Visualising a complex data solution landscape

Approaching organisational issues from an Interaction Design perspective

Olof Ingelsten

ingelstenolof@gmail.com

Interaction Design

Bachelor

22.5HP

Spring 2023

Supervisor: David Cuartielles

# Abstract

# Acknowledgements

I want to give my everlasting appreciation to coffee and the other chemical addictions that helped write this thesis.

# Table of Contents

Table of Contents

[1 Abstract 3](#_Toc133509107)

[2 Acknowledgements 3](#_Toc133509108)

[3 Table of Contents 3](#_Toc133509109)

[4 Introduction 5](#_Toc133509110)

[4.1 Context 5](#_Toc133509111)

[4.2 Aim 5](#_Toc133509112)

[4.3 Delimitations 5](#_Toc133509113)

[4.4 Structure of the thesis 5](#_Toc133509114)

[4.5 Problem statement 5](#_Toc133509115)

[4.5.1 Developing the research question through the process 5](#_Toc133509116)

[4.6 Related works 5](#_Toc133509117)

[5 Theory 5](#_Toc133509118)

[5.1 What is Interaction for Data Visualizaton? 5](#_Toc133509119)

[5.1.1 Interplay 8](#_Toc133509120)

[5.1.2 Person 8](#_Toc133509121)

[5.1.3 Data Interface 8](#_Toc133509122)

[5.1.4 Action, action-reaction, and reaction perceived as such 8](#_Toc133509123)

[5.1.5 Data-related intent 9](#_Toc133509124)

[5.2 Big Data Visualisation and Analytics: Future Research Challenges and Emerging Applications (Andrienko et al., 2020) 9](#_Toc133509125)

[5.2.1 Gennady & Andrienko. N 10](#_Toc133509126)

[5.2.2 Fekete. J-D 10](#_Toc133509127)

[5.2.3 Fisher. D 10](#_Toc133509128)

[5.2.4 Kraska. T 11](#_Toc133509129)

[5.2.5 Oulasvirta. A 11](#_Toc133509130)

[5.3 Voyagers and Voyeurs: Supporting Asynchronous Collaborative Information Visualization 12](#_Toc133509131)

[5.4 Constructive Visualisation 13](#_Toc133509132)

[5.4.1 Design challenges 13](#_Toc133509133)

[5.4.2 Components and processes of constructive visualisation 15](#_Toc133509134)

[5.5 Collecting thoughts and summarising learnings from research 15](#_Toc133509135)

[6 Design process and methods 15](#_Toc133509136)

[6.1 Planning 15](#_Toc133509137)

[6.1.1 User Centered Design 16](#_Toc133509138)

[6.1.2 Literature research and desktop research 16](#_Toc133509139)

[6.1.3 Critical reflection on the double diamond 16](#_Toc133509140)

[6.2 Early design stages 17](#_Toc133509141)

[6.2.1 Interviews as a research method 17](#_Toc133509142)

[6.2.2 Exploratory discussions 18](#_Toc133509143)

[6.2.3 Drawbacks 19](#_Toc133509144)

[6.2.4 Analysis process 20](#_Toc133509145)

[6.2.5 Results from interviews 21](#_Toc133509146)

[6.3 Sketching & Prototyping 25](#_Toc133509147)

[6.3.1 Prototyping methodology 25](#_Toc133509148)

[6.3.2 Three-point approach 25](#_Toc133509149)

[6.4 User testing 30](#_Toc133509150)

[6.4.1 Paper about user testing? 30](#_Toc133509151)

[6.4.2 Planning and user test process 30](#_Toc133509152)

[6.4.3 Iterating on the design 32](#_Toc133509153)

[7 Discussion 36](#_Toc133509154)

[7.1 The prototype 36](#_Toc133509155)

[8 Conclusion 38](#_Toc133509156)

[9 References 38](#_Toc133509157)

# Introduction

## Context

* General overview on the field of visualisation and the nhow it relates to Interaction Design
* Governance intro

## Aim

This project aims to explore the role and effect Interaction Design projects can have on large organisations. This is done through a user centred design project prompted by and in collaboration with IKEA of Sweden.

## Delimitations

* The prototype
* Stakeholders

## Structure of the thesis

## Problem statement

### Developing the research question through the process

## Related works

# Theory

## What is Interaction for Data Visualizaton?

In the article “What is Interaction for Data Visualization?” (Dimara & Perin, 2020) Dimara and Perin look at the definition of the term “interaction” both from the perspective of the field of Data visualisation (VIS) and from Human-Computer Interaction (HCI). They argue that there are several inconsistencies in how and when certain terminologies and methodologies are used in both fields regarding interactive data-driven visualizations. The basis of Dimara and Perin’s arguments is based on academic literature research in both VIS and HCI, as well as a questionnaire sent to several senior researchers in both fields. They start by defining an outline of how the field of VIS uses and perceives interaction as a medium of design.

From the perspective of VIS, they define three concept approaches for interaction: system-, task-, and human-centric. *System-centric* approaches in VIS disregard the specifics of the user, such as specific roles and in what way they interact with the artefact. Instead, this approach is more technology centred and is more concerned with program-specific operations. *Task-centric* approaches in VIS involve the user to a lesser degree, designing their organisational roles as developers and authors, and focusing more on low-level tasks within the artefact (such as re-arranging, filtering, inputting data etc.). Lastly, the *human-centric* approach in VIS is, in contrast to the task-centric, more concerned with *user intent*, and high-level interactions. With *intent,* Dimara & Perin means a “high-level cognitive activity” that can also transcend the scope of interactions within a given software, such as note-taking and mental images of previous interactive “history”. They identify that, in general, interaction within VIS is defined as a type of dialogue between a user and the visualisation, which is not entirely different from how an interaction designer could view a similar system. However, they identify that the core of the issue of interaction in VIS applications is the lack of traditional flexibility for the user in the flow of interaction. The internal definition of a user is mostly limited to a data analyst whose main tool of interaction is a computer with a mouse, thus greatly limiting the scope in which new designs could be made.

Dimara & Perin’s view on the state of interaction within HCI is based mainly on “*What is interaction*” (Hornbæk & Oulasvirta, 2017) who define “*seven concepts that characterize interaction in HCI*”, namely: Dialogue, Transmission, Control, Tool Use, Optimal Behaviour, Embodiment, and Experience. **I guess here I could go into a bit further detail, maybe?**

Through both of these perspectives and definitions of interaction, Dimara & Perin detail missing aspects of interaction, differences between VIS and HCI, and a common vocabulary for defining interaction in data visualisation design projects. The key differences they identified are as follows: They see a differentiation between HCI and VIS in the interpretation of which *“entities”* take part in the moment of interaction. HCI interactions detail mainly the interplay between the human and the computer, whereas VIS interprets the data component within the application as a third entity in addition to the others. There are contrasting ideologies in regard to how performant the flow of interaction should be. The design ideology within HCI is largely based on the notion of making effective flows and designing intuitive interactions, while the flow in data visualisation applications tends to be slower and more complex, given that the data should invite thoughtful reflection and provide the user with new insights beyond the visualisation itself. Another key difference identified is how both fields approach the *intent* of interaction. Intent within HCI is more aligned with the emotional state of the user in the flow of interaction and sees the entire flow as having a goal in mind and the tool as a means to that end. (Hornbæk & Oulasvirta, 2017) identifies that user intent in HCI often starts outside the actual flow of interaction for which we have designed. Demara & Perin compares this with a user ordering food online, the intent to get food exists before the user even interacts with the artefact that enables them to order. The artefact, in this case, could be a phone application with an intuitive interface and playful interactions that gives the user affordances to express their intent, but the intent, nonetheless, began outside of this flow. Dimara & Perin summarises this view as follows:

“So while a good design should reveal **how to use** the tool (e.g., using affordances), it should not impose **what to do** with it”.

The design of the application in the previous example could be designed for maximal ease of use for the user. However, the designer can only provide the tools for an intended experience, not how users actually want to experience it.

This is contrasted by VIS where the intent, defined as *data-related intent* by Dimara & Perin, is much narrower than HCI and, in a way, prone to change during the course of interaction. They call the intent data-related as visualisation is, by definition, a type of interface whose main purpose is to collect, and present abstract data in a way accessible by humans. The narrowness of intent in VIS is due to its main purpose of displaying data. However, this narrowness is complemented by its iterative nature, where the interplay of user, data, and artefact is ever-evolving as the user probes for new insights, which in turn affects the user’s intent.

Finally, flexibility within each design differ between HCI and VIS. Where interactions in HCI are most often low level functions that act as parts of a whole and allow for flexibility within the defined scope, flexibility in VIS designs encompass a larger scope and involves several aspects and parts of the interactive flow. Dimara & Perin define flexible and interactive data interfaces as follows:

“Flexibility within a data interface is the number of distinct, allowable actions of a person on the interface, as well as the number of interaction means with which the person can perform each action.”

With this definition, it is possible to map, describe, and compare visualisation methods through their means of interaction and possible actions.

By establishing a common ground for the definition of interactivity, Dimara & Perrin has defined scope and definitions in both HCI and VIS and found commonalities and differences in methodology, ideology, and terminology. With both fields’ views established, they synthesise a short definition for how interaction should be viewed in a setting of visualisation

“Interaction for visualisation is the interplay between a person and a data interface involving a data-related intent, at least one action from the person and an interface reaction that is perceived as such.”

