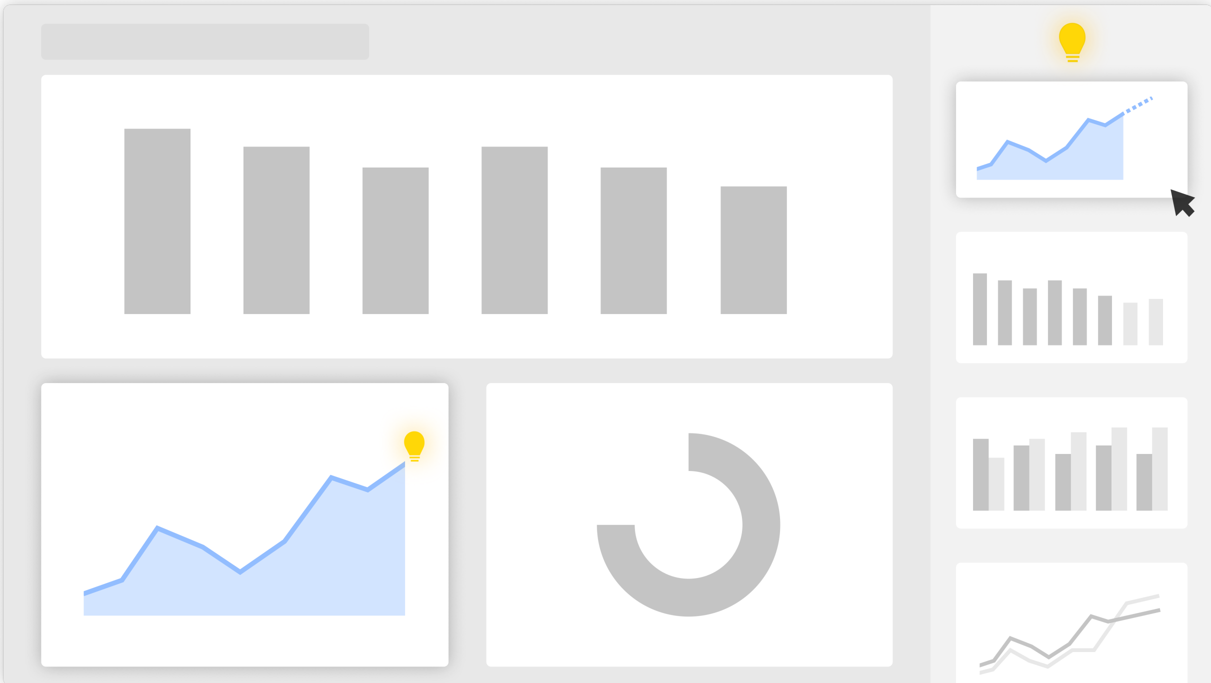
CHALMERS UNIVERSITY OF TECHNOLOGY | UNIVERSITY OF GOTHENBURG



# AIm$^2$: Exploring AI-Driven Insights in Visual Analytics

## Guidelines for Designing Insights to Aid in the Analytic Process

Master's thesis in Interaction Design and Technologies

MIRANDA ALDRIN MIRANDA BÅNNSGÅRD

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2020

Master's thesis 2020

# AIm$^2$: Exploring AI-Driven Insights in Visual Analytics

Guidelines for Designing Insights to Aid in the Analytic Process

MIRANDA ALDRIN
MIRANDA BÅNNSGÅRD

UNIVERSITY OF GOTHENBURG

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering
Chalmers University of Technology
University of Gothenburg
Gothenburg, Sweden 2020

AIm$^2$: Exploring AI-Driven Insights in Visual Analytics
Guidelines for Designing Insights to Aid in the Analytic Process
MIRANDA ALDRIN
MIRANDA BÅNNSGÅRD



Supervisor: Staffan Björk, Department of Computer Science and Engineering
Advisor: Maria Redström
Examiner: Olof Torgersson, Department of Computer Science and Engineering

Master's Thesis 2020
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: An illustration of a dashboard with available insights (illustration by the authors).

Typeset in LaTeX
Gothenburg, Sweden 2020

AIm$^2$: Exploring AI-Driven Insights in Visual Analytics
Guidelines for Designing Insights to Aid in the Analytic Process
MIRANDA ALDRIN
MIRANDA BÅNNSGÅRD
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg

## Abstract

The purpose of this thesis project was to explore the application of system-generated insight and research what needs to be considered when designing such systems. The project was exploratory in nature, and the process was designed to explore many different concepts and design challenges within the field. The scope of the project was very broad, as few other academic studies have looked into designing these types of systems. The broad scope allowed for a general exploration of the design space to find what design challenges there are. The thesis resulted in three artifacts: a high-fidelity prototype, a design language used for communicating the design, and a set of guidelines. The guidelines provide recommendations in order to aid the design of system-generated insights and the interaction with them, and cover the design process, general boundaries to be considered, as well as more insight specific guidelines. There are still many possibilities for further study within the field, but the hope is that the foundation built in this thesis will work as a stepping stone for future research.

Keywords: visual analytics, system-generated insights, AI-generated insights, human-AI interaction, information visualization, interaction design.

## Acknowledgements

We want to thank our supervisor Staffan, who has put up with our endless confusion and questions. It has been a real help to have you as our supervisor.

We would also like to thank the Company for all of the support we have received throughout the thesis. There have been many deign critiques and discussion regarding the subject, which have helped in the ideation process and throughout the rest of the project. We especially want to thank our mentor Maria, who has put up with many question over the course of the project and has patiently provided us with answers to all of them.

Lastly, we would like to thank the participants in our usability tests. Your input into the project have been detrimental to the success of the project.

Miranda Aldrin and Miranda Bånnsgård, Gothenburg, May 2020

# Contents

# List of Figures

# 1

# Introduction

With companies collecting more and more data, the tools for doing visual analytics are becoming more and more important. Visual analytics is the "science of analytical reasoning supported by interactive visual interfaces" [1], and there are multiple software one can use to do this. The idea is to easily be able to extract the relevant information from data by creating visualizations to aid in the analysis. The purpose of visual analytics is to enable and discover insights [2] made by the analyst, either through spontaneous "aha" moments or by discovering new relationships in a data set [3]. While the visualizations have been created using software, the creation of them is very user-driven. With the advances in artificial intelligence (AI) and machine learning, however, it is possible to use those techniques to make the analytical process more effective. Analytics software companies are starting to implement these advanced features into their tools by providing AI-driven insights to aid in the discovery of new insights by analysts. The insights can, for example, help the user find comparisons, correlations, outliers, and forecasts in a data set, which would otherwise easily have been missed. These insights can bring forward overlooked patterns, new perspectives, and help speed up the workflow.

Currently, many companies have implemented recommendation systems, and some have also implemented AI-driven insights. For example, Power BI [4], Tableau [5], and Qlik [6] all some sort of insight function. It is not only companies that are looking into this topic. Since it is a relatively new practice, there are no developed standards or design guidelines to follow. The features which already exist are not coherent and disrupt the user's flow when analyzing a visualization. Prototypes, such as Foresight [7] and Voyager [8], have also been produced to research the topic. However, they have focused on the implementation of algorithms behind the insights, rather than the interaction design surrounding them. Some of the more relevant research for this thesis is the classification of visualization recommendation systems (VisRec) by Vartak [9]. However, VisRec does not specifically look into AI-generated insights. Therefore, the thesis aims to broaden the research within AI-driven insights for visual analytics by providing guidelines and future recommendations within the field. The thesis will be done in collaboration with a data visualization and analytics software company. The company will be referred to as the Company, and their software will be called the Product from hereon out.

The thesis will explore the field of visual analytics as well as agentive technology throughout the project. Agentive technology is a relatively new term, coined by Christopher Noessel, when talking about narrow artificial intelligence, such as AI-driven insights in analytics software [10]. It is not assistive technology, which helps the user perform a task [10], but rather a background helper that does things for the user without the need for much input. Assistive technology is there to provide an automatic experience where the user takes action. In contrast, agentive technology takes more control and does the action for the user without the need for approval. The scale of user-driven, assistive, and agentive technology will be referred to as the level of automation. The lack of research about the level of automation for system-generated insights allows for exciting exploration in the thesis. However, with the emergence of more automation in a system, how do you design for transparency between the user and the system? Being transparent with the changes made by the system and if the AI has created any data is fundamental.

The project aims to produce guidelines for presenting AI-generated insights for visual analytics in order to retain the workflow of the analyst. The guidelines will be worked on iteratively, and design concepts and prototypes will be created in order to evaluate the guidelines. A final prototype will be created to showcase a possible implementation of the finalized guidelines. Both the final prototype and the guidelines will be the result of this thesis.

## 1.1 Research Question

There are visualization tools on the market that have some form of implementation of AI-powered insights, but it is still a relatively new concept, and there are no standards for it yet. In this thesis, ways of presenting AI-powered insights will be explored, making sure the insights are presented in a clear and inviting way that is not intrusive and interrupts the workflow of the user, working towards an enhanced user experience and improved outcome of the analytic process.

The thesis will aim to answer the following research question: *What should be considered when applying system-generated insights within visualizations in a visual analytics tool?*

## 1.2 Scope and Delimitations

Since the research question is quite broad, the scope of the project has been narrowed down by using a specific target group and by designing insights for certain types of graphs.

