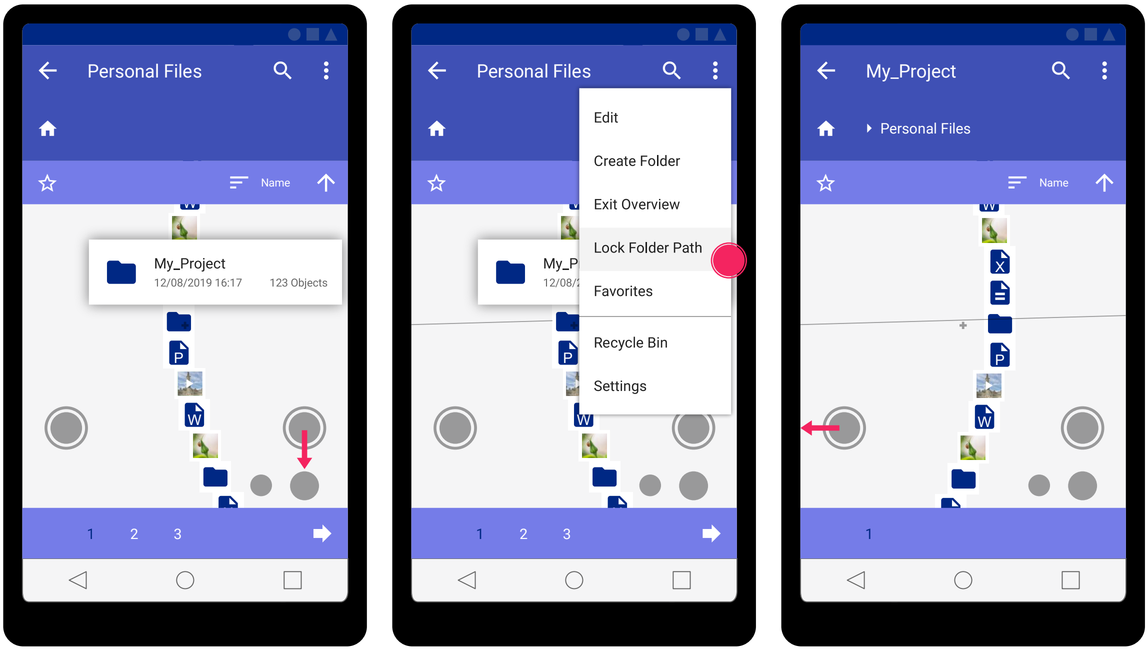



Figure 5.35: Wireframes 3D Overview (Space) - Navigation - Folders

*Note*. Wireframes visualising navigation between folders in 3D Overview mode. Designed for a mobile phone with a screen resolution of 360x640 pixels at 160 ppi. Wireframes created using Affinity Designer on Windows.

Figure 5.36 shows the transition to the tablet device. To fit a maximum of 60 uppercase characters the width for one node, for example, one file is scaled to a width of 640dp. The touch controls are the same as for the mobile phone (see above). Also, the visual appeal is the same for *device independence* (see user story ID 17). Hence, when the setting to show a preview of Node Layers for 2D Navigation is enabled the System User can see more nodes at once (*location independence*, see user story ID 18). Hence, I claim this provides a form of context+focus. The System User can also tap on an inactive preview Node Layer directly to switch to it.

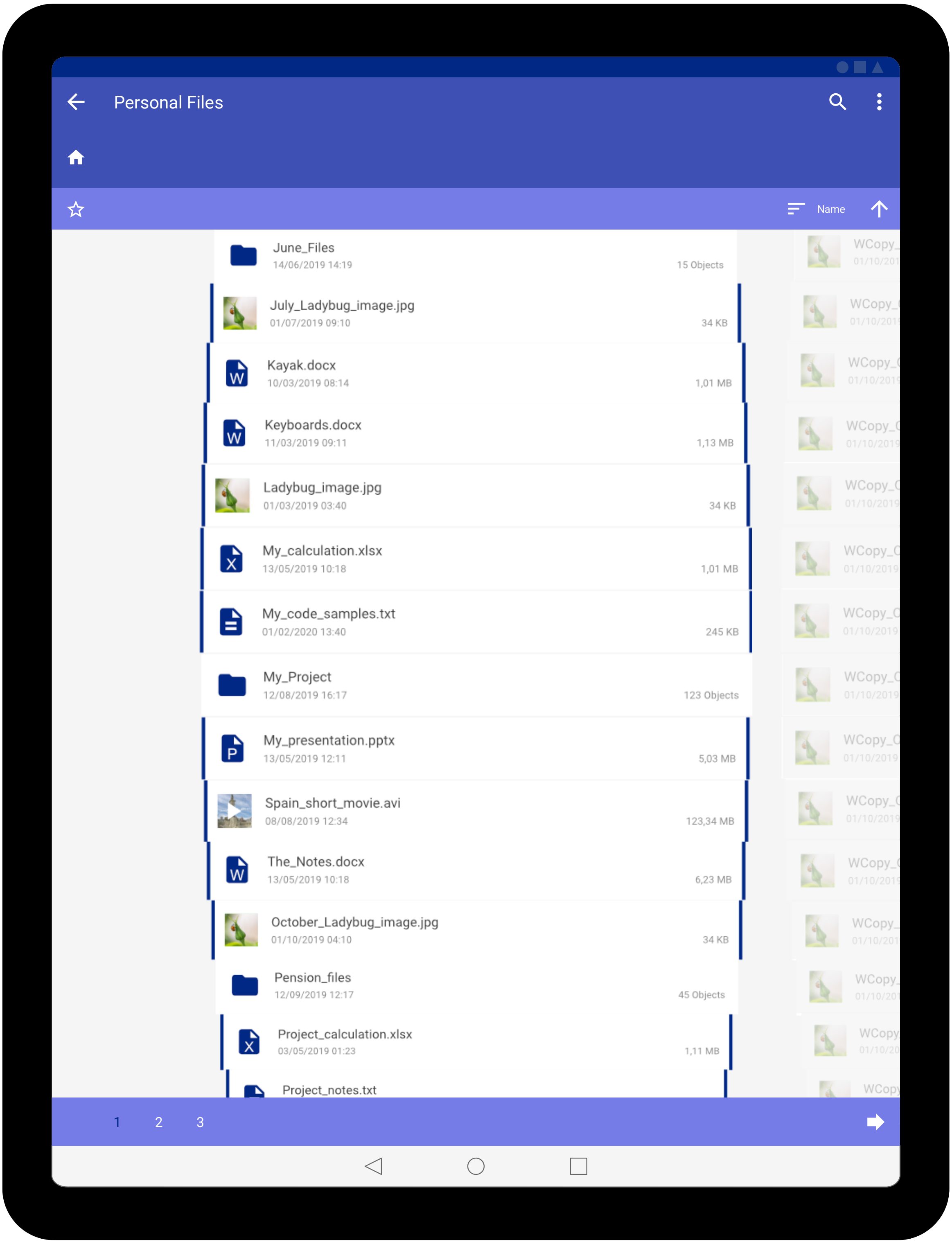



Figure 5.36: Wireframe 2D Navigation (Space) – Tablet Device

*Note*. Wireframe visualising 2D Navigation mode on a tablet device. Designed for a tablet with a screen resolution of 982x1309 pixels at 160 ppi. Wireframe created using Affinity Designer on Windows.

Consequently, Figure 5.37 shows the transition to the PC desktop. Similarly, to Figure 5.36 the setting to show a preview of Node Layers for 2D Navigation is enabled. Therefore, the System User can see even more nodes at once (*location independence*, see user story ID 18). The touch controls are the same as for the mobile phone. When a mouse or touchpad is used a double click opens a file or folder (node). This is true for all modes, for example, in 2D Navigation as well as in 3D Overview mode. The contextual menu can (also) be opened, for example, by using the right mouse button or a two finger tap on a touch pad. In comparison to Figure 5.27 it is possible to not only create a folder but also documents. Also, in edit mode (see Figure 5.34) the System User can change the name of files (pencil icon).

When using a mouse or touch pad and a keyboard the view direction in 3D Overview mode is controlled using the mouse while the arrow keys on the keyboard control movement. In 3D Overview mode the System User can unlock the mouse cursor by pressing the escape (esc) key, for example, to interact with the pop up window (see Figure 5.31). Clicking on the main window captures the mouse cursor again to continue exploration. The System User can also click on an inactive preview Node Layer directly to switch to it. The System User is also able to utilise common shortcuts, for example, instead of entering edit mode they can directly select multiple files holding down the control (ctrl) key or use drag and drop to move files to a different folder.

Worth noting that, possible other options in the contextual menu were not detailed at this stage to not divert an interviewees attention during evaluation. To exemplify, one may add another option to rename files and folder or options to open files with a certain application. However, this may lead to a detailed discussion about those standardised features which are not the focus of my design.

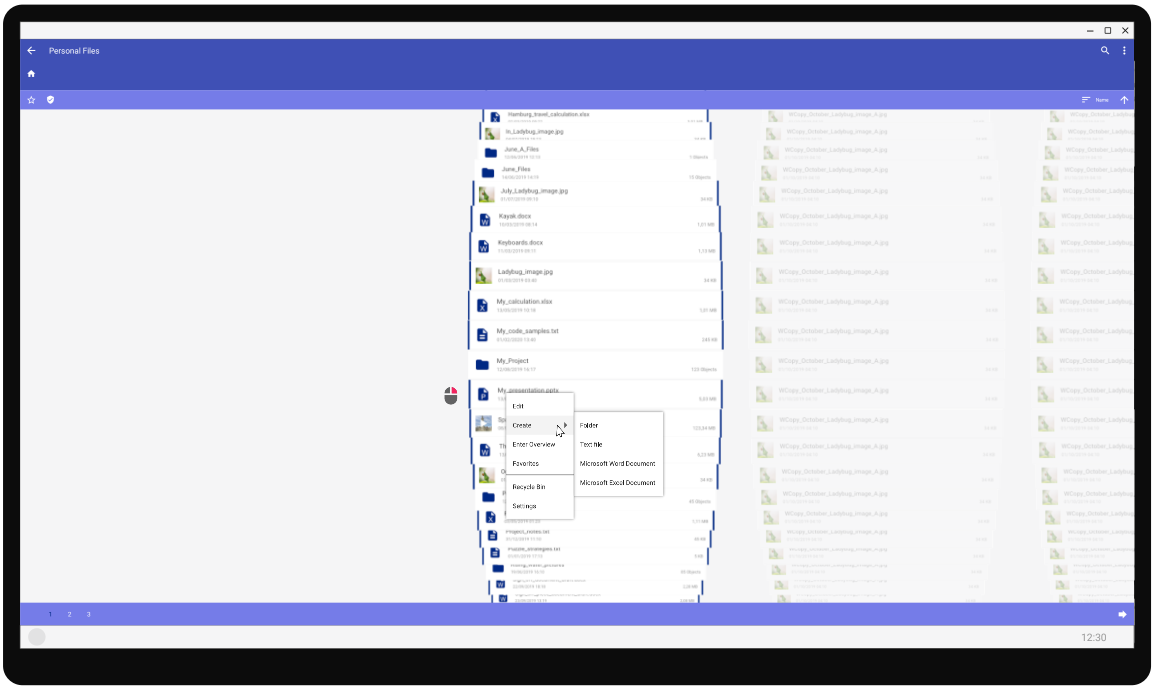



Figure 5.37: Wireframe 2D Navigation (Space) – PC Desktop Device

*Note*. Wireframe visualising 2D Navigation mode on a PC desktop device. Designed for a PC with a screen resolution of 1920x1080 pixels at 110 ppi (zoom for details). Wireframe created using Affinity Designer on Windows.

Figure 5.38 shows the utilisation of multiple instances on PC desktop (see *favourites and multiple instances*, user story ID 22). The System User can set a primary instance by clicking on the shield icon (upper left). This icon is only shown on PC desktop devices. The instance with the checked shield icon is considered the primary instance. Only for the primary instance the current position of the System User gets synchronised with other devices like mobile devices (*location independence*, see user story ID 18). Non-Primary instances can be used without restrictions but their position is not saved/synchronised (folder location). Files and folders can be moved between instances, for example, to select a file in one instance and move (copy operation) it to a second instance.

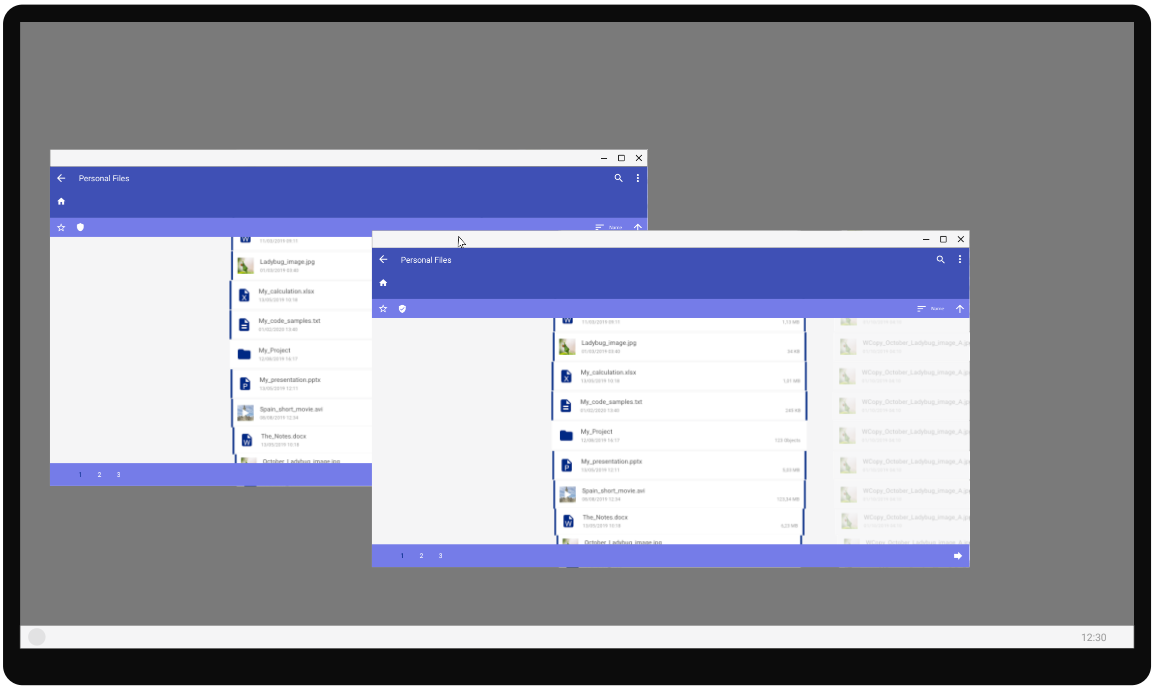



Figure 5.38: Wireframe 2D Navigation (Space) – PC Desktop Device – Multiple Instances

*Note*. Wireframe visualising 2D Navigation mode on a PC desktop device using multiple instances. Designed for a PC with a screen resolution of 1920x1080 pixels at 110 ppi (zoom for details). Wireframe created using Affinity Designer on Windows.

## 5.6 Chapter Summary

I conclude that my novel method addresses tasks for overview and navigation. In context to my task analysis, I carried out in Section 2.7 (see Table 2.2) my method supports the following tasks:

- Overview: Yes, it supports overview in 3D Overview mode where one can view a large, if not the complete dataset.
- Zoom: Yes, in 2D Navigation mode it provides a form of a fisheye lens effect because of the radial node layout. Hence, the node at the centre of the screen appears larger than its surrounding nodes. In 3D Overview mode the relative movement and perspective projection allow a form of continuous zoom where the System User controls the level of detail by moving the camera closer of farther away.

- Filter: Yes, my design considers the utilisation of such features, for example, sort.
- Details-on-demand: Yes, considered in 2D Navigation mode by utilising standardised, for example, contextual overflow menus allowing easy integration of another option to show file or folder details. Also, in 3D Overview mode the pop up window which is shown when the System User moves the crosshair over a node supports such tasks.
- Relate: Yes, as the node layout follows a consistent structure where the position of a Node Layer (files in a folder) always indicates the position of its parent node in its Node Layer (parent folder).
- History: Yes, for example by considering functions to navigate back to the last location. Also, the system is designed to track the System Users position at all times so they can, for example, switch devices at any time and still be where they left off.
- Extract: No, export features not considered at this stage.

Overall, my method is an adjacency layout in 3D. It utilises continuous zoom (3D Overview mode) and a form of a fisheye lens effect (2D Navigation) for context+focus. It also provides overview+detail functionality because of the display of the current path at the top of the screen (top app bar). Also, it provides the option to visualise additional, non-hierarchical relations (see Related Content feature). I argue that:

- My method has no node limit for visual clutter because of its radial layout. Node overlap cannot occur by accident and is always controlled by the System User, for example, when moving the camera in 3D Overview mode. Also, the width of a node is set to a fixed size. Therefore, it is irrelevant how long a text label becomes as this cannot introduce visual clutter.
- For interaction I support: scrolling, selecting items, contextual overflow menus, freely move and rotate in 3D space, editing and deleting items.
- My method utilises the available screen space very well as it uses the vertical space to show individual nodes in a Node Layer and the horizontal space, for example, to show

additional Node Layers. Also, even when looking at an empty folder the System User potentially sees the contents of other folders in the distance.

- It supports precise navigation, especially in context of the 2D Navigation mode as the interaction was designed to mimic a standard list-based type of interaction similar to a tree view (see Section 2.3.1).

To summarise, in context to my findings on what a new technique should provide (see Section 2.7) I conclude that my method supports overview and navigation tasks while utilising the available screen space. It is the first technique to mimic a traditional 2D list-based navigation while otherwise using a 3D native node layout when visualising hierarchical data. My method is also the first of such techniques designed with a focus on scalability from a technical perspective as the visualisation of the whole tree can be reduced to a user input like a selected path and also a design perspective to prevent visual clutter. Also, it is the first technique to be designed from the bottom up to be used on mobile and PC devices. Hence, to a large extent I contextualised and designed my technique for the **be here now** metaphor I introduced in Chapter 4.

# Chapter 6 Development and Evaluation of a Novel Method

## 6.1 Introduction

In this chapter I describe the development and evaluation of my novel method. In Section 6.2 I detail the design walk-through I carried out with a research participant for my conceptual design (see Section 5.4). Section 6.3 outlines the design review I carried out with a research participant for my wireframes (see Section 5.5). Consequently, in Section 6.4 I describe my performance results for rendering. Section 6.5 describes an edge case for my novel method due to radial scaling. In Section 6.6 I explain how I created a video-based prototype for my novel method. Section 6.7 describes the final evaluation stage for my video-based prototype. At the end, I provide a short summary in Section 6.8.

Similar to Chapter 4 all of my first cycle codes (Descriptive Codes) in the sections 6.2 and 6.3 are highlighted in *italics*. I utilised my codes from the previous Chapter 4 to describe my designs. All references to user story IDs are from my final user stories described in Section 4.7 and Table B4.

## 6.2 Design Walk-Through (Conceptual Design)

To evaluate my conceptual design (see Section 5.4) I chose to carry out a single semi-structured interview as a design walk-through. I presented the concepts to the interviewee and asked questions and for feedback. In detail, I planned the interview as follows:

- Present the conceptual design from the ecological perspective (see Figure 5.2). Ask for feedback, for example, general impression, challenges.
- Present the storyboard from the ecological, interaction and emotional perspective (see Figure 5.3 and Figure 5.4). Ask for feedback from all perspectives, for example, interaction challenges.
- Ask the participant for their view on:

  - The potential ability to customise the colour scheme and set custom icons (*bias*, see user story ID 6).
  - Features to set and maintain favourites and multiple instances. Question derived from feedback from my research's first supervisor feedback. Potentially a new user story with a mediocre priority rating (better) as these are not part of the core design metaphor.
  - The option to fully interact with their data in 3D Overview mode, for example, move files versus a view only (*customised structure*, see user story ID 2).
  - Automatically be moved away from the data (current folder) when they enter 3D Overview mode versus a simple unlock of the camera (*improve efficiency*, see user story ID 10).
  - The display of additional meta data in 3D Overview mode, for example, the number of files in a folder (*customised structure*, see user story ID 2).
  - The transform of nodes based on their distance to the user in the 3D Overview mode (semantic zoom), for example, that only enlarged file icons are shown without their labels at a certain distance (*improve efficiency*, see user story ID 10).
- Present conceptual design primarily from the interaction perspective (see Figure 6.1) asking for:
  - The participants view regarding different types of node layouts, for example, how do they perceive a horizontal circular layout versus a completely different geometric placement (linked to many user stories, for example, user story ID 10 and user story ID 7).
  - Their impression regarding an optimised visualisation based on the data type versus a static output (*type-based rendering*, see user story ID 14).

- Their view for or against a device dependent object visualisation regarding the width for a single node (*device independence*, see user story ID 17).

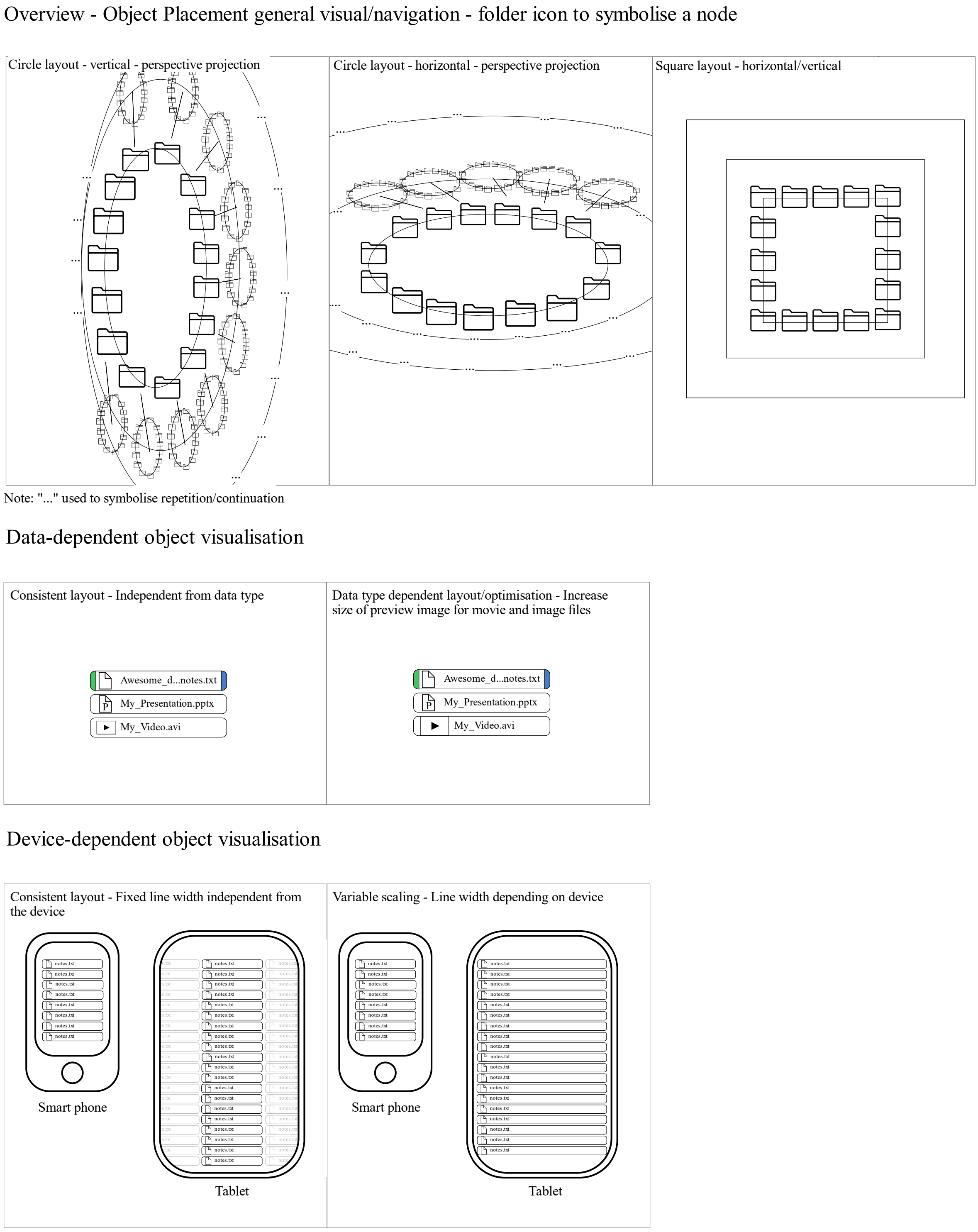



Figure 6.1: Conceptual Design Primarily From the Interaction Perspective

*Note*. English version of my conceptual design primarily from the interaction perspective. A German (original) version of this design was presented to interview participants. I translated the English version myself based on the German original.

After the interview I imported the transcript into NVivo for analysis. As a form of Provisional Coding, I utilised the following categories:

- ecological perspective
- interaction perspective
- emotional perspective

I chose these categories to sort the feedback into the corresponding design perspectives, for example, if the participant highlighted a feature impacting the ecology, I assigned a Descriptive Code for this feature request to the ecological perspective category.

Table C1 lists my Descriptive Codes after first cycle coding. Worth noting that, these codes were manually created and reviewed once by myself.

Worth noting that, Professional02 mentioned *responsive design* techniques in the interview. These techniques are aimed to adjust an apps user interface for different devices, for example, optimise for different input types or screen sizes (Walker & Radich, 2021).

In general, the feedback was rather positive. To exemplify, based on my conceptual design from the ecological perspective (see Figure 5.2) Professional02 saw the greatest benefit in the *centralisation* of information (*location independence*, see user story ID 18). Meaning, the added control or overview over one's data across devices. Also, they perceived my conceptual design from the ecological perspective to be complete without significant gaps like relevant system dependencies (*complete ecology*).

As an additional user story, I added *favourites and multiple instances* to my list of user stories described in Table 6.1:

**Table 6.1**

*User Stories - Addition*

| ID | User story: As a System User I want to … | Acceptance criteria for System User | Priority | Informed by leading pattern |
|---|---|---|---|---|
| 22 | have features available to me to utilise *favourites and multiple instances* across devices. | Set a favourite folder (parent node)<br>Select a favourite folder to be moved to this location<br>Use multiple instances of the application when using a PC desktop device | better: Not a core feature in the first iteration, it is not a commercial product | be here now |

*Note*. Added user story in reference to Table B4 (see Section 4.7).

Noteworthy, is the link which Professional02 established between *location independence* and the utilisation of favourites to extend the *centralisation* of information across devices. To quote Professional02 "*Which would also be really cool … such a favourite function. … that this is then (inc.) transmitted everywhere on the tablet or smart phone*".

From the ecological and in part interaction perspective we briefly discussed the idea to utilise *gestures to replace touch targets*. Based on this participatory design approach we concluded that this may be difficult to realise from the ecological perspective. Meaning, if the System User would use the method on a PC desktop device with a mouse and keyboard the lack of precise click targets may be hindering.

Also, Professional02 highlighted the importance of *responsive design*. This was also reinforced by an idea of them to utilise a *responsive node layout*. At its core they imagined the node layout to change from a vertical to a horizontal circular layout (see Figure 6.1) based on the current orientation of the device. Albeit an interesting idea I argue that this change would also change the complete control scheme of the system, for example, the scrolling direction. Therefore, I did not follow up on this concept.

Similarly, linked to the topic of *responsive design* Professional02 argued against a static width for a single node across devices. Precisely, they preferred a type of *node scaling* which utilises the whole width of the available screen space.

Overall, Professional02 preferred the vertical circular *node layout* over the presented alternative layouts (see Figure 6.1). Worth noting that, I also offered them the option to nominate other geometric layouts which I did not think of as a form of participatory design. However, they preferred the shown vertical circular *node layout* from the ecological and emotional perspective.

Also, Professional02 viewed the option to customise the colour scheme and icons as a valuable addition (*custom colour and icons*, see user story ID 6). Also, they highlighted the use of *familiar icons* to support their perceived usability of the system. Hence, this indicates support for options to allow the user to utilise user habits and to own the experience (*bias*, see user story ID 5 and user story ID 6). Especially, the option to customise the experience (see user story ID 6) may be an important next step to follow up on when deriving the method for commercial applications.

From the interaction perspective, Professional02 preferred a form of *automatic zoom in 3D Overview mode* to be more efficient. Interestingly, they highlighted the importance of accessibility options in the context of colour blindness (*colour blind accessibility*). Albeit important I cannot consider accessibility options in my design due to resource constraints.

Regarding the potential to optimise the visualisation of nodes based on their data type (see Figure 6.1) they preferred a static output instead (*consistent visualisation*). Consequently, I rejected user story ID 14 from my list of core ("*good*") user stories.

Professional02 stated that they prefer an implementation of a kind of pop up window mechanism for navigating in 3D Overview mode regarding the topic of *dynamic scaling in 3D Overview*. Meaning, they want to view the nodes as single icons or thumbnails and only if they focus individual objects a pop up window should appear to show details for the node, for example, file details. Hence, they described a task for details-on-demand. To quote Professional02: "*What I would prefer now, I believe so that it is clean and I believe such a view is first of all informative, really well constructed, I would then prefer that with the tooltip*".

When reflecting on the shown storyboard (see Figure 5.3 and Figure 5.4) Professional02 shared their thoughts regarding the *efficiency in 3D Overview* mode. They highlighted the added benefit from the perspective projection to see more relevant content in the background to potentially save time. To quote Professional02 "*In the third view, I think the solution is really cool with the 3D view that you look in the background, so to say, like a waiting list or waiting queue can already see the next interesting folder that could just come into question for me*". Hence, their feedback meets my design intention (*location independence*, see user story ID 18) while also highlighting the potential of the 3D Overview mode to *improve efficiency* (see user story ID 10).

Professional02 stated that they want to have access to the same *file and folder operations in 3D Overview* and 2D Navigation mode, for example, to delete a file. They argued that this is faster (*improve efficiency*, see user story ID 10) than switching back to 2D Navigation mode.

Based on their feedback Professional02 does not want additional meta data like the number of files in a folder to be displayed in 3D Overview mode (*meta data in 3D Overview*). To quote "*I do not believe that brings any added value for me personally*".

They also highlighted the importance of touch target sizing. Precisely, if the *size of touch targets* is too small it may impact usability (*error-free*, see user story ID 9). To quote Professional02 "*I believe that can be a bit problematic, I am a user who, for example, has two thick fingers*".

**6.3 Design Review (Wireframes)**

To evaluate the results for my wireframes (see Section 5.5) I carried out a single semi-structured interview as a design review aiming to identify challenges in my design primarily from the interaction perspective.

I structured the interview according to my user stories with a priority rating of good and connected task sequence models (see Section 4.7).

For the introduction I presented my conceptual design (see Section 5.4.1) and storyboard (see Section 5.4.2).

Also, I presented renderings (see Figure 6.2) created with Unity and Affinity Designer to explain the node layout (see Section 5.5.2). Important to note that, I counted each outer circle as one hierarchy level in those renderings where the children for root are at hierarchy level zero. I decided to communicate it like this to support the participants understanding of the layout. Regardless, this is technically wrong (see Section 5.5.2).