To further define the definition of interaction, several sub-components are needed with their own requirements that can define whether a visualisation interaction could be viewed as interactive.

### Interplay

An interaction in a data visualisation setting should allow for a back-and-forth dialogue with the system. The authors chose to differentiate *interplay* from *dialogue* as they find it to define sequential interactions when that may not necessarily be the case in VIS interactions.

### Person

Dimara & Perin talk about *entities* that engage in interactions. When defining the main entity interacting with the system, they call it a “person”, which, supposedly, is gender-neutral to a greater degree than the user while still allowing for non-human entities for interaction.

### Data Interface

By exchanging the terminology from the *visualisation system* to the *data interface*, Dimara & Perin open up a larger scope of interactive modalities other than the visual modality implied through the visualisation system. The term data interface also allows for greater agnostic interpretations that extend to scope to visualisations that are tactile, non-digital, auditory etc.

### Action, action-reaction, and reaction perceived as such

When referring to an action in a visualisation interaction, Dimara & Perin, similarly to how they approach the data interface, invite a wider approach to interpretation. Therefore, an *action* could be any interaction performed by the aforementioned person, whether it is physical, mental, high-level, or low-level, that results in a reaction from the interface.

As a continuation of action, action-reaction refers to the dialogue that exists between the person and the data interface. Specifically, it refers to the reactionary nature of natural dialogues between two people. The design challenge, however, is determining what type of reactionary modality best fits each interaction and how that is relayed to the user.

A necessary aspect for a reaction in a dialogue to be meaningful is that the recipient also *perceives* it as such. Suppose an interface reacts in a way where a clear cause and effect cannot be determined. In that case, the user becomes disconnected from the dialogue resulting in non-satisfying or non-effective interactions. Dimara & Perin summarised their view on perceiving reaction:

“...for a dialogue to occur the person must perceive the causality of their action /…/ To capture causality, we specify that the reaction must be perceived **as such**.”

### Data-related intent

As previously mentioned, the aspect of intent is important in both HCI and VIS and is defined here as *data-related intent*. This aspect of the visualisation interaction implies that the person interacting has a specific intent, before, after, or during the interaction that is related to the third entity in data visualisation interactions, namely, the data. Similarly, to the other components, the modality and temporality of the data-related intent are left open for interpretation; neither do they assign the intent to any specific entity, be it the person interacting or the designer creating the experience.

## Big Data Visualisation and Analytics: Future Research Challenges and Emerging Applications (Andrienko et al., 2020)

This is a report that collects thoughts and challenges by fourteen scientists in many different fields that relate to data visualisation and analytics like Information Visualisation, Human-Computer Interactions, Machine Learning, Data Management & Mining, and Computer Graphics. The report aims to provide an overview from many different expert perspectives on the current goals and challenges in the field. Each researcher was tasked to base their predictions and thoughts along two prompts, namely:

* *The top future research challenges in Big Data visualisation and analytics*
* *The top emerging applications in the context of Big Data visualisation and analytics*

A point shared by many of the researchers and the creators of the report for the future of Data Visualisations, or Data Interfaces as (Dimara & Perin, 2020) would call them, is the focus on their interactivity and accessibility. They stress the importance of how analysis of data often leads to results not seen at face value. Thus, future Data interfaces “*should provide sustainable insights and insight recommendations*”. In addition, (Andrienko et al., 2020) stress that the tools created for visualising data should be accessible to the degree that *non-data scientists* would be able to interact with and be provided with possibilities to gain new insights. (Andrienko et al., 2020) further argue that the tools currently available have been science and research focused and are built in such a way that they are extremely powerful and advanced, at the cost of usability. For the future of tools in data visualisation, greater care should be taken to construct user experiences that are *user-oriented* and intuitive to use while retaining their usefulness when applied.

In the following section, I will, in short, summarise each researcher’s main points and thoughts and how they could relate to the role of Interaction Design in Informational Design in Big Data applications. **The selection of researchers is limited to those who pose questions applicable to Interaction Design**

### Gennady & Andrienko. N

From the perspective of Gennady & Andrienko, using visualisations to outsource data processing should be continuously implemented throughout entire project processes, and not only at the end, as is usually standard. Since the primary function of data visualisation is to aid the user in seeing patterns and gaining otherwise missed insights, having a continuous dialogue with a visualisation system could effectively streamline current processes.

For future challenges and implementations of data visualisations, Gennady & Andrienko predict a need for new tools and methods to teach designing more complex systems than is currently available. With complex systems, they refer to non-trivial multi-dimensional data sources that require solutions beyond conventional graphing solutions. The research topic in VIS that Gennady & Andrienko conduct is mainly within *visually-driven analysis of spatio-temporal data*, which changes over several dimensions.

### Fekete. J-D

Fekete says, “To be effective, visualisation and visual analytics should be interactive, meaning that computing visual representations should happen in a few seconds, interacting on them should be responsive.”. This focus on the interactivity of data interfaces is further developed by (Dimara & Perin, 2020), as mentioned before. It displays an aspect of the design of data visualisations where Interaction Designers have opportunities to provide real value to the development of both fields.

Fekete further argues that implementing Progressive Data Analysis is an approach similar to what is done during the UX design of user flows on the internet. Large computational tasks are sectioned and divided into manageable chunks where the user can follow the process and deviate at any point instead of waiting for results.

### Fisher. D

Fisher talks about how we pose inquiries to the visualisation tools we use and how quicker responses from the tools would lead to a clearer dialogue between person and interface. With a focus on a dialogue between parts, the interaction becomes more iterative and promotes building insights. To further develop the field, Fisher urges designers to create design systems that allow for continuous interactions and that are laser-focused on specific tasks. By designing for specific use cases, Fisher argues, we as designers will better understand users' wishes, use patterns, how they wish to interact, and to what fidelity the tools need to show. Through this user-centric design ideology, Fisher believes that broader issues and questions in the field will become more manageable to answer and design.

### Kraska. T

Kraska talks about future challenges in data visualisation and how certain interactive elements could be implemented to increase ease of use in complex data structures. When designing visualisations in existing software, an issue arises when sufficiently complex data is used. Namely, the flexibility of use gets progressively harder, and insights gained rely heavily upon the expected outcomes by the user. To combat this issue, Kraska sees a design opportunity to design tools that do not restrict the user in how they choose to approach a question and allow for flexible changes during an interaction. The flexibility of such a system should then be quick in response regardless of the data size used. Similar to Andrienko, Kraska urges that the design of these systems should be accessible and user-centric in such a way that the use of visualisation interfaces should allow for non-data scientists to interact and gain insights. To make these data interfaces more accessible, designers can approach the interface's design from novel perspectives. These perspectives could be changes in the modality of interaction, as previously mentioned by Dimara & Perin, as well as the design of the interface itself. Kraska urges a change in design thinking from the current focus on technical solutions done in the backend to the user-centric ideologies of HCI and IxD:

 “Design the user interactions first and then figure out the system [the backend], which can actually support them.”

### Oulasvirta. A

Oulasvirta provides a technology-agnostic perspective and wishes to incorporate user-centred design methods as a core part of designing and understanding data visualisation and analytics. To approach VIS from a user-centred point of view, Oulasvirta suggests that a greater understanding of human perception is needed. By understanding how humans perceive their environments and how the human brain decodes data, we can design visualisations that are logically intuitive rather than computationally intuitive. By designing interfaces from this perspective, Oulasvirta says, we can present data and have tools to *explain* it.

For the future of VIS, Oulasvirta sees great value in building foundational research into the psychological aspects of human understanding rather than technological advancements. Oulasvirta summarises this perspective with a couple of questions that challenge the main issues in VIS and asks the designer to challenge their design practice:

“**Why** should a particular visualisation favored over another one in some context? **Why** should one choose particular design parameters over other ones? What are the **limits** or a particular type of visualisation, what can it do and -more importantly- what can it **not** do.”

## Voyagers and Voyeurs: Supporting Asynchronous Collaborative Information Visualization

(Heer et al., 2007) presents a design that implements aspects of asynchronous collaboration in a data visualisation program, *sense.us*. They make a point that the outcome of the design is to connect the cognitive and perceptual benefits of visualisation together with the benefits of interpersonal social interactions. To promote the social aspects of collaboration (Heer et al., 2007) implemented functions within the data visualisation software that allows users to comment, annotate, share insights, and discuss, all through a (for the time) novel method of interaction. From user tests, (Heer et al., 2007)could conclude that, through the implementation of social interactions and annotations, users engaged deeper in the data visualisations and helped each other to, make sense of, and gain new insights that otherwise were not presented through the visualisation. For future developments, (Heer et al., 2007) see a value in the addition of social aspects within the visual analysis and, through that, empower users to receive both in-depth and general knowledge in their visualisations.

(Heer et al., 2007) identifies three common features in visualisation designs that implement multi-user bookmarking functions. *Application bookmarks* allow users to save a certain state or position within a visualisation to allow for future reference. Application bookmarks can be shared between users or kept as private references. An example of an application bookmark is the pin functionality in Google Maps. Users can save a specific location and share that with other users to show directions, locations, or points of interest. A separate action that supports bookmarked visualisations is what (Heer et al., 2007) refer to as *independent discussions.* Independent discussions are places outside of the data interface which can point to certain parts within the interface, for example, sharing your location over a text message leads to a hyperlink that leads to the operating systems map application. A distinction of the independent discussions is that their interaction with the data interface is one-way, meaning that external sources access the interface while users within the interface cannot access the independent discussion. Finally, (Heer et al., 2007) define *embedded discussions*, as a functionality within the data interface itself that allows for streaming information that can be accessed and viewed by any user. An example of an embedded discussion would be the reviewing system in Google Maps, where users can leave comments and reviews on locations and shops for anyone to see, all within the application.