The Company has provided us with a target user group, defined by an analyst persona and a consumer persona. A persona called Serena, provided by the Company, is a useful target group as AI-generated insights fit well into the context of analysts doing ad-hoc analysis. Serena is the strategic analyst at a big company and often works with ad-hoc analysis, trying to find valuable knowledge within a data set. Ethan, the consumer, on the other hand, mostly receives dashboards and uses them to make decisions. He does not create the dashboards from scratch, so the scope includes both creating and using visualizations. Thus, the design will use these personas as its target user group, and will look at how AI-generated insights could be designed within this specific context.

Because of the collaboration with the Company, the project is also affected by the existing Product. While the thesis will try to look at system-generated insights for visual analytics in general, the final prototype and the concepts produced will be created with the Product in mind. This will set certain limitations on the design.

The generated insights depend on the context of the visualization and the question asked by the user. Because of this, the design will also mainly focus on AI-generated insights for three specific types of graphs: bar chart, scatter plot, and line chart. This way, it is possible to explore the design of insights within a smaller context, and the context can be looked at more in-depth.

## 1.3 Ethical issues

When designing agents, it is necessary to think about the ethical implications of the design and the results of the agent's tasks [10]. As a designer, one needs to think about the implications of an agent's actions and what damage they can do.

Data analysis is about asking questions and seeking an answer to those questions by exploring a data set [11]. The user doing the data analysis will be trying to gain knowledge from the data set, and to do so, they could use AI-generated insights, for example. However, these insights will be based on a trained agent, and a trained agent can be biased. The motive behind designing an agent will affect the outcome and the overall experience of it [12]. Even if the agent is not intentionally biased, it is difficult to see this bias before deploying it. While the agent may be less biased than the human counterpart [13], it still needs to be considered when designing agentive technology. A solution for these types of biases will not be created in this thesis. However, it is imperative to keep this in mind when implementing a whole system for generating insights and using it within the software.

Even if there are no biases in the agent, the generated insights could accidentally mislead the user if there are biases within the data set the user is analyzing. If people are going to be making big decisions based on these insights, it is crucial to think about the damage that can be made in cases such as these. Insights based on partial data can lead to biased decisions being made without the user knowing about it. If possible, designing for transparency regarding such biases would be one way of tackling the problem. While technically challenging, a calculation of bias could be one of the transparency measures needed in the system designed in this thesis, for example.

Since the goal is to design a concept for AI-generated insights into a data set or a visualization, it is also important to think about the ethical implications of auto-generated information. The design needs to be transparent about what insights are generated and what is taken straight from the data. An example of this could be if data is removed to make the visualization more accurate. The design also needs to be transparent with what data is removed and that the visualization is automatically generated rather than taken straight from the data set.

# 2

# Background

In this section, related work in different analytics software and tools are presented, with a focus on their implementation of AI-generated insights. The intended target user group of the project is also described.

## 2.1 Related Work

AI-generated insights are used throughout many types of business intelligence applications, and the application of insights can look very different depending on the tool. Different tools, including the Product, have been analyzed, and their uses of AI-generated insights are presented below.

### 2.1.1 Power BI Quick Insights

Power BI is a business intelligence tool developed by Microsoft. It can be used either as a desktop client or as a cloud service [4], with an AI-generated insights functionality called "Quick Insights." The functionality of the two applications differ, and the generated insights have different uses depending on the platform. Both applications have been analyzed for this thesis.

In the desktop application, Power BI can generate insights based on a specific data point, and point out possible causes for an increase or decrease in the data [14]. It can also point out correlations between different factors. When right-clicking after hovering over a specific data point, an "Analyse" option appears, with context-specific analysis, such as "Explain the increase" if there has been an increase, or "Find where the distribution is different" for bar charts. A new visualization is shown after the insights have been generated, providing statistical information about why the information is shown.

The cloud services version of Power BI differs quite a lot from the desktop counterpart. It allows the user to get more general insights on the whole data set or on the data that is displayed in a visualization, but not from a specific data point.

The Quick Insights produced from the whole data set results in a list of cards with

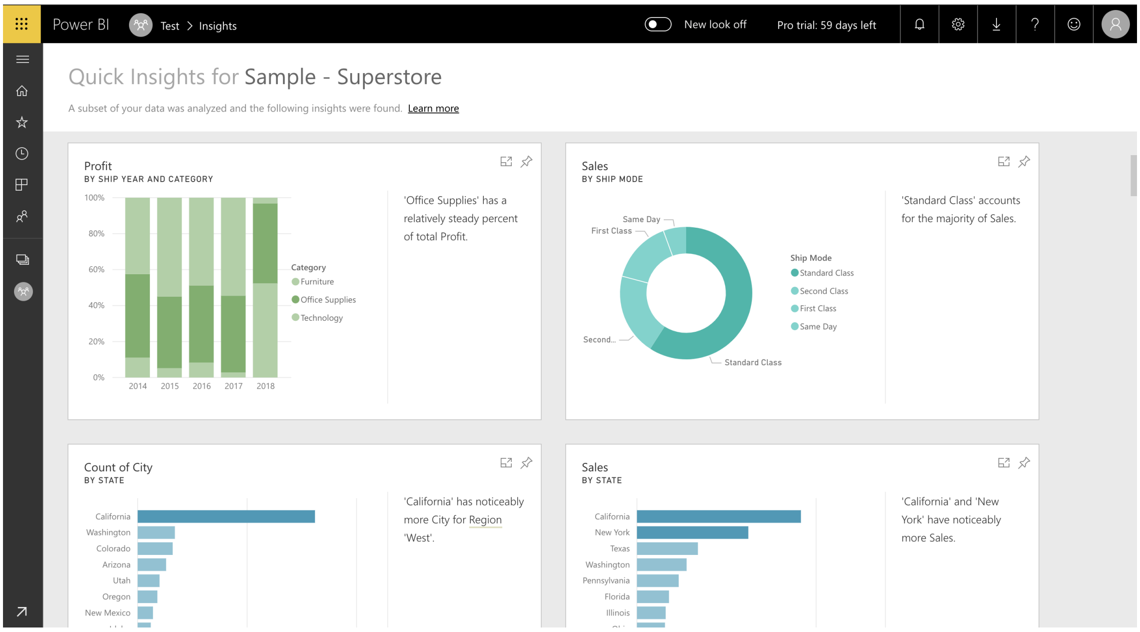



**Figure 2.1:** A screenshot of Power BI showing a list of generated insights [4]

suggested interesting graphs to use as a base for an analysis [15]. Each card shows a visualization and an explanation for why the user sees the graph, see Figure 2.1. The user can then choose which visualization to start from and can enter a focused view to explore the visualization more in-depth, or add it to a dashboard.

From a focused view of a visualization, it is possible to produce more specific Quick Insights from the data presented in the focused visualization. These insights could include things such as outliers, significant factors, or overall trends [16].

### 2.1.2 Tableau Explain Data

In the BI tool Tableau [17], it is possible to "get the why behind a data point" through AI-generated insights called Explain Data [5]. Explain Data's use is narrow, and data point focused, quite similar to the Quick Insights in the desktop version of Power BI. The user selects a data point that is of interest. Then AI-driven explanations are generated together with visualizations supporting those explanations, to give context for the insight, see Figure 2.2. The generated explanation and visualization can be opened and further examined in a new worksheet if the user chooses to do so.

### 2.1.3 Qlik Insight Advisor

While Tableau is similar to the desktop version of Power BI, the Qlik Insights Advisor in the Qlik Sense software is more similar to the cloud service of Power BI [6]. It is possible to get general insights over the whole data set, and there are also associative insights specific to visualizations viewed in a focused mode.

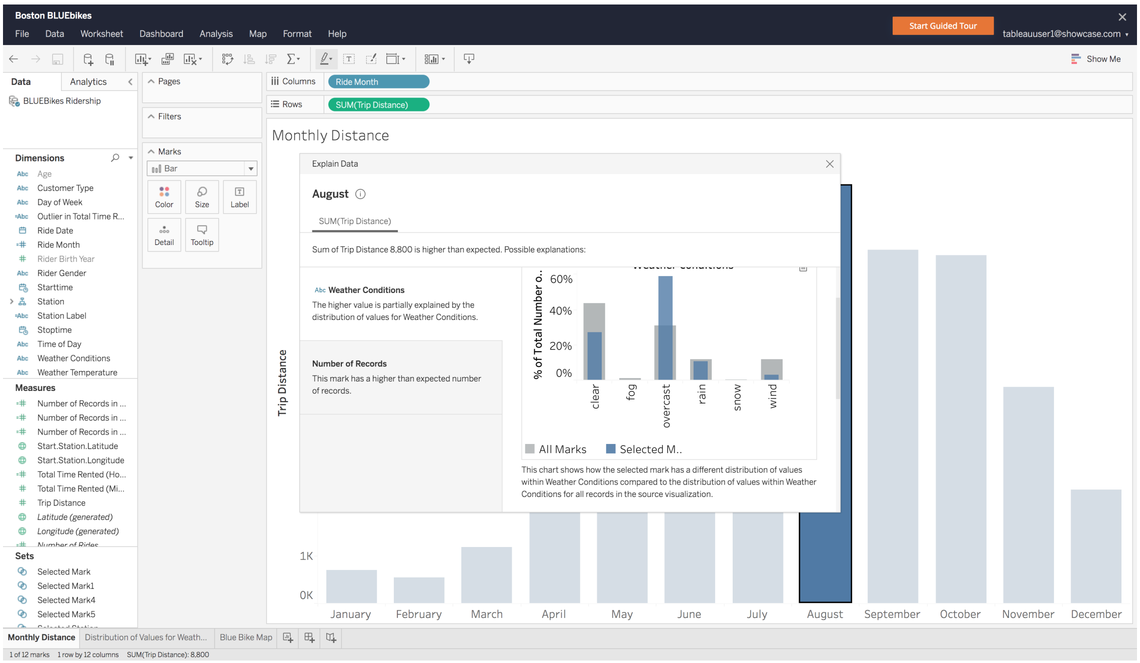



**Figure 2.2:** A screenshot of Tableau showing generated insights with explanation [17].

The general insights are generated based on the whole data set, presenting the suggestions in a list view with explanations for why each insight was generated, see Figure 2.3. It is possible to narrow down the general insights, to look at specific categories from the data set, without having to base it off of a visualization. There is also an integrated search bot to help the user search within the context of exploring the insights.