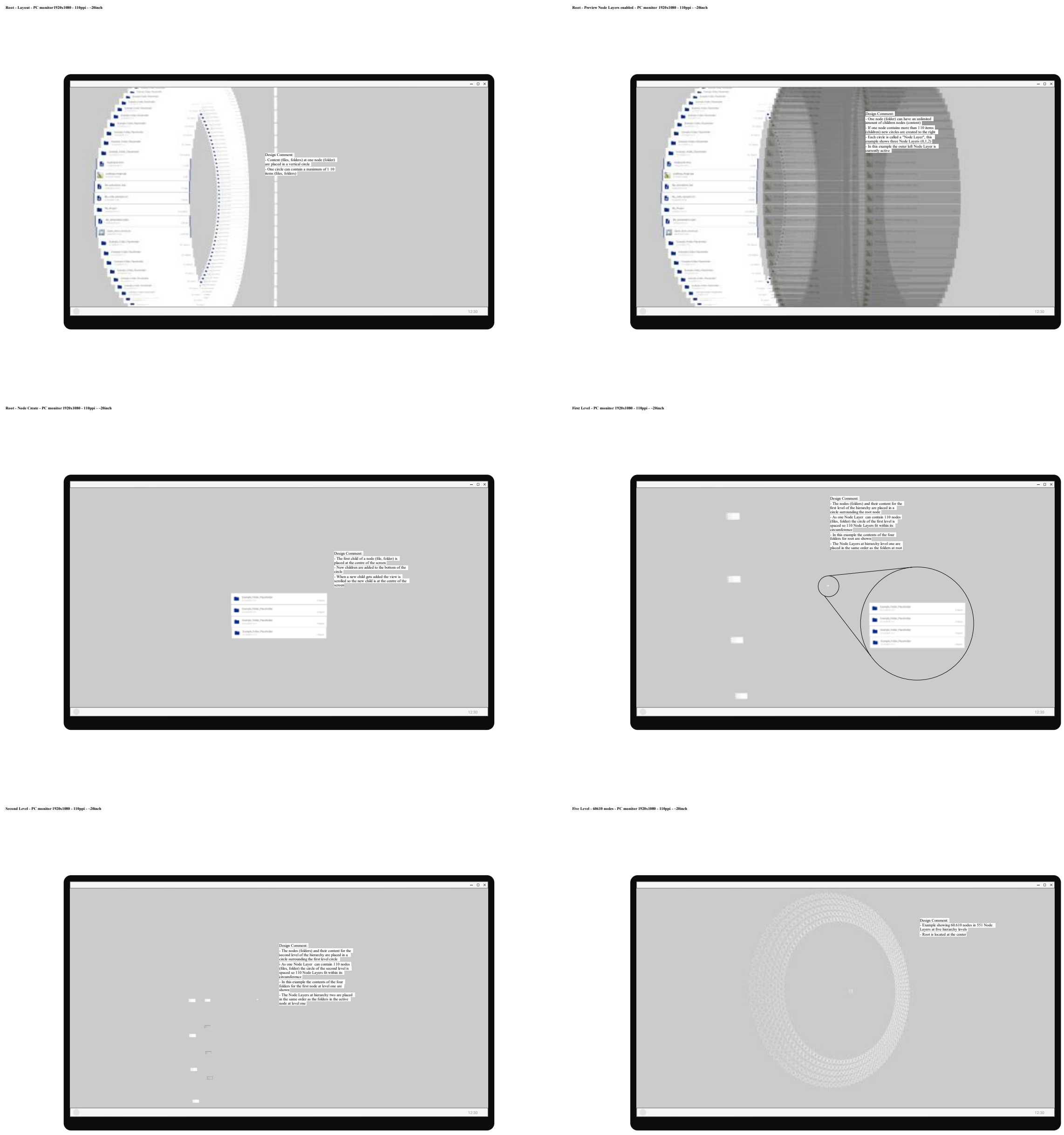


Figure 6.2: Node Layout – Design Review Material

*Note*. English version of wireframes to explain the node layout. The design comments were translated to German by myself based on the English original. The German version was presented to an interview participant. Each wireframe was designed for a PC with a screen resolution of 1920x1080 pixels at 110 ppi (zoom for details). Sizing information corrected compared to original. Wireframes created using Unity and Affinity Designer on Windows.

Next, I utilised my wireframes (see Section 5.5.3) to guide the research participant through the user stories. For presentation I consolidated my mobile phone wireframes into a form of *wireflow* prototype. Essentially, wireflows "are a combination of wireframes and flowcharts" (Laubheimer, 2016, Summary section). However, because my method has many different modes and perspectives, I just laid out the wireframes hierarchically according to their functionalities. Hence, I also did not draw connecting lines between each wireframe and interactive elements, for example, buttons.

Worth noting that, I addressed user story ID 5 only implicitly. Hence, the icons are shown on every wireframe. However, I put attention to the icons by highlighting the feature to change the icons and colour scheme of the application in the settings (see Figure 5.22). Albeit, this user story ID 6 does not have a good priority rating I claim it was a decent compromise to draw focus on the icons.

Also, I did not show scenarios for *immersion* (see user story ID 7) as I aimed to identify challenges mainly from the interaction perspective. Still, I asked the research participant to highlight potential challenges in general including the emotional perspective, for example, if something feels wrong to them.

After the interview I imported the transcript into NVivo for analysis. Again, as a form of Provisional Coding I utilised the following categories:

- ecological perspective
- interaction perspective
- emotional perspective

I chose these categories to sort the challenges into the corresponding design perspectives, for example, a topic for missing buttons which prevent further interaction would be assigned to the interaction perspective category.

Table C2 lists my Descriptive Codes after first cycle coding. Worth noting that, these codes were manually created and reviewed once by myself. As described, I aimed to identify precise challenges for my design. Therefore, I did not assign codes to passages of confirmation, for example, if the participant stated that something would work as presented.

Student01 made proposals to enhance the design. These were discussed during the design review as a form of participatory design. From the ecological perspective Student01 mentioned options to further customise the experience. Precisely, they proposed having multiple points in time which are continuously synchronised and are also customisable (*customised synchronisation*). Albeit, an interesting detail I decided to not consider it for my design as this goes against my design metaphor **be here now**. Hence, I argue that an infinite number of now again leads to disorientation and not abstraction and central control (see *location independence*, user story ID 18).

Also, they proposed visualising how the download status of documents may be visualised in the interface when talking about *document availability*. However, this is very device and use case specific. To exemplify, when working with internet-based applications this downloading of documents to a local device may not even be necessary. For essential data types (see user story ID 21) they briefly mentioned *MP3 files* but more from the perspective of their personal usage history.

Interestingly, they did not see *value* in using the method for their work practice. Also, Student01 said "*To be honest, I do not believe I would be more efficient. (…) But I can imagine it being just as efficient, yes.*". So, they can imagine shielding a similar task performance when compared to conventional applications (*improve efficiency*, see user story ID 10). However, they cannot imagine shielding a better performance.

From the emotional perspective Student01 talked about potential challenges in relation to user *habits*. Meaning, user acceptance may suffer because of the unfamiliar circular node layout. Hence, even though I considered this form of *bias* in my design (see for example user story ID 5) I cannot think of a way to eradicate any form of perceived change in my design.

Student01 proposed adding a form of label to the *action buttons* in 3D Overview mode to better visualise their functionality. Also, from the interaction perspective they proposed moving the fixed *camera position* in 2D Navigation mode so one could see more nodes at once. Hence, an interesting idea I argue that this will not work on mobile phones as the screen is just not wide enough. They also proposed a change for 3D Overview mode to adopt interaction techniques used to navigate online maps (*map navigation*). However, in comparison to a 2D map I cannot identify a suitable way to support such techniques in a 3D perspective view. To exemplify, when using gestures, the System User would have to swipe very often to rotate the view. They also asked for more *metadata* to be shown for each node, for example, when a document was last opened. Again, important but I am not designing a commercial product. Similarly, Student01 proposed a feature to *preview* content on demand. Very viable for a commercial product but potentially distracting at my stage of the research (see Section 5.4). Hence, they also proposed features on the topic I called *scripting* to allow the renaming of nodes according to a user specific business logic. Again, interesting in the context of a commercial product but not the core of my method. Student01 recommended to leave the *sorting* option enabled when editing with the focus on PC desktop devices. They also proposed a concept to control the movement *speed* in 3D Overview mode by long pressing on the joy sticks. Hence, this in an important detail but one that for me belongs to the settings (see Figure 5.22) category under general, for example, as a slider input. Hence, I argue that their proposed idea is a form of zoom which may work in a 2D map but not in a 3D perspective view. The final topic resolved around *user guidance* where Student01 proposed a form of initial guide through when the System User installs the application for the first time. This may help the user to overcome their *habits* and ease transitioning. However, such guidance may only become relevant when the method is rolled out as a commercial application or for large user studies. Therefore, such guidance is not in scope for my research.

## 6.4 Rendering Performance

To optimise rendering performance, I considered *prefabs* in Unity to realise my prototypes. A prefab can be viewed as a class blueprint which can be instantiated as separate copies in a scene (Ferrone, 2020). Hence, for my Unity renderings in Section 5.5.2 and 6.3 I utilised prefabs. I also utilised *GPU instancing* to further improve rendering performance in those renderings. GPU instancing optimises draw calls by batching copies of a mesh utilising the same material into one draw call (Unity Technologies, 2022a).

This approach may not be ideal for a production grade application as in reality the text of the label and the icons change for each node. Still, I argue that this approach may be a feasible one depending on the dataset if there is a lot of static content, for example, a folder icon which gets utilised for many nodes.

Regardless, for the overall system to work one needs an approach which can dynamically render text and icons to visualise the records from a database (see Section 5.4.1). To render text dynamically in Unity I utilised the integrated *TextMeshPro* solution. TextMeshPro is a solution to render text in Unity (Unity Technologies, 2022c). For the icon material I disabled GPU instancing. Figure 6.3 shows an example node with changeable text and metadata. Also, the ribbons and icon are separate objects to be changed dynamically. Figure 6.4 visualises 110 nodes for depth 1.


Figure 6.3: Example Node With Dynamic Elements

*Note*. Unity render result at 1920x1080 pixels simulating 2D Navigation mode with a fixed camera. Visualising one node with changeable text and elements. The bottom left shows the start-up time in seconds and the current frames per second.

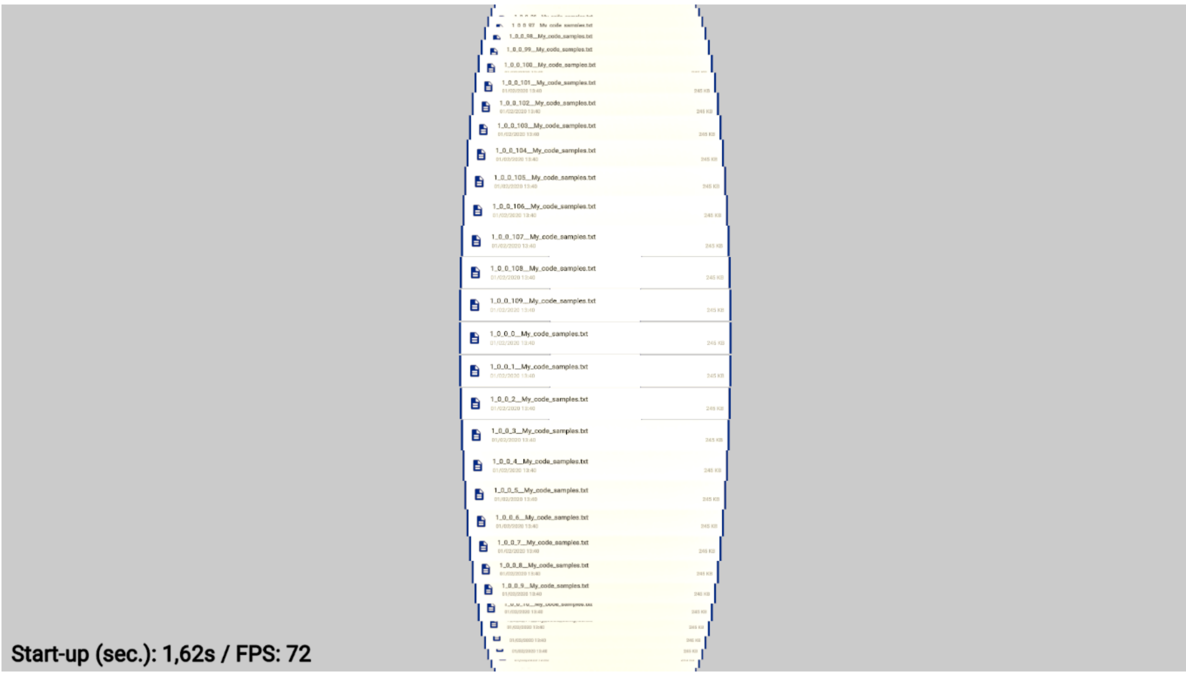



Figure 6.4: Example Node Layer With Dynamic Elements

*Note*. Unity render result at 1920x1080 pixels simulating 2D Navigation mode with a fixed camera. Visualising 110 nodes at depth 1 with changeable text and elements. The bottom left shows the start-up time in seconds and the current frames per second.

To evaluate the node limits for both approaches, using a prefab versus utilising dynamic elements, I measured the performance for both. In the Unity build settings, I chose to create development builds and auto connect the profiler. Also, I chose Direct3D11 to be the graphics API for Windows. Moreover, I disabled the Graphics Jobs option in the player settings to enable GPU usage profiling (Unity Technologies, 2022b). To simulate for the worst case, I moved the camera far out to see most of the nodes at the same time. I increased the tree depth level by one for each measurement. Each Node Layer has 110 nodes and I did not render preview Node Layers or related content, for example, using the History feature. Figure 6.5 shows an example for depth 2 using the prefab approach.

Table 6.2 lists the benchmark results using the prefab approach. Table 6.3 visualises the results for my dynamic approach. For the prefab approach the results are clearly limited by the CPU (see columns “*CPU*” and “*GPU*”). In comparison, the dynamic approach has to draw a lot more vertices. Therefore, the GPU also struggles to keep the framerate high, close to 60 FPS. However, because of the scalable node layout I argue these results may be improved significantly. For one at the distance from which I measured the results the System User can no longer see individual icons or text. Hence, one may implement those nodes in such a way that at a certain distance text and icons will not be rendered at all. Also, one may evaluate other approaches for simplification, for example, by replacing the meshes for individual nodes with larger singular meshes depicting a whole circle or sub circle. Hence, simplification methods to reduce mesh complexity as described by Ng and Low (2014) can improve rendering performance.

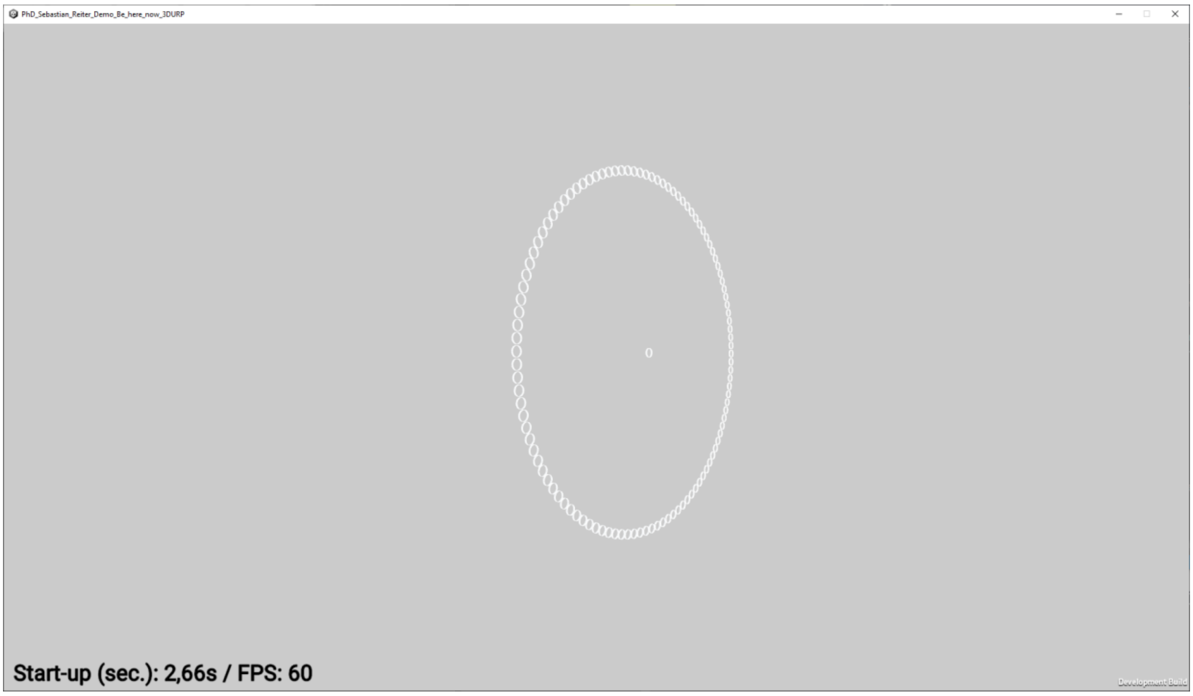



Figure 6.5: Example Result to Compare Rendering Performance

*Note*. Unity render result at 1920x1080 pixels simulating 2D Navigation mode with a moved camera. Visualising 12,210 nodes at depth 2 using the same prefab for each node. The bottom left shows the start-up time in seconds and the current frames per second.

**Table 6.2**

*Benchmark Results Utilising a Prefab Approach*

| Tree depth | Node count | Start-up time (sec.) | FPS (frame; approx.) | Tracked Memory (frame; MB; in use; approx.) | CPU (ms; frame; approx.) | GPU (ms; frame; approx.) |
|---|---|---|---|---|---|---|
| 1 | 110 | 2.60 | 59 | 91.9 | 16.65 | 18.53 |
| 2 | 12210 | 2.67 | 59 | 119.5 | 16.67 | 13.80 |
| 3 | 24310 | 2.74 | 60 | 147.1 | 16.65 | 11.77 |
| 4 | 36410 | 2.82 | 59 | 170.9 | 16.67 | 7.99 |
| 5 | 48510 | 2.90 | 59 | 202.1 | 16.67 | 2.59 |
| 6 | 60610 | 2.97 | 59 | 222.8 | 19.60 | 4.94 |
| 7 | 72710 | 3.04 | 59 | 248.8 | 31.27 | 8.24 |
| 8 | 84810 | 3.12 | 29 | 269.8 | 27.70 | 2.64 |
| 9 | 96910 | 3.22 | 29 | 316.0 | 29.70 | 6.56 |
| 10 | 109010 | 3.30 | 29 | 337.9 | 37.25 | 1.15 |

*Note*. Performance benchmarks utilising the prefab approach. Benchmarks were carried out using a workstation PC with a Ryzen 9 5950X CPU, 64GB RAM and a NVIDIA GeForce RTX 3090 GPU with 24GB VRAM on Windows 10 Pro x64. The development builds were created using Unity (editor version 2021.3.6f1) with the Direct3D11 graphics API. Measurements for the last four columns (FPS, Tracked Memory, CPU, GPU) were taken from a randomly selected frame. The last three columns (Tracked Memory, CPU, GPU) reflect the values as reported by the Unity Profiler for the selected frame.

**Table 6.3**

*Benchmark Results Utilising a Dynamic Approach*

| Tree depth | Node count | Start-up time (sec.) | FPS (frame; approx.) | Tracked Memory (frame; MB; in use; approx.) | CPU (ms; frame; approx.) | GPU (ms; frame; approx.) |
|---|---|---|---|---|---|---|
| 1 | 110 | 2.64 | 59 | 110.6 | 16.73 | 15.56 |
| 2 | 12210 | 13.19 | 11 | 1490 | 84.40 | 62.43 |
| 3 | 24310 | 34.38 | 5 | 1490 | 84.92 | 63.13 |
| 4 | 36410 | 67.16 | 3 | 4170 | 306.58 | 209.21 |
| 5 | 48510 | 111 | 2 | 5480 | 407.20 | 276.67 |
| 6 | 60610 | N/A | N/A | N/A | N/A | N/A |
| 7 | 72710 | N/A | N/A | N/A | N/A | N/A |
| 8 | 84810 | N/A | N/A | N/A | N/A | N/A |
| 9 | 96910 | N/A | N/A | N/A | N/A | N/A |
| 10 | 109010 | N/A | N/A | N/A | N/A | N/A |

*Note*. Performance benchmarks utilising the dynamic approach to render individual nodes, for example, with changeable text elements using TextMeshPro. Benchmarks were carried out using a workstation PC with a Ryzen 9 5950X CPU, 64GB RAM and a NVIDIA GeForce RTX 3090 GPU with 24GB VRAM on Windows 10 Pro x64. The development builds were created using Unity (editor version 2021.3.6f1) with the Direct3D11 graphics API. Measurements for the last four columns (FPS, Tracked Memory, CPU, GPU) were taken from a randomly selected frame. The last three columns (Tracked Memory, CPU, GPU) reflect the values as reported by the Unity Profiler for the selected frame. Benchmarks with “N/A” entries were not carried out as the performance was not feasible, especially regarding the frame rate (FPS).

## 6.5 Radial Scaling

One edge case I identified for my method is the visualisation of a very deep tree. Hence, given the radial distribution of the Node Layers the distance between them increases with the depth of the tree. Figure 6.6 visualises this challenge using cylinder meshes as placeholders for the individual Node Layers. This may pose a challenge when navigating in 3D Overview mode. However, I argue that this challenge may be mitigated by utilising a different camera lens like a fisheye and a combination of zoom. Also, I argue that the practical use cases for such deep trees are rather rare.

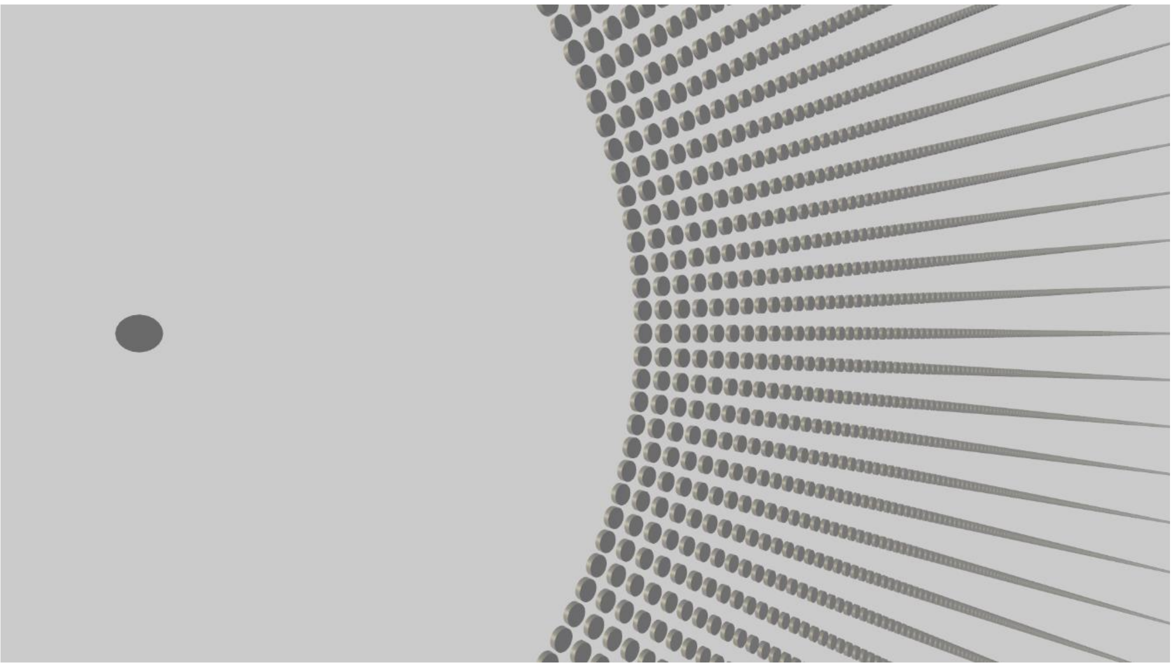


Figure 6.6: Visualising a Deep Tree

*Note*. Unity render result at 1920x1080 pixels simulating a tree with a depth of 999. The camera was moved to highlight the increasing distance between Node Layers as the tree grows deeper. Cylinder meshes were used to simulate the individual Node Layers.

**6.6 Developing a Scenario Based Prototype**

Based on the feedback from Student01 (see Section 6.3) I conclude that my design from the interaction perspective works as described. The rather small changes are:

- Leave sort enabled during edit mode
- Allow a setting to control the moving speed in 3D Overview mode
- Add a label or icon to the buttons in 3D Overview mode

Even though I did not have the resources to develop the whole prototype, for example, because of the cross-device design, I aimed to visualise the node layout and interaction experience based on a scenario as a form of a video-based prototype. As my scenario I chose a Student (sub role of System User, see user work role model in Section 4.7) aiming to work on a bug hierarchy (insect taxonomy) for an entomology project. I chose this scenario primarily because of the availability of images I could utilise for my prototype (see Appendix A). Overall, I aimed to address all of my user stories with a priority rating of good and the sub tasks associated to using a *customised structure* (see Section 4.7). To produce the video for my prototype I combined the Unity renderings with static image overlays. Worth noting that, for the renderings in Unity I used a static dataset (hard coded labels, set of images). Alternatively, I could have setup a database server to store the artificial dataset for my scenario or read the contents from a local folder on my development PC. Hence, storing the data on a remote server may have been beneficial to more accurately measure the start-up time of the prototype (see Section 6.4). However, for the purpose of this video-based prototype the storage type for the artificial dataset is negligible.

To prototype this scenario for a mobile device I chose to utilise the Simulator functionality built into Unity. As the target device for the Simulator, I chose the Samsung Galaxy S7 mobile phone. Hence, the Samsung Galaxy S7 has a resolution of "1440 x 2560 pixels" (GSMArena.com, n.d., DISPLAY section). Meaning, the resolution is exactly four times the target resolution of my design 360x640 pixels (see Section 5.5.1). For PC desktop I used a target resolution of 1920x1080 pixels. Worth noting that, for rendering the nodes in Unity I had to utilise a dynamic approach to render, for example, text labels. I chose this approach because it allowed me to more easily vary my otherwise static content, for example, so that files do not all share the same text label (see Section 6.4). Also, to show navigation and movement in Unity I applied animation, for example, to rotate the camera. Consequently, the simulation of movements and rotations of the camera in 3D Overview mode may not be totally accurate when compared to human interaction.

To set the scale for an individual node in the Simulator for a mobile phone in 2D Navigation mode I scaled the node at the centre of the screen to be 1440 pixels wide and 288 pixels high. This is equal to 360dp times four for width and 72dp times four for height (see Section 5.5.2). Worth noting that, to realise this in Unity I just moved the camera closer along the z-axis to 0, 1, -11.415 in World space (see Section 5.5.2). Therefore, I just had to adjust the width of the quad mesh representing the first child node. For PC desktop I just reused my values from Section 5.5.2, precisely, 440 pixels for width and 50 pixels for height. To exemplify, Figure 6.7 shows a first child node at the centre of the screen for PC desktop while Figure 6.8 visualises the same node on a mobile phone using the Unity Simulator. Worth noting that, for 3D Overview mode a single node has the same height as in 2D Navigation mode (72dp). However, I decided to increase the size of the respective icons or thumbnails from 56dp (see Section 5.5.3) to 64dp (64x64) for an improved visibility in 3D Overview mode.

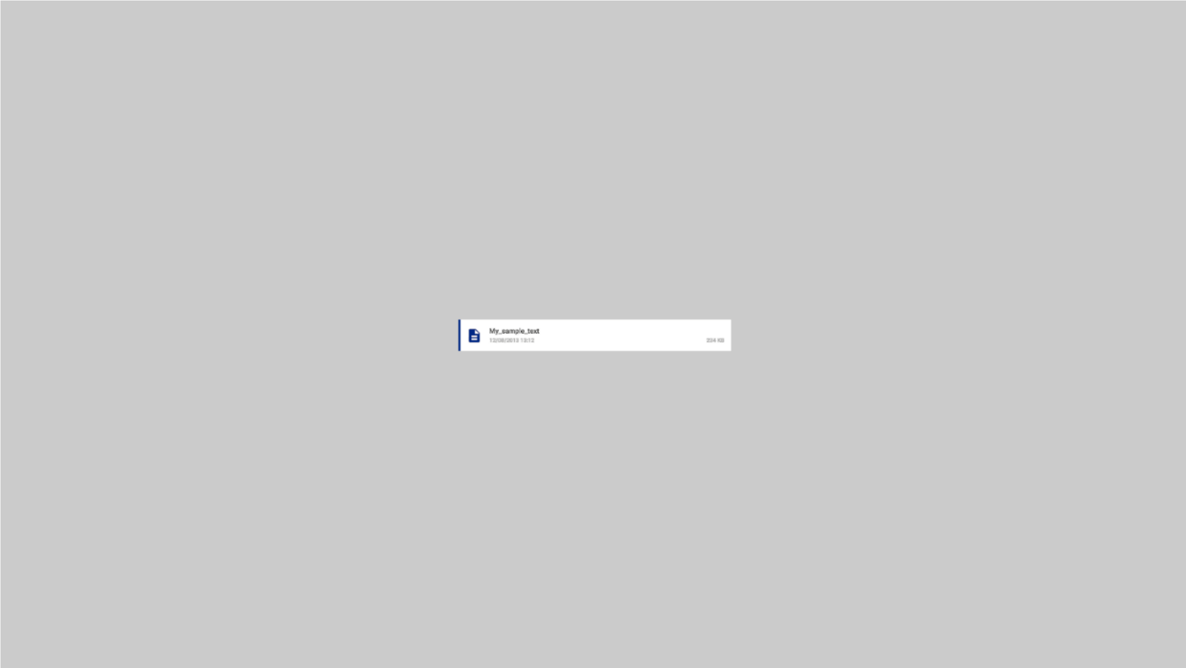



Figure 6.7: Scenario – Single Centre Node – PC Desktop

*Note*. Unity render result at 1920x1080 pixels simulating 2D Navigation mode with a fixed camera and the first child node at the centre for a PC desktop device.

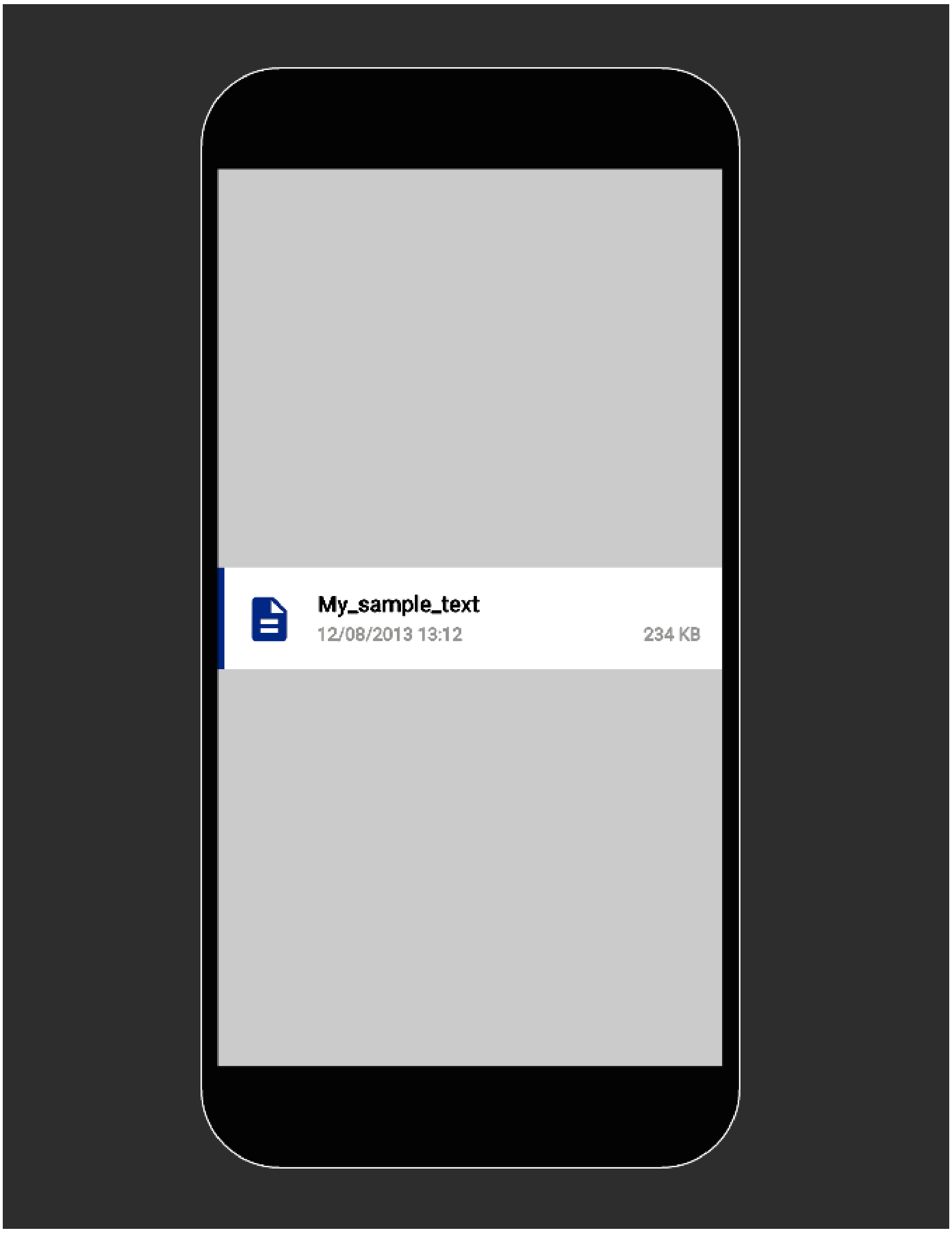



Figure 6.8: Scenario – Single Centre Node – Mobile Phone

*Note*. Unity render result using the Unity Simulator set to simulate a Samsung Galaxy S7 mobile phone. The view simulates the 2D Navigation mode with a fixed camera and the first child node at the centre for a mobile phone.