For the graphical annotations in the application, they used common tools and methods that their users already would be accustomed to. For example, they drew inspiration from the Microsoft Office package of tools, specifically the annotation tools in Microsoft PowerPoint. The tools empowered users to add free-form drawings, lines, arrows, shapes, and text annotations. (Heer et al., 2007)found that the annotated comment generated by users led to a greater common understanding of the content in the visualisations as users now had the opportunity to ask, and answer, questions directly at the point relevant in the visualisation. Users could provide each other with contextual information not necessarily present in the data set. The annotations also provided an additional data point for the interface itself as the amount of searchable data for the user expanded with each comment, thus making searching for specific data points easier. Several users expanded upon the base functionalities in the software and used existing functions to build more advanced systems. One such added system was the user implementation of narratives on the data sets. Through the use of annotations and links between different visualisations within the annotation, users could expand upon the existing data set by situating the data through historical narratives. One user built a narrative that explored the progression of female workers in historically male-dominated workplaces over a time period.

## Constructive Visualisation

(Huron et al., 2014) present in their paper new methods and terminologies for a democratisation of dat a visualisation for non-data scientist designers. They do this through Constructive Visualisation, a perspective on designing simple, dynamic, and expressive visualisations. Most important, constructive visualisations aim to empower users to, with simple building blocks, construct complex structures that embody their specific needs for visualisation. To divide Constructive Visualisations (Huron et al., 2014) identify several design challenges for visualisations through comparisons with existing VIS solutions and further develop their context through examples of research on the psychology of how constructivist approaches affect children's learning. From these perspectives (Huron et al., 2014) present components and processes for designing a Constructive Visualisation. Further, (Huron et al., 2014) analyse existing design solutions from the perspective of Constructive Visualisation and identify how they approach the design challenges from different angles and how they solve them.

### Design challenges

(Huron et al., 2014) identifies three design challenges for making constructive visualisations (visualised in Table 1):

**Keeping it simple**. Huron et al. compare and sees simple designs as something that users know intuitively and have known their entire lives. An example of an intuitive action, Huron et al. explain, is that of sketching. Regardless of age and skill, most people have the capacity to set pen to paper and express an idea through that medium. Here Huron et al. identify that a key to designing for simplicity is to identify and leverage actions that are intuitively connected to daily-, and life-long activities.

**Enabling expressivity**. Huron et al. define expressivity along the three freedoms defined by (Bertin & Barbut, 1968):

1. That there is a degree of freedom in defining the sign.
2. There is a degree of freedom in attributing properties to the sign.
3. And that there is a degree of freedom in how the sign can be arranged.

Adapting Bertin & Barbut’s degrees of freedom to a constructive visualisation, according to Huron et al., would entail an interface which would allow users to design non-destructively with reversible actions, allow users to change and adapt the interface to their needs. The interface should be accessible to users to make changes. Herein, Huron et al. define the term *plasticity* as “*/.../ the ability to re-model during the creation process*”.

**Incorporating dynamics**. Huron et al. explain that the greatest challenge is to design an interface in such a way that allows for simplicity and plasticity while still updating and adapting as the source of data changes. Currently, the most common way of designing dynamic visualisations is through coding, which is in itself a non-common practice that requires in-depth knowledge to use practically. Therefore, Huron et al. define the challenge of dynamics to be one that satisfies the previous challenges while still allowing for the non-proficient user to contribute and interact.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Visual Mapping Paradigm | Keep it Simple | Enabling expressivity | Incorporating data dynamics | Manipulation of Visual | Skills learning | CS System examples |
| Using | ✅ | ❌ | ✅ | Indirect | Medium | Excel, Tableau, Google Chart |
| Drawing | ✅ | ✅ | ❌ | Direct | Easy/Medium | Pen & Paper, Photoshop, Illustrator |
| Coding | ❌ | ✅ | ✅ | Indirect | Hard | Processing, D3.js, Infovis toolkit |
| Constructing | ✅ | ✅ | ✅ | Direct | Easy | Unknown |

Table 1. (Huron et al., 2014) situates constructive visualisation against existing VIS approaches through their perspective.

### Components and processes of constructive visualisation

From these design challenges, in combination with an analysis of lessons from constructivist ideas and research, Huron et al., propose the following definition for constructive visualisation, “*... the constructivist approach to designing information visualisation is the act of constructing a visualisation by assembling blocks, that have previously been assigned a data unit through a mapping.*”.

#### Components

**The basic unit: a token**. As defined by Huron et al., a token is the basic building block of which the data interface is compromised. The token can be either physical or virtual and take any shape, colour, volume, texture etc., as well as interactive elements such as moving the token. Each token's purpose is to represent a data element chosen through data mapping.

**Token grammar**. The token grammar of an interface defines different types of base tokens and how their attributes are mapped to the data. Multiple attributes of the same token can have different types of data mapped to them. Huron et al. stress here that the mapping of data and how the mapping is applied to the tokens are defined by the user designing the visualisation.

**Environment**.

**Assembly model**.

#### Processes

**Environment initialisation.**

**Mapping data to “tokens”, and data properties to token properties.**

**Assembling the tokens.**

**Evolution over time.**

## Collecting thoughts and summarising learnings from research

# Design process and methods

## Planning

In every design project, there are two realities when approaching conventional methodologies. One reality is purely theoretical and is often applied at the start of the project when planning research and prototyping methodologies. Often these methodologies are presented straightforwardly and generically to seem applicable to any design process. However, as the design process unfolds, the second reality sets in the reality where eventually unforeseen events require you to adapt and change your approach. This is not necessarily bad and requires you as a designer to improvise and use your skills as the process develops. Therefore, in the following section, I will describe and outline the design process from start to finish, with each methodological choice described as they were applied. The structure of the text follows the design process chronologically, with literature and user research leading into sketching and prototyping, which in turn ends in user testing and final prototype iterations.

### User Centred Design

* User Centred Design
  + (Abras et al., 2004; Mao et al., 2005; Norman & Draper, 1986)
* Qualitative research
  + (Brinkmann et al., n.d.; Fossey et al., 2002; Thorne, 2000)

### Literature research

The research process in this thesis includes both primary research through in-person user interviews and user tests, which is supplemented by secondary research in the form of a literature review of relevant studies in both Interaction Design and Visualisation Design fields. Secondary research in the form of a literature review is commonplace in academic research. It provides a way for the researcher to review existing research and either synthesise new theories or situate research in relation to each other (Martin & Hanington, 2012). The secondary research conducted in this thesis is mainly what (Ralph & Baltes, 2022) defines as an *Ad Hoc Review*. An Ad Hoc Review consists of a selection of papers chosen by the researcher to support, contextualise, develop, or otherwise situate the main topic of the research project. What differentiates the Ad Hoc Review from other literature review methodologies is its lack of systematised structure in how papers are picked. (Ralph & Baltes, 2022) calls the method for paper selection in Ad Hoc Reviews *purposive sampling*.

For this literature review, papers were not picked to support a position but also to situate and contextualise a design through existing perspectives in the field. The literature research was conducted through desk research using online databases such as *Google Scholar* and field-relevant databases such as the *ACM Digital Library* and *IEEE*. The process of filtering relevant research papers can be summarised as follows:

1. **How many citations does the paper have?** Older papers with many citations could be seminal in the field and therefore be relevant to include. Similarly, if a paper was released recently (e.g., 2020-present), it might not have many citations, but the findings could still be cutting-edge developments.
2. **When was the paper published?** When conducting literature research in human-computer interaction-related fields, many seminal papers (e.g., (Norman & Draper, 1986)) are becoming double-edged swords as their age is becoming apparent in the way they discuss “current” technology. However, many non-technical discussions about user-centred design hold today.
3. **How are the findings presented in the abstract?** This filtering category is the most abstract of the three and is based on the language used and how the research is presented in a paper. If the abstract of the paper is presented straightforwardly and presents the findings succinctly, then the paper is often decent enough for review.

### Critical reflection on the double diamond

The double-diamond research process is a widely adopted method for describing and planning a design project. Design Council UK popularised the method through in-depth methodological research of 11 large design companies (Design Council, 2007; Gustafsson, 2019). In short, the method divides the design process into four phases, *Discover*, *Define*, *Develop*, and *Deliver*. A recurring theme of the method is a divergence and convergence of thought and design. The *discovery* phase diverges and opens the design space for exploration and research, which converges during the *define* phase into more well-defined design opportunities. Similarly, the *develop* phase diverges, based on the design opportunities, to different prototyping iterations, which conclude through converging to a final design in the *deliver* phase (Design Council, 2019).