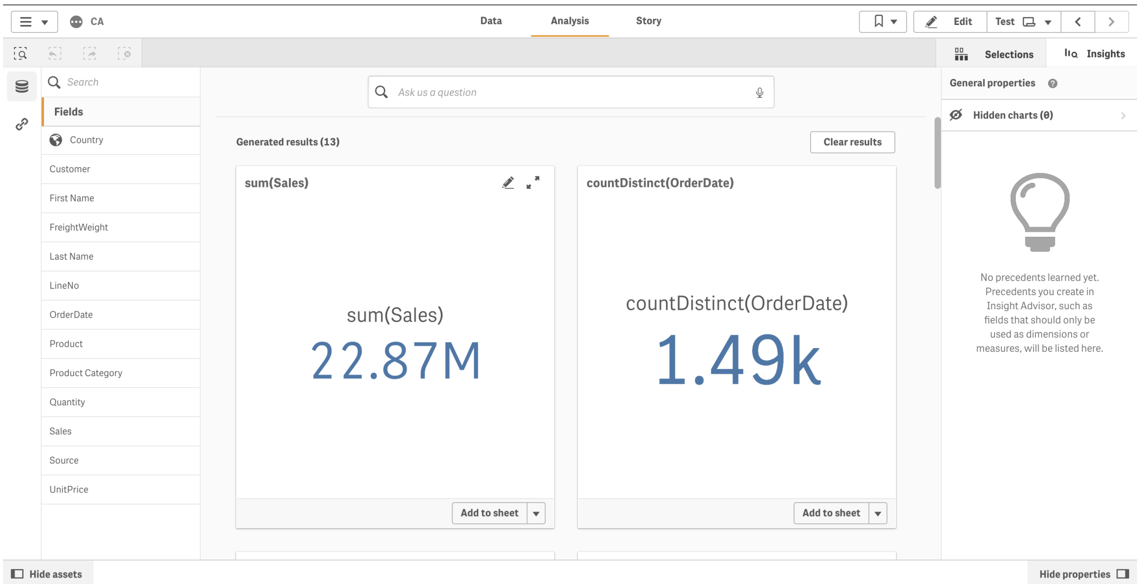



**Figure 2.3:** A screenshot of Qlik Sense showing a list of generated insights [18].

Associative insights are insights based on selected categories within the data set, either by selecting them from tables or the categories displayed in a visualization [19]. The insights are written explanations highlighting particularly interesting numbers in the selected data sets.

### 2.1.4 The Product

The Company has implemented AI-generated insights in the Product, seen by the user as a recommendation system. Similar to the above examples, the recommendations come in two different forms: more general and more narrow.

While there is no general recommendations based on the whole data set, there are recommendations based on the user picking a starting point for the AI to focus on. The Product then presents a list of statistically relevant relationships between the category, or categories, chosen, as well as other categories within the data set. The visualizations can then be added to a dashboard to be explored further.

The more narrow recommendations are based on selections within a visualization, and present interesting visualizations based on the relationship between the marked data and other categories within the data set.

The Product's recommendations currently focus more on interesting relationships, rather than explanations for fluctuations or outliers. It also does not provide explanations for why the recommendation was made.

### 2.1.5 Other related work

Research has also been done into the field of recommendations and insights outside of software companies. Two examples of relevant research is Voyager [8] and Foresight [7]. Voyager explores the realm of visualization recommendations by looking at the space of data dimensions and visual encoding [7]. The prototype provides breadth-oriented exploration to increase the coverage of a data set, and their approach focuses on data variation, rather than design variation [8]. It works by providing the user with a univariate summary when opening up the program, where they can then click on a variable that interests them and continue deep-diving into the content. Foresight, on the other hand, explores the space of insights and builds on ideas around previous research in automated analytics. Their prototype is built around insight queries and is structured similarly to Voyager. An initial list of possibly interesting insights is presented to the user, and they can then choose what to deep-dive into. While Voyager mentions certain design challenges, Foresight is even more focused on the implementation of the insight system rather than the design.

While both research projects resulted in high-fidelity prototypes, their focus seem to be on the implementation of algorithms for insight or recommendation systems. They also focus more on system-generated insights in standalone software, with the main purpose of suggesting insights. However, they do not address the application of insights in analytics software and within dashboards and visualizations.

## 2.2 Target user group

In this project, the target user groups will be the experienced analysts of an analytic tool, together with a consumers of already existing dashboards. The user groups will be the target of the design and guide design decisions.

**Figure 2.4:** Serena the Strategic Analyst and Ethan the Executive, two personas created by the Company

Two personas, created by the Company, will represent the user groups. The first persona Serena, the strategic analyst (see Figure 2.4), is described as an analyst at a big company, and her job is to combine and analyze data and make it understandable. She summarises, draws conclusions, and provides highlights from data and present it at meetings with managers. How the analysis is presented is also important to her in order to avoid misunderstandings. Transparency of the data and where the analysis came from is also of high importance since she might be required to answer questions that requires her to dig deeper into the data. She is not afraid to learn new ways of exploring and make use of the data and therefore motivated to learn more about the tool and its functions.

The second persona is Ethan, the Executive (see Figure 2.4). He is CFO at a big company, and he needs to make informed decisions for the company in order to be competitive. He is a busy person and must stay productive and often uses a tablet for his work. He meets with managers that provide him with ready presentations, and the data in dashboards must be clear and easily let him spot the crucial information.

# 3

# Theory

In this section, theory within the field of agentive technology, as well as the field of analytics, is presented. Visualization recommendation systems, as presented in [9], are also introduced, together with an overview of what insights are in visual analytics.

## 3.1 The Field of Analytics

The field of analyzing data is quite broad and includes many key terms, including analytics, data analysis, visual analytics, and business intelligence. These terms are often used interchangeably and are defined by multiple sources, which causes some confusion. Many of the terms are highly connected [1], which also contributes to the confusion. In this section, key terms within the field will be examined.

### 3.1.1 Business Intelligence and Analytics

Analytics is presented in [20] as a word with two definitions, "Analytics" and "analytics". Each serves a purpose within the field of analytics.

Eckerson [20] defines "Analytics" as an umbrella term for business decisions and actions driven by insights from data. However, he also admits that he has used the same definition for data warehousing, business intelligence, and performance management. Thus, "Analytics" will also be referred to as business intelligence within the context of this thesis.

Business intelligence (BI) was defined by Howard Dressner in 1989, with a definition that is very similar to the one mentioned in [20]: BI is an umbrella term for processes and tools which improve business decisions and can increase business performances [21], [22]. Over the years, the term has been redefined multiple times, either in a similar umbrella fashion as Dressner or a more narrow definition. For example, Negash and Gray [21] defined BI as systems that include data gathering, data storage, and knowledge management. Loshin, on the other hand, emphasizes the process of converting data to information, and information to knowledge, which helps drive profitable business actions [23]. The value of BI is the knowledge derived from the data, which can then be used as an asset, and according to Ranjan [24], the purpose of investing in BI is to start thinking proactively with the data collected by a company, rather than reactively.

The other definition of "analytics," however, describes the technologies and techniques used when analyzing data [20]. As the definition focuses on analysis within a business context, it is very tool oriented. Two types of analysis tools are presented: reporting tools and analysis tools. Reporting tools help the user monitor business metrics, for example, dashboards or scorecards. Analysis tools, however, help the user explore data in an ad hoc fashion. The type of tool to use depends on the question asked. While both types are connected within a business ecosystem, the trend is that there are paradigms where one type of tool is more relevant. For example, there is currently a trend of predicting business outcomes, asking the question, "what will happen?" Answering this question would require analysis tools rather than reporting tools.

Companies use BI software to aid in the BI process. According to Chaudhuri, Dayal, and Narasayya [25], BI software is a collection of decision support technologies, helping the user make better and faster decisions. The software is used to facilitate activities such as business analysis, forecasting, visualization, and querying [24]. The main objective of BI software is aiding in decision making by allowing businesses to know themselves and how they can improve [22].

### 3.1.2 Exploratory Data Analysis

The previously mentioned "analytics" focuses more on the process of analysis within a business context. However, data analysis can be seen as a similar but more general process. Cui [1] defines data analysis as a process of exploring data while trying to discover insights and use them to support decision making, through statistical models and analytic techniques. According to Adrienko and Adrienko [11], data analysis is driven by questions, motivating a party to analyze data and determine what methods to use, and then having to interpret the results.

Data analysis is divided into two categories: confirmatory data analysis (CDA) and exploratory data analysis (EDA) [1]. CDA is the process of confirming assumptions through statistical hypothesis testing. EDA, on the other hand, is a process of exploring data and finding patterns and other features. Adrienko [11] states that EDA mainly deals with analyzing with a purpose, possibly through a general question, or sets of questions, or just looking at what could be interesting in a data set. Unlike CDA, which uses visualizations to show results, EDA utilizes visualizations to explore and interact with data [1].