For my final video-based prototype I utilised static image overlays to simulate: app and system bars, buttons, menus, dialogs, virtual joysticks and buttons, additional notes, input text fields, pop ups and to highlight interaction. Figure 6.9 shows an example of such a static overlay used to produce my video. Worth noting that, I used OBS Studio to screen record the render results from Unity and Kdenlive to edit the final video, for example, to put the image overlays over the video tracks (see Appendix A).

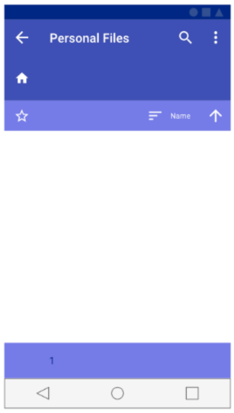



Figure 6.9: Scenario – Example Image Overlay – Mobile Phone

*Note*. Static image overlay used for creating the video-based prototype. The image was scaled to be used for a video in 1920x1080 pixels. Image created using Affinity Designer on Windows.

My scenario starts with the System User looking at their smart phone where the Space Personal Files is already open. They are in 2D Navigation mode and tap on the folder “*Bio_z_entomology*” (see Figure 6.10). The icons they see are familiar to them (*bias*, see user story ID 5). Worth noting that, they see their *customised structure* with all of their relevant data at one central location (*location independence*, see user story ID 18). Also, in my scenario the setting to show preview of Node Layers is enabled for 2D Navigation and 3D Overview mode to show a maximum of five Node Layers for preview. The option to show all Related Content and History in 3D overview mode is disabled (see descriptions for Figure

5.22). In general, the interaction is based on my wireframes (see Section 5.5.3) and where applicable the respective feedback from Student01 (see Section 6.3).

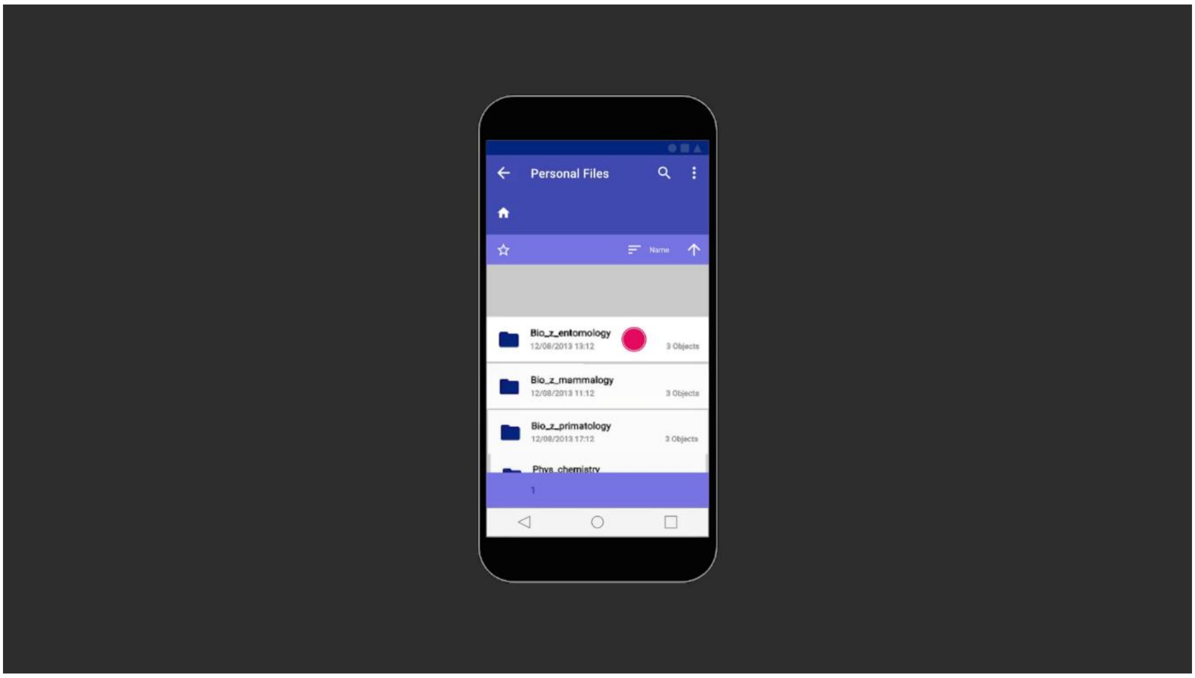



Figure 6.10: Scenario – Select First Folder in 2D Navigation – Mobile Phone

*Note*. Screenshot from the rendered video result (.mp4 at 1920x1080, 30 frames per second) simulating a System User interacting with my method. Created combining Unity render results and static image overlays. Unity Simulator set to simulate a Samsung Galaxy S7 mobile phone. The view simulates the 2D Navigation mode with a fixed camera where the first node (folder) gets tapped on (selected).

Consequently, they swipe to scroll down in the folder Bio_z_entomology to their next target folder "*3rd_semester*" (see Figure 6.11). Worth noting that, they could have also directly tapped on their target folder without scrolling. Hence, the nodes are sized accordingly to allow for precise interaction (*error-free*, see user story ID 9).

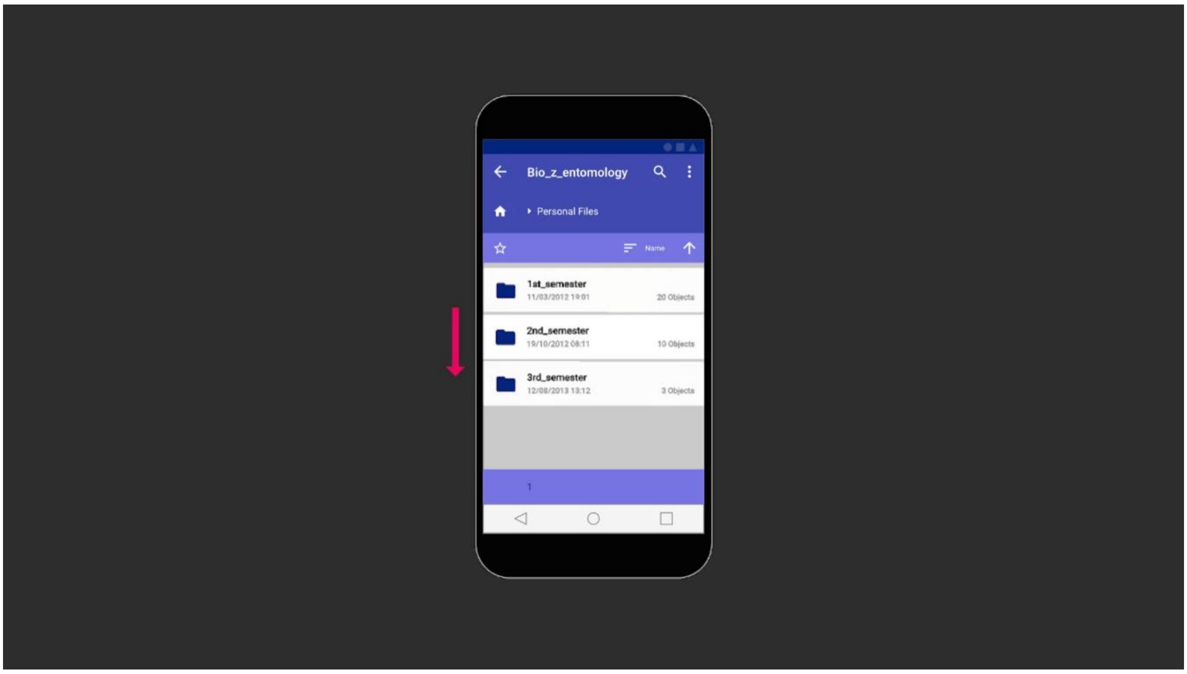



Figure 6.11: Scenario – Scroll to Select Sub Folder in 2D Navigation – Mobile Phone

*Note*. Screenshot from the rendered video result (.mp4 at 1920x1080, 30 frames per second) simulating a System User interacting with my method. Created combining Unity render results and static image overlays. Unity Simulator set to simulate a Samsung Galaxy S7 mobile phone. The view simulates the 2D Navigation mode with a fixed camera where the System User swipes to scroll down in a Node Layer.

Next, the System User selects the folder "*Bug_Hierarchy*" by tapping on it (see Figure 6.12). Worth noting that, even on the small screen of the mobile phone they see in the distance other content, in this case, the sub folders of "*Bio_z_primatology*" (see Figure 6.10). Hence, already a form of overview (*location independence*, see user story ID 18) in the form of context+focus.

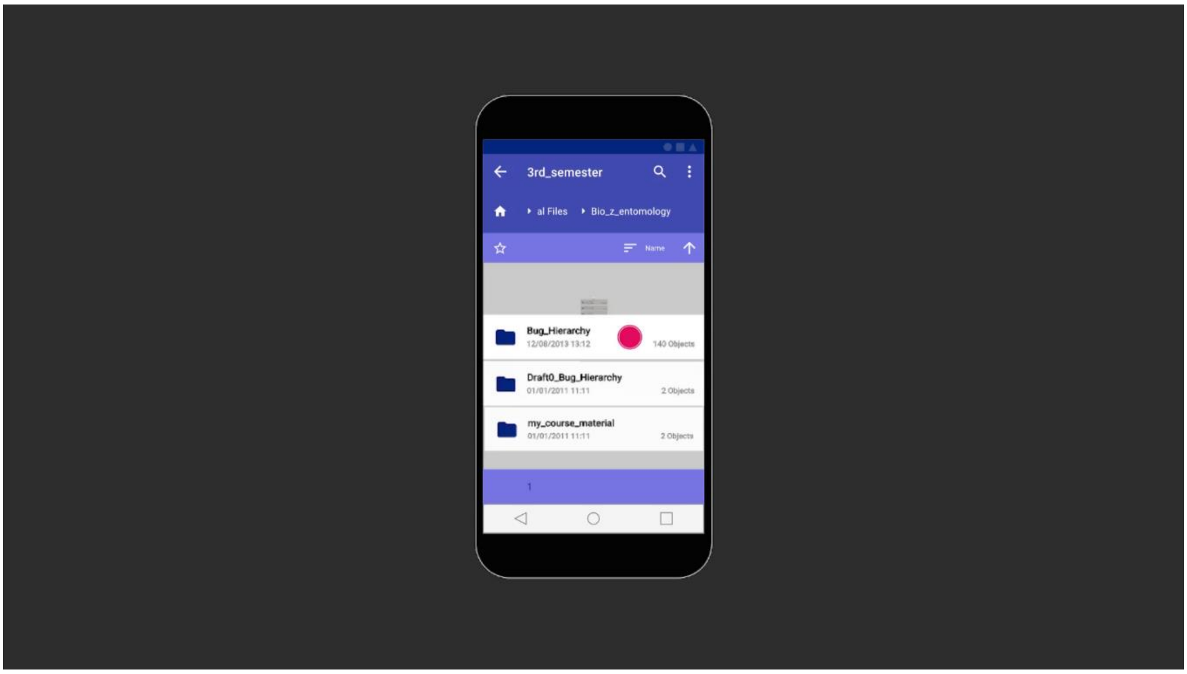


Figure 6.12: Scenario – Select Next Sub Folder in 2D Navigation – Mobile Phone

*Note*. Screenshot from the rendered video result (.mp4 at 1920x1080, 30 frames per second) simulating a System User interacting with my method. Created combining Unity render results and static image overlays. Unity Simulator set to simulate a Samsung Galaxy S7 mobile phone. The view simulates the 2D Navigation mode with a fixed camera where the first node (folder) gets tapped on (selected).

After briefly looking at the contents in their target folder Bug_Hierarchy they tap on the ribbon on the right side of their project notes file called "*!_Current_Notes.docx*" (see Figure 6.13). They aim to draw ideas from other previous, similar projects they have already done (*relate and analyse*, see user story ID 20).

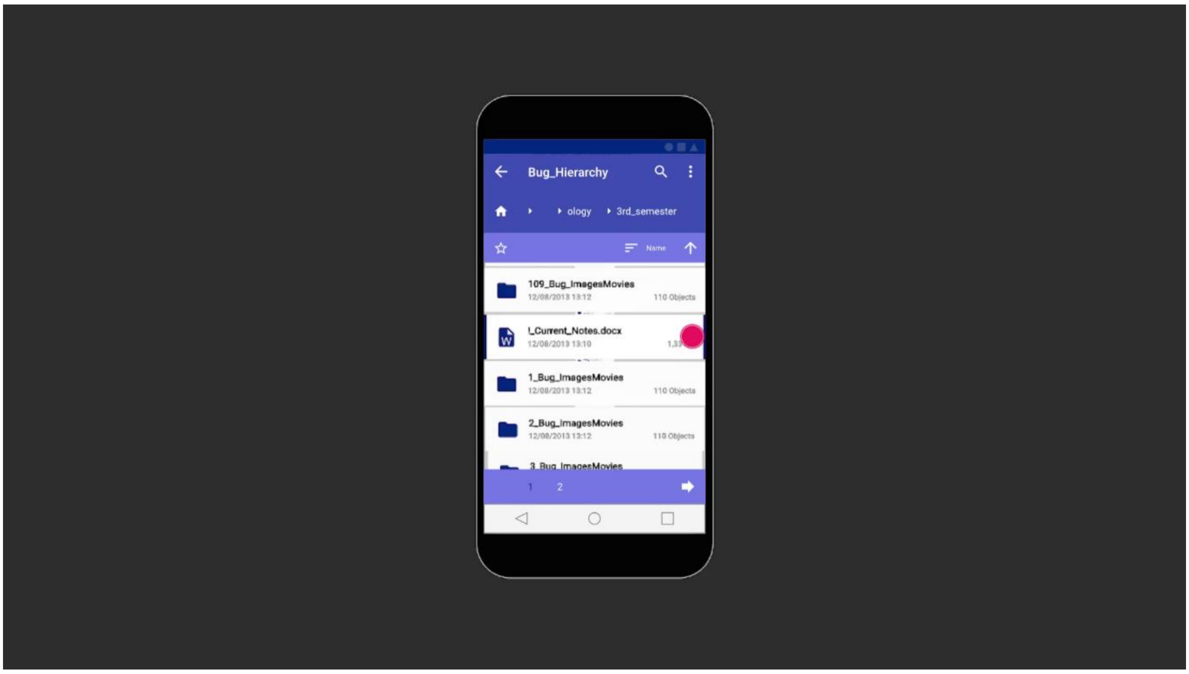



Figure 6.13: Scenario – Enter Related Content Mode for a File in 2D Navigation – Mobile Phone

*Note*. Screenshot from the rendered video result (.mp4 at 1920x1080, 30 frames per second) simulating a System User interacting with my method. Created combining Unity render results and static image overlays. Unity Simulator set to simulate a Samsung Galaxy S7 mobile phone. The view simulates the 2D Navigation mode with a fixed camera where the System User taps on the blue ribbon on the right side of a file to enter the mode to view Related Content for this file.

The System User can now see a related file to their notes which is a Microsoft PowerPoint presentation called “*final_pres_butterflies.pptx*” from a similar project they have done in the past (see Figure 6.14). Again, they can identify the context (overview) of their current location (*location independence*, see user story ID 18) because they can also see a large part of the tree for context+focus. After reviewing the content, they exit out of Related Content mode by tapping on the left ribbon of the file.

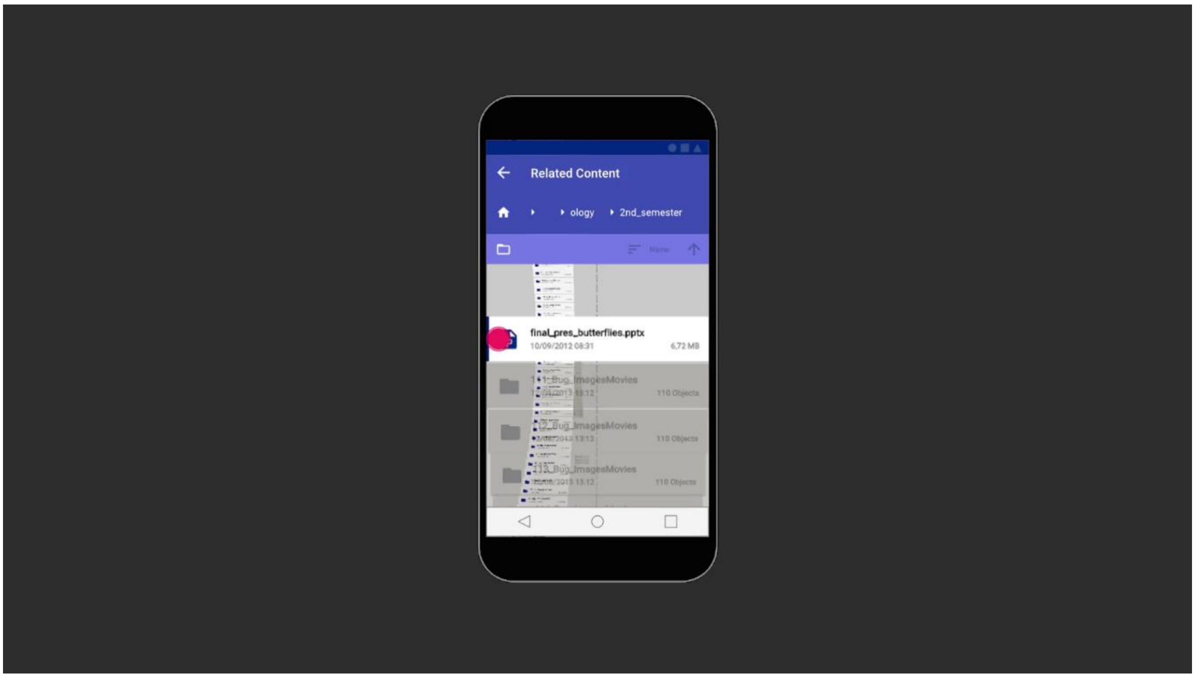



Figure 6.14: Scenario – Related Content Mode for a File in 2D Navigation – Mobile Phone

*Note*. Screenshot from the rendered video result (.mp4 at 1920x1080, 30 frames per second) simulating a System User interacting with my method. Created combining Unity render results and static image overlays. Unity Simulator set to simulate a Samsung Galaxy S7 mobile phone. The view simulates the 2D Navigation mode with a fixed camera where the System User taps on the blue ribbon on the left side of a file to exit out of Related Content mode.

Next, the System User taps on the blue ribbon on the left side of !_Current_Notes.docx to enter History mode (see Figure 6.15). Consequently, in Figure 6.16 they see an older version of the file (*additional relations*, see user story ID 16). They tap once more on the blue ribbon on the left side of the file to see an even older version of it. To exit out of History mode they tap on the close icon (see Figure 6.17).

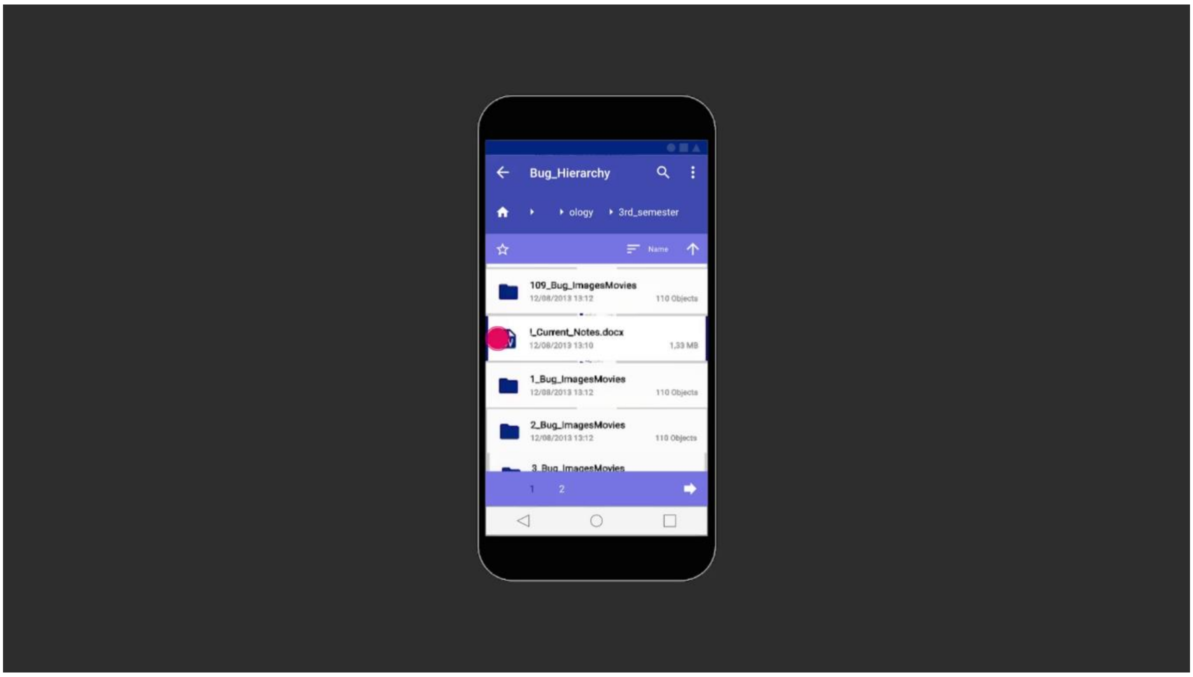



Figure 6.15: Scenario – Enter History Mode for a File in 2D Navigation – Mobile Phone

*Note*. Screenshot from the rendered video result (.mp4 at 1920x1080, 30 frames per second) simulating a System User interacting with my method. Created combining Unity render results and static image overlays. Unity Simulator set to simulate a Samsung Galaxy S7 mobile phone. The view simulates the 2D Navigation mode with a fixed camera where the System User taps on the blue ribbon on the left side of a file to enter History mode.

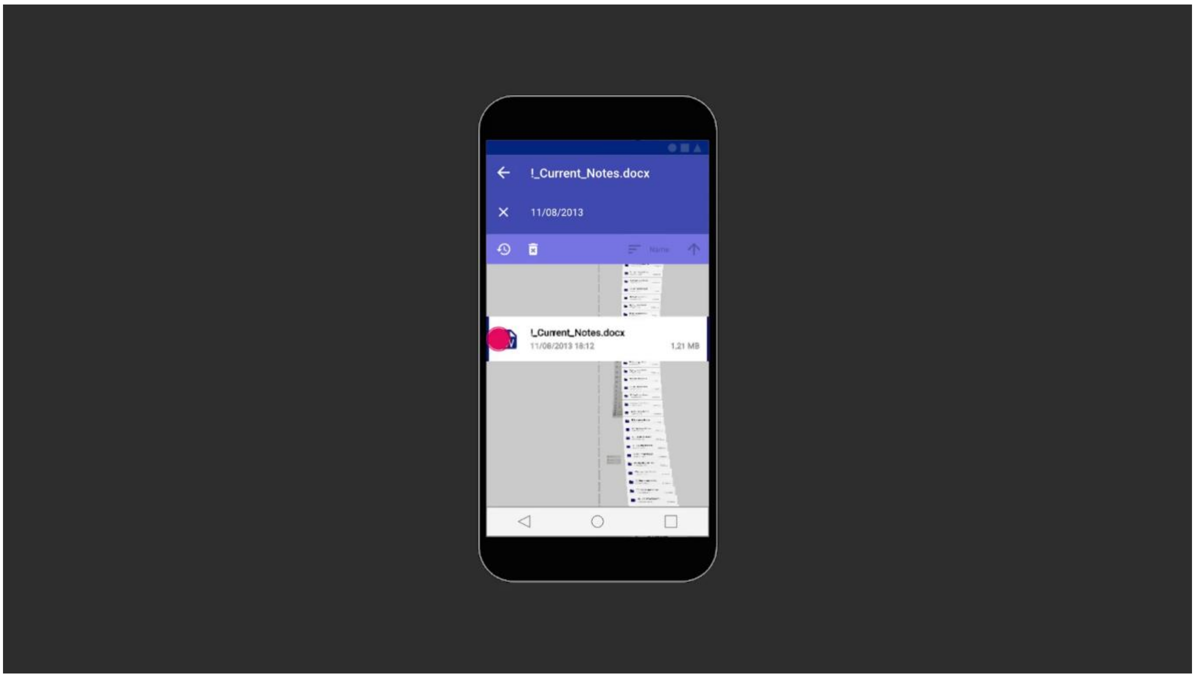



Figure 6.16: Scenario – History Mode for a File in 2D Navigation – Mobile Phone

*Note*. Screenshot from the rendered video result (.mp4 at 1920x1080, 30 frames per second) simulating a System User interacting with my method. Created combining Unity render results and static image overlays. Unity Simulator set to simulate a Samsung Galaxy S7 mobile phone. The view simulates the 2D Navigation mode with a fixed camera where the System User is in History mode and taps on the blue ribbon on the left to see an even older version of the file.

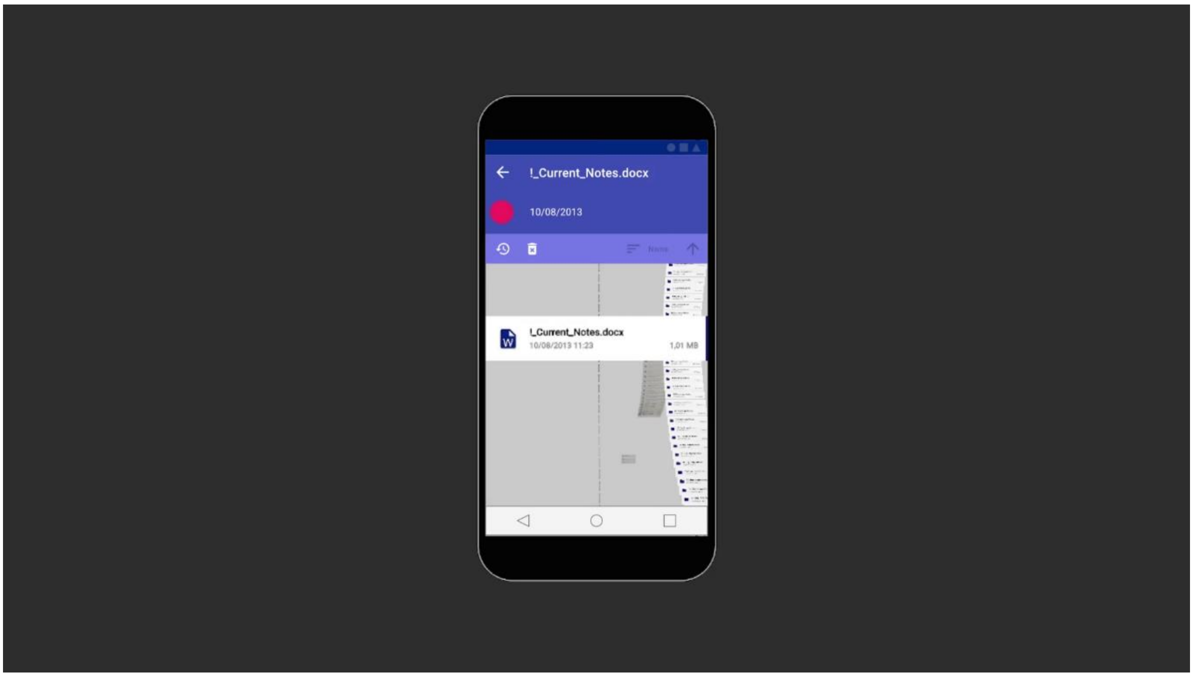



Figure 6.17: Scenario – Exit History Mode in 2D Navigation– Mobile Phone

*Note*. Screenshot from the rendered video result (.mp4 at 1920x1080, 30 frames per second) simulating a System User interacting with my method. Created combining Unity render results and static image overlays. Unity Simulator set to simulate a Samsung Galaxy S7 mobile phone. The view simulates the 2D Navigation mode with a fixed camera where the System User is in History mode and taps on the close icon to exit History mode.

Back in the folder Bug_Hierarchy, the System User opens the contextual overflow menu by tapping on it and selects the option to Enter Overview (see Figure 6.18) to work on their video and image material. In 3D Overview mode (see Figure 6.19) they can see more files and folders at once (*location independence*, see user story ID 18).

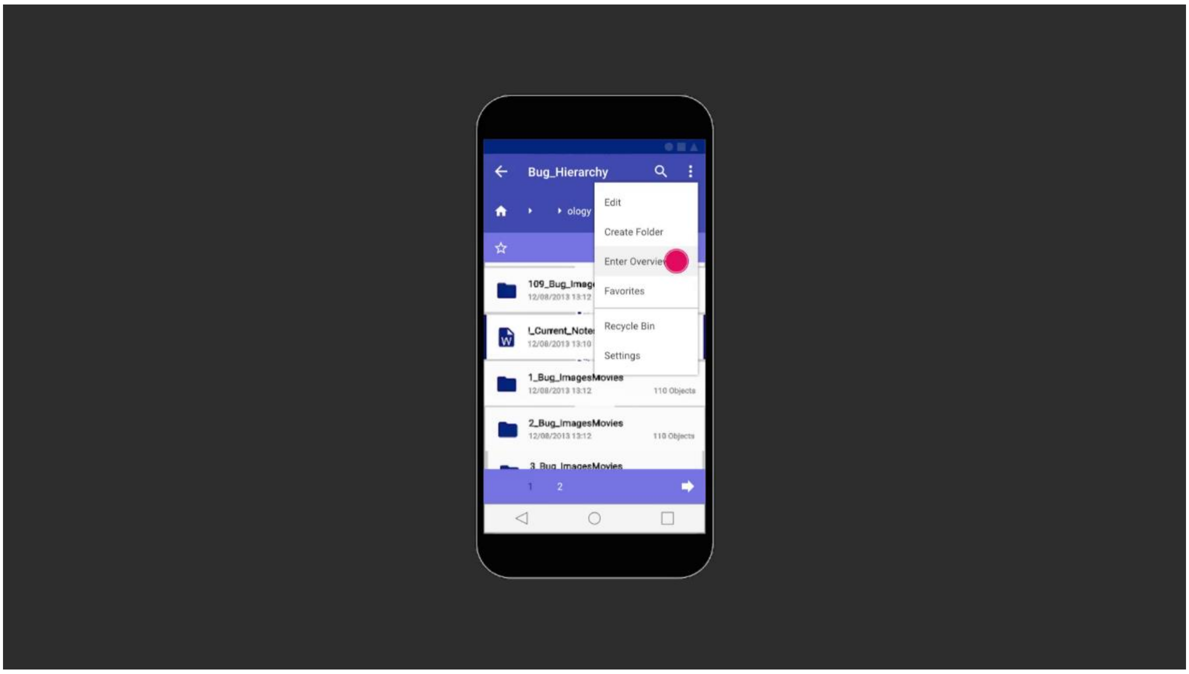



Figure 6.18: Scenario – Enter 3D Overview Mode From 2D Navigation – Mobile Phone

*Note*. Screenshot from the rendered video result (.mp4 at 1920x1080, 30 frames per second) simulating a System User interacting with my method. Created combining Unity render results and static image overlays. Unity Simulator set to simulate a Samsung Galaxy S7 mobile phone. The view simulates the 2D Navigation mode with a fixed camera where the System User taps on the option to enter 3D Overview mode.