As the Design Council (2019) themselves say, the common description of the double diamond is often not as linearly applied as the initial clean description could lead one to believe. A design process is often one of constant iteration, and a true-to-life double diamond process would need to include several loops leading back and forth from different parts (Gustafsson, 2019). Therefore, applying the double diamond as a general scheduling tool to describe the overall expected design process permits the design process to change and evolve as needed while retaining an overarching structure. However, the strengths of the method's generality are also its’ greatest weakness. As the design process unfolds and new insights determine new directions, the initial plan requires constant updating if you expect it to be constantly applicable. Unforeseen things happen in every project, and being locked into a specific design process does not permit a creative process. A more flexible mindset and approach is needed. The actual process of design is certainly one of divergence and convergence. However, it is much more chaotic and intricate than the visual model of the double diamond represents. The actual process could be likened to a fractal pattern of doing design and learning insights.

With these insights and critiques in mind, this thesis approached the double diamond pragmatically. I used it as a personal scheduling to check progress and as a tool for discussing project planning with stakeholders. The schedule (Figure 1) included milestone dates to keep in mind and key aspects and features to be developed throughout the project.

Diagram

Description automatically generated

Figure 1. The Double Diamond process applied to this project.

## Early design stages

### Interviews as a research method

Short intro to interviews?

(Knox & Burkard, 2009; Martin & Hanington, 2012; Wood, 1997)

#### Interview process

Meetings were booked with each UX team designer during the project's first official week for the initial exploratory interviews. This included UX Designers, UX Design Leads, and the Technical Manager. The meetings were recorded through audio and were analysed and summarised afterwards in an affinity diagram using the digital whiteboard tool *FigJam*.

#### Meeting the team

Before meeting each team member individually for the interview, I invited all the stakeholders to an introductory in-person meeting at the IKEA offices. The purpose of collecting everyone for this first group meeting was two-fold. Firstly, it was an opportunity to get acquainted with the entire team in a semi-informal setting and allow them to get to know me. Through this session, I had the opportunity to explain who I am, my intentions with the project, and my expectations, and meet their comments regarding the thesis.

Secondly, I explained the project's purpose and expected outcome based on the problem statement provided by the supervisor at IKEA. Further, I outlined the process and requirements of writing a bachelor's thesis and how I would collect and save data through the consent forms provided by Malmö University. A benefit of explaining this in front of the entire team is that, entering each meeting, everyone had a common baseline understanding of the project and its stakes. This gave me a canvas upon which I could direct each following discussion to be as efficient as possible without re-explaining the project at every meeting. However, although having each member up to date with the project allowed for relevant discussions, the interviewing process could have been made more efficient to yield better results due to reasons explained in the following section.

### Exploratory discussions

The individual meetings with each team member consisted of one-on-one sessions spanning 45-60 minutes in person or digitally (Four physical meetings, two digital). In addition, each interview was recorded through audio for later reference with consent from each interviewee. By recording each interview, I had the opportunity to focus more on driving interesting discussions with more fluid conversation rather than taking notes while talking, which can be difficult while conducting research solo.

The interview format was mainly *informal,* without any predetermined questions or questionnaires. It resembled an open discussion based on an initial prompt given to each interviewee, “*What type of value would a system solution overview bring to your work and design process?*” Afterwards, the discussion flowed dynamically as long as it was within the project's scope. Since the UX team are my intended stakeholders for this project and its intended outcome, I needed to learn their perspectives and values regarding how they are affected by the current digital landscape and how a mapping would impact them. Beyond learning the *Why?* of each user, I also wanted to know the *What?* regarding the mapping. When and where would they see themselves utilising a mapping tool, in what context, and why? What type of data would be relevant for them to see? By asking broad questions, I could situate and contextualise the project within the greater scope of the digital landscape at IKEA and collect valuable insights regarding the future implementation of a mapping prototype.

### Drawbacks

Through these informal interviews, valuable insights were gained. However, as previously mentioned, the way the interviews were conducted and how the data was processed could have been improved to save time and yield better results. Better results, in this case, would entail discussions that stayed within the project scope and were more structured than they were.

Although recording each session allowed for more engaging discussions, the following post-processing required considerable time. When listening back at 1.5-2 times speed, most interviews still needed an hour of busy work for taking notes, correcting, and summarising. To save time in the post-processing stage, a more semi-structured interview with more precise goals and questions could have provided more efficient (shorter) interviews, thus minimising time spent after the fact. Another benefit of structuring the interviews further is the data yield from each, which could be more in line with what is expected from the project outcome.

However, given the stakeholders within this project, semi-structured interviews are more beneficial than the traditional interview question format for this type of design research. Furthermore, responding dynamically and allowing the user more space to reflect upon their thoughts and practices demonstrated greater depth in each answer, which could be lost if you stick to a pre-determined formula. Another argument for urging the users themselves to reflect on the discussion is that I, as the “designer”, cannot make any grounded assumptions about the user's work and values. By following a too-structured method, I stand to fall for my biases and preconceptions, thus disallowing the users to speak on their behalf. Another thought on the informal interview is its sometimes improvisational feel which not all users responded to. The thought processes of some interviewees worked very well with this format and encouraged an open environment with a workshop-like feel where spontaneous whiteboard collaborations arose. On the other hand, some users were less engaged with this type of interview and would perhaps have benefited from a more structured method.

Finally, booking six meetings in one week, especially the first week, was not the most efficient use of that time. As all interviewees were asked the same initial prompt combined with the shallow knowledge of their experiences, most users responded with similar answers. This is okay, as it demonstrates commonalities between the users and a direction for the project. However, this insight was gained after the first three interviews leading the remaining three to be more confirming discussions based on assumptions gained through the first three. By staggering the interviews weekly, more specific questions could be asked as new knowledge was gained. Although initially, this was not practical due to planning and business-related issues. Getting a hold of a user for an interview is more challenging as IKEA operates on a meeting-based culture, thus requiring almost a two-week head start to find times to meet.

### Analysis process

#### Affinity diagramming

Affinity diagramming is a method of data analysis that focuses on externalising insights and helps designers to structure information by categorising among common themes (Martin & Hanington, 2012). Insights and observations from user interviews are written down on individual Post-it notes and scattered on an open surface. The post-its’ are then clustered by commonalities and themes. These themes are not determined beforehand but arise through the process. The themes can then be analysed and used to generate design opportunities and design issues that can be iterated upon.

As previously mentioned, all interviews were recorded through audio for later analysis so the discussions could flow better and be more focused. Taking notes after the fact, together with a recording, allows for greater detail in insights and quotes to be captured, which otherwise could be missed during the discussion. However, note that the insights recorded did not compromise an entire written transcription of each discussion, as that would take too much time and not yield more or different insights. This method of interview analysis provides the researcher with a high degree of flexibility in both the interview session and the analysis but comes with some drawbacks. The main drawback found through this method is the insights and questions learned after the fact, which should have been noticed during the interview that could have led to other valuable discussions. Nonetheless, such questions can be answered in a follow-up meeting with the user, requiring more time, preparation and planning.

Chart

Description automatically generated

Figure 2. Finished layout after applying the affinity diagram method.

After taking notes, analysing, and condensing the raw data to manageable insights through the affinity diagram (Figure 2), seven themes could be identified: *Data & Input*, *Purpose of the Overview*, *Finding Information*, *Issues & Problems*, *Governance*, *Previous Mappings*, and *Practicalities*. Some themes were large enough to be divided further into sub-themes which will be expanded upon later. All insights are not directly associated with or relevant to this project but provide a richer understanding of the context in which the project is situated and will be presented as such. The design opportunities and insights gained from this exercise and learnings from the literature will inform the later sketching, designing, and prototyping stages.

### Results from interviews

In the following section, the resulting insights and learnings from the interviews will be presented as is with some explanation, further analysis will be developed upon in the discussion and through later design activities and prototyping.

#### Data & Input

This, together with *Purpose of the Overview,* is the most extensive theme regarding insights and collected data which is not unexpected as the interviews focused heavily on the themes within. The insights gained from Data & Input are technical and inform the design space accordingly; therefore, they are framed as questions rather than straightforward observations. Later in the *Purpose of the Overview* section, the technical aspects will be contextualised through the stakeholder's needs and thoughts.

**Context & Connectivity**

* How do multiple solutions work together in a process, is it linear, circular, or iterative?
* How does data flow between solutions, and what type of data is it?
* What is the context of the solution, and where does it feed data?
* What tools or processes share data?

**User Information**

* Who are the primary users of the solution, what roles use it, and who is the solution owner?
* What is the primary function of a specific role within the solution, and what is its expected outcome?
* What does the user's workflow look like within the solution process from start to finish?
* What other tools do the users use within the process?

**Solution specifics**

* What is the primary function and purpose of the solution?
* What types of workflows and problems does the solution solve?
* What types of data are handled within the solution?

**Interactions**

* Several users used the analogy of “zooming” in and out when discussing interacting with the mapping.
* The data displayed should not be from a technical perspective, e.g. usage statistics or business goals and costs.
* The system should allow for flexibility in interaction and input to enforce a greater sense of ownership in interaction.
* Animations could augment and enforce certain overview aspects, e.g., data flow between solutions.

#### Purpose of the Overview

The purpose of the mapping is different depending on whom you ask, as differing perspectives have different stakes depending on their work. However, the purpose of the mapping can be divided into two prominent themes, *The larger context* and *The personal context*. In the same sense as the mapping impacts the UX Designers' personal workflow and approach to new projects and discussions, it also affects a larger context as multiple designers interact both with internal and external stakeholders.