Within EDA there are two types of tasks: elementary tasks and synoptic tasks [11]. Performing elementary tasks involves looking at specific elements within a data set, such as looking up values corresponding to a starting value of a data category. Synoptic tasks, on the other hand, requires consideration of the entirety of a data set, or subset. It is performed to see behaviors and find patterns within the data. In both tasks, however, an analyst tries to derive insights from the data by "visualizing correlations, outliers, empirical distribution and density functions, clusters" [7].

While the definition of EDA focuses on the task of analyzing data, rather than the tools used, Adrienko and Adrienko [11] mention some tools to aid in the data analysis process. The classification of tools resulted in five categories: visualization, display manipulation, data manipulation, querying, and computation. As argued in [11], it is necessary to use a variety of tools to analyze data properly. They also stress the importance and the great role of visualization, even when working with the other four tools.

### 3.1.3 Visual Analytics

While data analysis is the process of analyzing data, [1] defines visual analytics as "the science of analytical reasoning supported by interactive visual interfaces." It has its roots in "Knowledge Discovery and Data Mining," proposed in 1989 as a result of the increase in the amount of data to analyze. Visual analytics combines multiple fields, such as visualization, human-computer interaction, data analysis, statistics, perception and cognition, and analytical reasoning.

Depending on the definition, the goal of visual analytics is very similar to EDA [1]. However, visual analytics also tries to bring in human cognition and intelligence to produce the results. It emphasizes the analysis of data and discovery of knowledge within the data, rather than just the presentation of numbers.

According to [1], some challenges of visual analytics include scalability and interaction.

Scalability of visual analytics is the ability to effectively display many data points or dimensions at the same time [1]. With more and more data having to be analyzed, it becomes a problem for the human ability to interact and analyze that data. Currently, most research focuses on larger displays for the analysis, however other types of methods could be more effective at helping with the problem of scalability. Scalability is limited both by human and machine capabilities.

As mentioned in [1], the interaction-based challenges are grounded in the investigation of the cognitive and perceptual impacts of incorporating human judgment in the analytics process. The challenge lies in exploring new interactions and taking advantage of advances in technology and devices.

### 3.1.4 Insight in Visual Analytics

According to Card et al. [26], "the purpose of visualization is insight," and Thomas and Cook argue that the purpose of visual analytics is to enable and discover insight [2]. However, because of its abstract nature, insight is challenging to define [3]. What does insight mean in the field of analytics?

In cognitive neuroscience, an insight is perceived as an "aha" or "eureka" moment. It is a moment where a person moves from a state of not knowing to knowing how to solve a problem [3]. It is even possible to quantify this unit of discovery by observing the person's neural network. Chang et al. calls it a "spontaneous insight," and they argue that insight can also be defined as a unit of knowledge, where an analyst becomes informed of a previously unknown relationship in a data set [3]. Insight can be achieved by providing "overview, adjust, detect patterns, and match mental model" [3]. Demiralp et al. [7], creators of Foresight presented in section 2.1.5, provides some concrete examples of what insights can be:

"Examples of insights include a high linear correlation between attributes x and y, high concentration about the mean of x-values, the presence of extreme x-value outliers, a strong clustering of (x;y)-values according to z-values."

Chang et al. [3] also argue that both types of insights are useful in visual analytics. With certain deep knowledge within a topic comes a chance of making discoveries through insights. However, spontaneous insights also allows the analyst to learn new knowledge because it opens up new directions.

## 3.2 Visualization Recommendation Systems

Modern visual analytics tools should provide the user with assistance when it comes to producing visualizations, taking into account the user's preferences and requirements [27]. Vartak et al. [9] proposes visualization recommendation (VisRec) systems to enable faster visual analytics. The system works by recommending visualizations that highlight patterns or trends of interest. The necessity of VisRec systems comes down to two main factors, the size of the data sets and the varying skill level of the users performing data analysis. The study focused on systems-oriented challenges with VisRec systems, but acknowledges that there are other types of challenges within these types of systems that still need to be addressed, such as user interface and interaction problems.

Some limitations of current tools are proposed by [9]. Because of the large data set, it is difficult for the user to traverse all of the information and manually visualize everything. Not being able to visualize everything can result in some of the data set never be explored. Insufficient means to specify trends of interest is another limitation. It is currently difficult for users to specify what they are looking for within the context of their analysis. Current tools also lack the context of a "bigger picture," making it difficult to see if a pattern or a trend is an anomaly specifically for this attribute, or if a similar trend can be seen in other attributes. Providing the "big picture" context of a pattern or trend would provide useful information for the user, which could easily be missed if a search for correlations had to be done manually.

### 3.2.1 Recommendation Axes

To determine the usefulness of a visualization, Vartak [9] proposes five factors that need to be considered when designing a VisRec system. These factors are called recommendation axes.

**Data characteristics** are interesting values, trends, and patterns, which are then used when determining if a visualization is interesting for a user. These characteristics include summaries, correlations, patterns and trends, and advanced statistics. The recommended visualizations can either show one of these characteristics, or be used as a measure to rank the usefulness of a visualization.

Another axes to look at is the **intended task or insight** of the user. These aspects include the style of analysis, the subject of analysis, or the goal of analysis [9]. One way of obtaining the intended task is through user-driven inputs such as a drop-down menu. However, it could also be possible to infer the intent by looking at the user's actions.

Determining the **semantics or domain knowledge** will also dictate the usefulness of a visualization. The recommendation system needs to understand what type of trends are obvious, and which ones are novelties. Domain knowledge can include what types of information are usually interesting, but also outside factors, which are not part of the data set.

**Visual ease of understanding** is a dimension that determines the usefulness of a visualization based on if the data is visualized most intuitively.

VisRec systems also need to determine the **user preferences and competencies** of its users. It needs to be designed to handle different types of visual literacy and statistical ability and adapt to the user's preferences.

### 3.2.2 Recommendation Criteria

In order to recommend visualizations, a measure of the quality of the recommendations is also needed, such as a recommendation criteria.

**Relevance** This criteria is determined by the recommendation axes and involves the factors mentioned in 3.2.1.

**Surprise** A metric that measures the novelty of a recommendation and determines whether it is original, unexpected, or if something is out of the ordinary.

**Non-obviousness** Determines whether the recommendations match with the semantics and domain knowledge of the data set.

While the above criteria are mainly for the visualizations themselves, Vartak [9] also describes some essential criteria for the quality of a whole visualization set. The diversity and the coverage of the set are important criteria used to determine the quality of a whole set.

## 3.3 Agentive Technology

The tools used for many activities are going from being "dumb" tools, to becoming smarter and starting to act on their own, through artificial intelligence [10]. Current artificial intelligence technology is quite "weak", and is called artificial narrow intelligence (ANI). This type of AI is good at doing one particular thing and cannot apply its knowledge to other contexts than those which it already knows of. The AI is made up of one or more agents, or narrow artificial intelligence that acts on behalf of its user.

According to [10], an agent's intelligence can be determined by several factors. Agents monitor data streams and try to make smart inferences depending on the read data. For example, an intelligence agent can plan by looking at multiple different options and choosing the best one, and it can also adapt depending on the circumstances. Advanced agents can also refine their behaviors as the technology is being used more and more. The intelligence of an agent is a spectrum, and it can be more or less intelligent, depending on the system.

Much like the agent's intelligence is a spectrum, the agency of an artifact can be more or less agentive than others. Agentive technology consists of one or more agents who watch a continuous data stream and respond with some type of artificial intelligence to act on behalf of its users. Each part of that definition can be tweaked for an artifact to become more or less agentive, such as the intelligence or the size of the data stream.

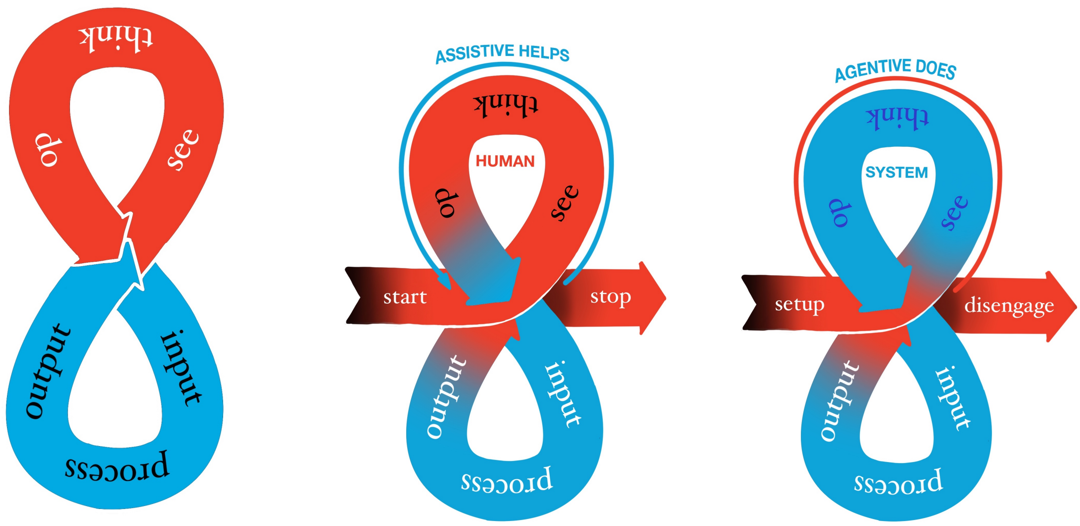
Some key aspects of agentive technology is that an agent does a task as per the preferences of the user. The design must be focused on an easy setup for the user and give the user information touchpoints throughout. An agent within the agentive technology will be working in the background, out of sight for the user, and will perform actions on behalf of the user. The technology also allows the user to keep the agent on track through preferences or other input.