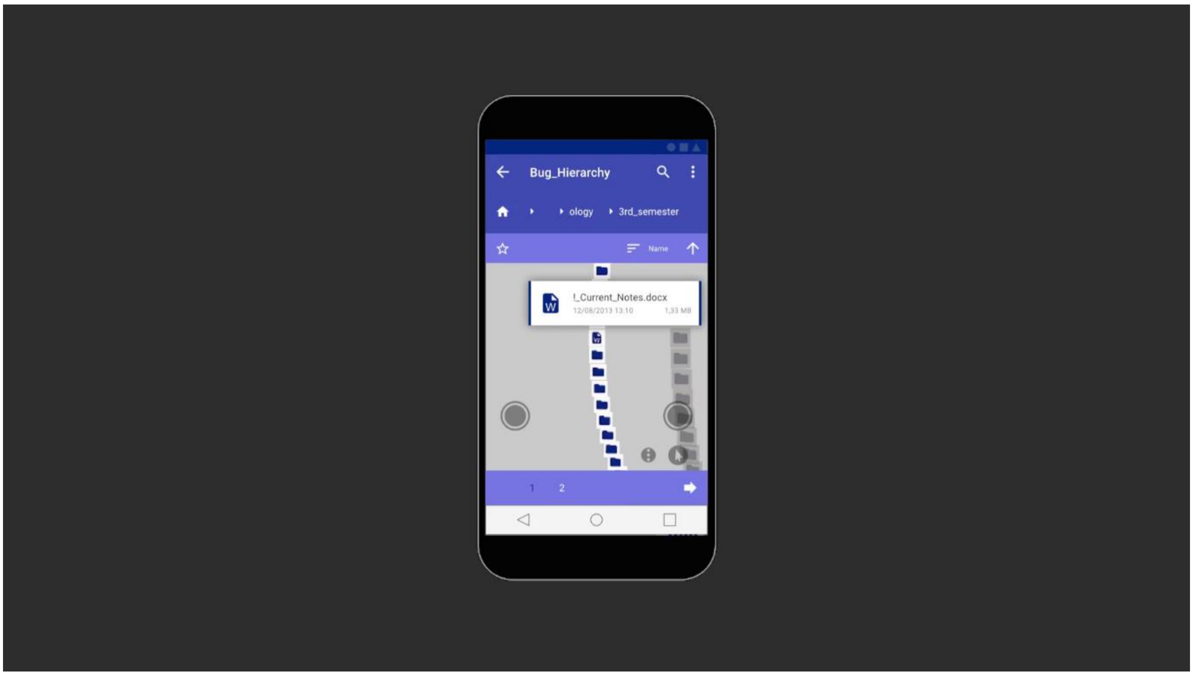



Figure 6.19: Scenario – 3D Overview Mode – Mobile Phone

*Note*. Screenshot from the rendered video result (.mp4 at 1920x1080, 30 frames per second) simulating a System User interacting with my method. Created combining Unity render results and static image overlays. Unity Simulator set to simulate a Samsung Galaxy S7 mobile phone. The view simulates the 3D Overview mode with a movable camera.

Next, in 3D Overview mode the System User taps on the blue ribbon on the right side in the pop up window for !_Current_Notes.docx (see Figure 6.20). Consequently, the System User enters the Related Content mode for this file, this time in 3D Overview. After reassuring that they have not forgotten anything they exit the mode again by tapping on the back arrow at the top left (see Figure 6.21).

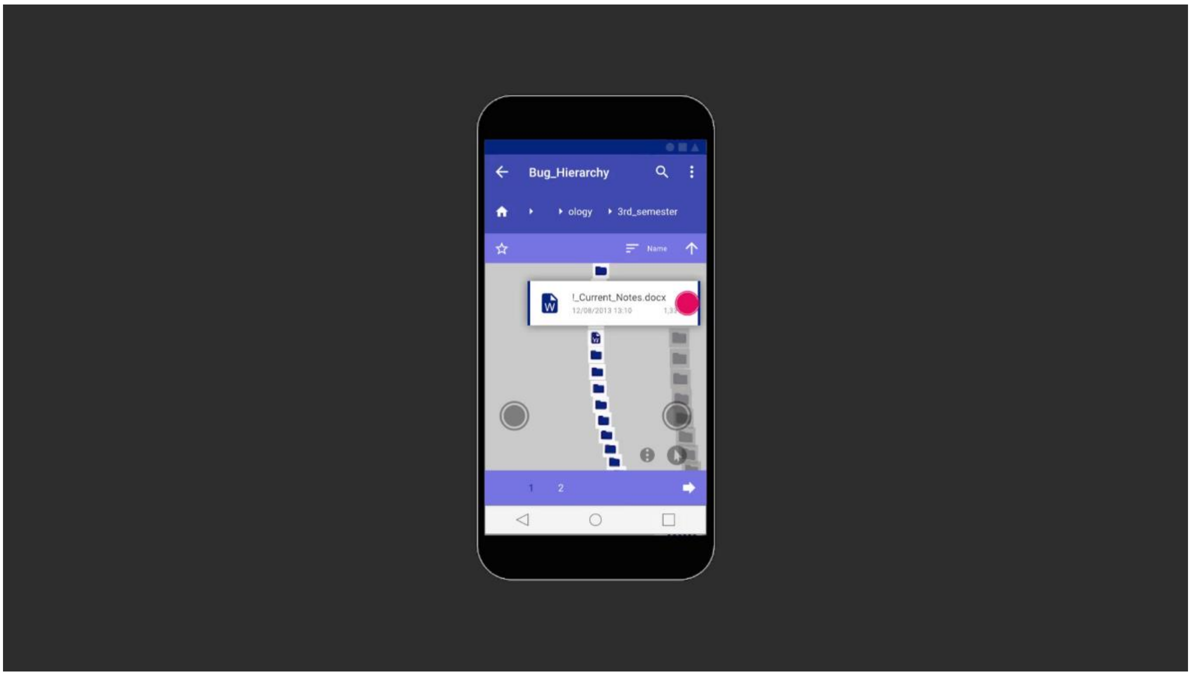



Figure 6.20: Scenario – Enter Related Content Mode in 3D Overview – Mobile Phone

*Note*. Screenshot from the rendered video result (.mp4 at 1920x1080, 30 frames per second) simulating a System User interacting with my method. Created combining Unity render results and static image overlays. Unity Simulator set to simulate a Samsung Galaxy S7 mobile phone. The view simulates the 3D Overview mode with a movable camera where the System User taps on the blue ribbon on the right side to enter Related Content mode.

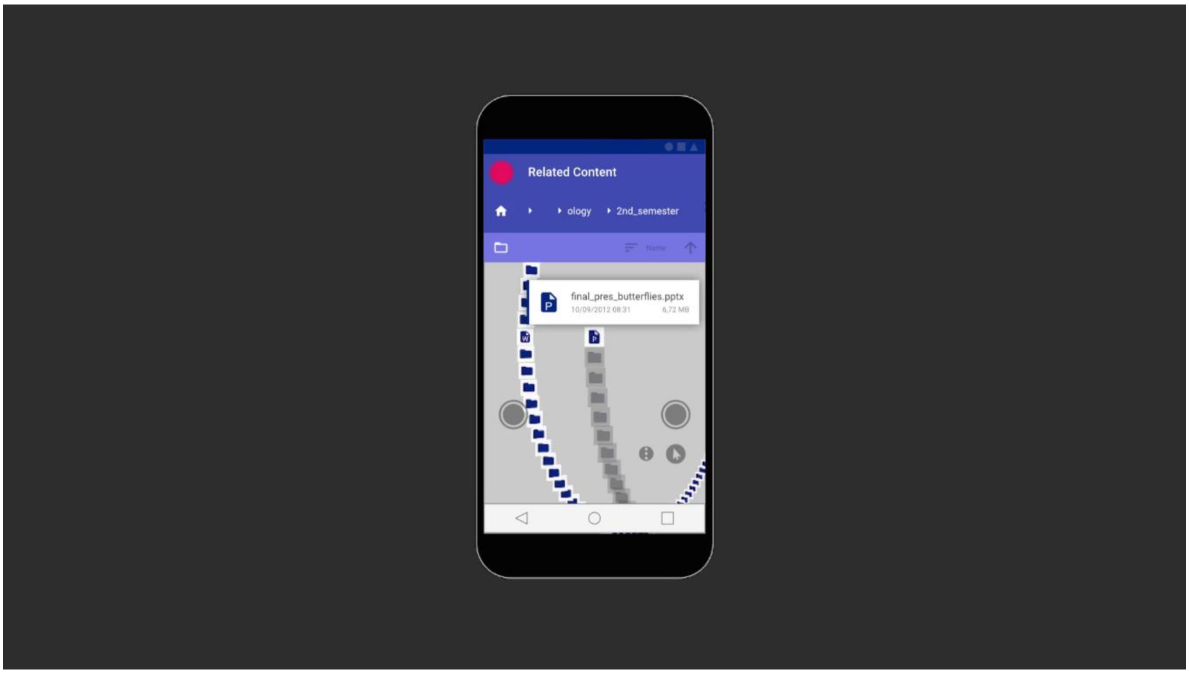



Figure 6.21: Scenario – Exit Related Content Mode in 3D Overview – Mobile Phone

*Note*. Screenshot from the rendered video result (.mp4 at 1920x1080, 30 frames per second) simulating a System User interacting with my method. Created combining Unity render results and static image overlays. Unity Simulator set to simulate a Samsung Galaxy S7 mobile phone. The view simulates the 3D Overview mode with a movable camera where the System User exits out of Related Content mode by tapping on the back arrow.

Next, the System User utilises the virtual joysticks to walk around (*immersion*, see user story ID 7) in their dataset (see Figure 6.22). They arrive at the first subfolder of Bug_Hierarchy called "*1_Bug_ImagesMovies*" (see Figure 6.23). Worth noting that, as soon as the crosshair is moved over a Node Layer of a subfolder it is set to active and the path at the top changes.

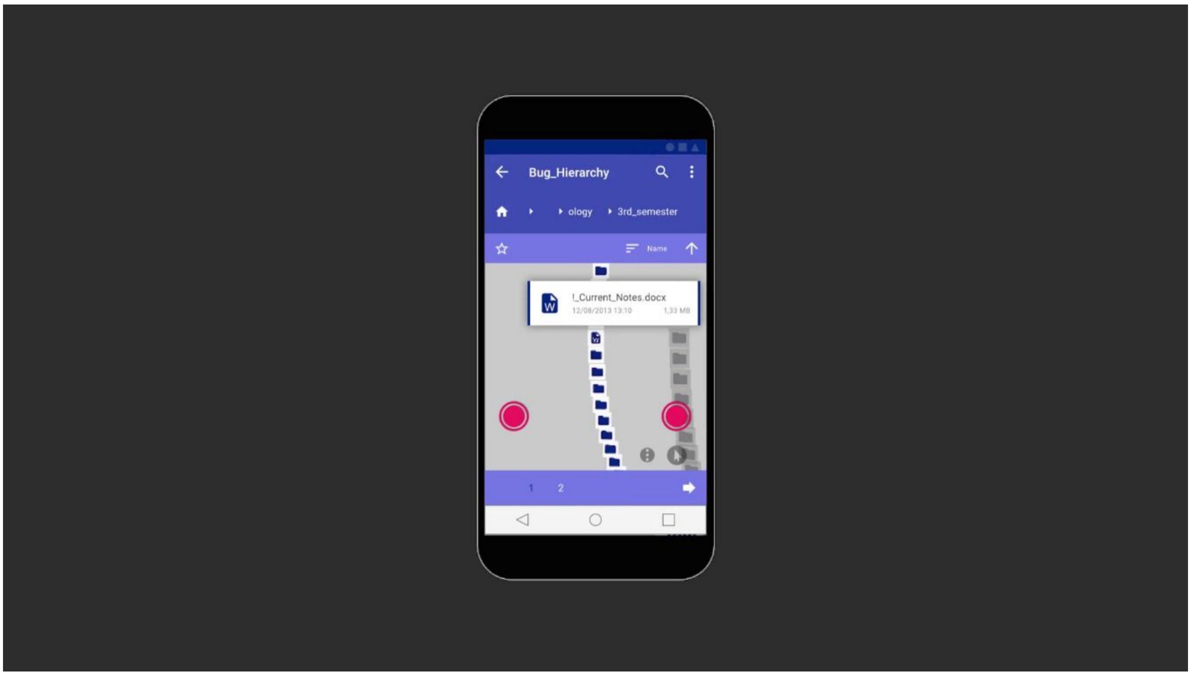



Figure 6.22: Scenario – Initialise Movement in 3D Overview – Mobile Phone

*Note*. Screenshot from the rendered video result (.mp4 at 1920x1080, 30 frames per second) simulating a System User interacting with my method. Created combining Unity render results and static image overlays. Unity Simulator set to simulate a Samsung Galaxy S7 mobile phone. The view simulates the 3D Overview mode with a movable camera where the System User utilises the virtual joysticks to initiate moving in 3D space.

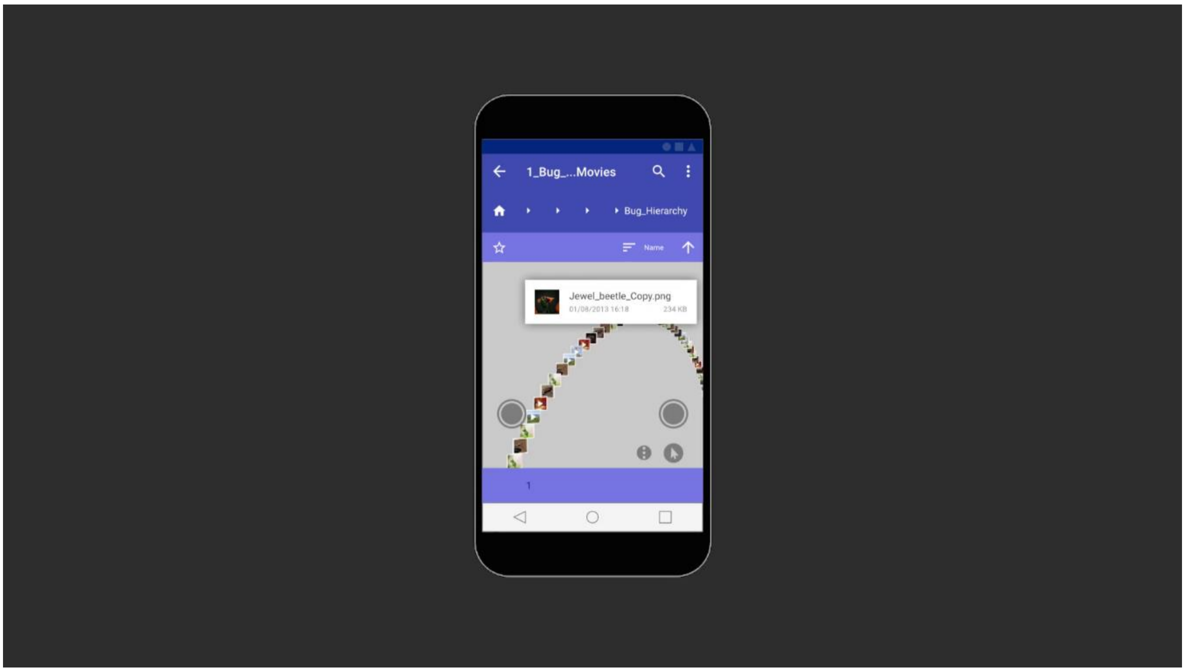



Figure 6.23: Scenario – Subfolder in 3D Overview – Mobile Phone

*Note*. Screenshot from the rendered video result (.mp4 at 1920x1080, 30 frames per second) simulating a System User interacting with my method. Created combining Unity render results and static image overlays. Unity Simulator set to simulate a Samsung Galaxy S7 mobile phone. The view simulates the 3D Overview mode with a movable camera where the System User arrives at a subfolder with many videos and images.

Next, the System User switches to a PC desktop device using a mouse and keyboard as the input devices (see Figure 6.24). They notice that they are still at the same position where they left off from their mobile phone while they can see even more files and contextual information (*screen space utilisation*, see user story ID 12) in the distance for context+focus (*location independence*, see user story ID 18). The user interface is very similar to the mobile phone (*device independence*, see user story ID 17).

Figure 6.24: Scenario – Switch to PC Desktop Device in 3D Overview – PC Desktop

*Note*. Screenshot from the rendered video result (.mp4 at 1920x1080, 30 frames per second) simulating a System User interacting with my method on a PC desktop device. Created combining Unity render results and static image overlays. The view simulates the 3D Overview mode with a movable camera.

Consequently, the System User aims to clean up the folder they are currently in to utilise the videos and images for their project. Therefore, they hold down the control (ctrl) key on their keyboard and left click to select three files (see Figure 6.25). They change their view direction by moving the mouse to put the crosshair above each file they want to select. Then, they press the delete key (del) to move those three files to the recycle bin of the Space. Worth noting that, on a PC desktop device the System User does not need to confirm their intention to delete these files again in a dialog. Figure 6.26 visualises the view after the files got deleted.

Figure 6.25: Scenario – Select Three Files in 3D Overview – PC Desktop

*Note*. Screenshot from the rendered video result (.mp4 at 1920x1080, 30 frames per second) simulating a System User interacting with my method on a PC desktop device. Created combining Unity render results and static image overlays. The view simulates the 3D Overview mode with a movable camera where the System User just selected three files.


Figure 6.26: Scenario – After Deletion of Three Files in 3D Overview – PC Desktop

*Note*. Screenshot from the rendered video result (.mp4 at 1920x1080, 30 frames per second) simulating a System User interacting with my method on a PC desktop device. Created combining Unity render results and static image overlays. The view simulates the 3D Overview mode with a movable camera where the System User just deleted three files.

Next, the System User notices that cleaning up this folder may take a lot of time and that they only want to utilise two items for their project. Therefore, they select these two items by again holding down the control (ctrl) key while left clicking on the items they want to copy to a new location. With the two items selected they press the control (ctrl) and C key to copy the items to the clipboard (see Figure 6.27).

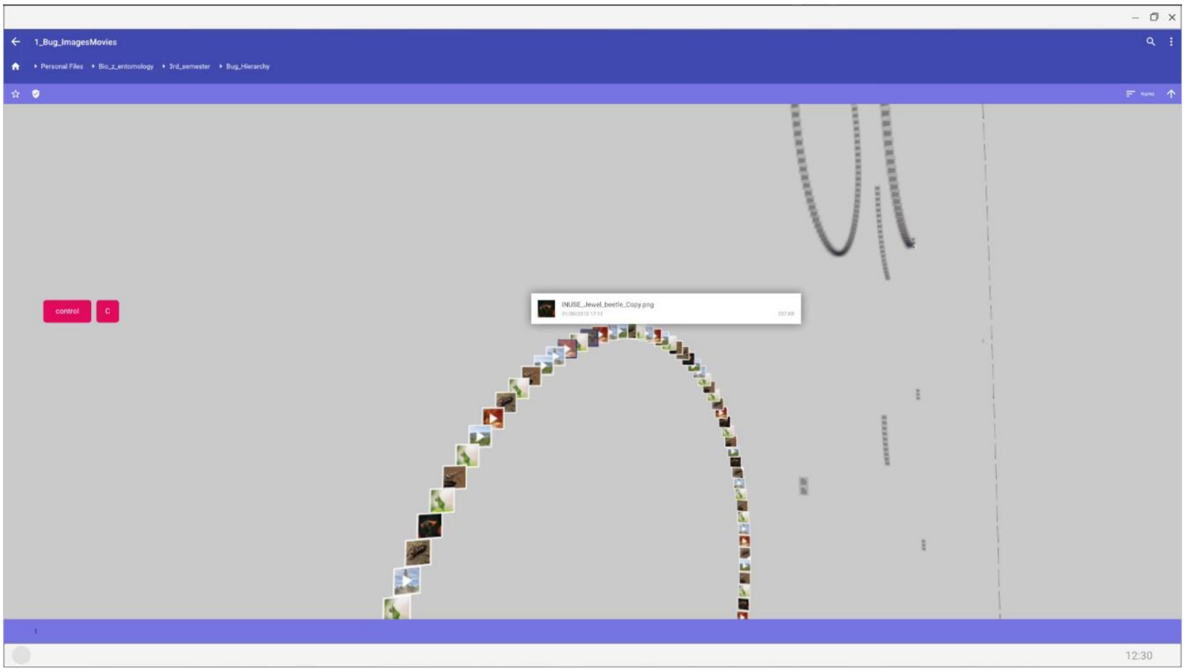



Figure 6.27: Scenario – Copy two Files in 3D Overview – PC Desktop

*Note*. Screenshot from the rendered video result (.mp4 at 1920x1080, 30 frames per second) simulating a System User interacting with my method on a PC desktop device. Created combining Unity render results and static image overlays. The view simulates the 3D Overview mode with a movable camera where the System User copies two files to the clipboard.

With the two items in the clipboard the System Users navigates back to the Bug_Hierarchy folder by moving their mouse towards the respective Node Layer to rotate the camera (see Figure 6.28). When they arrive at their target location the path and highlighting of the nodes changes accordingly (see Figure 6.29).


Figure 6.28: Scenario – Change Camera Rotation in 3D Overview – PC Desktop

*Note*. Screenshot from the rendered video result (.mp4 at 1920x1080, 30 frames per second) simulating a System User interacting with my method on a PC desktop device. Created combining Unity render results and static image overlays. The view simulates the 3D Overview mode with a movable camera where the System User rotates the camera by moving their mouse.

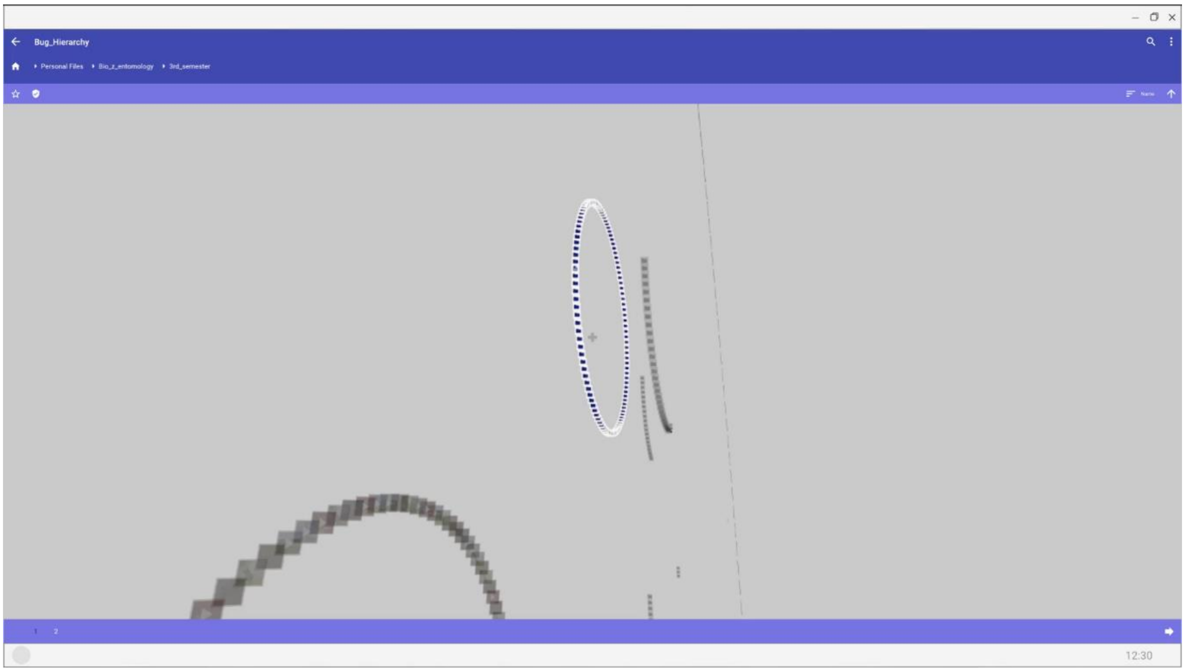


Figure 6.29: Scenario – Focus Change in 3D Overview – PC Desktop

*Note*. Screenshot from the rendered video result (.mp4 at 1920x1080, 30 frames per second) simulating a System User interacting with my method on a PC desktop device. Created combining Unity render results and static image overlays. The view simulates the 3D Overview mode with a movable camera where the System User changed the focus to a new folder.

Next, the System User decides to switch Node Layers in the folder Bug_Hierarchy by first moving the mouse towards the second Node Layer they want to select (see Figure 6.30). As the System User comes closer to the second Node Layer it gets highlighted in the bottom app bar. Consequently, the System User left clicks once on the inactive Node Layer to switch to it (see Figure 6.31). Worth noting that, at this very moment not only the highlighting changes but also the folder contents are exchanged at the next depth level to show the contents of the folders for this second Node Layer (*improve efficiency*, see user story ID 10).

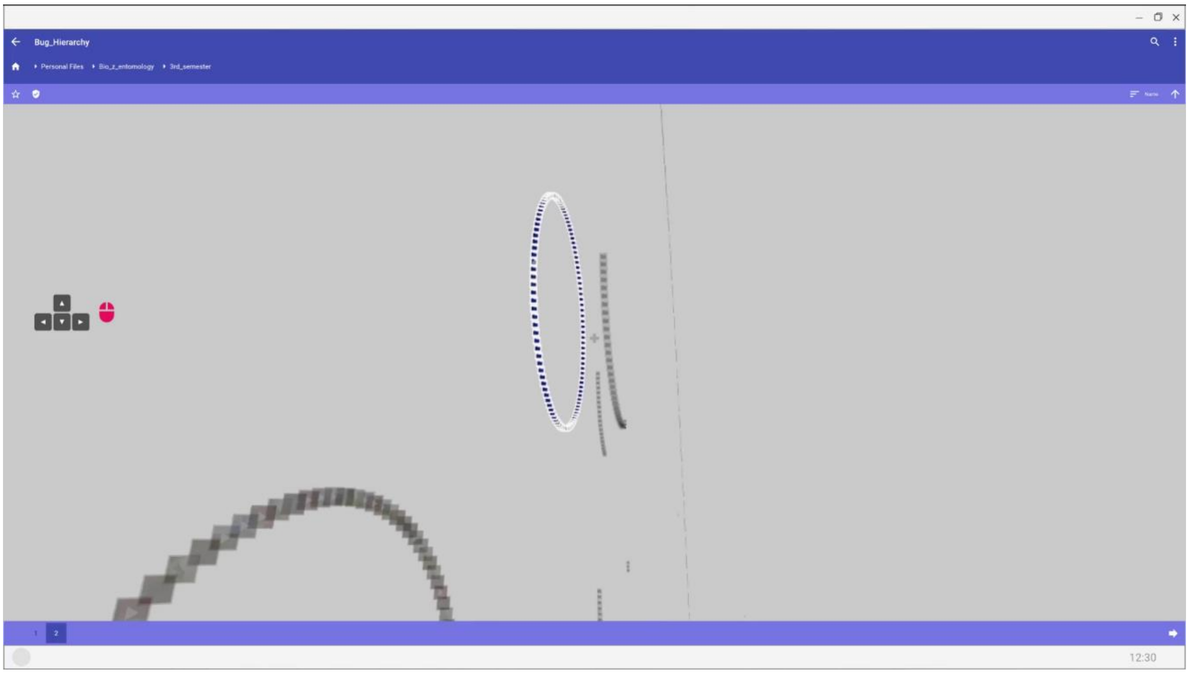



Figure 6.30: Scenario – Change Camera Rotation Towards Second Node Layer in 3D Overview – PC Desktop

*Note*. Screenshot from the rendered video result (.mp4 at 1920x1080, 30 frames per second) simulating a System User interacting with my method on a PC desktop device. Created combining Unity render results and static image overlays. The view simulates the 3D Overview mode with a movable camera where the System User rotates their view towards the second Node Layer of a folder.

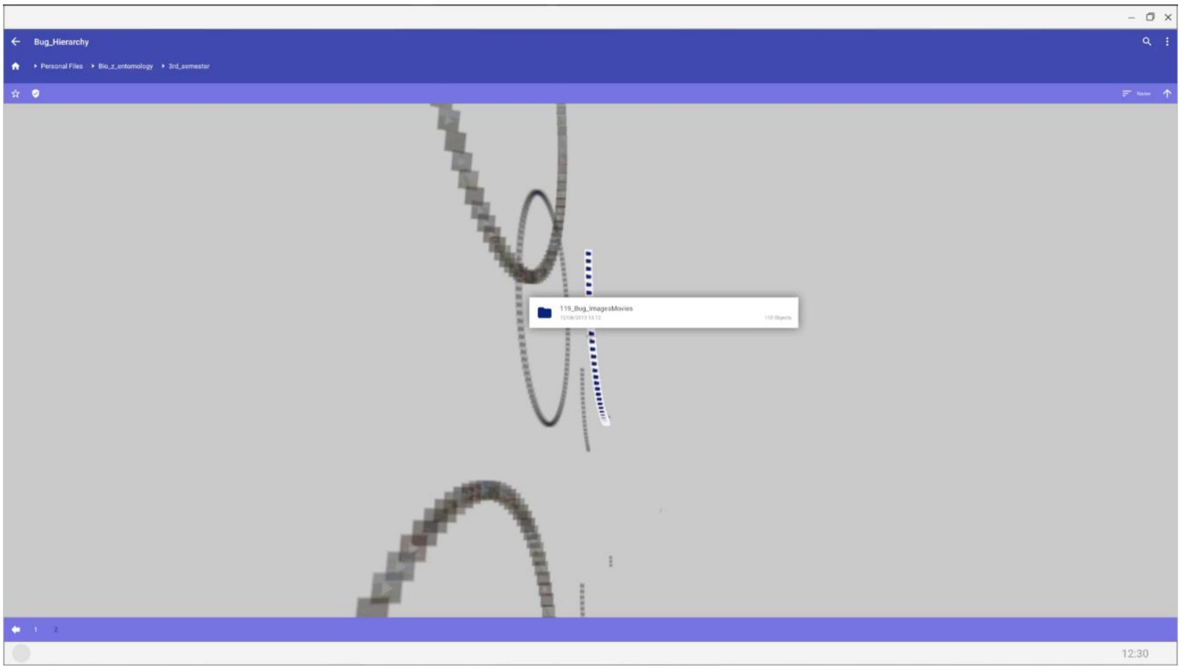



Figure 6.31: Scenario – Switch to Second Node Layer in 3D Overview – PC Desktop

*Note*. Screenshot from the rendered video result (.mp4 at 1920x1080, 30 frames per second) simulating a System User interacting with my method on a PC desktop device. Created combining Unity render results and static image overlays. The view simulates the 3D Overview mode with a movable camera where the System User switched to a second Node Layer of a folder.