**The larger context**

When discussing the digital landscape, mentioning the larger scope of people affected and included in this mapping type would be necessary. A recurring theme amongst the designers was the need for a common understanding of the contexts in which they worked. Through a better understanding of the larger context, some users hypothesised that projects could be defined better and have more focused design processes than they currently have.

* Communication in the UX Team needs to be more cohesive regarding the digital landscape.
* Most UX Designers need a clearer picture of the digital landscape.
* A deeper understanding of the digital landscape would aid in the design of new solutions.
* Multiple differing perspectives between UX Designers and end users lead to a need for more consensus in discussions. “*There should always be a shared consensus between us designers and our users, and if it doesn’t happen all the time, it* ***should*** *happen all the time!*”.

**The personal context**

Beyond interpersonal communication and understanding, several users wished to have a better understanding for their own sake when approaching new projects.

* Gain a deeper understanding of the users.
* Get an understanding of the digital landscape from multiple perspectives.
* Find related solutions to build more effective workflows and draw new connections.
* Concretising projects and determining project directions.
* The overview can be used as a personal “*quality assurance*” to situate oneself within the larger scope of the digital solution landscape.

#### Finding Information

When asked how the designers find information about specific solutions in their work, they all said they prefer to contact and discuss it with someone. In these cases, the reason for preferring direct contact is that the existing digital sources of information regarding each solution are spread out over multiple databases, each displaying different types of data differently. The discrepancy in information availability and accessibility has led to a culture where people prefer “old-fashioned” methods of gaining information, namely booking meetings, sending messages, and e-mails. However, a limiting factor in this emergent system is that one's access to information directly correlates with the size of one's personal network, leading to situations where junior co-workers are unintentionally restricted from relevant data compared to older co-workers with a more extensive network.

* Information access depends on the personal network of each designer.
* Multiple intranet portals with bad search engines and UX lead to them being unused as sources of information.
* Most designers contact other people through e-mails, Microsoft Teams chats, and book meetings to get the information they are after.

#### Issues & Problems

Throughout the interviews, the issues described through the problem statement issued by IKEA become more prominent and concretised. These issues affect the UX Team and their work processes but are also evident throughout the organisation in the touchpoints between teams. There are multiple reasons why these issues exist within IKEA. One is the need for consistent access to solution information within the organisation. Through the mapping design made within this project's scope, some of these issues could be addressed and lessened but would require an organisational effort to fix completely. Beyond difficult-to-find information, there is a need for better communication between teams. Project specifics are not directly available within the UX Team, requiring designers to actively seek out and learn about others' projects at their leisure. This often leads to situations where multiple teams work with significant overlap.

* There is a lack of communication and shared data between teams.
* The lack of consensus and landscape contextualising often leads to overlapping projects.
* Those who discover overlaps are usually senior designers with a more extensive understanding of the digital landscape through work experience.
* Much time is spent on projects researching problems already solved by other solutions.
* Seeing a greater context outside your project space is difficult. “*We tend to work a lot in silos*”.

#### Governance

The leading issue in why previous mappings failed was the unclear picture of *mapping governance*, who owned the mapping? The governance theme affects a scope much larger than defined in this project. However, it is nonetheless critical to the reception and future deployment of mapping solutions, as will be developed later in the section *Previous Mappings*. The main stakeholders of this project are the UX Team, and the prototype is therefore focused on their perspective. However, the future development of a complete mapping application would entail engaging developers and designers outside of the UX Team, which further stresses the question of *governance*. When several teams design, develop and maintain the software, who is the owner?

* Who *owns* the mapping?
* How could you connect UX Designers in different projects?
* How would you present the map to an outer stakeholder?
* How to present multiple perspectives for a common understanding?
* Empower junior designers with their learning and independence.

#### Previous Mappings

Before this project, some attempts had been made at mapping the solution landscape. However, they have failed to survive longer than the projects they were designed for several reasons. One issue was that the mappings were too specific and designed from a single perspective in the project, providing valuable insights. However, it was created only within the project scope, leading to much time spent on research and design that was thrown out afterwards. Another issue was that the design of these mappings often was not created by the UX Team but resulted from an external consultant leading to a situation with no clear mapping owner. Lastly, the largest factor in why no mapping has been maintained is that the input method for each has been primarily through manual inputs requiring many work hours spent to keep the mapping up to date.

* Too specialised and constrained to specific projects.
* Focused on singular perspectives.
* Created for the UX Team but not by the UX Team.
* After the project ended, there was no sense of ownership over the mapping.
* Manual input was too time demanding and led to the mappings not being maintained.

#### Practicalities

As a part of the informal interview format, some discussions led to topics regarding practicalities surrounding the project and subsequent prototypes.

* Users would mainly use the mapping at the start of projects.
* Values the insights gained as a point of departure.
* Prefers to use tools already in their toolbox.

#### In summary

## Sketching & Prototyping

### Prototyping methodology

#### Prototyping as an Interaction Designer

(Buxton, 2007; Houde & Hill, 1997; Lim et al., 2008; Martin & Hanington, 2012)

#### Wizard of Oz

Wizard of Oz (WOZ) prototyping is a methodological tool when a designer mocks up an interaction flow without implementing full functionality, just as the methods namesake from the 1939 musical with the same name. (Dahlbäck et al., 1993) defines the method as “*simulating intelligent interfaces (and sometimes also systems) using so-called Wizard of Oz -studies, i.e. having a person simulate the interface (and system)*”. Applications of this method tend to manifest in a spectrum of prototype fidelities. However, the general end goal is the same, suggesting to the user that the system they interact with is larger than the sum of its parts. However, as time and technology change, so to can methodology. As the implementation of the WOZ in this project is not as strictly defined as (Dahlbäck et al., 1993) said in the ’90s. It could be argued that interactive prototypes made in Figma allude to a larger implemented space than is available within the prototype scope. Therefore, as the prototype designer, I define the limited scope within the prototype with click, hover, and mouse-event states. I offload the “wizarding” to the software and can observe the user interacting from a third-person perspective.

#### Material exploration

### Three-point approach

The prototyping process, in the early stages, implemented a three-point approach using multiple mediums, each with benefits and drawbacks to the prototyping process. The prototyping mediums used were:

1. Sketching designs on paper
2. Visual & semi-interactive prototypes in Figma
3. Interactive prototypes in code

By working iteratively through the different processes, I could quickly sketch concepts and ideas on paper, sort through them to see which best would fit the intended interaction, mock up the designs in Figma and then implement those that were most viable into the live code prototype. This workflow enabled me to work as an ideator, a designer, and a developer while doing design hand-offs between each stage. Furthermore, by working iteratively and filtering the design proposals through his regimen of different mediums, I also played on my strengths as a designer, which is mainly visual, so when the design came to the point of implementation in code it would be the design most fitting for both my skills and the design in general.

The choice of these three mediums over other potential means of ideation and prototyping comes mainly down to their accessibility, availability, and specialised areas of function, as displayed in (Table 2). As the table suggests, the limitations of one medium are covered by another, and they complement each other in the workflow I have chosen to use here.

|  |  |  |  |
| --- | --- | --- | --- |
| *Medium* | *Prototyping speed* | *Visual fidelity* | *Interactivity* |
| Paper | Fast | Low | Limited |
| Figma | Average | High | Semi |
| Code | Slow | Medium | Full |

Table 2. The benefits and drawbacks of the prototyping methods

Each medium was used according to its strengths in comparison to the others. Sketching on paper is by far the fastest of these methods regarding the progress you can make concerning the amount of effort required. By limiting ideation sessions (e.g. 5 minutes of sketching per input), many concepts could be prototyped quickly (Figure 3). Furthermore, by disregarding visual specifics and keeping the sketches low-fidelity, advanced visuals and interactions can be abstracted and represented without a larger time investment than other mediums. However, the limiting factor of paper sketches is the need for more complex interactions, and user flows.

Diagram

Description automatically generated

Figure 3. Sketch prototypes

After enough sketches, the most viable options were picked to be further iterated in Figma. Which sketch to be picked was based on its relevance to the total user experience regarding the problem statement and the user research. Building custom components using Figma’s component library function allowed me to quickly implement a design system that could be iterated upon to create different variations and designs (Figure 4). The benefit of iterating further in higher fidelity using Figma was that certain design flaws hidden in the abstractions of the paper sketches showed themselves. Another benefit of working in a higher fidelity was the change of mindset and focus that came with it. While the paper sketches mainly focused on broader interactions and layout, the higher fidelity in the Figma prototype brought forth designs that focused on the content displayed and how the user would see and interact with it. The layout design and content displayed were based on user reflections from the initial interviews and related visualisation software such as Google Maps (*Google Maps*, n.d.) and Connected Papers (*Connected Papers | Find and Explore Academic Papers*, n.d.). According to the research done by Field (Heer et al., 2007), creating a design reminiscent of software, the users are used to and comfortable with you can leverage their prior experience and create an environment with a lower point entry.



Figure 4. Iterations on the layout and design language

A built-in function in Figma is the ability to mock-up a semi-interactive user flow. Without programming in code it is possible to design clickable user flows. The benefit of making interactive flows in Figma in comparison to an actual code implementation is the speed of implementation. With the design system made with modular components, interactions and flows that would require a more considerable time investment to code can be first prototyped and tested. The interactive prototypes made in Figma provide an insight to the look and feel (Houde & Hill, 1997) of how a fully working implementation would work. However, despite the prototypes *looking* like a working implementation, it is important to note that the prototypes only exist statically within the limited scope in which they were created and are unable to respond dynamically to changing data and more complex interactions. The limitations of the prototyping system show most clearly when prototyping interactions that require multiple states (Figure 5), where even interactions such as filtering between four options result in an exponential number of states that needs manual designing.