As stated in [10], agentive technology is not the same as assistive technology. Assistive technology aids the user by doing much of the computing for the user, but the user still has to be the one to do the final step. While parts of agentive technology include assistive qualities, the agent will act for the user, acting entirely on their own for certain parts.

Agentive technology is also not just automating redundant tasks. It takes agency over a task and performs it for the user, but within a scope and limitations set by the user. It services the human, and depends on certain inputs, while automation serves to cut out the human altogether.

### 3.3.1 Designing Agentive Technology

Noessel [10] introduces a new model for describing interactions with agentive technology. A previous simplified model for other technology is the see-think-do loop, describing the loop of human interaction with an artifact. The user sees something, then think about how to respond to that thing, and finally does something about it. When talking about the model from a human-computer interaction context, the loop is also interconnected with a similar loop for the computer. The computer takes an input, from the user's "do" action, processes that input, and then outputs it so the user can see the result. The two loops connect, creating a figure eight, which continuously loops, seen in Figure 3.1a. When assistive technology is added to an application, a start and stop point of the interaction is added to the loop. The idea is then for the assistive technology to assist the user during the see-think-do loop, as seen in Figure 3.1b.




**(a)** User-driven see-think-do loop

**(b)** Assistive see-think-do loop

**(c)** Agentive see-think-do loop

**Figure 3.1:** See-think-do loops presented by Noessel [10]

The agentive model, however, includes a setup and a disengage part at the beginning and end of the figure eight mentioned above. The user controls these two parts, while the two loops mentioned above will be completely taken over by the agent, as seen in Figure 3.1c. However, the design should still account for the user to be able to set up the agent's capabilities and limitations, see what the agent is doing by monitoring what is happening and receiving updates about successes and problems, as well as helping the agent perform tasks by triggering certain events or even take over for the agent completely if the user would want.

Two major challenges one faces when designing for human-agent interaction is conveying capabilities and conveying limitations [10]. For digital agents, the lack of physical affordances will prove to be a challenge when trying to convey what the agent can do.

### 3.3.2 Guidelines for Human-AI Interaction

Amershi et al. [28] presents a list of guidelines for human-AI interaction, based on a study of 150 AI-related design recommendations, and evaluated through both heuristic evaluations and user testing. The overall concepts are very similar to what Noessel [10] highlights as important aspects when designing agentive systems.

From the applications evaluated using these guidelines, all participants found at least one guideline, which was either enforced or violated in each application. As concluded in [28], this shows a broad relevance factor of the guidelines, as they can be applied to many types of applications. However, some types of applications, such as voice assistants, had many reported "does not apply" instances. Roughly 40% of the guidelines were commonly found in AI applications, indicating the existence of intersecting mechanisms within AI systems.

The clarity of the guidelines was based on misrepresentations and confusion during evaluation. Guidelines were deemed unclear when they were systematically prone to errors. The final set of guidelines, seen below, is an updated set, updated for the parts that needed clarification.

The final guidelines are a set of 18 recommendations (see [28, p. 3]), divided into four categories: initially, during interaction, when wrong, and over time.

**Initially**

- G1 **Make clear what the system can do.** The system should help the user understand what it is capable of doing.
- G2 **Make clear how well the system can do what it can do.** Communicate how often the AI makes mistakes and its limitations.

**During Interaction**

- G3 **Time services based on context.** Interruption of the service should be based on the user's current context and adapt depending on the task and the environment.
- G4 **Show contextually relevant information.** Base the display of information on the task and the environment of the user.
- G5 **Match relevant social norms.** The experience should be based on the user's expected behaviour based on social and cultural context.
- G6 **Mitigate social biases.** The AI system should not enforce undesired or unfair stereotypes or biases.

**When Wrong**

- G7 **Support efficient invocation.** The user should easily be able to make a request to the system.

- G8 **Support efficient dismissal.** When necessary, the user should be able to dismiss the system.
- G9 **Support efficient correction.** When the system is wrong, the user should be allowed to correct, edit, or refine it.
- G10 **Scope services when in doubt.** The AI system's services should downgrade when uncertain of the user's goal.
- G11 **Make clear why the system did what it did.** Provide an explanation of the AI's behaviour and why something was done is a certain way.

**Over Time**

- G12 **Remember recent interactions.** A short term memory within the system should allow the user to reference a recent command or task.
- G13 **Learn from user behaviour.** The user's experience should be personalized and learn from their behaviour over time.
- G14 **Update and adapt cautiously.** Do not update or adapt the system's behaviour in such as way as to disrupt the user's experience.
- G15 **Encourage granular feedback.** Allow the user to provide feedback regarding their preferences within the system.
- G16 **Convey the consequences of user actions.** Show how changes in preferences will immediately change the future behaviour of the system.
- G17 **Provide global controls.** The system should be globally customizable, controlling what the system monitors and how it behaves.
- G18 **Notify users about changes.** Clearly inform users when the functionality of the system is updated.

The above set of guidelines is the result of an iterative process, evaluating both the relevance and clarity of each guideline, and then updating the set of guidelines as needed. The study shows that a general set of guidelines apply to human-AI interactions. However, the resulting set is a generalized set of guidelines, and there is a trade-off between generality and specialization. Specialized guidelines could be needed within certain high risk or highly regulated fields, such as automotive vehicles or financial systems. The general set of guidelines could be used as a basis for the development of more domain-specific guidance.

# 4

# Methodology

In this section, design research, along with design practice, will be presented. Also, the design process and its challenges will be described, along with suitable design methods relevant for the project.

## 4.1 Design Research

When doing design research, there are multiple methodologies one can choose to follow, depending on the research project, to produce design theory. There have been definition attempts for many types of methodologies. Three examples of methodologies are defined by Frayling as *Research into art and design*, *Research through art and design*, and *Research for art and design* [29, p. 5]. The two first definitions are the most commonly used methodologies, according to Frayling, while *Research for art and design* is, for example, the material gathered as a reference when creating art or design.

*Research into art and design* is the most straightforward, and most common, of the two methodologies [29]. It results in a further understanding into art or design, such as a theoretical perspective or historical research.

While some projects can use the above methodology to produce design theory, *Research through art and design* is more suitable for modern projects involving development work and action research, where both the design and the process are used to communicate the result [29].

Thus, this thesis will rely on *Research through art and design* as a way of producing the resulting design theory, but will specifically use the methodology defined as *Research through design*, recognizing the design process as a research process, bridging theory and practice. [30]. The resulting designs are then seen as representing the designer's decisions and as a justification of possible solutions to certain problems [31]. Gaver [31] argues that design research should result in multiple design examples, which can then be annotated with theory, rather than take their place. It is an interwoven relationship between the practical and the theoretical.

According to [31], the resulting design theory can be of several different types, such as conceptual work, a manifesto, a framework for design, or characterization of research through design itself. Often it is a combination of types.

Research Through Design often results in conceptual work, allowing the designer to describe the design in general terms and can be used to communicate their idea. The concept does not need to be fully developed, as descriptions of rationales or influences can also count as conceptual work. As the conceptual work is discussed, they become classified as theories.

A design manifesto is another example of theory produced from research through design practices. According to [31], design manifestos describe design practice, and are created to illustrate the approach and build an account of the process for the future.

Similar to design manifestos, frameworks for design also suggest practices to be pursued in the future. However, they tend to "downplay both their theoretical commitments and normative stance." [31] A framework can suggest methods to approach each stage of designing products. It "implies a conceptual orientation."

Research Through Design has previously been disputed as being unclear regarding "protocols, descriptions and guidelines for its processes, procedures, and activities" [31], when compared to scientific research. However, Gaver [31] has tried to clear up this confusion by providing an argument for why the design theory produced from Research Through Design should not be considered scientific, but rather be a different type of theory.

This thesis will focus on producing design theory as annotations to the artifacts produced. The theory will act as a rationale for the design choices and contains the artifacts produces and the design space. It will not try to pursue an ideal version of science, but rather perform research through design to aid in the discovery of the problem space and development of artifacts. The thesis will seek the "ultimate particulars," the truths of design.

## 4.2 Design Practice

Design methods and approaches started to develop in the mid 20th century, in an attempt to narrow down the essence of designing to be able to share reliable and standardized methods within the community [32]. Many methods and processes were proposed, and one important thing to note is that they were very different from each other. Jones proposes that this is a clue in itself, in regard to design methods, as the variety of discussed methods might reflect the variety needed when thinking of design methods. One thing these methods have in common, however, is the fact that they focus not on the result, but rather the ingredients of creating the design. However, what is the result of these methods? According to Jones, design is "to initiate change in man-made things" [32, p. 4]. As previously mentioned, there are multiple proposed processes to bring about change, but two possible approaches within interaction design, and are most relevant to this thesis, are user-centered design and activity-centered design.

### 4.2.1 User-Centered Design

The underlying philosophy of user-centered design is "placing the users at the center of design decisions" [33, p. 1]. The decisions should not be based on what the users say they want, but rather through educated observations of user's behaviors and preferences for various contexts in an application.