The System User aims to create a new folder in Bug_Hierarchy to store their videos and images they ultimately want to utilise in their project. Therefore, they right click with their mouse to open the contextual overflow menu. This unlocks the mouse cursor. Consequently, they select (left click) the options "*Create*" and "*Folder*" to create a new folder (see Figure 6.32). A dialog appears where they enter the name for their folder utilising the keyboard (see Figure 6.33). They name their folder "*Z_FIN_Bug_ImagesMovies*" and confirm the creation of the folder by pressing the enter key. This operation results in the new folder being created. The camera is automatically moved to the new location of this folder and it gets highlighted. From this position the System User left clicks twice (double click) to automatically move to its location and see the content of this folder (see Figure 6.34).

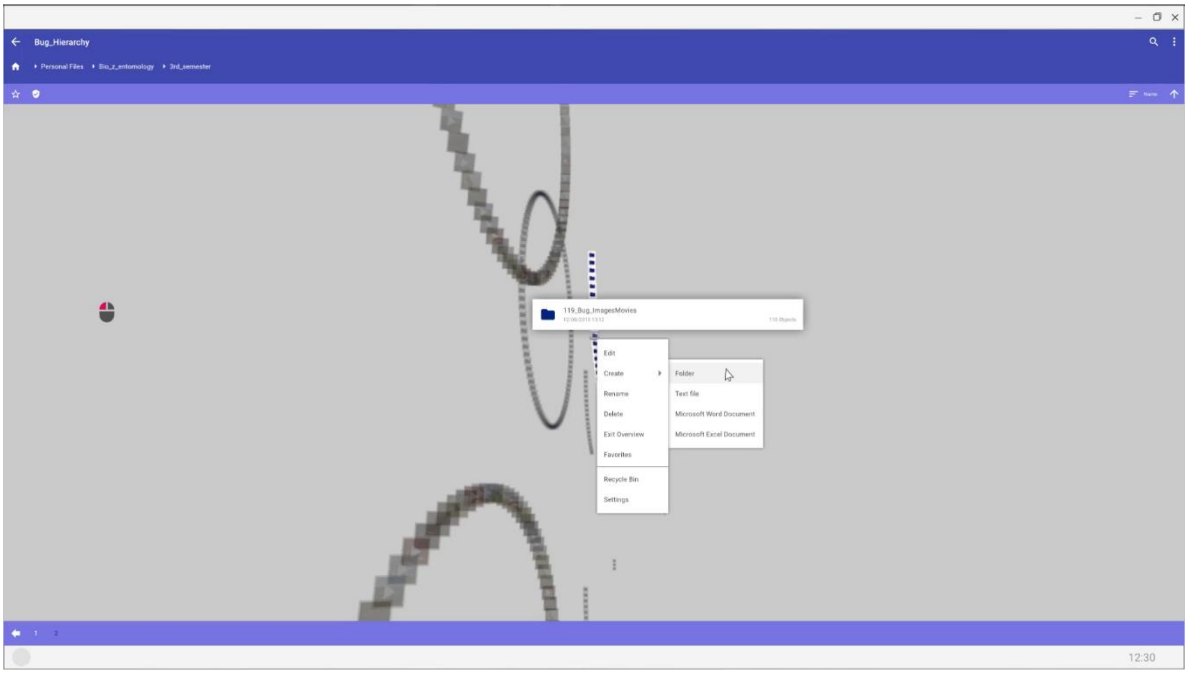



Figure 6.32: Scenario – Select Option to Create a new Folder in 3D Overview – PC Desktop

*Note*. Screenshot from the rendered video result (.mp4 at 1920x1080, 30 frames per second) simulating a System User interacting with my method on a PC desktop device. Created combining Unity render results and static image overlays. The view simulates the 3D Overview mode with a movable camera where the System User selects the option to create a new folder.

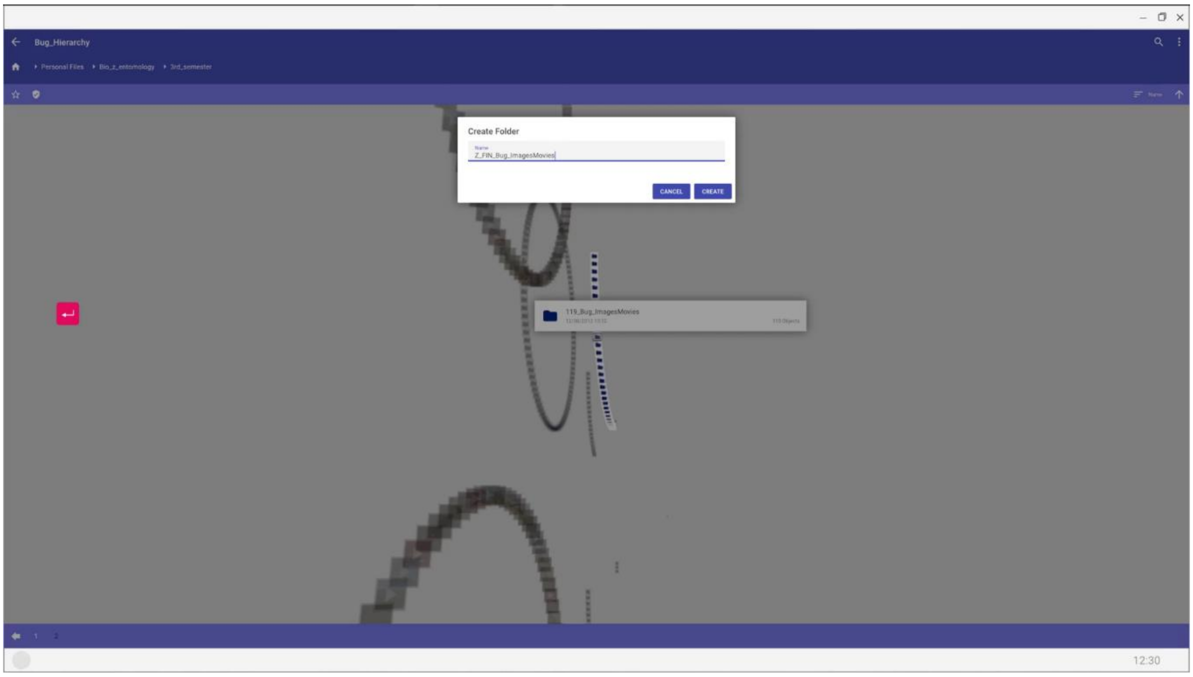



Figure 6.33: Scenario – Dialog to Create a new Folder in 3D Overview – PC Desktop

*Note*. Screenshot from the rendered video result (.mp4 at 1920x1080, 30 frames per second) simulating a System User interacting with my method on a PC desktop device. Created combining Unity render results and static image overlays. The view simulates the 3D Overview mode with a movable camera where the System User confirms the creation of a new folder.


Figure 6.34: Scenario – Navigate to Folder in 3D Overview – PC Desktop

*Note*. Screenshot from the rendered video result (.mp4 at 1920x1080, 30 frames per second) simulating a System User interacting with my method on a PC desktop device. Created combining Unity render results and static image overlays. The view simulates the 3D Overview mode with a movable camera where the System User carries out a double click with the left mouse button to move to a folders location.

In this new empty folder Z_FIN_Bug_ImagesMovies the System User presses on the control (ctrl) and V key to paste the two files they copied earlier (see Figure 6.35). Consequently, the two files are inserted into the folder (see Figure 6.36).

Figure 6.35: Scenario – Empty Folder in 3D Overview – PC Desktop

*Note*. Screenshot from the rendered video result (.mp4 at 1920x1080, 30 frames per second) simulating a System User interacting with my method on a PC desktop device. Created combining Unity render results and static image overlays. The view simulates the 3D Overview mode with a movable camera where the System User views an empty folder and initiates a paste operation.

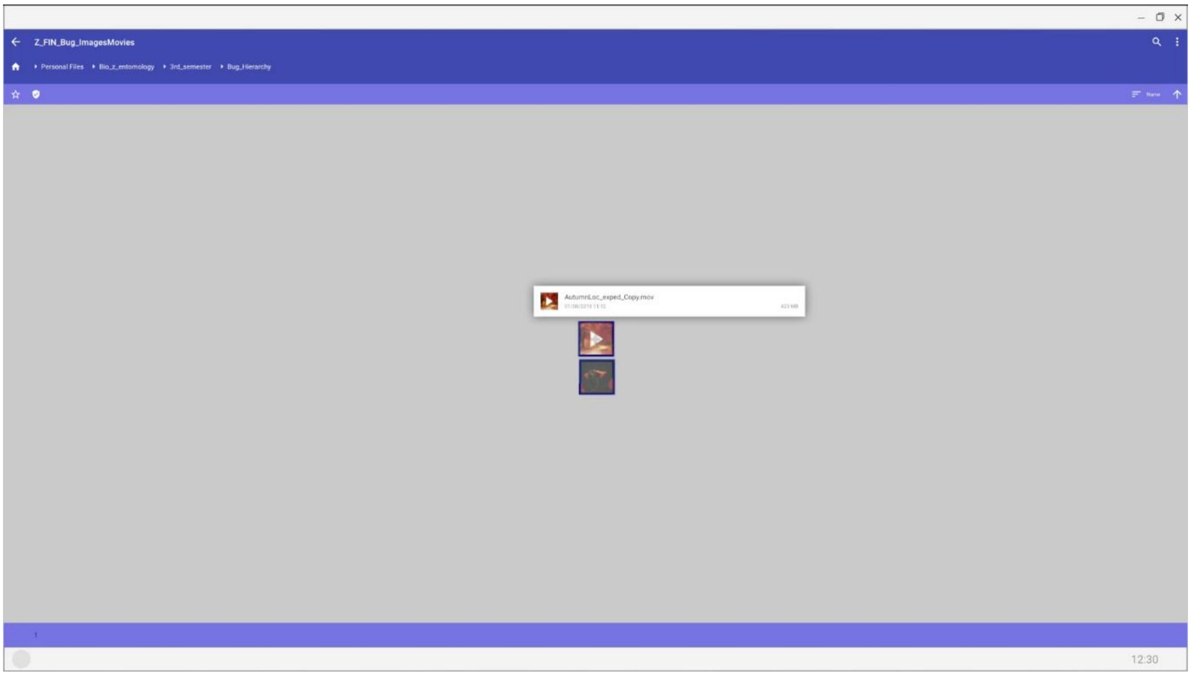


Figure 6.36: Scenario – Result of Paste Operation in 3D Overview – PC Desktop

*Note*. Screenshot from the rendered video result (.mp4 at 1920x1080, 30 frames per second) simulating a System User interacting with my method on a PC desktop device. Created combining Unity render results and static image overlays. The view simulates the 3D Overview mode with a movable camera where the System User just pasted two files from the clipboard.

Next, the System User aims to identify more suitable images and videos for their project. Therefore, they utilise the arrow keys on the keyboard to move away from the folder Z_FIN_Bug_ImagesMovies and their mouse to rotate the camera to the left (see Figure 6.37). Therefore, they gain an efficient overview of the videos and images in the other folders (*improve efficiency*, see user story ID 10), for example, in folder "*138_Bug_ImagesMovies*" (see Figure 6.38).


Figure 6.37: Scenario – Reposition Camera in 3D Overview – PC Desktop

*Note*. Screenshot from the rendered video result (.mp4 at 1920x1080, 30 frames per second) simulating a System User interacting with my method on a PC desktop device. Created combining Unity render results and static image overlays. The view simulates the 3D Overview mode with a movable camera where the System User changes their perspective using their keyboard and mouse.

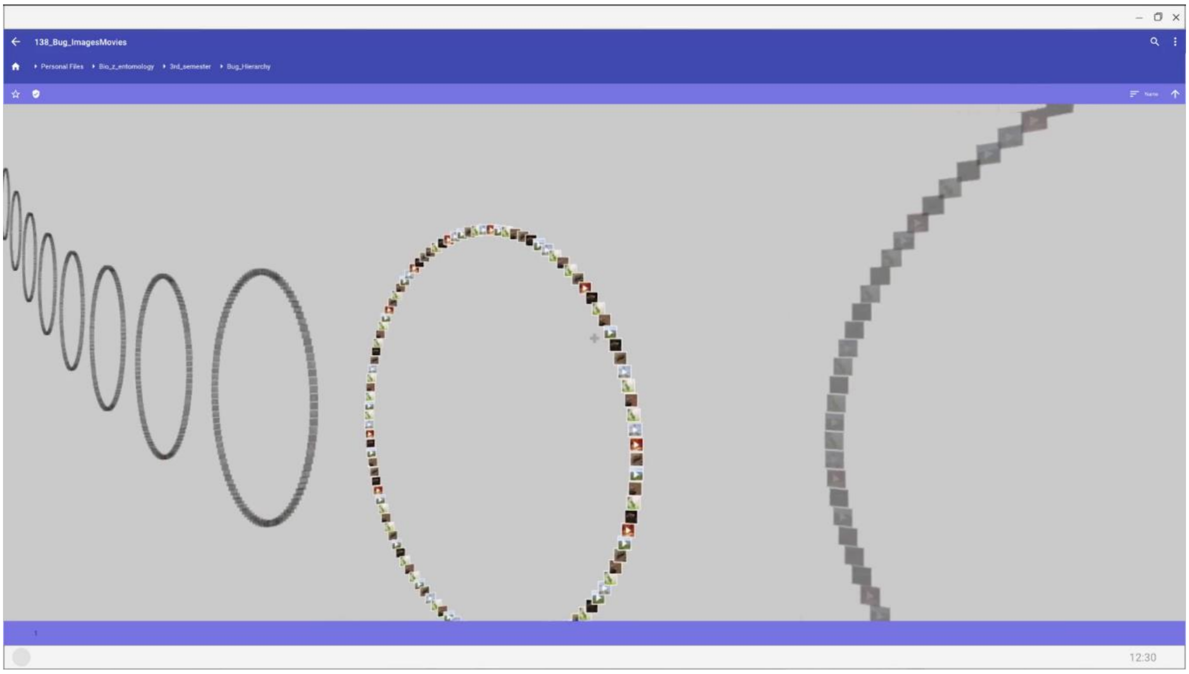



Figure 6.38: Scenario – Rotate Further to the Left in 3D Overview – PC Desktop

*Note*. Screenshot from the rendered video result (.mp4 at 1920x1080, 30 frames per second) simulating a System User interacting with my method on a PC desktop device. Created combining Unity render results and static image overlays. The view simulates the 3D Overview mode with a movable camera where the System User rotates the camera further to the left using their mouse.

The System User halts at the folder "*137_Bug_ImagesMovies*" to utilise its videos and images. They right click with their mouse to open the contextual overflow menu and select the option Exit Overview using a left click with their mouse (see Figure 6.39). Consequently, they switch to the 2D Navigation mode for this folder (see Figure 6.40).

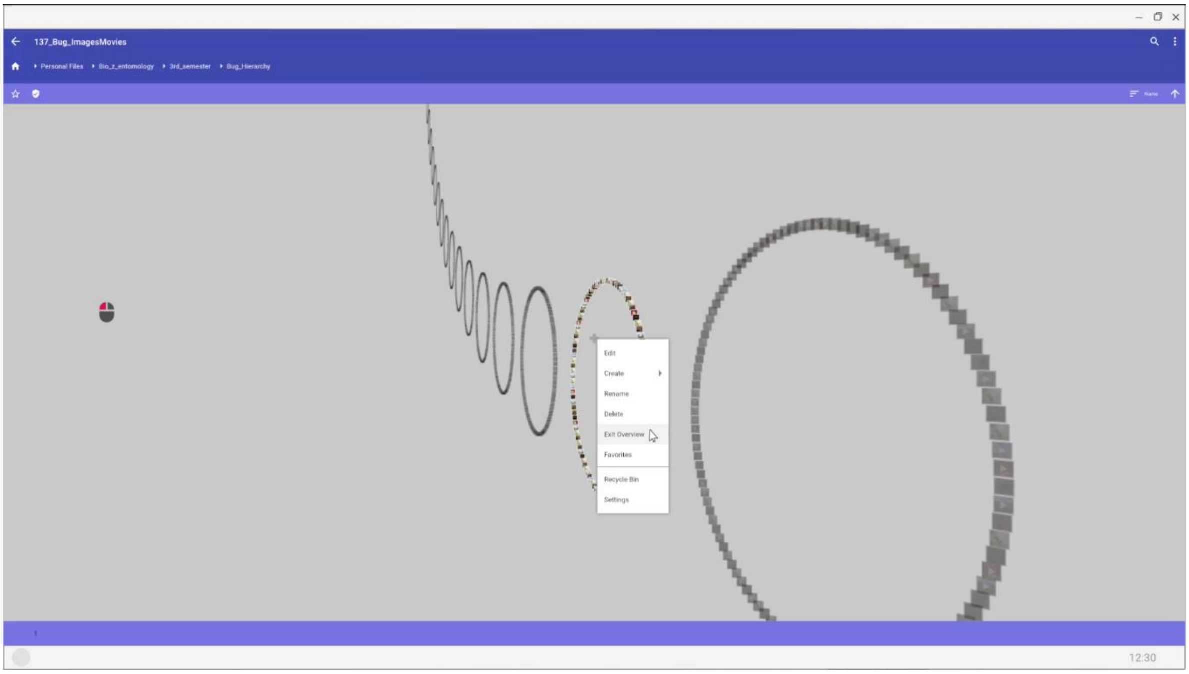



Figure 6.39: Scenario – Exit out of 3D Overview – PC Desktop

*Note*. Screenshot from the rendered video result (.mp4 at 1920x1080, 30 frames per second) simulating a System User interacting with my method on a PC desktop device. Created combining Unity render results and static image overlays. The view simulates the 3D Overview mode with a movable camera where the System User selects the option to exit out of 3D Overview mode.

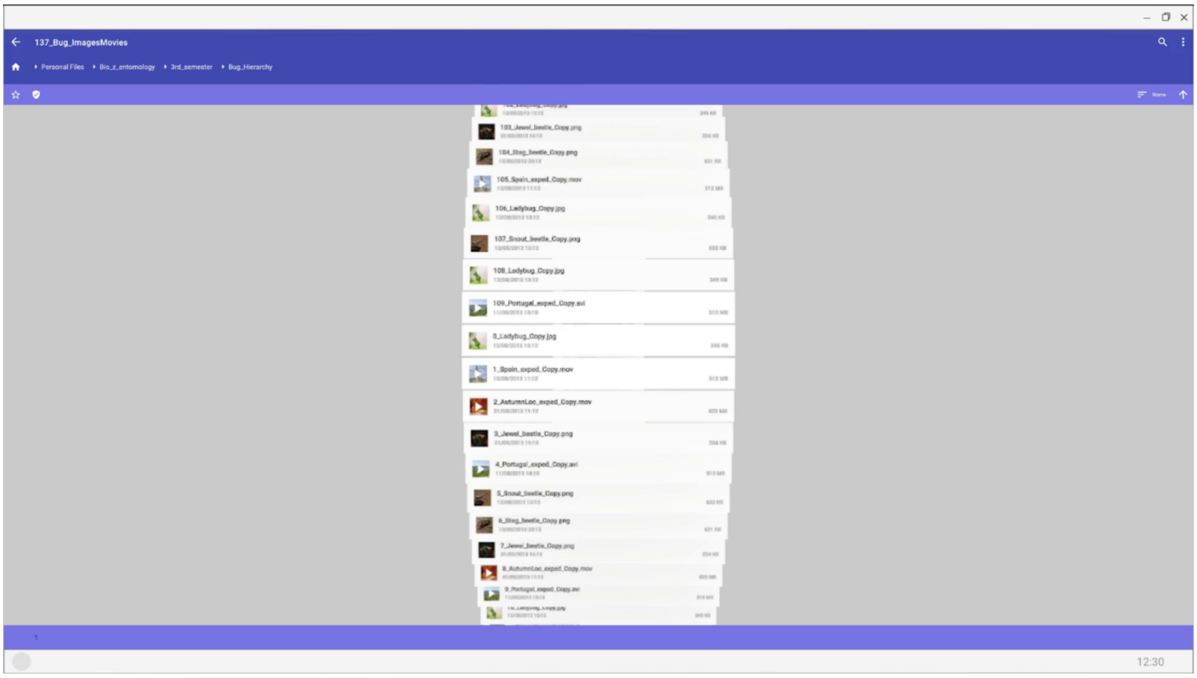


Figure 6.40: Scenario – Contents of a Folder in 2D Navigation – PC Desktop

*Note*. Screenshot from the rendered video result (.mp4 at 1920x1080, 30 frames per second) simulating a System User interacting with my method on a PC desktop device. Created combining Unity render results and static image overlays. The view simulates the 2D Navigation mode with a fixed camera where the System User just exited out of 3D Overview mode.

In the folder 137_Bug_ImagesMovies the System User decides to utilise the video called "*4_Portugal_exped_Copy.avi*" for their project. To identify all the files, they want to utilise they decide to rename the respective files. Therefore, they right click on the first file they want to use to open the contextual overflow menu and left click on the option "*Rename*" (see Figure 6.41). A dialog appears where they change the name of the file to "*4_Portugal_exped_Copy_TAKEvPRES.avi*" using their keyboard (see Figure 6.42). They confirm their change by pressing the enter key. Consequently, the name of the file gets changed in the visualisation (see Figure 6.43).

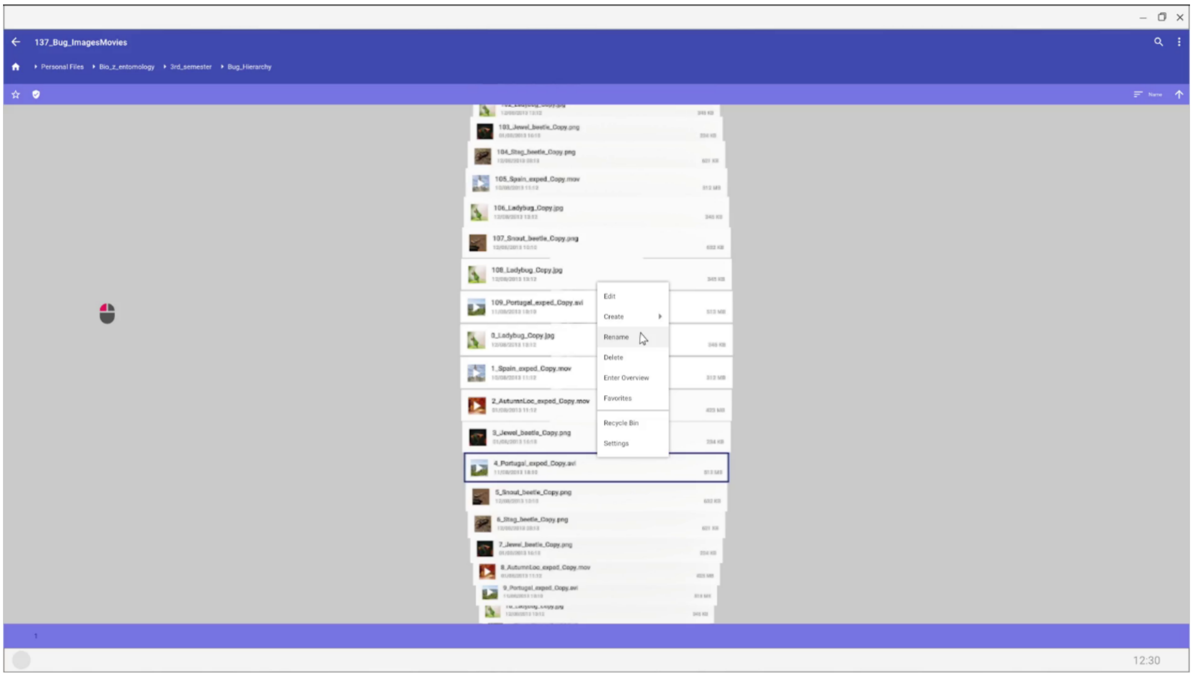


Figure 6.41: Scenario – Initiate Renaming a File in 2D Navigation – PC Desktop

*Note*. Screenshot from the rendered video result (.mp4 at 1920x1080, 30 frames per second) simulating a System User interacting with my method on a PC desktop device. Created combining Unity render results and static image overlays. The view simulates the 2D Navigation mode with a fixed camera where the System User selects the option to rename a file.

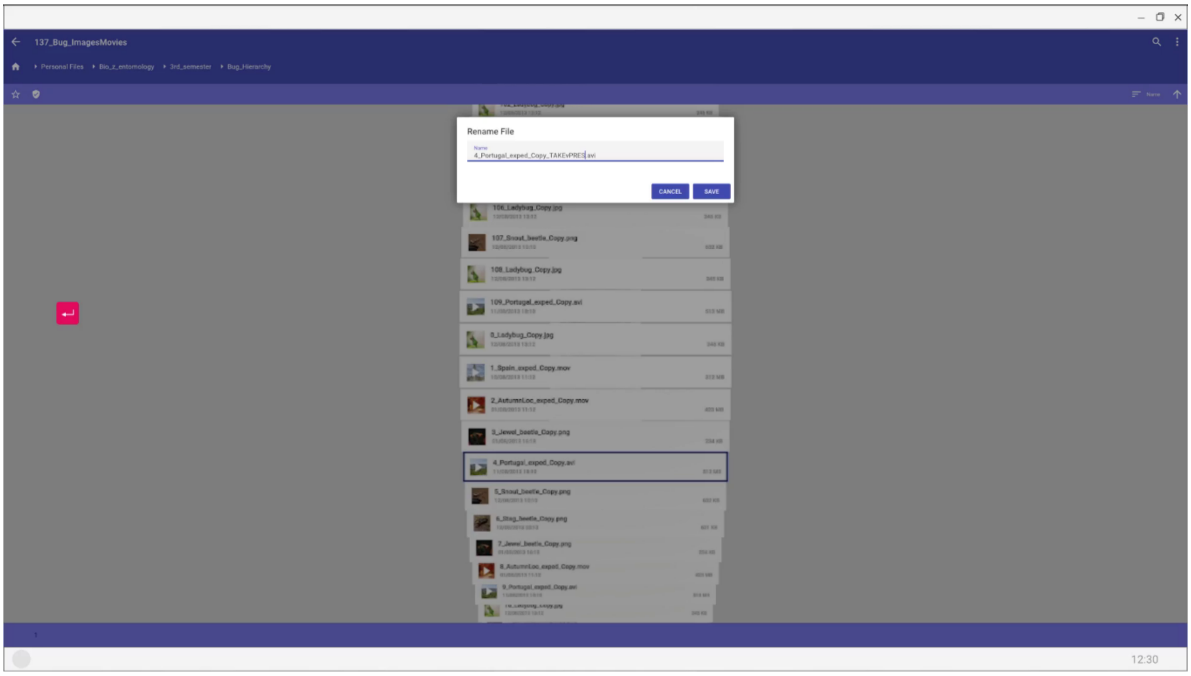



Figure 6.42: Scenario – Dialog to Change a File Name in 2D Navigation – PC Desktop

*Note*. Screenshot from the rendered video result (.mp4 at 1920x1080, 30 frames per second) simulating a System User interacting with my method on a PC desktop device. Created combining Unity render results and static image overlays. The view simulates the 2D Navigation mode with a fixed camera where the System User just confirms the change of a file name.

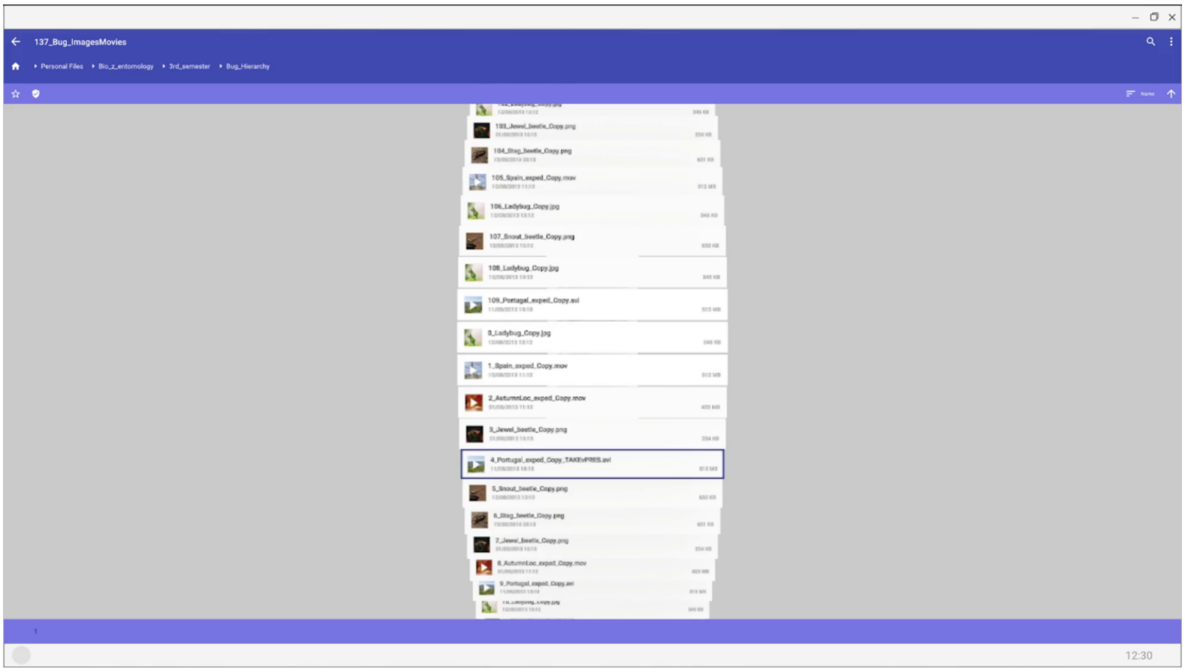


Figure 6.43: Scenario – Renamed File in 2D Navigation – PC Desktop

*Note*. Screenshot from the rendered video result (.mp4 at 1920x1080, 30 frames per second) simulating a System User interacting with my method on a PC desktop device. Created combining Unity render results and static image overlays. The view simulates the 2D Navigation mode with a fixed camera where the System User just changed the name of a file.

## 6.7 Final Evaluation

For evaluation I aimed to present my scenario-based, video-based prototype (see Section 6.6) to participants and subsequently ask them to fill out a questionnaire. Participants were staff and students of the University of Gloucestershire. I utilised as a standard questionnaire the User Experience Questionnaire (UEQ) (see Section 3.8.3). For presentation I chose to produce a video including a voice over to show and explain the prototype. My presentation was structured as follows:

- Introducing the overall aim of my new system including my conceptual design from the ecological perspective. Therefore, I showed a final version of my conceptual design from the ecological perspective where I adjusted the orientation of the node layout for all three devices at the bottom (see Figure 6.44). Worth noting that, for the aim I highlighted the following points which I aimed to improve with my system:

- The option to view data on any device (mobile or PC desktop) using one interface.
- The idea to remain at the same position when switching devices.
- To remove the need to manage multiple storage locations like Google Drive.
- Be able to see more files and folders at once as the visualisation utilises the available screen space of the respective device.
- Allow precise navigation in 2D Navigation mode similar to common list-based interaction common in standard applications.
- Additional 3D Overview mode to explore data by walking through it while the overall layout stays the same.
- The option to utilise contextual information in this native 3D view, for example, to see versions of files or similar content.