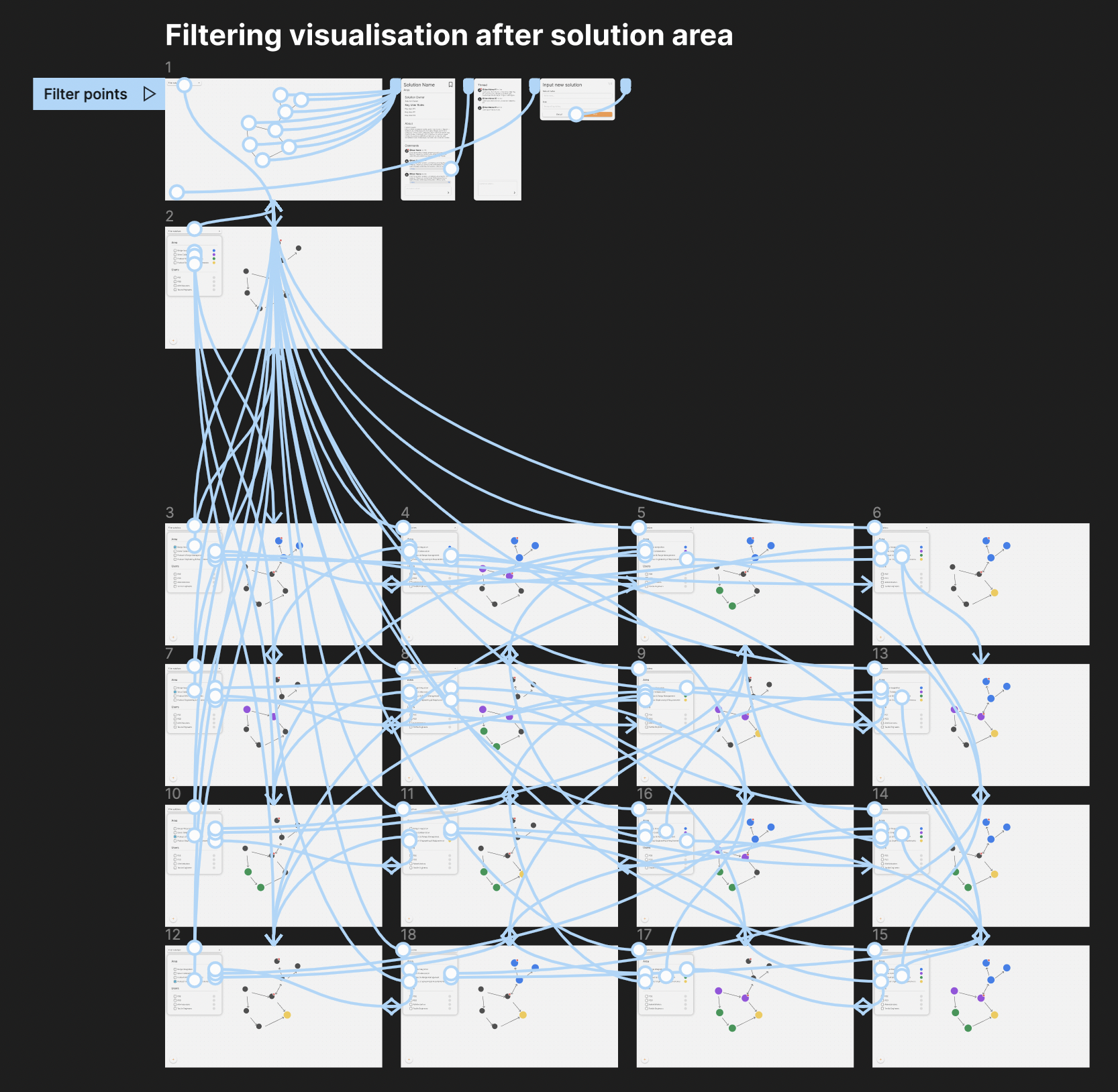


Figure 5. Filtering four options requires many intermediate states.

In that sense, the prototypes made in Figma are *Wizard of Oz* (Dahlbäck et al., 1993) prototypes; abstractions and visualisations of more complex structures than what is used to power them. Therefore, to prototype more complex interactions that uses dynamic input and output where there is an action-reaction from the prototype it is necessary to implement in code.

The purpose of making prototypes in code is to complement the limitations of the Figma prototypes. However, as previously mentioned, programming takes more time in comparison to the other, more visual, prototyping methods and the focus is more on *implementation* rather than *look and feel (Houde & Hill, 1997)*. The implementation prototypes build upon the *force-graph* library created by Vasco Asturiano (Asturiano, 2018/2023), a JavaScript library built upon the visualisation library D3.js (Bostock, n.d.) and allows users to implement dynamic *force-directed graphs*. Besides using JavaScript, the prototype is built upon a base HTML website with CSS styling.

The early prototyping stages when doing an implementation prototype consist largely of material exploration and technical exercises to probe both the viability of the method for the developer and how inline the method is with the expected outcome. By exploring and iterating on simple concepts separately, such as reading data from JSON, dynamically updating nodes with JSON data, filtering data points, and implementing user input interactions, I could later consolidate learnings and the built-up code base to construct a more comprehensive user flow (Figure 6).



Figure 6. Early functional prototype for first user tests

The data used in the public prototype (Ingelsten, n.d.)(Figure 6) is comprised of made-up solution names, solution owners, and roles, as the information provided by IKEA is under confidentiality law. However, even though the data is a placeholder, the interactions and overall user experiences are the same.

For the coming user tests, there are two major prototypes to be tested, the Figma prototype and the implementation prototype. Both approach the design from different perspectives. The Figma prototype is focused on visual aesthetics and the interactions therein, and it aims to promote discussions with the users about the different representations of the data set. While taking visual cues from the Figma design, the implementation prototype's focus is to show the user the interaction interplay between the data interface, data, and user through which a discussion of interaction aesthetics could be held. By separating the prototype into two more specialised prototypes, I aim to have more focused and directed discussions, taking a lesson from the informal interviews in the research stage.

## User testing

### Paper about user testing?

* Har någon på Zotero

### Planning and user test process

User testing was conducted in two sessions with four of the users that were in the initial interviews. However, additional testing was done more informally through other in-person meetings and e-mail communication. The testing sessions were staggered with a few days between. This was partly due to the practicalities of finding time with the users and allowing iteration between sessions. After exclaiming an interest in trying the prototype, additional testing opportunities arose spontaneously through regular discussions where a user tested and provided feedback. Similarly, the test done over e-mail was due to one of the initial users falling ill before the meeting.

The process of conducting the user test with both the implementation and visual prototype was similar in structure to the informal interviews held during the early research study. In short, the tests were conducted as follows:

1. Reiterate for the user the purpose of the design regarding the initial problem statement and what we discussed in our last meeting.
2. Explain the purpose and difference of the implementation prototype compared to the visual.
3. Guide the user through the prototypes while having an open design discussion regarding their moment-to-moment thoughts about the interactions, the visualisation, and the data displayed.
4. Lastly, a discussion with the user where they could go back through the prototypes and point to aspects that could be changed or improved.

After the user test concluded, I compared and showed the notes I took to ensure I had not missed or misinterpreted the user’s input. The process for the spontaneous user tests followed a similar process, although shortened due to their specific circumstances.

Due to organisational secrecy, the data used in the prototypes is completely fake and fabricated. Testing with internal users, I was allowed to use real data. However, since most prototype testing was conducted through a publicly accessible website, the data had to be sanitised to *resemble* the original data set rather than *represent* it. To counteract possible user input regarding this change, the data was formatted in a way that seemed like real solutions at a cursory glance. This was done by imitating the naming schema commonly used in IKEA products, e.g., a capitalised Swedish word, often a location or noun, for example: “*MÖCKELN*”, “*INSEXNYCKEL*”, or “*AGUNNARYD*”.

Since the main user group on this project consists of UX Designers, the user testing method relied on the user’s expertise in design. By pivoting the tests to allow the user to freely discuss the prototype from their perspective as a UX Designer, I could get feedback that improved practical aspects such as visuals and interaction and aspects of the design that regarded user flow and accessibility. The user’s familiarity with design methodologies common to both UX and Interaction Design promoted an environment where, instead of the usual designer/user dynamic, the users and I could approach the prototypes as design equals. Although this project is not strictly intended to be about or aims to be about co-design, implementing co-design methodologies and principles can yield results that would otherwise be missed if you held a more traditional view of the user/designer relationship.

Lastly, in the following section, where I discuss the prototype iterations brought forth during user testing, one main aspect should be noted: how the user testing process unfolded and how it intertwined with the sketching process that happened simultaneously. Each user testing session had one or two main prototypes that were the intended subjects for testing. However, due to the reasons stated above about the users also being designers, after testing the finished prototypes, often discussions lead to impromptu displays of unfinished sketches to demonstrate aspects of the design I was working on but was not ready to call “finished”. A result of this is an iterative workflow which is more “back and forth” between prototype versions rather than staged, as is usually common in user testing. Several prototype iterations share multiple user testing groups with their feedback and input. Therefore, the following section will focus on each distinctive *generation* of prototypes rather than user testing sessions.