The user-centered design approach originated in the 1980s in Donald Norman's research lab at the University of California, San Diego. Many others have defined it throughout the years, but the general idea, is to put the user at the center of the design. Since Norman proposed it, the approach has been adopted by numerous designers, and in 1999 ISO 13407 was introduced, which was later updated to ISO 9241-210 [34], [35]. ISO 9241-210 defines a standard for human-centred design, using human instead of user as to allow the inclusion of all types of stakeholders affected by the design, but is often used synonymously with user-centered design. ISO's human-centred design is similar to previously mentioned definitions, and is an approach to interactive systems development with a focus on the users, their needs, and their requirements, and uses methods to apply ergonomics and usability to designs.

ISO 9241-210 provides a framework for working with user-centered design, and states some core principles to follow when adopting a human-centred approach [35, p. 6]:

(a) "the design is based upon an explicit understanding of users, tasks, and environments"

(b) "users are involved throughout design and development"

(c) "the design is driven and refined by user-centred evaluation"

(d) "the process is iterative"

(e) "the design addresses the whole user experience"

(f) "the design team includes multidisciplinary skills and perspectives"

Preece, Rogers, and Sharp [36] present a list of principles for a user-centered approach that involves similar concepts as the ISO standard, saying there should be a focus on human skill and judgment, directly relevant to the activity in hand, and will support the user. The three principles mentioned are early focus on users and tasks, empirical measurement, and iterative design. The principles are similar to the above mentioned, with some minor tweaks and more overarching principles. This is generally the case when looking at definitions or recommendations for a user-centered approach, with the main similarity being the user as a focal point.

The rationale to adopt a user-centered design is because of its economic and social benefits for both users and the companies producing the products [35]. Usable systems tend to be more appreciated and do well commercially, and reduces the risk of the product failing to meet requirements and being rejected by users. Being able to spot problems with the design at an early stage is beneficial according to [36, p. 284], as it is 100 times more expensive to find and fix software problems after delivery of the product.

### 4.2.2 Activity-Centered Design

In 1999 Gifford and Enyedy proposed a new type of design approach for Computer Supported Collaborative Learning (CSCL) environments, activity-centered design. Activity-centered design is based on the insights of distributed cognition and Activity Theory, and "emphasizes the design of computer-mediated environments to support and structure the interactions and interdependencies of an activity system" [37, p. 1]. It was formed as an alternative to other design approaches often used for CSCL environment, domain centered design, and learner centered design. As opposed to those approaches, activity-centered design is proposed to focus on designing activities that help students to carry out goal-directed actions using mediating material and social structures. According to [33], activity-centered design does not involve studying what the user must be able to do in the application, but rather how the application can enable the user to do those things. The focus is moved from understanding the users, to looking at them as participants in an activity.

While the term has been around for quite some time, there is still a lack of further research within the area of activity-centered design. It is proposed as a theoretical framework, allowing for future development of concrete design principles [37]. Unfortunately, those design principles have yet to be formed, and according to [33], no texts have been published outlining possible processes, methods, and deliverables, which an activity-centered approach could produce. More work is needed to properly see how this approach can be used on a day-to-day basis.

## 4.3 The Design Process

Design problems are often considered to be wicked problems [38]. One of the characterizing properties of wicked problems, mentioned by Ritter & Webber [39], are that wicked problems cannot be defined since the solution of the problem is also one definition of it. Another property is that a wicked problem is unique in the sense that, even though there might be similarities between the problems, there is no guaranteed success when applying the same solution to them. There are no right or wrong solutions to a wicked problem, just suggestions of solutions that can be graded from good to bad, from better to worse. Furthermore, there is no stopping rule to a wicked problem since there is not knowing if it is the correct solution, just if the solution is good or bad [39].

These characteristics of wicked problems are also the challenges with design problems. Since a design problem is unique in a way, there is no general solution or algorithm that can be used. A common way of tackling the complexity of the design domain, and thus the "wickedness" of design problems, is to use an iterative process [40]. Using an iterative process is also mentioned in section 4.2.1, as one of the principles for User-centered Design.

An example of an iterative process specific to interaction design and UX, is a lifecycle template called "the Wheel", presented by [40]. It consists of four different activities: Analyse, Design, Prototype, and Evaluate (see Figure 4.1). The activity Analyse consists of identifying user needs, understand the domain, and establish requirements. In the Design phase, different design ideas can be generated and explored through, for example, ideation and brainstorming. For the Prototype activity, prototypes with different levels of fidelity can be produced, and in the Evaluate activity, the design is tested to see if it meets the user needs and requirements [40].

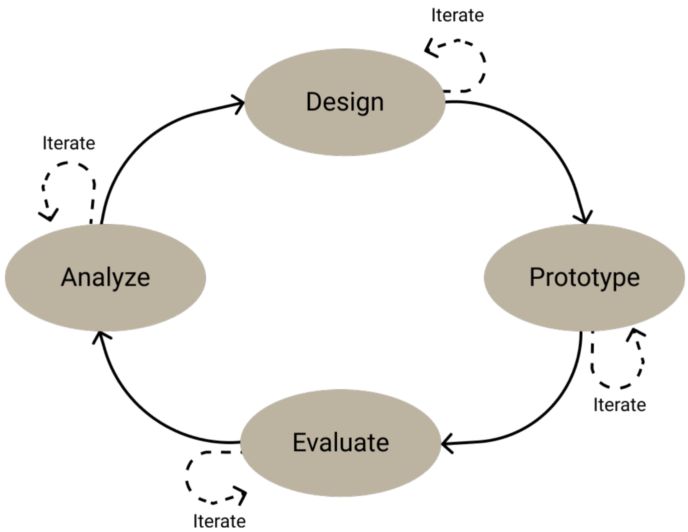



**Figure 4.1:** The Wheel - an example of iterative process presented in [40]

The flow between the activities is not necessarily in this order. They can be performed in any order, any number of times. The border between the activities is also diffuse since a sketch in the Design activity can be seen as a prototype as well, or an evaluation of some kind can take place in another activity than Evaluate activity [40].

The iteration (both within an activity or the whole process) is important in order to improve and refine the design through the process. Feedback provides the direction towards a more successful design. The challenge lies in moving forward through the activities, and balance the different activities against each other depending on goals and resources [40].

Throughout the iterative design process, it is good to be able to communicate about the design. Therefore, a design language can be used. A design language can be seen as the building blocks for design [41]. It is a tool for communication, for example, thinking, talking, discussing, and expressing a design. It can help with the communication between designers, between designers and clients, between designers and producers. Being able to communicate about a design also brings a better understanding of it. The communication also leads to being able to solve design problems easier. There are different levels of formality for design languages, ranging from formal and precise to more informal, like being just a byproduct in a team of designers when they communicate and solve design problems.

## 4.4 Design Methods

The lifecycle template "the Wheel" mentioned in section 4.3, is what its name implies, a template [40]. This means that for each design project, the process needs to be adapted to that specific project with its specific requirements, objectives, and resources. There are no rules for how to select the process and its methods for a project since there are as many processes as projects. Various factors affect the choice of methods to use, for example, what kind of product or system that is going to be designed and the resources in at hand [40]. The process is also affected by the phase it is in. In the beginning, there might be more emphasis on analysis, whereas, in the end, the weight is on evaluation. Making all these decisions is a challenge and also a balance of the needs, constraints, and limited resources.

There are frameworks and guidelines to help with this. In [42], a 3-dimensional framework is suggested to help decide on what research methods to use and when to use them. The different dimensions, which are Attitudinal vs. Behavioral, Qualitative vs. Quantitative and Context of Product Use, divide the methods into different categories. From here, it is possible to pick the best suitable research method depending on different goals.

In the following subsections, methods relevant for each activity in the Wheel (Analyse, Design, Prototype, and Evaluate), will be presented. The grouping of the different methods is not as distinctive as they might seem when they are divided between the different categories. Just because one method is listed under the Evaluate category, do not mean that it would not be suitable to use it under the Analyse phase.

### 4.4.1 Analyse

In this section, some methods for the Analyse activity are listed. The Analyse activity is about understanding user needs, explore problem space and extract requirements [40].

#### Literature Reviews

In a literature review, the researcher gathers information about the given topic, like a product or domain, through various sources like books, journals, articles, dissertations, and websites [30]. It is more about extracting relevant information and bring a suitable part into the own project than summarizing all information from every source. It is crucial in a review that sources are relevant and credible. Internet grants easy access to much information, but with this also comes the responsibility of critically examine the sources of information, and it is also important to correctly reference the review.

## Observation

Observation is a method to collect information and discover patterns about an artifact, environment or people, by observing the relevant situation [30]. Observation can be semistructured, where the observer observes and documents, or more structured, with predefined categories and checklists. It is important to keep in mind the tendency to see what one is looking for and instead observe with an open mind and to not draw conclusions about underlying motivations just from the behavior, but to verify this through example interviewing participants.

## Competitive Research and Competitive Analysis

Competitive research can be used to investigate current solutions among competitors and learn what works and what does not work for users in order to get a competitive advantage [43]. In order to accomplish competitive research, and to collect the data effectively and systematically, it is recommended to use a matrix in the form of a spreadsheet. This makes it easier to compare products and keep track of all information that needs to be collected and not forget something. Each row in the matrix will correspond to a competitor and each column containing a UX or market attribute. What attributes to compare depends on the product. It can, for example, be description, revenue streams, primary categories, content type, personalization features, and heuristic evaluation. The important part is that the positive and negative aspects of the UX of each product is evaluated.