- Outlining the scenario for my video-based prototype and guiding the participants through the usage of the prototype. My guidance through this section was similar to my descriptions in Section 6.6. Worth noting that, after the scenario I added one static image as an appendix. This image shows an example using the method on a tablet device (see Figure 6.45).
- At the end, participants were asked to fill out the UEQ using a link to SurveyMonkey which I provided.

Worth noting that, the final video I showed to research participants included the following remarks at the start of the part showing the scenario (grammatical and spelling mistakes were corrected):

- *"Selection and movement in the video are simulated using animation"*
- *"Loading times during navigation were shortened (cut)"*

- *"Navigation speed was reduced"*
- *"Prototype built using a static data set"*
- *"Supplemental/Alternative functionalities/controls like manage "Favourites" are not implemented/shown in detail"*
- *"Prototype implemented using Unity engine; static images were used as overlays for app and system bars, buttons, menus, dialogs, virtual joysticks and buttons, notes, input text fields, pop ups and to highlight interaction"*
- *"Prototype designed for Android and Microsoft Windows desktop PC"*

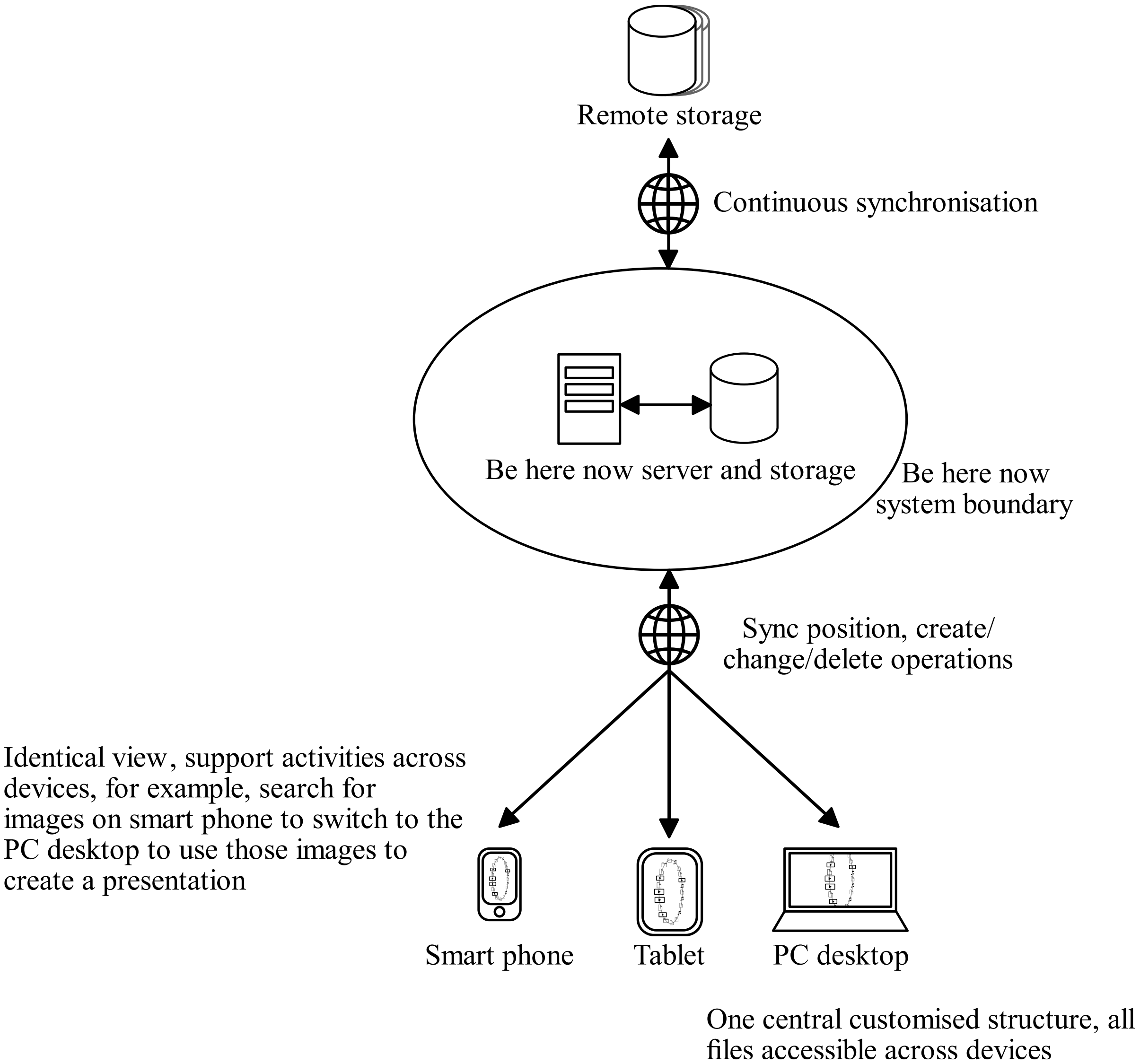



Figure 6.44: Final Conceptual Design From the Ecological Perspective

*Note*. My final conceptual design from the ecological perspective for my novel method.

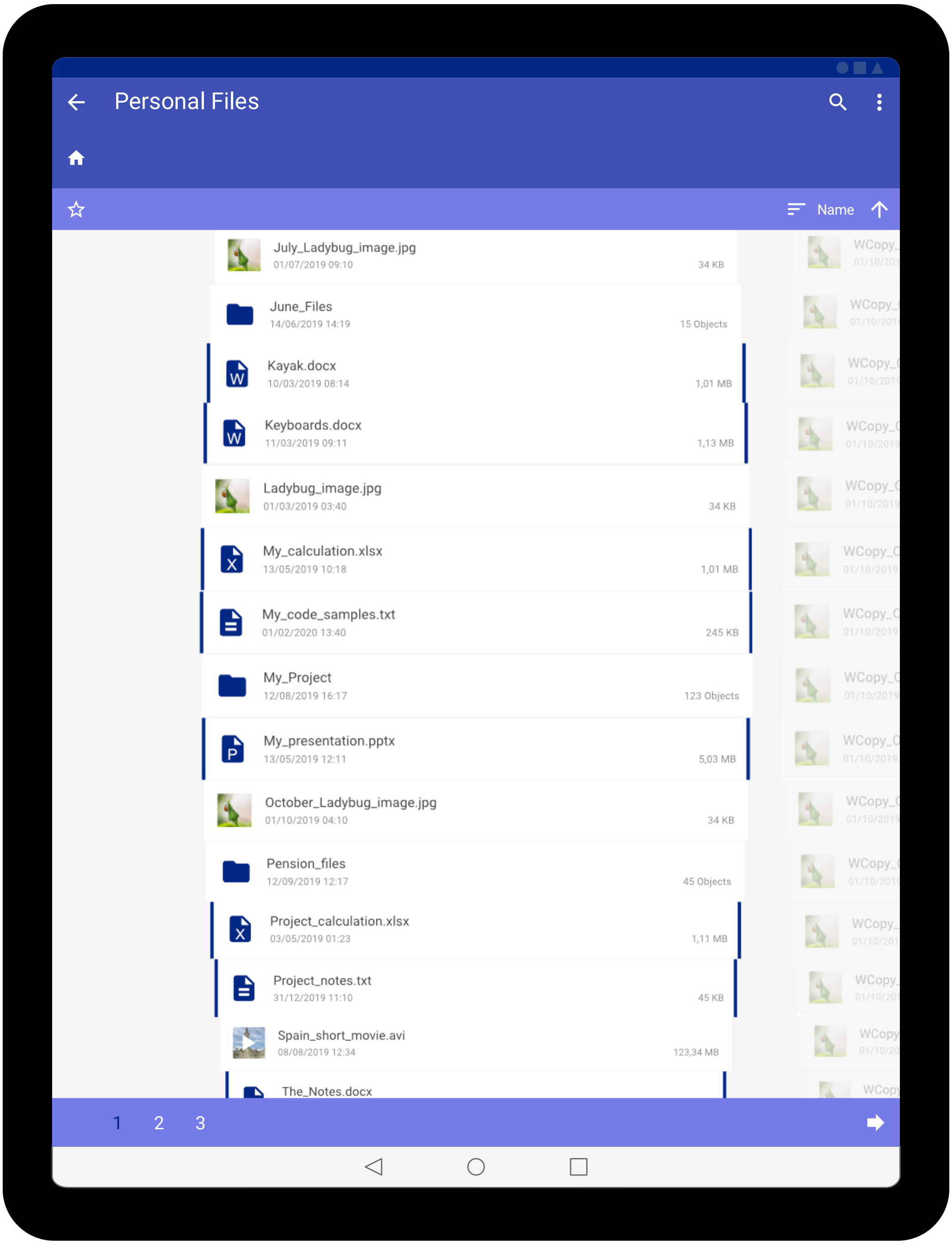


Figure 6.45: Video Appendix – 2D Navigation on a Tablet Device

*Note*. Wireframe showing 2D Navigation mode on a tablet device. Designed for a tablet with a screen resolution of 982x1309 pixels at 160 ppi. Wireframe created using Affinity Designer on Windows.

To contextualise my results I utilised the UEQ handbook by Schrepp (2023). According to Schrepp (2023) the UEQ consists of six scales and 26 items (questions). They explained, that the first scale attractiveness summarises the overall impression of a system.

The second scale perspicuity may indicate how approachable or learnable a product is. The efficiency scale measures if users can carry out their tasks efficiently. The fourth scale dependability may show if the user feels that they are in control regarding the interaction. The fifth scale stimulation may indicate if the users are excited and motivated to actually use the system. Finally, the novelty scale indicates if the solution is received as being creative and innovative or in other words interesting.

As described in Appendix A I utilised the Microsoft Excel based Data Analysis Tools to work on my results. 13 participants completed my questionnaire. I imported their answers into the Data Analysis Tools for a first review regarding data quality. There is a feature in the Data Analysis Tools to identify inconsistent answers based on two heuristics. According to Schrepp (2023) the first heuristic measures how consistent the answers of a participant are for one scale. They explained that a response should be considered as suspicious if two or three scales have a distance of more than three between the worst and the best answer within one scale. Also, they described that they measure the number of identical answers of a participant as a second heuristic. They recommended to exclude the answers of a participant if they chose the same answer, for example, the first option, more than fifteen times.

According to their recommendations I have excluded two users from the result dataset who have chosen the same answer 17 times. Consequently, I analysed the answers of 11 participants using the Data Analysis Tools.

The Data Analysis Tools calculates the means for each scale and visualises the result in a bar chart. According to Schrepp (2023) the usual interpretation of the results is that values between -0.8 and +0.8 are considered as a neutral evaluation while values greater than 0.8 are considered positive and values below -0.8 represent a negative evaluation. They explained that the scales range from -3 to +3, so from very bad to very good. Also, they detailed that the error bars in their bar chart represent the 95 percent confidence intervals of the scale mean.

Figure 6.46 shows the bar char for my results. It visualises the following values:

- Attractiveness: 0.076 (mean), 0.529 (confidence interval)
- Perspicuity: -0.341 (mean), 0.768 (confidence interval)
- Efficiency: -0.477 (mean), 0.688 (confidence interval)
- Dependability: 0.250 (mean), 0.418 (confidence interval)
- Stimulation: 0.545 (mean), 0.663 (confidence interval)
- Novelty: 0.932 (mean), 0.648 (confidence interval)

To summarise, the results show a neutral evaluation result for all scales except for novelty where the evaluation can be considered as positive. I consider the results for the confidence intervals to be quite large. Hence, I argue that this is most likely a result of the rather small sample size of only 11 participants. This dependency between the number of participants and the size of the error bar was also described by Schrepp (2023).

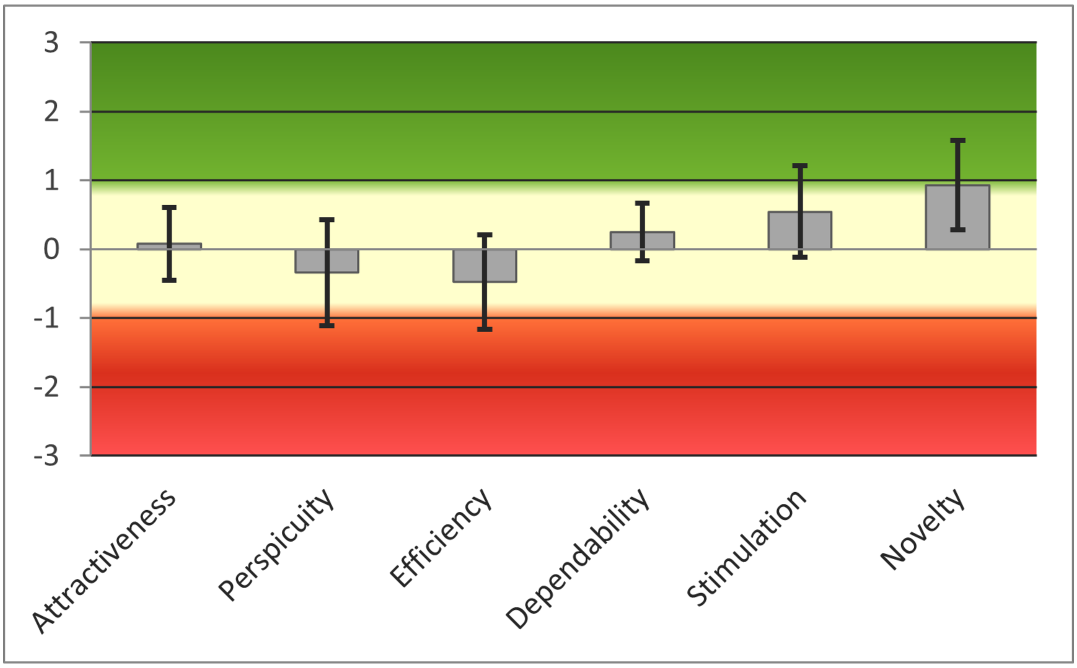



Figure 6.46: UEQ Results – Means of the Scales – Bar Chart

*Note*. Bar chart visualising the means of the UEQ scales for my results. The error bars show the 95 percent confidence interval of the mean. This result was created using the UEQ Data Analysis Tools in Microsoft Excel.

Figure 6.47 visualises the individual means for all items (questions). The results are quite consistent with the scale means. There is only one exception for the perspicuity scale where the item not understandable versus understandable has a mean of 0.5 while the other items are rather on the opposite side, for example, the item confusing versus clear has a mean of -0.9. This might be due to misinterpretation for one of the questions.

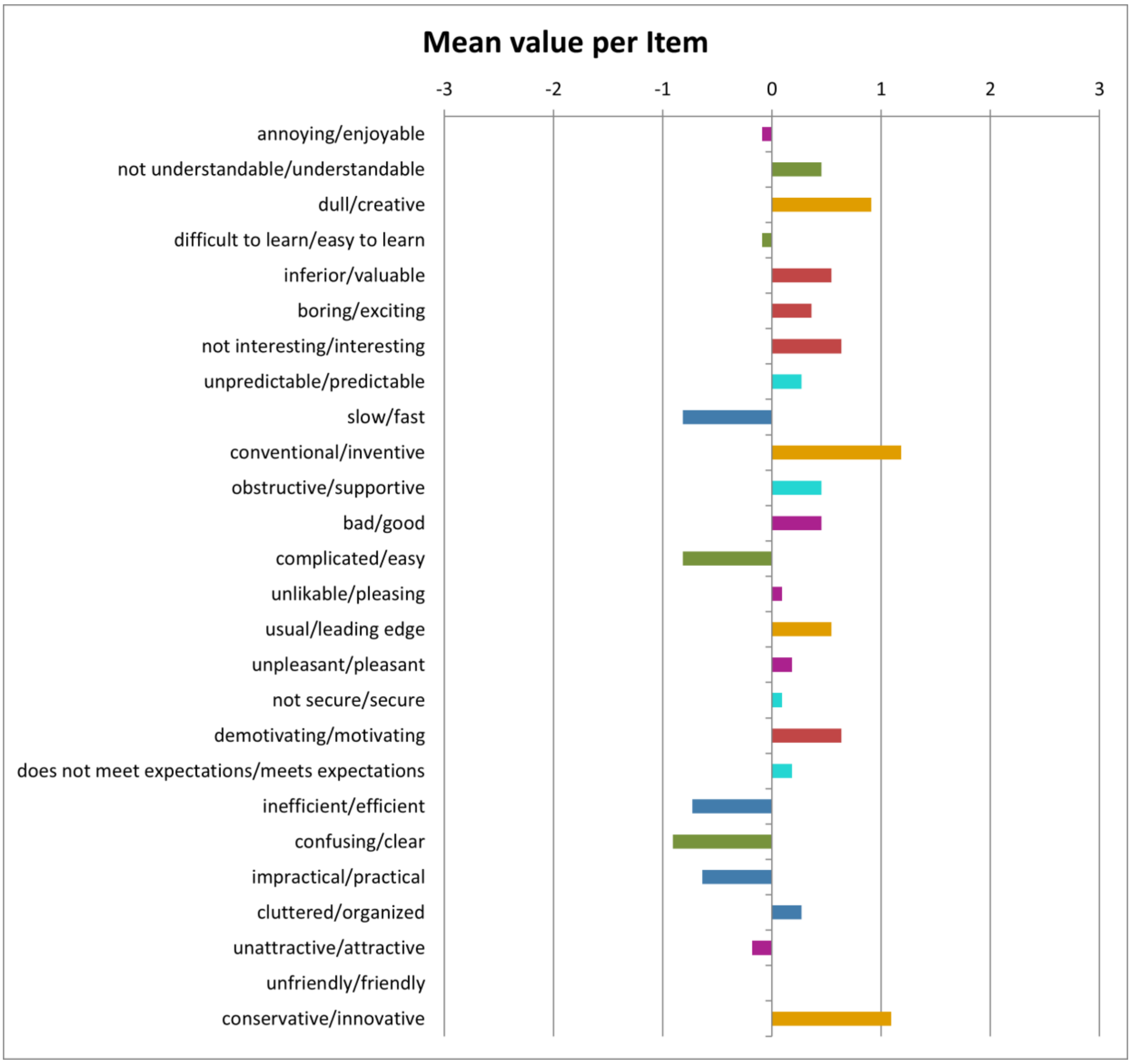



Figure 6.47: UEQ Results – Mean Value per Item – Bar Chart

*Note*. Bar chart visualising the means of the individual UEQ items (questions) for my results. The colours of the individual bars show to which scale the items belong to. This result was created using the UEQ Data Analysis Tools in Microsoft Excel.

Schrepp (2023) described that they also included a benchmark into the Data Analysis Tools. According to them this benchmark contains the data for 452 UEQ product evaluations. They explained that their benchmark classifies a product into five different categories per scale. These are: excellent: 10 percent range of the best results; good: 10 percent are better and 75 percent are worse; above average: 25 percent are better, 50 percent are worse; below average: 50 percent are better, 25 percent are worse; bad: in 25 percent of the worst results.

Worth noting that, in the version 12 of the Data Analysis Tools I used there is a description in one of the worksheets (Benchmark) where it is stated that the benchmarks dataset consists of data from 468 studies. I assume that the information in the handbook (452 product evaluations) has not been updated accordingly.

Figure 6.48 shows the result for the benchmark against other UEQ product evaluations. Hence, the results for all scales fall into the category bad except for stimulation which is below average and novelty which is above average. I somehow expected such results especially for scales like efficiency as the participants were only shown a slowed down video demonstration of a simulated System User using the method exploring their files and folders. Also, my sample size of only 11 participants is rather small. However, the rating for novelty is quite surprising. Hence, it at least indicates that my method was perceived as somehow innovative and interesting. Consequently, I argue that for further research one may develop a complete prototype for users to actually interact with. Especially for efficiency I would expect drastically different results as the 2D Navigation mode of my method is not that different from navigating one’s files and folders in more conventional applications like the Microsoft Windows Explorer. Also, I argue that this benchmark may be biased because I argue that most results in this benchmark are most likely coming from commercial projects, for example, from attempts to improve the usability of existing, more conventional applications. Hence, such a benchmark may be more useful if the authors of the Data Analysis Tools at least would differentiate between results from commercial versus research projects. Still, for transparency I included the results.

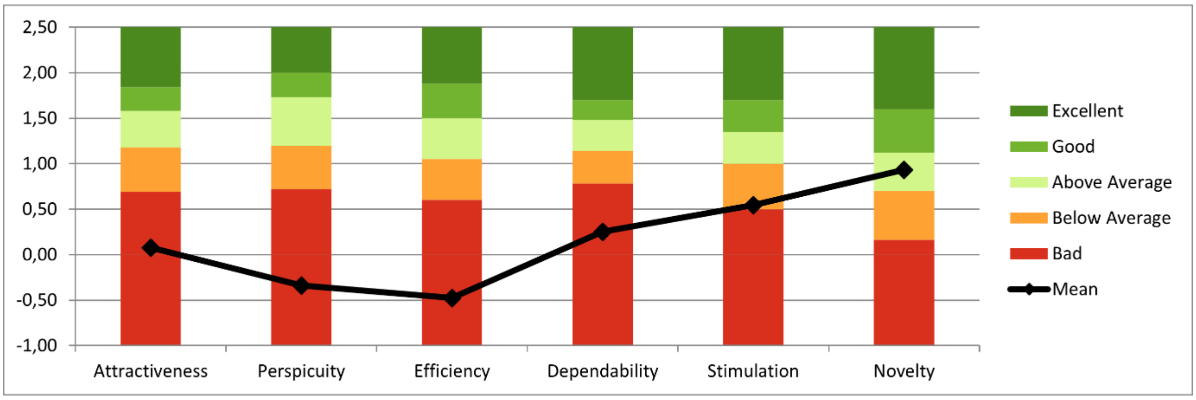


Figure 6.48: UEQ Results – Benchmark – Bar Chart

*Note*. Bar chart visualising the benchmark results comparing my results against other results available in the Data Analysis Tools. This result was created using the UEQ Data Analysis Tools in Microsoft Excel.

## 6.8 Chapter Summary

In summary, the evaluation of my novel method shows promising results which warrant further research. I argue that:

- Regarding qualitative results I received rather positive feedback for my design walk-through which I carried out for my conceptual designs (see Section 6.2). To exemplify, the participant highlighted the 3D Overview mode for *location independence* (see user story ID 18). Based on the design review I carried out I conclude that my design from the interaction perspective works as described subject to only small changes (see Section 6.3).
- The node limits for performance when using Unity were evaluated in Section 6.4 (see Table 6.2 and Table 6.3). My results indicate that rendering dynamic text and icons reduces the rendering performance drastically. However, as one core aspect of my

design was scalability this should not be a large issue in most situations. Meaning, each level of depth in the tree is only loaded and rendered depending on user interaction. Therefore, only in the rarest instance is the complete tree rendered at once.

- The edge case I identified in Section 6.5 can indeed become an issue if the tree to visualise is very deep. However, I argue that the practical use cases for such deep trees are rather rare.
- For quantitative results my UEQ results showed a neutral evaluation result for almost all scales except for novelty which can be considered as a positive evaluation result (see Section 6.7). I argue that the results are good enough to warrant further research, for example, developing a completely usable prototype for further evaluation.

# Chapter 7 Conclusion and Further Work

## 7.1 Achievements

In summary my research project addresses various aspects regarding the visualisation and interaction with hierarchical datasets.

In Chapter 2 I critically describe and evaluate various techniques to visualise and interact with hierarchical data. I covered 2D, 3D and techniques for mobile devices. Also, I assessed those techniques using a task analysis. I synthesised the results of my assessment and conclusions from the available literature into requirements for new techniques to visualise and interact with hierarchical data.

In Chapter 3 I outline the methodology I applied for my research project. This includes a critical discussion on why I have chosen certain methods over alternative approaches. Also, I describe my scope and target user group for my project. The software I have used to carry out my research project is listed in Appendix A. Worth noting that, I also describe various challenges I have faced during my research in this chapter.

Chapter 4 describes the steps undertaken to understand user needs. For one, I describe the models I have created, for example, work role model. Also, I detail the design of my interviews and online questionnaire. I synthesised the results into various patterns, models and user stories informing my design going forward.

In Chapter 5 I describe my design based on the requirements from Chapter 2 and my synthesised user stories and models from Chapter 4. I outline all the design activities I have carried out from the initial design to the final wireframes.

In the final Chapter 6 I describe various evaluation results, for example, for a design review with a research participant. Moreover, I outline my development and design work from a technical perspective including a final evaluation utilising a standardised questionnaire (UEQ) and a video-based prototype.

Overall, I achieved to complete all of my research objectives from Section 1.2:

- I carried out a critical investigation into the existing methods to visualise and interact with hierarchical data. I provided a task analysis comparing the methods to each other.
- I designed a novel method to visualise and interact with hierarchical data based on the identified requirements from my literature review. Accordingly, I also included my synthesised actual user needs from my own research.
- I developed a prototype for my novel method to identify technical challenges and limitations. I also developed a complete scenario-based, video-based prototype utilising animation techniques for a mobile phone and a PC desktop.
- I evaluated my novel method consistently from understanding user needs to each design activity and the technical prototyping.

## 7.2 The Contributions to the New Knowledge Generation

The contributions I have outlined in Section 1.3 have been achieved. This thesis presents two main practical contributions. The first one is the identification of actual user needs in the context of viewing and interacting with hierarchical data focussing on file and folder management. This user-centred design approach may lead to a method which is perceived as meaningful and effective by the user when compared to other existing techniques (Shen et al., 2019). The second one is the design and development of a scalable technique to visualise and interact with hierarchical data. Scalability in this context means for one that visual clutter may not occur when the tree grows in size. Moreover, the method is scalable from a technical perspective by offering the possibility to render only parts of the whole tree. The novel method may be utilised in commercial scenarios, for example to provide a new form of online shopping. Also, it may be useful in academia, for example when visualising data clusters.

In detail the main contributions are:

- Identification of actual user needs in the context of viewing and interacting with hierarchical data focussing on file and folder management.
- Design and develop a novel method to visualise and interact with hierarchical data utilising a user-centred design approach explicitly challenging the predominance of the common tree view.

I achieved the following:

- My novel method is the first technique to visualise and interact with hierarchical data to utilise a user-centred design approach derived from user experience (UX) design.
- Based on my research to understand user needs I introduced a rather novel design metaphor.
- My design is novel because it visualises individual nodes in detail while also being scalable without introducing visual clutter. Moreover, it utilises the available screen space.
- My method is novel as it was designed from the start for both mobile and PC devices.
- With my technique, I introduced a novel concept to visualise and interact with hierarchical data by enabling a conventional 2D navigation in an otherwise native 3D view. Hence, it combines both benefits from a 3D overview and 2D navigation in just one view where a switch between the modes is simply realised by moving the camera and scaling the node labels.

## 7.3 Discussions of Limitation and Future Work

Because of the scope of my research and resource constraints I can identify various areas for improvement.

With available resources a fully usable prototype may be developed. This requires at least the development of a server application with two frontend applications, one for mobile and one for PC desktop. However, it may be possible to reduce the development effort by creating just one frontend application which scales accordingly depending on the device. Consequently, this application has to be evaluated especially in regards to its usability. Hence, if the task performance and precision (low error rates) does not at least match conventional solutions like the Microsoft Windows Explorer my novel method may be expandable from the end-user’s point of view.

For further evaluation it may also be interesting to consider the UX component meaningfulness especially in regards to further, more advanced prototypes. To observe a research participant using the method over a period of time, for example, a week, where they are only allowed to use the method to carry out their file and folder management may provide valuable insights. Such an approach may also help to lessen the effect of a potential bias towards the Desktop metaphor or more generally speaking their existing habits.

Also, it may be desirable to change the target scope from file and folder management to other tasks related to internet, office, communication or organisation (see Section 4.8). Hence, there may be tasks where a 3D Overview mode is received as being more useful than in tasks related to file and folder management.

Another limitation is the rather small sample size of only 11 participants in the UEQ evaluation. I argue that the small sample size led to rather large confidence intervals (see Section 6.7). Also, the evaluation was based on a video-based prototype (video presentation) rather than an actual usable prototype. Hence, this may have impacted the results of the evaluation (see Section 3.9 and Section 6.7).

Also, offering financial incentives towards research participants can be viewed as another limitation of this research project. However, I argue that the incentives were of a rather small amount and were ethically justified as the participants did not expose themselves to harm of any kind (see Section 3.5).

Moreover, it may be interesting to evaluate the application of my novel method to virtual reality devices. Hence, this would mean that the novel method may be fully cross device compatible.

## References


Abdelazeem, B., Abbas, K. S., Amin, M. A., El-Shahat, N. A., Malik, B., Kalantary, A., & Eltobgy, M. (2022). The effectiveness of incentives for research participation: A systematic review and meta-analysis of randomized controlled trials. *PLOS ONE*, *17*(4). https://doi.org/10.1371/journal.pone.0267534

Abello, J., Van Ham, F., & Krishnan, N. (2006). ASK-GraphView: A Large Scale Graph Visualization System. *IEEE transactions on visualization and computer graphics*, *12*(5), 669-676. https://doi.org/10.1109/tvcg.2006.120

Alsakran, J., Chen, Y., Zhao, Y., Yang, J., & Luo, D. (2011). STREAMIT: Dynamic visualization and interactive exploration of text streams. *2011 IEEE Pacific Visualization Symposium*, 131-138. https://doi.org/10.1109/pacificvis.2011.5742382

Andrews, K. (1995). Case study. Visualising cyberspace: information visualisation in the Harmony Internet browser. *Proceedings of Visualization 1995 Conference*, 97-104. https://doi.org/10.1109/infvis.1995.528692

Andrews, K. (2002). Visual exploration of large hierarchies with information pyramids. *Proceedings Sixth International Conference on Information Visualisation*, 793-798. https://doi.org/10.1109/iv.2002.1028871

Andrews, K., & Kasanicka, J. (2007). A Comparative Study of Four Hierarchy Browsers using the Hierarchical Visualisation Testing Environment (HVTE). *2007 11th International Conference Information Visualization (IV '07)*, 81-86. https://doi.org/10.1109/iv.2007.8

Andrews, K., Kienreich, W., Sabol, V., Becker, J., Droschl, G., Kappe, F., Granitzer, M., Auer, P., & Tochtermann, K. (2002). The InfoSky visual explorer: Exploiting Hierarchical Structure and Document Similarities. *Information Visualization*, *1*(3-4), 166-181. https://doi.org/10.1057/palgrave.Ivs.9500023

Apple Inc. (2022). *iPad Tech Specs*. Apple. Retrieved October 10, 2022, from https://www.apple.com/uk/ipad-10.2/specs/

Baron, M. E. (1969). A Note on the Historical Development of Logic Diagrams: Leibniz, Euler and Venn. *The Mathematical Gazette*, *53*(384), 113-125. https://doi.org/10.2307/3614533

Bederson, B. B., Shneiderman, B., & Wattenberg, M. (2002). Ordered and quantum treemaps: Making effective use of 2D space to display hierarchies. *ACM Transactions on Graphics*, *21*(4), 833-854. https://doi.org/10.1145/571647.571649

Bell, T. E., & Thayer, T. A. (1976). Software requirements: Are they really a problem? *Proceedings of the 2nd international conference on Software engineering*, 61–68.

Belter, M., & Lukosch, H. (2022). Towards a Virtual Reality Math Game for Learning In Schools - A User Study. *2022 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, 808-809. https://doi.org/10.1109/vrw55335.2022.00255

Blender Foundation. (n.d.). *Blender 4.1*. Blender. Retrieved June 14, 2024, from https://www.blender.org/

Borromeo, N. A. (2021). *Hands-On Unity 2021 Game Development* (2nd ed.). Packt Publishing.