### Iterating on the design

#### Implementation Gen 1 & 2 (1.5)

I presented *Implementation Gen 1 and Gen 2 (1.5)* (Figure 7 below) to the assigned supervisor at IKEA, the supervisor from MAU, fellow Interaction Design bachelors, and later two of the UX Designer users at IKEA. Gen 1 & 2 (1.5) are bundled together as they were tested simultaneously due to reasons developed later. The version of Gen 2 tested in this first round was an early implementation and was iterated further for the second round of user testing.

Two major themes arose from this first testing phase, visual and interactive. Testers found the overall visual design of the prototype to be unappealing to the point of distracting from the intended interactions. The lack of labelling on each solution made finding specific solutions more difficult. Similarly, in this initial iteration, users could neither see the direction of the data flow nor what type of data. This was a highly requested addition from end users.

The filtering interactions made possible through the checkboxes at the top of the screen were naively implemented. They resulted in a jarring change in the visualisation, further confusing the testers (Figure 8 below). One tester also requested a more in-depth filtering system with tags connected to each solution. Users also requested a clearer way of visualising what filters were currently active in the visualisation. A part missing from these prototypes was the implementation of data flow and content, which was one of the requested functions from the user research. Users reiterated this through testing as well.

Overall, end users enjoyed the prototype conceptually, although not visually, and could see how it could be implemented into their workflow. One user said regarding the difference of this type of visualisation compared to the Excel sheets they are accustomed to: “*At the moment, the digital solution landscape is quite abstract for me, and I don’t get how any of it fits together. This really makes it concrete.*”.

Some of the issues brought up by users through testing were not, for example, the visual aesthetics, completely unsurprising, which prompted a visual redesign borrowing inspiration from the technical sketches developed “between” prototype iterations. Note that the main difference between Gen 1 and Gen 2 is visual, with similar filtering interactions. Therefore, when approaching the first round of user tests with end users, I was prepared to constructively meet the users’ feedback with sketches prepared in advance. This allowed the user and I to discuss the prototype with the sketch as a “canvas” for ideation.

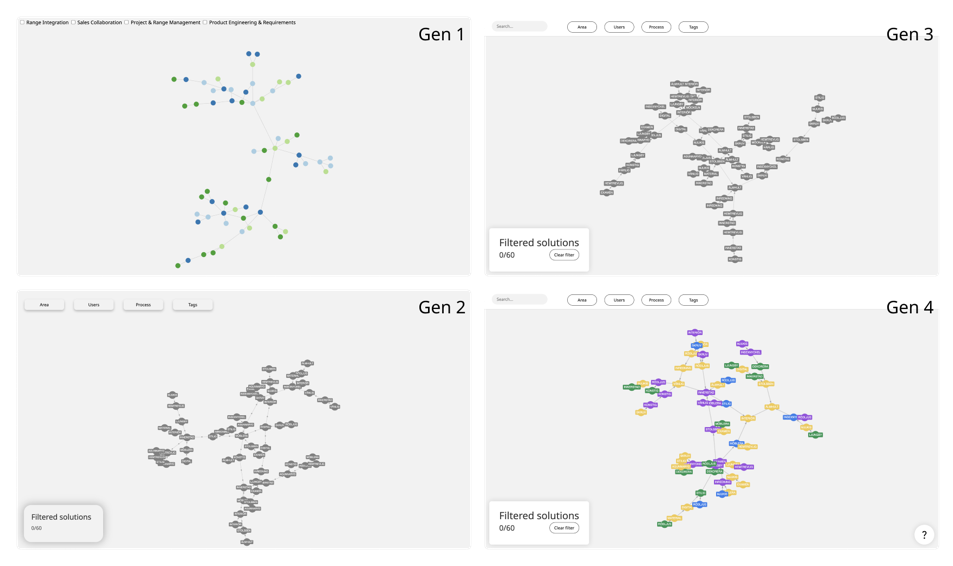


Figure 7. Implementation prototype iterations 1-4.

#### Implementation Gen 2 & 3

After the first round of user testing, further technical and visual iterations were done to arrive at the final iteration of *Gen 2*. The user feedback in this session still involved some aesthetic discussions but focused more on interactions and user flow.

Visually, users liked the higher fidelity aesthetics of Gen 2 compared to Gen 1. However, the colours chosen (white, grey, red, and orange) were found distracting as they did not follow IKEA’s design system *SKAPA* as the users expected them to. Similarly, from the point of accessibility, the colours did not allow for easy reading due to differing contrasts. As a further effect of the visual design having poor contrast, high usage of box shadows, inconsistent text sizing, and use of white space, users found the prototype to have much visual noise, distracting from the intended user flow.

To address earlier feedback about data flow, Gen 2 implemented a particle system that visualised the data flow between solutions along the lines connecting each node. However, though implicitly understood by the users on a micro scale between individual solutions, the particles on a zoomed-out macro scale more resembled thousands of ants and did not convey the data flow in any practical sense. This implementation showed users *how* and *where* data flowed but missed the crucial aspect of *what* data flows between solutions. This was not part of the data set provided and had to be mocked up in the following iteration.

As each prototype got iterated upon and became more detailed, so did the feedback. Some user feedback regarded bugs in the prototype, which was a limitation of my proficiency in programming. However, at this level of fidelity, discussions regarding the interactions no longer lingered on basics or bugs but looked forward to possible future implementations of new interactions and data connections. Such future implementation. Requests from users included the ability to specify their own processes and tags, a way to clear filters, clearer user flows for filtering, a search function, a way to edit/ add solutions, the ability to “favourite” solutions, and other data-specific requests.

* Fattas det något här?

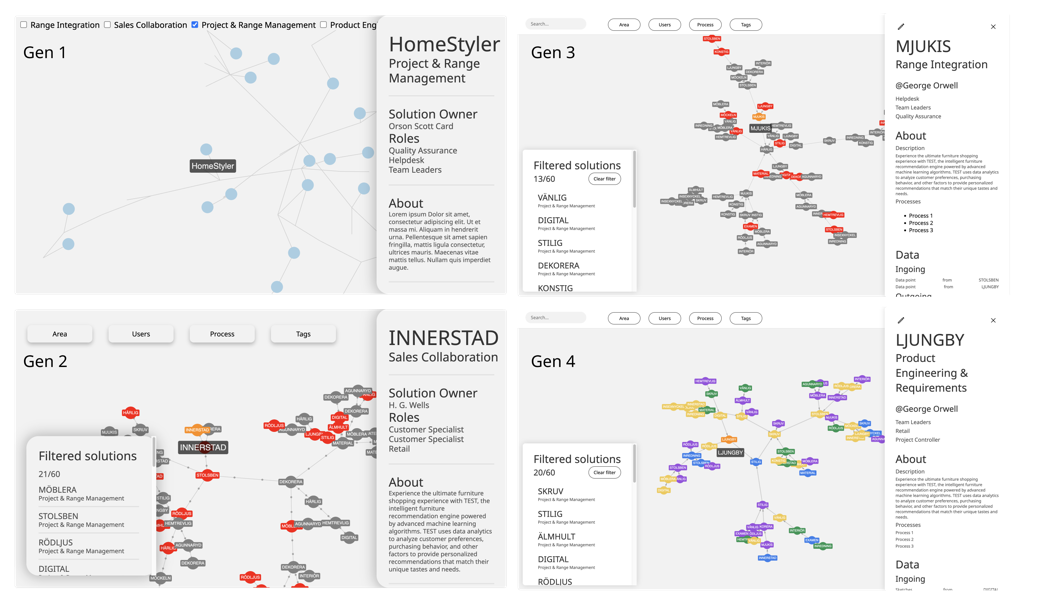


Figure 8. Filter and detail states of each prototype iteration.

#### Implementation Gen 4

Consolidating feedback from all user tests that had not been implemented and organising them according to implementation viability, a final prototype generation could be iterated to be presented as the final proof of concept to stakeholders. A more detailed account of the final prototype will be in the discussion.

#### Figma prototypes

As previously discussed, in addition to the implementation prototypes, several visual concepts were designed in Figma to show impractical concepts to implement through code (Figure 9). These were shown and discussed simultaneously with the implementation prototypes, and the results are intertwined with those of the implementation prototypes. In general (Figure 9, pictures 1-5):

Diagram

Description automatically generated

Figure 9. Visual Figma prototypes. (1) Area’s overlapping selection, (2) Process selection, (3) User overlap, (4) Manual editing of solutions, (5) Solution changes over time, future/past

1. User conceptually liked the idea of visualising area overlap with a literal-coloured area. However, in the way implemented (Figure 9, picture 1) the colour semantics did not equate to user expectations and made the overall visual impression “muddy”. Furthermore, as some users pointed out, this example shows a small, best-case scenario and would not likely scale well with the more chaotic structure of the actual solution landscape.
2. End users most requested this prototype. Seldomly does one UX Designer work with the entire solution landscape. More commonly, each designer works with 2-5 solutions on a project-by-project basis, meaning having all solutions displayed at once is unnecessary. Users specified a wish to be able to specify their own processes on top of existing processes at the workplace. However, removing all other solutions besides those specified in the process filter was not a good alternative; the other solutions remained, albeit toned down.
3. A common workflow need for designers when approaching new solutions is understanding what users are active there and what they do. Users understood the distinctions and overlapping users in the example, and several stated how this would allow them to understand the solution user’s workflow better.
4. Since most users find information through interpersonal connections instead of using digital databases, having the option to edit and add solution data manually would empower designers to utilise the wealth of verbal information that is present. This sentiment of enabling and working with the existing and preferred data collection method instead of trying to fix it resonated with users as they felt that having the option to edit and add would empower and strengthen their autonomy in the solution landscape.
5. Finally, one part of the visualisation’s utility is to see the solution landscape as it is currently. However, when planning and designing for new solutions, it is valuable to know how the surrounding landscape changes over time. Therefore, this visual prototype was iterated with user feedback to allow the designer to “peek” into the future of the solution landscape to see how it changes as solutions get replaced, new solutions get added, and solutions get removed.