After the competitive research, it is time for the competitive analysis and to come to some conclusions on what can give a competitive advantage in the own product or service to be designed. It is about transforming data into insights, such as what are the current trends, good or bad features and discover opportunities, and not just about making the comparisons or how to copy competitors. There are four recommended steps in order to perform a competitive analysis. Firstly, highlight interesting cells in the matrix by color-coding them. It can, for example, be the right solution for something or a problematic feature that many competitors struggled with. The second step includes creating subgroups of the products in order to increase efficiency when comparing, sorting, and filtering the products. By discover differences or similarities between the products might help in identifying what makes a product successful or failing, and thus what gives a product a competitive advantage. In the third step, it is time to fill out the last column in the matrix. This column is about summarizing each of the competing products, identifying what it is that makes the product more or less successful. What the best practices for UX and business models that were found among the products? What are the common trends, and what seems to make a successful product? The fourth and final step is about writing the Competitive Analysis Findings Brief, which is a summary of the complete analysis with recommendations about the future of the potential product. Is there a market for the potential product, and if yes, what are the opportunities in order to be a success?

### Personas

Personas is a tool for design teams describing user behavior and is a way of helping designers to focus on the design and to aid communication [30]. A persona consists of name, photo/sketch, and a description of his or her goals, attitudes, and behaviors. The persona is synthesized from the data of real users and derived from their similarities and patterns of behavior.

### 4.4.2 Design

Design is a very broad term, but in this context as one of the phases in the Wheel, it refers to the activity of generating and exploring different design ideas. **Ideation** is when designers come up with different suggestions to solve a design problem [40]. This is the creative phase of the design process, and when design ideas start to evolve. In [40], ideation is referred to as a collaborative and fast-paced activity. Ideation is about exploring and creating various design alternatives, and thus, it is important to iterate many times and generate many ideas in the form of, for example, sketches and prototypes. The ideation process is about generating ideas, but it also about reviewing them. It is important not to mix these two modes, since judgment during idea generation might block inspiration.

**Brainstorming** is one form of ideation where the combined creative power of a group is used [40]. There are different kind ways of implementing brainstorming, for example, **Bodystorming** [30]. What is important is that ideas are quickly generated, and the more, the better, all ideas are valuable [40]. One idea can trigger other ideas, as well. The session often starts with a discussion to set up the rules, decide a time limit, and establish goals. Document ideas by quick sketches and annotations, preferably one idea per paper/post-it in order to be able to move around the ideas freely.

**Sketchstorming** is another method for ideation [44]. It is a good method for coming up with many ideas. People are divided into smaller groups, where they individually create sketches without words or symbols. Then, one at a time, the sketches are presented to the rest in the smaller group, and the rest guess on what the sketch is illustrating, discussing the ideas. This method generates ideas both in the form of sketches, and the guesses and discussion around those sketches. **Crazy 8's** is another ideation method for coming up with a lot of ideas [45], [46]. It is about getting creative, not produce perfect sketches. A paper is folded three times to create eight sections. Then during eight minutes, the participants sketch eight ideas that afterwards are discussed.

A **Design Charette** could be described as a type of workshop, where the participants together generate the best possible ideas [30]. It is a good idea to involve people from different disciplines, not just designers, but for example project stakeholders and developers as well. The participants are divided into smaller groups, where they perform 10 minutes of sketching together, then two people move to new groups, and the last person remains. Everyone brings forward their best idea in the new groups, and the design ideas mix and merge into the best one. Ideas are explored and evaluated several rounds, the best ones evolving to be better and better, in an iterative process [30].

### 4.4.3 Prototype

By producing prototypes, designers can concretize a design idea, either for exploration of design ideas, communication purposes or for evaluation [47]. A prototype can take on many forms, depending on purpose and stage in the design process. A rough sketch on paper can be used in earlier stages, and later in the design process, the prototype might look more like the final product [47].

A **low-fidelity prototype** is a simple prototype intended to be fast and cheap to produce, and also be easily modified [47]. They can be made of for example paper or cardboard, without the look or functionality of the final product, but instead being flexible and often used early in the design process for testing and exploring different ideas. An example of low-fidelity prototyping are sketches. Also, these combined can create another method, storyboards, which is a series of sketches. They illustrate how a user might achieve a specific task. A third example is Wizard of Oz [30], where a human operator imitates the responses of a product/software for a user, instead of having actually to implement the product.

**High-fidelity prototypes** are often a bit more complicated compared to low-fidelity prototypes, and often closer to the look and functionality of the final product [47]. Creating these types of prototypes take more time and are more common later in the design process. Since it is time-consuming, a trade-off here could be about the range and depth of the functionality for the prototype. Horizontal prototyping is when there is much functionality, but it is less implemented, compared to vertical prototyping when there is less functionality, but the existing features are more implemented [47].

The level of fidelity depends on what kind of question the designer wants to have answered, and there is always a trade-off since one prototype can not answer all the questions [47].

**Parallel prototyping** is the process of quickly creating several low-fidelity prototypes simultaneously in order to experiment and explore different design ideas [30]. By exploring several designs at the same time, it helps the designer not getting too fixated and converge around one idea too early in the design process, as well as it gives more possible solutions to the design problem. Then the best from each idea can be merged into one final and optimized design.

### 4.4.4 Evaluate

In the Evaluate phase, the design is tested and analyzed in order to see if it meets the user needs and requirements.

#### Heuristic Evaluation

Heuristic evaluation, sometimes known as expert evaluation, is an evaluation method not requiring users [47]. Time shortage or difficulty finding users can be reasons for not involving users [47]. However, heuristic evaluation can also be applied in order to discover usability problems before letting real users test an interface [30]. Instead of users, a researcher, sometimes referred to as an expert, evaluates the design or interface against a set of heuristics, based on design guidelines or best practices for usability in order to discover usability problems [47]. This method can be used in any phase of the design process [47], and can help detect baseline usability problems early in the project [30]. One of the negative aspects of heuristic evaluation is that actual users can only detect some problems. It is also important to take care when creating the heuristics for the evaluation since poorly developed criteria for the evaluation will not help in finding the relevant usability issues [47]. A way of decreasing the chance of missing key problems is to have several researchers do the evaluation but also to combine this method with others, for example, usability testing with actual users.

#### Usability testing

Usability testing is a method for evaluating an interface in order to discover potential problems [30]. A test consisting of a set of tasks is constructed, created to reflect real user goals. A test participant performs the tasks, while often asked to use the Think-aloud technique, to express their process and thoughts out loud to the test facilitator. The interface is tested to see if the user might struggle or experience confusion and frustration. The tasks must be constructed in a way that does not influence the test participant's actions.

#### Questionnaire

Questionnaires collect self-reported data from users about their behaviors, feelings, and perceptions [30]. The formulation of questions is essential since it will affect the answers as well. Open-ended questions can give a deeper insight while closed-ended questions are easier to analyze. Questionnaires are often used and complemented with other research methods, for example, observation, in order to verify (or challenge) the subjective data from the questionnaires. This method can be used in several stages of the design process, for example, Analyse or Evaluation.

#### Interviews

An interview is a method for gathering users data and can be divided into four categories: unstructured, semi-structured, structured, and group interviews (often called focus groups) [47]. This method for gathering qualitative data is often combined with other methods in order to verify, for example, quantitative data from a questionnaire [30]. The level of structure of the interview depends on the goal and what type of information the researcher want to gain from the interview [47].

**Unstructured** interviews are more informal and with open questions [47]. This allows for more flexibility even though the researcher has a list of subjects to talk about [30]. It can feel more comfortable for the participant since it is more of a conversation, but this also puts pressure on the researcher to lead the conversation in order to get the relevant information [30].

The **structured** kind of interviews are more rigorous and involve a script of questions to be answered and with a set of possible predefined answers [47]. They are best used when there is a specific goal, and specific questions can be formed to achieve that goal [47]. It is easier to manage both time and questions with a structured interview, and it is also easier to analyze the data afterward [30]. The negative part is that it can make the interviewee more uncomfortable since a structured interview can seem formal and impersonal.

**Semi-structured** interviews is a mix of structured and unstructured interviews [47]. Here a combination of open and closed questions can be used, and the set of questions is more as a reminder for the interviewer to cover all the topics of the interview [36].

The fourth type of interview, group interview, is also called **focus group**, is where a group of people talk to each other lead by a moderator [47]. This sort of interview requires a good moderator that can guide, encourage, and make the group comfortable. A benefit from a group of people is that this can trigger a more natural conversation [30] but a problem with this sort of interview is that a group of people often tends to agree with each other which is not helpful when a researcher or designer tries to elicit different users' needs and motivations [47].

### Design Critique

A way of getting feedback on a design is a design critique [48]. This can be done anytime in the design process and be used several times in different iterations. The design is presented to critiquer(s) that can give feedback and provide discussion in order to improve the design.

## 4.5 Tools

In this section, some of the available tools relevant for the project and design methods above will be described. Most of the tools are software or platforms for different types of prototyping or evaluation.

### 4.5.1 Prototyping tools

There are many different tools available for the different stages of the design process, from creating rough sketches to pixel-perfect wireframes [49]–[55].

**Balsamiq Wireframes** is a software for wireframing and enables the user to quickly sketch mock-ups and wireframes [49]. Wireframes in Balsamiq have the look of black and white sketches, which can be an advantage when making low-fidelity prototypes, and it is too early in the design process to focus on details or colors.

**Marvel** is another platform for designing and prototyping [50]. This web application allows the user to create wireframes but also to link them together into prototypes. The projects can also be easily shared for collaboration and feedback.