Bose, S. (2021, February 22). *What is the ideal screen size for responsive design?* BrowserStack. Retrieved July 11, 2022, from https://www.browserstack.com/guide/ideal-screen-sizes-for-responsive-design

Bozhanov, I. (2020, December 18). *Populating the tree using AJAX and lazy loading nodes*. GitHub. Retrieved February 17, 2022, from https://github.com/vakata/jstree/wiki#populating-the-tree-using-ajax-and-lazy-loading-nodes

Brooke, J. (1996). SUS: A 'Quick and Dirty' Usability Scale. In P. W. Jordan, B. Thomas, B. A. Weerdmeester, & I. L. McClelland (Eds.), *Usability Evaluation in Industry* (pp. 189-194). Taylor & Francis Ltd.

Bruls, M., Huizing, K., & van Wijk, J. J. (2000). Squarified Treemaps. In W. C. de Leeuw & R. van Liere (Eds.), *Data Visualization 2000* (pp. 33-42). Springer. https://doi.org/10.1007/978-3-7091-6783-0_4

Burch, M., Vramulet, A., Thieme, A., Vorobiova, A., Shehu, D., Miulescu, M., Farsadyar, M., & Van Krieken, T. (2020). VizWick: a multiperspective view of hierarchical data. *Proceedings of the 13th International Symposium on Visual Information Communication and Interaction*, 1-5, Article 23. https://doi.org/10.1145/3430036.3430044

Card, S. K., & Nation, D. (2002). Degree-of-interest trees: a component of an attention-reactive user interface. *Proceedings of the Working Conference on Advanced Visual Interfaces*, 231–245. https://doi.org/10.1145/1556262.1556300

Carriere, J., & Kazman, R. (1995). Research report. Interacting with huge hierarchies: beyond cone trees. *Proceedings of Visualization 1995 Conference*, 74-81. https://doi.org/10.1109/infvis.1995.528689

Chacón-Vázquez, O., Montané-Jiménez, L. G., Ochoa-Rivera, C. A., & Hernández-Ocaña, B. (2021). Design of Platform to Support Workflow Continuity in Multi-Device Applications. *Advances in Science, Technology and Engineering Systems Journal*, *6*(1), 1091-1099. https://doi.org/10.25046/aj0601121

Chen, C. (2005). Top 10 unsolved information visualization problems. *IEEE Computer Graphics and Applications*, *25*(4), 12-16. https://doi.org/10.1109/mcg.2005.91

Chhetri, A. P., Zhang, K., & Jain, E. (2015). A mobile interface for navigating hierarchical information space. *Journal of Visual Languages & Computing*, *31*, 48-69. https://doi.org/10.1016/j.jvlc.2015.10.002

Cockburn, A., Karlson, A., & Bederson, B. B. (2009). A review of overview+detail, zooming, and focus+context interfaces. *ACM Comput. Surv.*, *41*(1), Article 2. https://doi.org/10.1145/1456650.1456652

Cockburn, A., & McKenzie, B. (2000). An Evaluation of Cone Trees. In S. McDonald, Y. Waern, & G. Cockton (Eds.), *People and Computers XIV - Usability or Else!* (pp. 425-436). Springer. https://doi.org/10.1007/978-1-4471-0515-2_28

Cockburn, A., & Savage, J. (2004). Comparing Speed-Dependent Automatic Zooming with Traditional Scroll, Pan and Zoom Methods. In E. O'Neill, P. Palanque, & P. Johnson (Eds.), *People and Computers XVII - Designing for Society* (pp. 87-102). Springer. https://doi.org/10.1007/978-1-4471-3754-2_6

Craig, P. (2015). Interactive animated mobile information visualisation. *SA '15: SIGGRAPH Asia 2015 Mobile Graphics and Interactive Applications*, 1-6, Article 24. https://doi.org/10.1145/2818427.2818458

Craig, P., & Huang, X. (2015). The Mobile Tree Browser: A Space Filling Information Visualization for Browsing Labelled Hierarchies on Mobile Devices. *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*, 2240-2247. https://doi.org/10.1109/cit/iucc/dasc/picom.2015.331

Craig, P., & Kennedy, J. (2008). Concept relationship editor: a visual interface to support the assertion of synonymy relationships between taxonomic classifications. *Proceedings Volume 6809, Visualization and Data Analysis 2008*, *6809*, Article 680906. https://doi.org/10.1117/12.765540

Denzin, N. K. (2017). *The research act: a theoretical introduction to sociological methods*. Routledge. (Original work published 1970)

Dodge, M. (n.d.). *Fly Through the Web*. Mappa.Mundi Magazine. Retrieved March 15, 2022, from https://mappa.mundi.net/maps/maps_018/index.html#hotsauce

Dresing, T., Pehl, T., & Schmieder, C. (2015). *Manual (on) Transcription. Transcription Conventions, Software Guides and Practical Hints for Qualitative Researchers. 3rd English Edition* (3rd ed.) http://www.audiotranskription.de/english/transcription-practicalguide.htm

Etemad, K., Baur, D., Brosz, J., Carpendale, S., & Samavati, F. F. (2014). PaisleyTrees: A Size-Invariant Tree Visualization. *EAI Endorsed Transactions on Creative Technologies, 1*(1). https://doi.org/10.4108/ct.1.1.e2

Fairchild, K. M., Poltrock, S. E., & Furnas, G. W. (1988). SemNet: Three-Dimensional Graphic Representations of Large Knowledge Bases. In R. Guindon (Ed.), *Cognitive Science and Its Applications for Human-computer Interaction* (1st ed., pp. 201-234). Psychology Press.

Fang, G., Bhardwaj, N., Robilotto, R., & Gerstein, M. B. (2010). Getting Started in Gene Orthology and Functional Analysis. *PLoS Computational Biology*, *6*(3), Article e1000703. https://doi.org/10.1371/journal.pcbi.1000703

Ferrone, H. (2020). *Learning C# by Developing Games with Unity 2020* (5th ed.). Packt Publishing.

Furnas, G. W. (1986). Generalized fisheye views. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 16–23. https://doi.org/10.1145/22627.22342

Google. (2022, April 20). *Material Icons Guide*. Google Developers. Retrieved October 11, 2022, from https://developers.google.com/fonts/docs/material_icons

Google. (n.d.-a). *App bars: bottom*. Material Design. Retrieved October 11, 2022, from https://material.io/components/app-bars-bottom

Google. (n.d.-b). *Applying color to UI*. Material Design. Retrieved October 11, 2022, from https://material.io/design/color/applying-color-to-ui.html#usage

Google. (n.d.-c). *COLOR TOOL*. Material Design. Retrieved October 11, 2022, from https://material.io/resources/color/#!/?view.left=0&view.right=0&primary.color=3f50b5

Google. (n.d.-d). *Components*. Material Design. Retrieved October 11, 2022, from https://material.io/components

Google. (n.d.-e). *Lists*. Material Design. Retrieved October 12, 2022, from https://material.io/components/lists

Google. (n.d.-f). *Pixel density*. Material Design. Retrieved October 10, 2022, from https://material.io/design/layout/pixel-density.html#pixel-density

Google. (n.d.-g). *System icons*. Material Design. Retrieved October 18, 2022, from https://material.io/design/iconography/system-icons.html

Google. (n.d.-h). *Text legibility*. Material Design. Retrieved October 11, 2022, from https://material.io/design/color/text-legibility.html#legibility-standards

Google. (n.d.-i). *The type system*. Material Design. Retrieved October 10, 2022, from https://material.io/design/typography/the-type-system.html#type-scale

Google. (n.d.-j). *Understanding layout*. Material Design. Retrieved October 13, 2022, from https://material.io/design/layout/understanding-layout.html#principles

Google. (n.d.-k). *Update to Material Design 3*. Material Design. Retrieved October 8, 2022, from https://material.io/

Graham, M., & Kennedy, J. (2010). A Survey of Multiple Tree Visualisation. *Information Visualization*, *9*(4), 235-252. https://doi.org/10.1057/ivs.2009.29

Greenberg, S., Carpendale, S., Marquardt, N., & Buxton, B. (2012). *Sketching User Experiences*. Morgan Kaufmann.

Grix, J. (2010). *The foundations of research* (2nd ed.). Palgrave Macmillan.

GSMArena.com. (n.d.). *Samsung Galaxy S7*. GSMArena. Retrieved January 11, 2023, from https://www.gsmarena.com/samsung_galaxy_s7-7821.php

Hakala, T., Lehikoinen, J., & Aaltonen, A. (2005). Spatial interactive visualization on small screen. *Proceedings of the 7th international conference on Human computer interaction with mobile devices & services (MobileHCI '05)*, 137-144. https://doi.org/10.1145/1085777.1085800

Hao, J., Zhang, K., & Huang, M. L. (2007). RELT – Visualizing Trees on Mobile Devices. In G. Qiu, C. Leung, X. Xue, & R. Laurini (Eds.), *VISUAL 2007: Advances in Visual Information Systems* (pp. 344-357). Springer. https://doi.org/10.1007/978-3-540-76414-4_34

Harshfield, N., Chang, D.-J., & Rammohan. (2015). A Unity 3D framework for algorithm animation. *2015 Computer Games: AI, Animation, Mobile, Multimedia, Educational and Serious Games (CGAMES)*, 50-56. https://doi.org/10.1109/cgames.2015.7272955

Hartson, R., & Pyla, P. S. (2019). *The UX Book: Agile UX Design for a Quality User Experience* (2nd ed.). Elsevier Inc.

Hinderks, A., Schrepp, M., & Thomaschewski, J. (n.d.). *UEQ User Experience Questionnaire*. UEQ-Online. Retrieved December 29, 2022, from https://www.ueq-online.org/

Holten, D. (2006). Hierarchical Edge Bundles: Visualization of Adjacency Relations in Hierarchical Data. *IEEE transactions on visualization and computer graphics, 12*(5), 741-748. https://doi.org/10.1109/tvcg.2006.147

Hornbæk, K., Bederson, B. B., & Plaisant, C. (2002). Navigation patterns and usability of zoomable user interfaces with and without an overview. *ACM Transactions on Computer-Human Interaction, 9*(4), 362-389. https://doi.org/10.1145/586081.586086

Hughes, T., Hyun, Y., & Liberles, D. A. (2004). Visualising very large phylogenetic trees in three dimensional hyperbolic space. *BMC Bioinformatics, 5*, Article 48. https://doi.org/10.1186/1471-2105-5-48

Ilmberger, W., Schrepp, M., & Held, T. (2009). Was verursacht den Zusammenhang zwischen Ästhetik und Usability? In H. Wandke, S. Kain, & D. Struve (Eds.), *Mensch und Computer 2009: 9. fachübergreifende Konferenz für interaktive und koooperative Medien - Grenzenlos frei* (pp. 383-392). Oldenbourg Wissenschaftsverlag. https://doi.org/10.1524/9783486598551.383

Johnson, B., & Shneiderman, B. (1991). Tree-maps: a space-filling approach to the visualization of hierarchical information structures. *Proceeding Visualization '91*, 284-291. https://doi.org/10.1109/visual.1991.175815

Johnson, J. (2021). *Designing with the Mind in Mind: Simple Guide to Understanding User Interface Design Guidelines* (3rd ed.). Morgan Kaufmann.

Johnson, J., Roberts, T. L., Verplank, W., Smith, D. C., Irby, C. H., Beard, M., & Mackey, K. (1989). The Xerox Star: a retrospective. *Computer*, *22*(9), 11-26. https://doi.org/10.1109/2.35211

Johnson, S. (1999). *Interface Culture: How New Technology Transforms the Way We Create and Communicate*. Basic Books. (Original work published 1997)

K Desktop Environment e. V. (n.d.). *Kdenlive*. KDE-Anwendungen. Retrieved June 14, 2024, from https://apps.kde.org/de/kdenlive/

Karstens, B., Kreuseler, M., & Schumann, H. (2003). Visualization of complex structures on mobile handhelds. *Proceedings of International Workshop on Mobile Computing*.

Kobsa, A. (2004). User Experiments with Tree Visualization Systems. *IEEE Symposium on Information Visualization*, 9-16. https://doi.org/10.1109/infvis.2004.70

Kreuseler, M., Lopez, N., & Schumann, H. (2000). A scalable framework for information visualization. *IEEE Symposium on Information Visualization 2000. INFOVIS 2000. Proceedings*, 27-36. https://doi.org/10.1109/infvis.2000.885088

Kruskal, J. B., & Landwehr, J. M. (1983). Icicle Plots: Better Displays for Hierarchical Clustering. *The American Statistician*, *37*(2), 162-168. https://doi.org/10.2307/2685881

Lamping, J., Rao, R., & Pirolli, P. (1995). A Focus+context Technique Based on Hyperbolic Geometry for Visualizing Large Hierarchies. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 401-408. https://doi.org/10.1145/223904.223956

Laricchia, F. (2022, August 23). *Tablet shipments market share by vendor worldwide from 2nd quarter 2011 to 2nd quarter 2022*. Statista. Retrieved October 10, 2022, from https://www.statista.com/statistics/276635/market-share-held-by-tablet-vendors/

Laubheimer, P. (2016, December 4). *Wireflows: A UX Deliverable for Workflows and Apps*. Nielsen Norman Group. Retrieved October 21, 2022, from https://www.nngroup.com/articles/wireflows/

Laugwitz, B., Held, T., & Schrepp, M. (2008). Construction and Evaluation of a User Experience Questionnaire. In A. Holzinger (Ed.), *HCI and Usability for Education and Work* (pp. 63-76). Springer. https://doi.org/10.1007/978-3-540-89350-9_6

LaViola Jr., J. J., Kruijff, E., Bowman, D. A., Poupyrev, I., & McMahan, R. P. (2017). *3D User Interfaces: Theory and Practice* (2nd ed.). Addison-Wesley.

Lazar, J., Feng, J. H., & Hochheiser, H. (2010). *Research Methods in Human-Computer Interaction*. John Wiley & Sons Ltd.

Mackinlay, J. D., Robertson, G. G., & Card, S. K. (1991). The Perspective Wall: Detail and Context Smoothly Integrated. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 173-179. https://doi.org/10.1145/108844.108870

Macquisten, A., Smith, A. M., & Johansson Fernstad, S. (2020). Evaluation of Hierarchical Visualization for Large and Small Hierarchies. *2020 24th International Conference Information Visualisation (IV)*, 166-173. https://doi.org/10.1109/iv51561.2020.00036

Marriott, K., Chen, J., Hlawatsch, M., Itoh, T., Nacenta, M. A., Reina, G., & Sturzlinger, W. (2018). Immersive Analytics: Time to Reconsider the Value of 3D for Information Visualisation. In K. Marriott, F. Schreiber, T. Dwyer, K. Klein, N. H. Riche, T. Itoh, W. Stuerzlinger, & B. H. Thomas (Eds.), *Immersive Analytics* (pp. 25-55). Springer International Publishing. https://doi.org/10.1007/978-3-030-01388-2_2

McGill, S. M., Cholewicki, J., & Peach, J. P. (1997). Methodological considerations for using inductive sensors (3space isotrak) to monitor 3-D orthopaedic joint motion. *Clinical Biomechanics*, *12*(3), 190-194. https://doi.org/10.1016/S0268-0033(97)00063-6

McNeill, P. (1997). Paying People to Participate in Research: Why not? *Bioethics*, *11*(5), 390-396. https://doi.org/10.1111/1467-8519.00079

Mellström, C., & Johannesson, M. (2008). Crowding out in Blood Donation: Was Titmuss Right? *Journal of the European Economic Association*, *6*(4), 845-863. http://www.jstor.org/stable/40282686

Mochizuki, T., & Furukawa, Y. (2022, January 20). PlayStation Creator Kutaragi Snubs Metaverse and VR Headsets. *Bloomberg*. https://www.bloomberg.com/news/articles/2022-01-20/playstation-creator-kutaragi-snubs-metaverse-and-vr-headsets

Muller, M. J., & Kuhn, S. (1993). Participatory design. *Communications of the ACM*, *36*(6), 24-28. https://doi.org/10.1145/153571.255960

Munzner, T. (1997). H3: laying out large directed graphs in 3D hyperbolic space. *Proceedings of VIZ '97: Visualization Conference, Information Visualization Symposium and Parallel Rendering Symposium*, 2-10. https://doi.org/10.1109/infvis.1997.636718

Muse Group & contributors. (n.d.). *Audacity is the world's most popular audio editing and recording app*. Audacity. Retrieved April 7, 2024, from https://www.audacityteam.org/

Nebeling, M., Mintsi, T., Husmann, M., & Norrie, M. (2014). Interactive development of cross-device user interfaces. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2793–2802. https://doi.org/10.1145/2556288.2556980

Ng, K.-W., & Low, Z.-W. (2014). Simplification of 3D Triangular Mesh for Level of Detail Computation. *2014 11th International Conference on Computer Graphics, Imaging and Visualization*, 11-16. https://doi.org/10.1109/cgiv.2014.9

Norman, D. A. (2005). Human-centered design considered harmful. *Interactions*, *12*(4), 14-19. https://doi.org/10.1145/1070960.1070976

Park, S.-M., & Kim, Y.-G. (2022). A Metaverse: Taxonomy, Components, Applications, and Open Challenges. *IEEE Access*, *10*, 4209-4251. https://doi.org/10.1109/access.2021.3140175

Patton, J., & Economy, P. (2014). *User Story Mapping: Discover the Whole Story, Build the Right Product* (1st ed.). O'Reilly Media.

Paulovich, F. V., & Minghim, R. (2008). HiPP: A Novel Hierarchical Point Placement Strategy and its Application to the Exploration of Document Collections. *IEEE transactions on visualization and computer graphics*, *14*(6), 1229-1236. https://doi.org/10.1109/tvcg.2008.138

Pavlopoulos, G. A., Soldatos, T. G., Barbosa-Silva, A., & Schneider, R. (2010). A reference guide for tree analysis and visualization. *BioData Mining*, *3*, Article 1. https://doi.org/10.1186/1756-0381-3-1

Pinandito, A., Az-Zahra, H. M., Fanani, L., & Putri, A. V. (2017). Analysis of web content delivery effectiveness and efficiency in responsive web design using material design guidelines and User Centered Design. *2017 International Conference on Sustainable Information Engineering and Technology (SIET)*, 435-441. https://doi.org/10.1109/siet.2017.8304178

Pixabay GmbH. (2021, December 13). *Pixabay Terms of Service*. Pixabay. Retrieved December 28, 2022, from https://pixabay.com/service/terms/

QSR International. (n.d.). *Unlock insights with the leading qualitative data analysis software*. NVivo. Retrieved December 27, 2022, from https://www.qsrinternational.com/nvivo-qualitative-data-analysis-software/home

Reiter, S. (2011). *Entwurf und Programmierung einer Anwendung zur Datensichtung und Selektion im dreidimensionalen Raum [Unpublished Diplom (FH) thesis]. Munich University of Applied Sciences.*

Rekimoto, J., & Green, M. (1993). The Information Cube: Using Transparency in 3D Information Visualization. *Proceedings of the Third Annual Workshop on Information Technologies & Systems (WITS'93)*, 125-132.

Robertson, C. (n.d.). *Roboto*. Google Fonts. Retrieved October 10, 2022, from https://fonts.google.com/specimen/Roboto/about

Robertson, G. G., & Mackinlay, J. D. (1993). The Document Lens. *Proceedings of the 6th Annual ACM Symposium on User Interface Software and Technology*, 101–108. https://doi.org/10.1145/168642.168652

Robertson, G. G., Mackinlay, J. D., & Card, S. K. (1991). Cone Trees: Animated 3D Visualizations of Hierarchical Information. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 189-194. https://doi.org/10.1145/108844.108883

Saldana, J. (2021). *The Coding Manual for Qualitative Researchers* (4th ed.). SAGE Publications.

Sancho-Chavarria, L., & Mata-Montero, E. (2018). A 3D Approach for the Visualization and Edition of Hierarchies: The Case of Biological Taxonomies. *2018 IEEE 38th Central America and Panama Convention (CONCAPAN XXXVIII)*. https://doi.org/10.1109/concapan.2018.8596366

Schrepp, M. (2023). *User Experience Questionnaire Handbook: All you need to know to apply the UEQ successfully in your projects Author: Dr. Martin Schrepp Version 9 (02.03.2023)* https://www.ueq-online.org/Material/Handbook.pdf

Schulz, H.-J., Hadlak, S., & Schumann, H. (2011). The Design Space of Implicit Hierarchy Visualization: A Survey. *IEEE transactions on visualization and computer graphics*, *17*(4), 393-411. https://doi.org/10.1109/tvcg.2010.79

Serif (Europe) Ltd. (n.d.). *Wir sind Affinity.* Affinity. Retrieved April 7, 2024, from https://affinity.serif.com/de/

Shen, H., Bednarz, T., Nguyen, H., Feng, F., Wyeld, T., Hoek, P. J., & Lo, E. H. S. (2019). Information visualisation methods and techniques: State-of-the-art and future directions. *Journal of Industrial Information Integration*, *16*, Article 100102. https://doi.org/10.1016/j.jii.2019.07.003

Shneiderman, B. (1996). The eyes have it: a task by data type taxonomy for information visualizations. *Proceedings 1996 IEEE Symposium on Visual Languages*, 336-343. https://doi.org/10.1109/vl.1996.545307

Shneiderman, B., & Wattenberg, M. (2001). Ordered treemap layouts. *IEEE Symposium on Information Visualization, 2001. INFOVIS 2001.*, 73-78. https://doi.org/10.1109/infvis.2001.963283

Spence, R., & Apperley, M. (1982). Data base navigation: an office environment for the professional. *Behaviour & Information Technology*, *1*(1), 43-54. https://doi.org/10.1080/01449298208914435

Stasko, J., & Zhang, E. (2000). Focus+context display and navigation techniques for enhancing radial, space-filling hierarchy visualizations. *IEEE Symposium on Information Visualization 2000. INFOVIS 2000. Proceedings*, 57-65. https://doi.org/10.1109/infvis.2000.885091

Sukamto, R. A., Wibisono, Y., & Agitya, D. G. (2020). Enhancing The User Experience of Portal Website using User-Centered Design Method. *2020 6th International Conference on Science in Information Technology (ICSITech)*, 171-175. https://doi.org/10.1109/icsitech49800.2020.9392044

SurveyMonkey. (n.d.). *Erstellen Sie blitzschnell kostenlose Umfragen und Formulare*. Retrieved April 7, 2024, from https://de.surveymonkey.com/

Sutcliffe, A., & Gault, B. (2004). Heuristic evaluation of virtual reality applications. *Interacting with Computers, 16*(4), 831-849. https://doi.org/10.1016/j.intcom.2004.05.001

Taivalsaari, A. (1999). *The Event Horizon User Interface Model for Small Devices* (Report No. TR-99-74). Sun Microsystems Laboratories.

Teo, T., Norman, M., Adcock, M., & Thomas, B. H. (2017). Data fragment: Virtual reality for viewing and querying large image sets. *2017 IEEE Virtual Reality (VR)*, 327-328. https://doi.org/10.1109/vr.2017.7892309

The Khronos Group Inc. (n.d.). *WebGL: LOW-LEVEL 3D GRAPHICS API BASED ON OPENGL ES*. Retrieved December 26, 2022, from https://www.khronos.org/webgl/

The Omni Group. (n.d.). *OmniGraffle*. Retrieved June 15, 2024, from https://www.omnigroup.com/omnigraffle

The University of Edinburgh School of Geosciences. (2013). *Interview Consent Form*. The University of Edinburgh. Retrieved April 8, 2022, from https://www.ed.ac.uk/files/imports/fileManager/Interview_Consent_Form.pdf

Tidwell, J., Brewer, C., & Valencia, A. (2020). *Designing Interfaces* (3rd ed.). O'Reilly Media.

Tominski, C., Abello, J., Van Ham, F., & Schumann, H. (2006). Fisheye Tree Views and Lenses for Graph Visualization. *Tenth International Conference on Information Visualisation (IV'06)*, 17-24. https://doi.org/10.1109/iv.2006.54

Tu, Y., & Shen, H.-W. (2007). Visualizing Changes of Hierarchical Data using Treemaps. *IEEE transactions on visualization and computer graphics*, *13*(6), 1286-1293. https://doi.org/10.1109/tvcg.2007.70529

Unity Technologies. (2022a, October 21). *GPU instancing*. Unity Manual. Retrieved October 28, 2022, from https://docs.unity3d.com/Manual/GPUInstancing.html

Unity Technologies. (2022b, October 21). *GPU Usage Profiler module*. Unity Manual. Retrieved October 29, 2022, from https://docs.unity3d.com/Manual/ProfilerGPU.html

Unity Technologies. (2022c, October 21). *TextMeshPro*. Unity Manual. Retrieved October 28, 2022, from https://docs.unity3d.com/Manual/com.unity.textmeshpro.html

Unity Technologies. (n.d.). *Unity 2021 LTS*. Unity. Retrieved October 12, 2022, from https://unity.com/releases/2021-lts

Van Ham, F., & Van Wijk, J. J. (2002). Beamtrees: compact visualization of large hierarchies. *IEEE Symposium on Information Visualization, 2002. INFOVIS 2002.*, 93-100. https://doi.org/10.1109/infvis.2002.1173153

Van Verth, J. M., & Bishop, L. M. (2016). *Essential Mathematics for Games and Interactive Applications* (3rd ed.). CRC Press Taylor & Francis Group.

VideoLAN Organization. (n.d.). *VideoLAN, ein Projekt und eine gemeinnützige Organisation.* Retrieved April 7, 2024, from https://www.videolan.org/vlc/

Walker, J., & Radich, Q. (2021, June 24). *Responsive design techniques*. Microsoft. Retrieved October 5, 2022, from https://learn.microsoft.com/en-us/windows/apps/design/layout/responsive-design

Wang, W., Wang, H., Dai, G., & Wang, H. (2006). Visualization of large hierarchical data by circle packing. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 517–520. https://doi.org/10.1145/1124772.1124851

Warchamp7. (n.d.). *OBS Studio*. OBS. Retrieved April 7, 2024, from https://obsproject.com/

Ware, C., & Mitchell, P. (2005). Reevaluating stereo and motion cues for visualizing graphs in three dimensions. *Proceedings of the 2nd symposium on Applied perception in graphics and visualization*, 51–58. https://doi.org/10.1145/1080402.1080411

Wijaya, A. C., Munandar, M. W. A., & Utaminingrum, F. (2019). Usability Testing of Augmented Reality For Food Advertisement Based On Mobile Phone Using System Usability Scale. *2019 International Conference on Sustainable Information Engineering and Technology (SIET)*, 266-269. https://doi.org/10.1109/siet48054.2019.8986118

Wilkinson, M., & Moore, A. (1997). Inducement in Research. *Bioethics*, *11*(5), 373-389. https://doi.org/10.1111/1467-8519.00078

Wiss, U., Carr, D., & Jonsson, H. (1998). Evaluating three-dimensional information visualization designs: a case study of three designs. *Proceedings. 1998 IEEE Conference on Information Visualization. An International Conference on Computer Visualization and Graphics (Cat. No.98TB100246)*, 137-144. https://doi.org/10.1109/iv.1998.694211

Woodburn, L., Yang, Y., & Marriott, K. (2019). Interactive Visualisation of Hierarchical Quantitative Data: An Evaluation. *2019 IEEE Visualization Conference (VIS)*, 96-100. https://doi.org/10.1109/visual.2019.8933545

Yang, J., Ward, M. O., Rundensteiner, E. A., & Patro, A. (2003). InterRing: A Visual Interface for Navigating and Manipulating Hierarchies. *Information Visualization, 2*(1), 16-30. https://doi.org/10.1057/palgrave.ivs.9500035

Yin, J., & Gao, Y. (2016). The design and research of 3D desktop interface based on the Pen + Touch. *2016 IEEE International Conference on Real-time Computing and Robotics (RCAR)*, 129-133. https://doi.org/10.1109/RCAR.2016.7784013

Zhou, D., Huang, J., & Schölkopf, B. (2005). Learning from Labeled and Unlabeled Data on a Directed Graph. *Proceedings of the 22nd International Conference on Machine Learning*, 1036–1043. https://doi.org/10.1145/1102351.1102482

Zimmerman, T. G., Lanier, J., Blanchard, C., Bryson, S., & Harvill, Y. (1986). A hand gesture interface device. *SIGCHI Bull.*, *17*(SI), 189–192. https://doi.org/10.1145/30851.275628

Zutlevics, T. (2016). Could providing financial incentives to research participants be ultimately self-defeating? *Research Ethics*, *12*(3), 137-148. https://doi.org/10.1177/1747016115626756

# Appendix A

## Software and Licenses

I utilised NVivo (see QSR International, n.d.) for coding and analysis. In detail I used it for:

- Coding my transcripts.
- Documenting my *analytic memos*. According to Saldana (2021) an analytic memo documents ones coding processes and choices.
- Creating coding matrices to analyse intersections between codes.
- Carrying out cluster analysis for word similarity between codes.
- Creating hierarchy charts visualised as treemaps to compare the amount of coding references.
- Carrying out a word frequency analysis visualised as a word cloud to identify frequently used terms. Worth noting that, for word frequency queries I utilised the NVivo default list of stop words in German. I manually added the following custom stop words to this list: “unv inc lacht laughs hm bejahend affirmative also man sind den dem hat bin #00 P I”.

My choice for NVivo was solely informed by the recommendation of my supervisor as other students of my University were already using it to great success.

For technical prototyping I utilised the Unity engine, version 2021 (2021.3.6f1) LTS (see Unity Technologies, n.d.) for Microsoft Windows. Otherwise, I also used the following standard applications on Windows:

- Microsoft Excel for data analysis, for example, recording benchmark results.
- Microsoft Visual Studio 2019 for C# development in Unity and debugging.
- Affinity Designer, version one (see Serif (Europe) Ltd, n.d.) to design my wireframes, sketches, textures, conceptual designs and models.

- Microsoft Word for writing.
- Microsoft Teams for remote interviews.
- Clarivate EndNote version 20 to manage my references and citations.
- Kdenlive (see K Desktop Environment e. V., n.d.) to edit and render the video for my prototype.
- OBS Studio (see Warchamp7, n.d.) to screen record remote interviews and to screen record render results for prototyping (software also used on Mac OS X).
- VLC media player (see VideoLAN Organization, n.d.) to take screenshots.
- Audacity (see Muse Group & contributors, n.d.) for voice recordings.
- Microsoft OneDrive to share material, for example, a video.
- Microsoft PowerPoint to present content to research participants.
- COLOR TOOL (see Google, n.d.-c) to select a colour scheme for my prototypes.

I used the following desktop computer components to create my thesis, including benchmarking:

- CPU: AMD Ryzen 9 5950X
- RAM: 64 GB
- GPU: NVIDIA GeForce RTX 3090 24GB VRAM
- OS: Windows 10 Pro x64

To carry out my technical prototyping and design work, for example, wireframes I also utilised the following:

- Images of a ladybug, Spain, a snout beetle, a jewel beetle, a stag beetle, Portugal and an autumn scene were all taken from Pixabay. The license from Pixabay grants “an irrevocable, worldwide, non-exclusive and royalty free right to use, download, copy, modify or adapt the Content for commercial or non-commercial purposes. Attribution of the photographer, videographer, musician or Pixabay is not required" (Pixabay GmbH, 2021, License for Content - Pixabay License section).

- The family of fonts called *Roboto* by Robertson (n.d.). These fonts are licensed using the Apache License, version 2.0 (Robertson, n.d.).
- For icons the *Material Icons* by Google (2022). These icons are licensed under the Apache License, version 2.0 (Google, 2022).
- To analyse and report on the results of the UEQ I utilised the Microsoft Excel based Data Analysis Tools. These Excel sheets are provided by Hinderks et al. (n.d.). I used the latest version 12 of the tools with a file time stamp of the 19th of February 2023.