# Discussion

## The prototype

* **The prototype**
  + Quick run-through of the final proof of concept
  + Through the lens of research in the lit. review.
  + User testing
    - Meeting User feedback
    - Benefits/drawbacks of testing with different users each session
    - Invited users to dream bigger and ideate freely about future possibilities and solutions.
      * The prototype was never the end-all-be-all and was not interpreted as such 🡪 It is a proof of concept that has opened up a new design space at the organisation.
* **Working with a company/outer stakeholder**
  + Access to users
    - Very clear user base and needs.
    - You need to jump through hoops to get in touch.
    - Fitting in to a new organization culture.
    - Working with that culture and finding solutions around (eg. Three week process to set up accesses and emails that still don’t work)
    - Working with UX Designers as stakeholders
      * A different approach compared to other types of users
  + Using real data vs fake
    - Users wished for real data in prototypes.
    - Got access to incomplete sets of information.
    - Can’t share data publicly, company secrets 🡪 user testing gets really difficult.
    - Mockup similar fake data as a compromise
    - Requires careful explaining when doing tests
  + Balancing the stakeholders needs with the thesis needs
    - Creating an end-product that produces value for the stakeholder while remaining true to the thesis.
    - Clear communication and compromises with the stakeholder 🡪 thesis needs to come first
      * Discussions about what constitutes a satisfactory result for both parts.
      * How does it impact the dynamic between designer and stakeholder in a user centric design process when the designer has an intrinsic stake in the project
* **What have I learned about visualization?**
  + The actual visualization is not as important as the data around it.
    - Working with data, filtering, interacting, and researching users’ needs provided the valuable foundation the visualization is bult upon. The visualization is a means to an end.
  + It can change the way people see their organization and their place within it.
* **Pivoting to coding an implementation prototype**
  + Initial idea was to keep it strictly visual –> what changed?
  + How can coding a prototype add to my and the end -user experience?
* **What are the ethical aspects?**
  + Transparency within the company
  + Democratic empowerment to people working at the company
  + Who is accountable? If you don’t see other solutions/aren’t aware of better paths then who is responsible? Usually accountability falls on the unknowing worker.
  + New workers are overwhelmed when orgs get to large 🡪 one solution can be categorization through VIS
  + Meta discussion about ethics
    - How can Interaction Design projects/practices make workplaces more accessible?
    - Using design methodology for onboarding, empowering, and belonging.
* **Governance**
  + A real issue in large companies
  + At a certain size the company act more as an organism 🡪 not one clear driving force but several that all build a whole
  + “Accountability” when tools change and people move around -> You can’t always foresee changes but how can you maintain some relevant governance in digital solutions
* **Other thoughts**
  + Where else can the prototype be used
    - Inside IKEA / Outside use cases.
    - Any type of data that is serializable into JSON could work.
    - When does a solution become too generic?
    - Is there a limit to when being generic is bad, because that can also be a strength.

# Conclusion

# References

Abras, C., Maloney-Krichmar, D., & Preece, J. (2004). User-Centered Design. *Encyclopedia of Human-Computer Interaction*, 14.

Andrienko, G., Andrienko, N., Drucker, S. M., Fekete, J.-D., Fisher, D., Idreos, S., Kraska, T., Li, G., Ma, K.-L., Mackinlay, J. D., Oulasvirta, A., Schreck, T., Schmann, H., Stonebraker, M., Auber, D., Bikakis, N., Chrysanthis, P. K., Papastefanatos, G., & Sharaf, M. A. (2020, March 30). *Big Data Visualization and Analytics: Future Research Challenges and Emerging Applications*. BigVis 2020: Big Data Visual Exploration and Analytics. https://hal.inria.fr/hal-02568845

Asturiano, V. (2023). *Force-graph* [JavaScript]. https://github.com/vasturiano/force-graph (Original work published 2018)

Bertin, J., & Barbut, M. (1968). *Sémiologie graphique: Les diagrammes, les réseaux, les cartes*. La Haye, Mouton.

Bostock, M. (n.d.). *D3.js—Data-Driven Documents*. Retrieved 12 April 2023, from https://d3js.org/

Brinkmann, S., Jacobsen, M. H., & Kristiansen, S. (n.d.). *Historical Overview of Qualitative Research in the Social Sciences*.

Buxton, W. (2007). *Sketching user experiences: Getting the design right and the right design*. Elsevier/Morgan Kaufmann.

*Connected Papers | Find and explore academic papers*. (n.d.). Retrieved 11 April 2023, from https://www.connectedpapers.com/

Dahlbäck, N., Jönsson, A., & Ahrenberg, L. (1993). Wizard of Oz studies: Why and how. *Proceedings of the 1st International Conference on Intelligent User Interfaces - IUI ’93*, 193–200. https://doi.org/10.1145/169891.169968

Design Council. (2007). *Eleven lessons: Managing design in eleven global brands. A study of the design process*. Design Council UK. https://www.designcouncil.org.uk/our-work/skills-learning/resources/11-lessons-managing-design-global-brands/

Design Council. (2019, May 17). *Framework for Innovation: Design Council’s evolved Double Diamond*. https://www.designcouncil.org.uk/our-work/skills-learning/tools-frameworks/framework-for-innovation-design-councils-evolved-double-diamond/

Dimara, E., & Perin, C. (2020). What is Interaction for Data Visualization? *IEEE Transactions on Visualization and Computer Graphics*, *26*(1), 119–129. https://doi.org/10.1109/TVCG.2019.2934283

Fossey, E., Harvey, C., Mcdermott, F., & Davidson, L. (2002). *Understanding and Evaluating Qualitative Research*.

*Google Maps*. (n.d.). Google Maps. Retrieved 11 April 2023, from https://www.google.com/maps/@55.5880384,13.0322337,14z

Gustafsson, D. (2019). *Analysing the Double diamond design process through research & implementation* [Master of Arts, Aalto University]. https://aaltodoc.aalto.fi/bitstream/handle/123456789/39285/master\_Gustafsson\_Daniel\_2019.pdf?sequence=1&isAllowed=y

Heer, J., Viégas, F. B., & Wattenberg, M. (2007). Voyagers and voyeurs: Supporting asynchronous collaborative information visualization. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 1029–1038. https://doi.org/10.1145/1240624.1240781

Hornbæk, K., & Oulasvirta, A. (2017). What Is Interaction? *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, 5040–5052. https://doi.org/10.1145/3025453.3025765

Houde, S., & Hill, C. (1997). What do Prototypes Prototype? In *Handbook of Human-Computer Interaction* (pp. 367–381). Elsevier. https://doi.org/10.1016/B978-044481862-1.50082-0

Huron, S., Carpendale, S., Thudt, A., Tang, A., & Mauerer, M. (2014). Constructive visualization. *Proceedings of the 2014 Conference on Designing Interactive Systems*, 433–442. https://doi.org/10.1145/2598510.2598566

Ingelsten, O. (n.d.). *IxD Thesis 2023*. Retrieved 12 April 2023, from http://www.inglestone.art/

Knox, S., & Burkard, A. W. (2009). Qualitative research interviews. *Psychotherapy Research*, *19*(4–5), 566–575. https://doi.org/10.1080/10503300802702105

Lim, Y.-K., Stolterman, E., & Tenenberg, J. (2008). The anatomy of prototypes: Prototypes as filters, prototypes as manifestations of design ideas. *ACM Transactions on Computer-Human Interaction*, *15*(2), 1–27. https://doi.org/10.1145/1375761.1375762

Mao, J.-Y., Vredenburg, K., Smith, P. W., & Carey, T. (2005). The state of user-centered design practice. *Communications of the ACM*, *48*(3), 105–109. https://doi.org/10.1145/1047671.1047677

Martin, B., & Hanington, B. (2012). Universal methods of design: 100 ways to research complex problems, develop innovative ideas, and design effective solutions. *Choice Reviews Online*, *49*(10), 49-5403-49–5403. https://doi.org/10.5860/CHOICE.49-5403

Norman, D. A., & Draper, S. W. (Eds.). (1986). *User centered system design: New perspectives on human-computer interaction*. L. Erlbaum Associates.

Ralph, P., & Baltes, S. (2022). Paving the way for mature secondary research: The seven types of literature review. *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 1632–1636. https://doi.org/10.1145/3540250.3560877

Thorne, S. (2000). Data analysis in qualitative research. *Evidence-Based Nursing*, *3*(3), 68–70. https://doi.org/10.1136/ebn.3.3.68

Wood, L. E. (1997). Semi-structured interviewing for user-centered design. *Interactions*, *4*(2), 48–61. https://doi.org/10.1145/245129.245134