I used SurveyMonkey (see SurveyMonkey, n.d.) to carry out all of my questionnaires. I also used Apple Keynote on Mac OS X to present content to research participants. For initial prototyping and early designs I also used Blender (see Blender Foundation, n.d.) and OmniGraffle (see The Omni Group, n.d.).

Worth noting that, I attempted to embed all of my figures, for example, wireframes, as vector-based images (.svg). Unfortunately, this was not always possible due to technical challenges, for example, when using Affinity Designer some exports had missing elements. Therefore, I had to utilise image (.png) exports in many cases for my figures.

# Appendix B

## Descriptive Codes, Questionnaire Design and User Stories

**Table B1**

*Descriptive Codes Phase 1*

| Descriptive Code | Code description | Code path | Quotes (evidentiary warrant) |
|---|---|---|---|
| *customised structure* | Use custom file hierarchies, for example, create (sub-)folders with a naming convention | file and folder management context/tasks | Professional01: "*My Mac at home is more organised, classic, file structure, I go into it and go classic, classic structure (...) naming and so. It is really much more organised*" |
| *no structure* | Do not use a custom file hierarchy, for example, store everything in one folder | file and folder management context/tasks | Professional01: "*my MacBook I have quite a lot on the desktop. And organise myself completely on the desktop to have fast access also in the cloud.*" |
| *search* | Use search feature to find files or folders | file and folder management context/tasks | Interviewer: "*Yes, you said you rarely use the search.*" Professional00: "*Sometimes, so rather rarely*" |

| Descriptive Code | Code description | Code path | Quotes (evidentiary warrant) |
|---|---|---|---|
| *bias* | User habits, customs to support innovation | file and folder management context/user stories/emotional impact | Professional01: "*So the keyboard, the classic keyboard comes from an insane past, yes. Has actually nothing left to do with reality but we still have it.*" |
| *immersion* | Feeling of immersion in a scenario, for example, being in the situation | file and folder management context/user stories/emotional impact | Professional00: "*one feels totally put into the situation.*" |
| *improve efficiency* | Create a faster, more efficient solution | file and folder management context/user stories/usability | Professional00: "*If of course, if something would be offered which (...) is more convenient, is more comfortable, is faster.*" |
| *improve search* | Improve existing search feature for better results | file and folder management context/user stories/usability | Professional01: "*As we have learned it from Amazon, that they can know beforehand what we want. Based on what we search for.*" |

| Descriptive Code | Code description | Code path | Quotes (evidentiary warrant) |
|---|---|---|---|
| *screen space utilisation* | Improve screen space utilisation, for example, to show more information on mobile devices | file and folder management context/user stories/usability | Professional01: "*Telephone is too small*" |
| *speech recognition* | Allow speech recognition as an input method | file and folder management context/user stories/usability | Professional00: "*The Alexa. But of course, that is [laughs] absolutely a dream of the future (...) that one says Alexa or some tool tell me please short (...) the registration numbers of BMW, from the first half year 2019.*" |
| *additional relations* | Additional relations for nodes | file and folder management context/user stories/usefulness | Professional01: "*my main topic is always. What is the latest version of V1 comma nanana because many of those folders are processed by other people, yes.*" |

| Descriptive Code | Code description | Code path | Quotes (evidentiary warrant) |
|---|---|---|---|
| *device independence* | One coherent user interface across devices | file and folder management context/user stories/usefulness | Professional01: "*What would be nice, if all those desktops across my three devices would be the same.*" |
| *location independence* | Support a central management for files, folders (store, central control) | file and folder management context/user stories/usefulness | Professional01: "*one always used to know exactly where to go, yes. That was just one place [laughs], yes. Maybe more hard drives.*" |
| *relate and analyse* | Support activities (multiple connected tasks), for example, create a PowerPoint presentation based on input from an e-mail | file and folder management context/user stories/usefulness | Professional01: "*This is a mixture, most of the time it is so conceptional work over PowerPoint, yes. More or less, so, create graphics which shall represent an idea. This is what takes a lot of time, just because I need a bit more to put it on paper. Also, to put on PowerPoint.*" |

| Descriptive Code | Code description | Code path | Quotes (evidentiary warrant) |
| --- | --- | --- | --- |
| *applications 2D, 3D* | Use of 2D vs. 3D applications (Provisional Code) | general context | Interviewer: "*We also talked about the programs you use in your daily routine. That these are still more 2D, list based.*" Professional00: "*Yes, still.*" |
| *pictures* | Work with images, pictures or graphics in general | general context/data types | Professional00: "*With pictures and at the end one carries out an appraisal session with the hotel manager.*" |
| *relational data* | Work with relational data, for example, relational databases | general context/data types | Professional00: "*Yes, I have done a lot with Access.*" |
| *text* | Work with text (I also consider PowerPoint presentations as text) | general context/data types | Professional01: "*But essentially, a lot of Excel spreadsheets.*" |
| *mobile devices* | Use of mobile devices (smart phone) (Provisional Code) | general context | Professional00: "*I will take the mobile phone first*" |

| Descriptive Code | Code description | Code path | Quotes (evidentiary warrant) |
|---|---|---|---|
| *emotional impact* | Emotional impact as a general decision criterion (Provisional Code) | general context/selection criteria | Interviewer: "*Without wasting time, good. And as said things like style and yes (…) design are more second-tier.*" Professional00: "*Second-tier, it must not disturb, distract me.*" |
| *price* | Price as a general decision criterion | general context/selection criteria | Professional00: "*But matter of expense was backmost. Hardly played a role when it was within a certain coverage. It was a very, very last point.*" |
| *usability* | Usability as a general decision criterion (Provisional Code) | general context/selection criteria | Professional00: "*If it loads faster or when one can use stronger compression or there are all kinds of aspects which played a role there.*" |
| *usefulness* | Usefulness as a general decision criterion (Provisional Code) | general context/selection criteria | Professional01: "*if the usefulness is not there, one cannot reach the next step over the usability or the design*" |

| Descriptive Code | Code description | Code path | Quotes (evidentiary warrant) |
|---|---|---|---|
| *creative content creation* | Create/edit/review audio, video content or scripting, programming | general context/tasks | Professional01: "*Oh, yes. Less now but more conceptional with photos, yes.*" |
| *internet* | Use of the internet, for example, social media, information, media consumption or shopping | general context/tasks | Professional00: "*Yes, sure. I also look into the internet, sure.*" |
| *office content creation* | Create/edit/review office documents, for example, Excel spreadsheets | general context/tasks | Professional01: "*Excel spreadsheets or PowerPoint I always have to create for presentations.*" |
| *private communication* | Private/personal communication, for example, short messages, e-mails | general context/tasks | Professional01: "*First, I check what happened at night on WhatsApp or on my e-mails.*" |
| *private organisation* | Private/personal planning of activities, for example, appointments | general context/tasks | Professional01: "*Look at my calendar and organise my day*" |

| Descriptive Code | Code description | Code path | Quotes (evidentiary warrant) |
|---|---|---|---|
| *project management* | Manage projects, for example, mind mapping, time management | general context/tasks | Professional01: "*I use it as a tool to control the digital world of my real world.*" |
| *work communication* | Work related communication, for example, short messages, e-mails | general context/tasks | Professional00: "*Exactly, if it is urgent, it is either, the people are called back.*" |
| *work organisation* | Work related planning of activities, for example, appointments | general context/tasks | Professional01: "*Apps for mobility and organisation through the day.*" |
| *virtual reality devices* | Usage of virtual reality devices (Provisional Code) | general context | Professional00: "*Yes, very interested. I consider it to be an extremely exiting topic and it is also a topic with a very, very strong future. Based on my opinion. Especially (...) in various areas of consumption.*" |

*Note*. List of result Descriptive Codes after first cycle coding.

**Table B2**

*Questionnaire Design*

| ID | Question | Rationale based on phase 1 |
|---|---|---|
| 1 | Your age in years? (Optional question) | Out of scope: I decided against considering personas, detailed subroles or user classes |
| 2 | Your gender? (Optional question) | Out of scope: I decided against considering personas, detailed subroles or user classes |
| 3 | Your e-mail address. Please note that you need to fill out this field in order to receive your voucher. We will only use this information to verify your participation in the survey. (Optional question) | Out of scope: Data not utilised, just forwarded to the staff of the University of Gloucestershire |
| 4 | Do you have a side job (in addition to being a student)? If so in which area? | Out of scope: I decided against considering personas, detailed subroles or user classes |
| 5 | Are you interested in new advanced graphical user interfaces for computer interaction (HCI)? | Is there a broad potential for new visualisation techniques (possibilities of the future) linked to my pattern **faith in virtual reality**? |

| ID | Question | Rationale based on phase 1 |
| --- | --- | --- |
| 6 | For which of the following regular (more than once per week) tasks (work and/or home usage related) do you use IT devices (for example, a smart phone, a laptop etc.)? | Do regular tasks really revolve around **office, communication and organisation**? |
| 7 | How are most of the interfaces of your regularly used (work and/or home usage) applications designed? | Are the programs indeed mostly 2D based, see **work practice ecology is text-based and 2D**? |
| 8 | Which type of data do you use the most (in regards to time spent) – digital and/or non-digital at work and/or home? | Is *text* really that dominant, see **work practice ecology is text-based and 2D**? |
| 9 | Which IT device do you mostly use in general (work and/or home)? Please insert numbers to reflect percentage (for example, 60 to 40). | Are mobile devices really that important in the participants daily routines, see **mobile device a cluttered companion**? |

| ID | Question | Rationale based on phase 1 |
| --- | --- | --- |
| 10 | Do you mostly use cloud based (for example, Google Docs) or locally installed (for example, Microsoft Office) applications to complete your regular tasks (work and/or home usage related)? | Does the result support the need for *device independence*? |
| 11 | Which data locations do you use in a professional and/or personal context for your regular (used more than once per week) digital documents (for example, .pdf, .doc, .jpg etc.)? | Does the result support the need for *location independence*? |
| 12 | Please name your personal favourite software application (mobile or desktop). You can name any application (work and/or home usage related). | Out of scope: Just to inform question ID 13 |
| 13 | Were the following criteria relevant for your decision regarding your favourite application? | Are they choosing **usable software with perceived value**? |

| ID | Question | Rationale based on phase 1 |
|---|---|---|
| 14 | When you are looking for your relevant files (for example, photos, spreadsheets, mails, messages) on your PC how do you find them? | Is a *customised structure* preferred? |
| 15 | Do you use only one application to find/navigate all your relevant files on your PC (for example, Windows-Explorer)? | Does the result support the need for *device independence*? Meaning, do they view this topic from a central point of view? |
| 16 | When you are looking for your relevant files (for example, photos, spreadsheets, mails, messages) on your mobile device (for example, smart phone) how do you find them? | Mostly *search* and *no structure*? |
| 17 | Do you use only one application to find/navigate all your relevant files on your mobile device (for example, iPhone documents)? | Does the result support the need for *device independence*? Meaning, do they view this topic from a central point of view? |

| ID | Question | Rationale based on phase 1 |
|---|---|---|
| 18 | What do you value the most in an application to navigate through your data (for example, photos, spreadsheets) on your IT device (mobile or PC)? | In the context of file and folder management is emotional impact really rather irrelevant, see **usable software with perceived value**? |
| 19 | What are you currently missing when navigating / searching for your files (for example, photos, spreadsheets, mails, messages) on your IT device (mobile and/or PC)? | Does the result support the **be here now** metaphor? |
| 20 | Do you regularly (at least once a week) use virtual reality and/or augmented reality software/products? | Does the result support the **faith in virtual reality** pattern? |

| ID | Question | Rationale based on phase 1 |
|---|---|---|
| 21 | Can you imagine a near future (max. 10 years from now) were some kind of virtual and/or augmented reality device (for example, AR glasses) will replace all of our current devices (smart phones, laptops, desktops etc.) completely (in all areas)? | Does the result support the **faith in virtual reality** pattern? |
| 22 | Do you (or did you once try to) order your data on your pc in a similar way as you do in real life (for example, like items on your desk)? | Does the result support the **be here now** metaphor? |

*Note*. Questions utilised in the online questionnaire including a rationale on why these questions were asked based on the findings from phase 1. Grammatical and spelling errors (if any) in the questions ("*Question*" column) were corrected for this table view. Also, the readability was improved by replacing "*e.g.*" with "*for example,*".

**Table B3**

*Descriptive Codes Phase 3*

| Descriptive Code | Code description | Code path | Quotes (evidentiary warrant) |
|---|---|---|---|
| *access management* | Managing access rights, for example, share a file or folder | file and folder management context/tasks | Student01: "*What also happens is that I so to say for me, for my learning process create any folder and there are several documents in it and I just want to share a single document from it and then so to say only specifically release access to it.*" |
| *customised structure* | Use custom file hierarchies, for example, create (sub-)folders with a naming convention | file and folder management context/tasks | Professional02: "*So first I put first yes so to say create something like directories or respectively large folder structures.*" |
| *search* | Use search feature to find files or folders | file and folder management context/tasks | Professional02: "*but just to make it go fast I actually use the Windows search because I actually already know what the name of the file is and then I just have to type it in and then that will come so to say.*" |

| Descriptive Code | Code description | Code path | Quotes (evidentiary warrant) |
|---|---|---|---|
| *user history* | Utilise history features, for example, recent files | file and folder management context/tasks | Student01: "*And what also happens is that one so to say last opened files are simply opened (again?) via quick access.*" |
| *bias* | User habits, customs to support innovation | file and folder management context/user stories/emotional impact | Professional02: "*But what I actually wanted to say now was instead of now for example preview text what now is in a text file is that there it just says PDF than I know ok it is now a PDF file and if it is now an Excel file than there is just this Excel symbol with green and that's how I can sort it by colour and then maybe I am more effective too while searching.*" |
| *immersion* | Feeling of immersion in a scenario, for example, being in the situation | file and folder management context/user stories/emotional impact | Professional02: "*You are so to say almost in the middle of what is happening and then can somehow decide for yourself what I want to click on now and what not.*" |

| Descriptive Code | Code description | Code path | Quotes (evidentiary warrant) |
|---|---|---|---|
| *security concerns* | Fears, for example, data loss or data privacy | file and folder management context/user stories/emotional impact | Student01: "*So that nothing is lost or so to say for emergencies and exactly and would say or claim of me that I value it a lot.*" |
| *error-free* | Support error-free navigation, for example, minimise input errors | file and folder management context/user stories/usability | Student01: "*But then I can be sure that I am not suddenly somewhere else caused by a shaker still nearby or maybe even still on it.*" |
| *improve efficiency* | Create a faster, more efficient solution | file and folder management context/user stories/usability | Student01: "*Exactly that you do not have to look at everything and have to watch ok it is that shortly wait three seconds, preview. Is that what I am looking for in there and then to the next.*" |

| Descriptive Code | Code description | Code path | Quotes (evidentiary warrant) |
|---|---|---|---|
| *improve search* | Improve existing search feature for better results | file and folder management context/user stories/usability | Professional02: "*So with the search I think it's important now, for example, that you do not just type but that there is also an integrated, intelligent search.*" |
| *screen space utilisation* | Improve screen space utilisation, for example, to show more information on mobile devices | file and folder management context/user stories/usability | Student01: "*So I think that it's really like you said actually only the display size*" |
| *speech recognition* | Allow speech recognition as an input method | file and folder management context/user stories/usability | Professional02: "*A voice assistant when searching that would somehow also be a very cool function.*" |

| Descriptive Code | Code description | Code path | Quotes (evidentiary warrant) |
|---|---|---|---|
| *type-based rendering* | Optimise render results and features based on object type, for example, different layouts for text-based content versus pictures | file and folder management context/user stories/usability | Professional02: "*Exactly if it were a video or an image then I would find it helpful but with text files rather less.*" |
| *additional relations* | Additional relations for nodes | file and folder management context/user stories/usefulness | Student01: "*And also to have a version history. So that I, what do I know if the computer somehow crashes that one can still access various older versions.*" |
| *device independence* | One coherent user interface across devices | file and folder management context/user stories/usefulness | Student01: "*But I also use the device independence is also very important to me and exactly.*" |

| Descriptive Code | Code description | Code path | Quotes (evidentiary warrant) |
|---|---|---|---|
| *location independence* | Support a central management for files, folders (store, central control) | file and folder management context/user stories/usefulness | Professional02: "*But I know that it is predetermined and one has to always communicate where something was put.*" |
| *reduction* | Features to reduce results based on criterion, for example, filter operation for a data type | file and folder management context/user stories/usefulness | Professional02: "*And depending on what file format I am looking for now can fixate, for example, is it now a PDF ok then I search for PDF for the title.*" |
| *relate and analyse* | Support activities (multiple connected tasks), for example, create a PowerPoint presentation based on input from an e-mail | file and folder management context/user stories/usefulness | Professional02: "*And on the topic usability tests and I did an internship a year ago in the same department and also did usability tests and also collected data.*" |

| Descriptive Code | Code description | Code path | Quotes (evidentiary warrant) |
|---|---|---|---|
| *search* | Feature a search option | file and folder management context/user stories/usefulness | Interviewer: "*Do you mean that in a context, I'll say, of a search function which has to exist, right? So, I am assuming you would need that too, yes?*" Student01: "*Yes*" |
| *mobile devices* | Use of mobile devices (smart phone) (Provisional Code) | general context | Student01: "*I have exactly the same access to the cloud via my mobile device and yes I would also say I use it almost every day, yes.*" |

*Note*. List of result Descriptive Codes after first cycle coding.

**Table B4**

*User Stories*

| ID | User story: As a System User I want to … | Acceptance criteria for System User | Priority | Informed by leading pattern |
|---|---|---|---|---|
| 1 | be able to share files or folders with others (*access management*). So that I can share my work or state of work with (Temporary,) Third-party Users. | Can share files or folders with others | best: Not the most important feature in work practice, rare use | **structure is control** |
| 2 | utilise my own *customised structure* to store and find my files. So that I can I effectively find files relevant to me. | Create nodes like folders<br>Restructure (move at least) a single node to another position<br>Delete (at least) one node and its linked children<br>Edit nodes, changing the text label<br>Select a node of interest | good: Essential work practice | **structure is control** |

| ID | User story: As a System User I want to … | Acceptance criteria for System User | Priority | Informed by leading pattern |
|---|---|---|---|---|
| 3 | *search* for files or folders. So that I can find files I misplaced. | Search for a single file or folder | better: Not the most important feature in work practice | **structure is control** |
| 4 | have a *user history* available to me. So that I can quickly (re-)open the files or folders I used last. | View or utilise the history for last used nodes | better: Not the most important feature in work practice, rare use | **structure is control** |
| 5 | work with a familiar view, for example, familiar icons for folders. So that I feel more secure when using new software (*bias*). | Recognise familiar symbols and associate meaning | good: Essential for a familiar experience | **be here now** |
| 6 | work with a familiar view by adjusting for familiarity, for example, adjust icons. So that I feel a sense of owning the experience (*bias*). | Change symbols or colours of nodes | better: Not the most important feature in a first iteration | **be here now** |

| ID | User story: As a System User I want to … | Acceptance criteria for System User | Priority | Informed by leading pattern |
|---|---|---|---|---|
| 7 | feel immersed in the experience (*immersion*). So that I feel joy when using an application. | Feeling of immersion, being there, being in control | good: Essential to the design metaphor | **be here now** |
| 8 | feel save when using the application, for example, no data loss or not collecting private information (*security concerns*). So that I have trust in the application it is not harmful to me. | System architecture, for example, ensuring data redundancy and local setup | better: Not a core feature in the first iteration, it is not a commercial product | **be here now** |
| 9 | manage my files and folders free from errors (*error-free*). So that I am not distracted from reaching my goal. | Measure similar/low error-rates compared to industry standards like Windows Explorer | good: Essential for user acceptance | **be here now** |

| ID | User story: As a System User I want to … | Acceptance criteria for System User | Priority | Informed by leading pattern |
|---|---|---|---|---|
| 10 | manage my files and folders as fast as possible (*improve efficiency*). So that I am not wasting any of my time. | Measure similar/high task performance compared to industry standards like Windows Explorer | good: Essential for user acceptance | **be here now** |
| 11 | have advanced search capabilities to find my files and folders, for example, utilising artificial intelligence (*improve search*). So that I am not wasting my time searching. | Measure similar/high task performance compared to industry standard search engines like Apple Spotlight | best: Not developing advanced search engines | **be here now** |

| ID | User story: As a System User I want to … | Acceptance criteria for System User | Priority | Informed by leading pattern |
|---|---|---|---|---|
| 12 | have solutions optimised to utilise the available screen space (*screen space utilisation*). So that I can use my devices more efficiently, for example, my smart phone. | View more nodes compared to industry standards like iPhone Files app | good: Essential for user acceptance | **be here now** |
| 13 | use my voice for controlling the application (*speech recognition*). So that I can use it more efficiently when I am on the go. | Ask application to select a node | best: Not a core feature in the first iteration | **be here now** |

| ID | User story: As a System User I want to … | Acceptance criteria for System User | Priority | Informed by leading pattern |
|---|---|---|---|---|
| 14 | have different optimised layouts for data types, for example, section for thumbnail picture only for videos or images but not text (*type-based rendering*). So that I can work with my files more efficiently. | Can view relevant information for nodes depending on their data type | good: Essential for user acceptance | **be here now** |
| 15 | have features for data types, for example, preview features just for images or videos but not text (*type-based rendering*). So that I can work with my files more efficiently. | Can utilise preview feature (or others) depending on their data type | best: Not a core feature in the first iteration | **be here now** |

| ID | User story: As a System User I want to … | Acceptance criteria for System User | Priority | Informed by leading pattern |
|---|---|---|---|---|
| 16 | have support for *additional relations*, for example, to see older versions of a file. So that I can separate certain important tasks from my current activity. | Identify related information to nodes, for example, an older version of a file | good: Essential for user acceptance | **be here now** |
| 17 | have one coherent interface across all of my devices (*device independence*). So that I can use my data independent from a device specific ecology. | Recognise a very similar structure for the interface and interaction | good: Essential for user acceptance | **be here now** |

| ID | User story: As a System User I want to … | Acceptance criteria for System User | Priority | Informed by leading pattern |
|---|---|---|---|---|
| 18 | have central control and overview for all of my files and folders across all of my storage locations (*location independence*). So that I do not have to maintain multiple data locations. Also, I then have the option for an overview across all of my data, for example, to improve communication when working with others. | Create a new file to understand the concept<br>Utilise overview feature(s) to see a large part of the structure | good: Essential for user acceptance | **be here now** |
| 19 | apply filters to my data (*reduction*). So that I can fixate on, for example, certain data types when looking for files. | Apply a filter for a data type | better: Not a core feature in the first iteration | **be here now** |

| ID | User story: As a System User I want to … | Acceptance criteria for System User | Priority | Informed by leading pattern |
|---|---|---|---|---|
| 20 | have support when carrying out my activities by visualising linked information for files (*relate and analyse*). So that I can consider relationships to potentially unknown or historic datasets, for example, to consider results from older similar projects. | View semantically related information to a node (document) | good: Essential for user acceptance | **be here now** |
| 21 | be able to work with (at least) videos and text-based content. So that I can work with the data types most relevant to me. | Select a node of type video and text | good: Essential for user acceptance | **work practice ecology is 2D videos and text** |

*Note*. User stories derived from my codes and patterns.

# Appendix C

## Descriptive Codes

**Table C1**

*Descriptive Codes for the Design Walk-Through*

| Descriptive Code | Code description | Code path | Quotes (evidentiary warrant) |
|---|---|---|---|
| *centralisation* | Design to centralise data storage, operations and navigation | ecological perspective | Professional02: "*So where I see the greatest benefit right now is when I, for example PC as you explained it earlier if I there now would search for pictures or then just hours later then on the tablet or smart phone that I can still continue from where I was last time*" |
| *complete ecology* | Completeness of the conceptual design from the ecological perspective, for example, missing dependencies | ecological perspective | Interviewer: "*that you now here let's say from the ecological perspective something is still missing so external dependencies, more systems or such topics, not. You do not notice anything now, no problem*" Professional02: "*No*" |

| Descriptive Code | Code description | Code path | Quotes (evidentiary warrant) |
|---|---|---|---|
| *favourites and multiple instances* | A feature to set and utilise favourites across devices including the option to use multiple instances of the application | ecological perspective | Professional02: “*Which would also be really cool … such a favourite function. … that this is then (inc.) transmitted everywhere on the tablet or smart phone*” |
| *gestures to replace touch targets* | Gestures to replace touch targets from the ecological perspective | ecological perspective | Interviewer: “*With the mouse that becomes … a problem again, does it not*” Professional02: “*Exactly*” |
| *node layout* | Preferences regarding node layouts from the ecological perspective | ecological perspective | Interviewer: “*works well across devices, you think.*” Professional02: “*Yes, exactly*” |

| Descriptive Code | Code description | Code path | Quotes (evidentiary warrant) |
|---|---|---|---|
| *node scaling* | Scaling of nodes based on the available screen space | ecological perspective | Professional02: "*I actually think I prefer the second view because it just looks a lot tidier to me. … the first one is already a bit confusing and overwhelming … But I think like this I now really have what I want to watch centrally displayed*" |
| *responsive design* | Optimise the design for different devices, for example, input controls | ecological perspective | Professional02: "*it says here once the identical view, meaning it is on tablet, PC and smart phone then probably the same. On the PC … the desktop view and I imagine it like this now that on tablet and smart phone then this, how is this called, … that this thing is responsive, so the view.*" |
| *responsive node layout* | Dynamic node layout based on the device and orientation of the device | ecological perspective | Professional02: "*So if I change the view from portrait to landscape mode then maybe one can display it horizontally*" |

| Descriptive Code | Code description | Code path | Quotes (evidentiary warrant) |
|---|---|---|---|
| *custom colour and icons* | Customising options to change the colour scheme and icons | emotional perspective | Professional02: "*So I think such a customising would not even be so bad*" |
| *familiar icons* | Familiar icons to support usage | emotional perspective | Professional02: "*The one with the P when it's presented in such a way that one can directly see ok that is a PowerPoint presentation, I find it very valuable. Because that always helps me because I am more of a visual person*" |
| *node layout* | Preferences regarding node layouts from the emotional perspective | emotional perspective | Professional02: "*Yes it just feels a lot better, so (...) the view is much more pleasant for me now to look at. One can do it horizontally but not mandatorily*" |
| *automatic zoom in 3D Overview* | Features to automatically move away the camera when the 3D Overview mode gets enabled | interaction perspective | Professional02: "*No, (...) actually also the second option. It is just more time efficient. I do not have to do much, it just happens automatically*" |

| Descriptive Code | Code description | Code path | Quotes (evidentiary warrant) |
|---|---|---|---|
| *colour blind accessibility* | Accessibility challenges when designing | interaction perspective | Professional02: "*but maybe one can change that as well because of the green colour, there are also people who have a red-green deficiency*" |
| *consistent visualisation* | Visualisation of nodes based on their data type | interaction perspective | Interviewer: "*You do not see it as, is no, no usability gain, it will not make you faster*" Professional02: "*Yes, exactly*" |
| *dynamic scaling in 3D Overview* | Dynamic scaling of nodes based on user distance in 3D Overview mode | interaction perspective | Professional02: "*What I would prefer now, I believe so that it is clean and I believe such a view is first of all informative, really well constructed, I would then prefer that with the tooltip*" |

| Descriptive Code | Code description | Code path | Quotes (evidentiary warrant) |
|---|---|---|---|
| *efficiency in 3D Overview* | Efficiency in 3D Overview mode when browsing files | interaction perspective | Professional02: "*In the third view, I think the solution is really cool with the 3D view that you look in the background, so to say, like a waiting list or waiting queue can already see the next interesting folder that could just come into question for me*" |
| *file and folder operations in 3D Overview* | Preferences regarding file and folder operations in 3D Overview mode | interaction perspective | Professional02: "*Actually the first one, I believe it is much faster than jumping back … to the other view*" |
| *meta data in 3D Overview* | Display of additional meta data in 3D Overview mode, for example, number of files in a folder | interaction perspective | Professional02: "*I do not believe that brings any added value for me personally*" |
| *size of touch targets* | Importance of touch target sizes | interaction perspective | Professional02: "*I believe that can be a bit problematic, I am a user who, for example, has two thick fingers*" |

*Note*. List of result Descriptive Codes after first cycle coding.

**Table C2**

*Descriptive Codes for the Design Review*

| Descriptive Code | Code description | Code path | Quotes (evidentiary warrant) |
|---|---|---|---|
| *customised synchronisation* | Features to allow multiple synchronised states to choose from | ecological perspective | Student01: "*Maybe so (like?) standard views that you can customise and can call up when required or something*" |
| *document availability* | Working with documents which have to be locally available | ecological perspective | Student01: "*then what I believe would still be exciting is the memory status, so if you open a document then you have to, so to say momentarily download it locally once. That one can see somehow ok is that downloaded locally or is that just in the cloud right now*" |
| *MP3 files* | Usage of MP3 files | ecological perspective | Student01: "*except then MP3 but that is not really up to date anymore*" |
| *value* | Relevancy of the novel method | ecological perspective | Student01: "*So I do not quite see the added value to a current, to a conventional folder structure*" |

| Descriptive Code | Code description | Code path | Quotes (evidentiary warrant) |
|---|---|---|---|
| *habits* | User habits, customs | emotional perspective | Student01: "*No, I believe that works. So, it could just be unfamiliar because I believe if you currently somehow add a folder somewhere in known programs then they are simply listed from top to bottom and not from the middle.*" |
| *action buttons* | Action buttons in 3D Overview mode | interaction perspective | Student01: "*Maybe I do not know there should be an emoji in there or an icon or maybe even such a word select or something.*" |
| *camera position* | Camera positions in 2D Navigation mode | interaction perspective | Student01: "*No, I see well. I believe it would be even cool if you could see the whole circle so to also see the files behind.*" |

| Descriptive Code | Code description | Code path | Quotes (evidentiary warrant) |
|---|---|---|---|
| *map navigation* | Map like navigation in 3D Overview mode | interaction perspective | Student01: "*Yes, because I believe so direct gestures are always more intuitive than indirect gestures. Well then, I mean one knows it when one is, for example, on a map online somewhere in the browser and can only move back and forth if one clicks on a plus and minus then and every time one presses on the plus it goes so to say a certain amount further in, zooms in.*" |
| *metadata* | Metadata for documents | interaction perspective | Student01: "*Ok, then I believe it would be important to see when the document was opened or edited the last time*" |
| *preview* | Preview content | interaction perspective | Student01: "*Then what I imagine is also cool if one somehow presses for a long time so that one sees such a short preview*" |

| Descriptive Code | Code description | Code path | Quotes (evidentiary warrant) |
|---|---|---|---|
| *scripting* | Customisation, for example, to rename multiple selected folders | interaction perspective | Student01: "*that one somehow if one is in a node somehow can select 10 folders and then so to say can activate or deactivate them in a Word document, whether they should be numbered or not.*" |
| *sorting* | Sort features | interaction perspective | Student01: "*Yes, I only have one situation in mind now, if one so to say has the, yes, the open file explorer on the laptop, then one can sort according to different sorting criteria with one click*" |
| *speed* | Navigation speed in 3D Overview mode | interaction perspective | Student01: "*that one says if one puts one's thumb on the controller and stays on it for, I do not know a second then a slider opens and the more one goes up, so if one goes up very slightly then he zooms in very slowly*" |

| Descriptive Code | Code description | Code path | Quotes (evidentiary warrant) |
|---|---|---|---|
| *user guidance* | User support, for example, an integrated user guide or manual | interaction perspective | Student01: "*Well, I think it would be enough if one so to say downloads it fresh, which then gives a guide through and then so to say shown once when you click on the ribbon then that happens and so on*" |

*Note*. List of result Descriptive Codes after first cycle coding.