**Figma** is a tool that lets the user design wireframes and connect them into prototypes [51]. It offers collaboration in real-time and allows several users to work on the same project.

**Adobe XD** is also a software for design and prototyping and part of the Adobe Creative Cloud subscription [52]. The user can produce wireframes and prototypes but also share them with other users. Since it is part of the Adobe Creative Cloud subscription, it works well with other Adobe software.

**Axure RP** is a software for creating interactive prototypes [53]. It is also possible for the user to create HTML mock-ups, allowing the user to add functionality to the prototype without any coding.

### 4.5.2 UserTesting

UserTesting.com is an online service for remote user tests in order to evaluate products such as websites, mobile applications or prototypes with real users [56]. At this platform, tests are set up with tasks to be completed by the test participants and possibly requirements and screening questions in order to get test participants from the preferred user group. The test participant completes the test using the think-aloud technique, while audio and screens are recorded. It is possible to add questions as well, closed or open, for the user to answer during/after the test. The results of the tests are available right after the participants are done with all the tasks and questions.

# 5

# Planning

For this master's thesis, an iterative design approach was chosen in order to answer the research question. An iterative design would also help to address the challenge of design problems being wicked (as mentioned in section 4.3). The project was planned to be divided into three iterations, preceded by a pre-study. Each phase was planned to consist of different design activities, and after each iteration, a set of guidelines would be developed or refined. The planned result of the project was the final version of the guidelines, along with a high-fidelity prototype.

## 5.1 Pre-study

The first phase would be about preparing for the design work by doing a literature review on the topic of AI-generated insights, recommendations in analytics, and how to work with agentive technologies in general. The literature review was also going to cover design methodology and relevant methods for the project. A competitive analysis, with a focus on UX attributes, would also be performed on different analytics software, including the Product, in order to explore the existing solutions on AI-driven insights and recommendations and their context in visual analytics.

## 5.2 First Iteration

In this phase, gathering data and setting up requirements, was planned to lead to the first version of guidelines. Interviews with internal stakeholders, such as product owners, developers, and designers, would be held in order to gain further understanding about the target user group, the field of analytics, the Product, and current challenges and opportunities on the subject of AI as a tool in visualizations. The combined insights from the pre-study, along with interviews from the different stakeholders, would help in the formulation of the first set of guidelines.

## 5.3 Second Iteration

The second iteration was planned to start with ideation and concept generation based on the takeaways and developed guidelines from the first iteration. This would be done by workshops and brainstorming, and afterward, the concepts were going to be evaluated in a design critique. Then some more ideation would be done by parallel prototyping based on the most promising concepts from the critique. Low-fidelity prototypes would be produced, continued with usability testing for evaluation of the prototypes. The outcome from the evaluation would then be used to update and refine the guidelines.

## 5.4 Third Iteration

In the third and final iteration, ideation was planned to be more focused on details and the updated guidelines from the previous iteration. From this, a high-fidelity prototype would be created. The aim of this prototype was to explore the guidelines into more detail and, after evaluation, provide the final, refined, and updated version of the guidelines.

## 5.5 Time Plan

The workload between the 20 weeks of this master's thesis had been divided into three main parts. The first four weeks have been a pre-study, consisting of a literature review and writing a planning report. Twelve weeks of the actual project work would follow, with three guideline iterations, where the majority of the time was going to be spent in the third iteration working on implementing a high-fidelity prototype and finalizing the guidelines. The last four weeks were planned to only focus on finalizing the thesis, with time for preparing for the presentation and opposition. Writing would have taken place continuously throughout the project. A Gantt Chart of the time plan is presented below.


Spring 2020: Weeks 1-20
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
Pre-Study
Literature Review
Competitive Analysis
Planning Report Writing
Planning Report
First Iteration
Data Gathering
Setting up requirements
First Guidelines
Second Iteration
Ideation and concept generation
Evaluation
Low-Fidelity Prototype
Updated Guidelines
Third Iteration
Ideation
Design
Implement High-fidelity prototype
Evaluation
Analysis
Final Guidelines
Report Writing
Report
Report 90%
Final Report
Finalize thesis
Prepare Presentation
Presentation
Prepare Opposition
Opposition
CURRENT WEEK

# 6

# Execution and Process

This chapter describes the process of the thesis. It was divided into several phases, starting with a pre-study followed by four iterations. Each iteration consisted of similar design activities: ideation, evaluation, and a reflection of the results, followed by an updated set of guidelines. During the iterative process, many prototypes were produced, and the concepts were iterated upon through each iteration. While standard practise is to see the prototype as the main result of an iteration, this thesis also produced and iterated over guidelines in every iteration.

The thesis progression mostly followed the plan presented in Section 5.5. The most significant change was that the second iteration was split into two separate iterations, resulting in dividing the project into a pre-study and four iterations. This was done since the first part of the planned second iteration included producing designs and evaluating them. The evaluation of the concepts affected the guidelines, and thus an extra iteration of guidelines was produced. However, the realized process is still very similar to the planned tasks, apart from iterating on the guidelines one extra time.

## 6.1 Pre-Study

The first part of the project consisted of a pre-study to study the field of analytics and agentive technology, while also getting a deeper understanding of the problem area of insights in visual analytics. The phase was primarily spent on a literature study, and also formulating an initial set of guidelines.

### 6.1.1 Literature Review

Multiple books and research papers were surveyed to get an understanding of what areas would be interesting to explore. The Company had books within the field of analytics and visualizations, as well as books for different interaction design topics, such as interview techniques, user research, and the design process. Reading these books and suggested papers gave a broad understanding of the subject at hand, particularly the field of analytics, as it was a new topic.

While literature was found for both the field of analytics as well as within interaction design, it was more difficult to find more specified topics, specifically for system-generated insights. The topic of agentive technology and human-AI interaction was explored and provided some further information about the topic. Google Scholar [57], a search engine for searching scholarly literature from many different types of databases, including Science Direct [58], IEEE Xplore [59] and ACM Digital Library [60] databases, was used to search for relevant keywords. IEEE Xplore and ACM Digital Library were also used. Topics researched in these databases include, but are not limited to, the following keywords:

- Visual analytics
- Data analytics
- Data insights
- Automated
- Predictive analytics, visualizations, recommendations, insights
- Situational insights
- Metadata insights
- Dashboard design
- Dashboard guidelines
- Agentive, assisted, augmented technology
- Visualization recommendation system
- Business information
- System generated insights
- AI generated insights

Within the time limit of the project, using the keywords mentioned above, the research yielded no previous research specifically within the design of system generated insights. Some relevant topics, such as the VisRec system mentioned in Section 3.2, were found, as well as research done within the design of producing insights from a more programmatic perspective, referenced in Section 2.1.5 and 3.1.4. The lack of research for guidelines and design frameworks for system-generated insights points towards a relatively unexplored problem space.

### 6.1.2 Reflections

Through the literature study, it was possible to get a broader understanding of the topic at large. However, it was clear that there was a lack of information regarding the application of system-generated insights. The lack of information made it difficult to know where to start.

The idea of insights is not well defined, as Chang et al. [3] states, and while they go on to define two types of insights, spontaneous insights and knowledge insights, the idea of insights is still quite abstract. What is explored in the thesis is the application of system-generated insights, thus partly removing the analyst from the equation. The purpose of the thesis was that the prototypes produced would provide system-generated insights to the user, to aid them in the analysis process, and help the analyst to reach insights themselves. The system-generated insights are there to make it easier to see connections within the data and to both provide the user with spontaneous insights and more knowledge of the data set.

The original scope of the project was to only look at the level of automation of applying system-generated insights. However, because of the lack of previous research, a broader scope was adopted to explore the design space of system-generated insights. The research question was changed to the one stated in Section 1.1, and the scope of the project became more broad and exploratory within the field.

While a narrower scope would have lead to a more focused study, it would have been difficult to specifically look at only the level of automation for insights, as there was no foundation for what system-generated insights were or how they are applied. However, the process was influenced by the idea of the level of automation, and it played a significant role in the exploration of the application of system-generated insights.

The most useful findings of the pre-study were the guidelines for human-AI interaction [28], Noessel's research in agentive technology [10], as well as the VisRec system [9]. These were some of the most influential resources for the initial set of guidelines presented below. While other prototypes have been produced, mentioned in Section 2.1.5, none of them have explored insights in such a broad context as within a visual analytics tool, which is supposed to work with many different types of users with different goals.

### 6.1.3 Guidelines v. 0

An initial version of the guidelines was created based on the literature review done in the pre-study. Some guidelines are based on the human-AI guidelines from Microsoft [28] as well as the recommendations mentioned by Noessel [10]. However, some guidelines are also based on the findings within the field of visual analytics and the VisRec system presented by Vartak [9].

There were a total of fifteen guidelines produced, divided into four categories: interactions with the agents, communication with the user, user experience, and analytical reasoning.

#### Interactions with the Agents

The guidelines in this category handle the interactions between the user and the agent, such as the setup, control of the agent, and the feedback provided by the user. Essentially, it focuses on the user's input to the agent.

G1 **The AI should be easy to set up.**

Noessel [10] states that in when setting up an agent, the user should be able to communicate their goals and preferences, set the permission and authorization of the system, and test the agent before deploying it to do the actual work.

G2 **User should be able to dismiss an agent.**

Dismissing an agent should be possible, according to the human-AI guideline G8 [28]. Noessel [10] also talks about dismissing the agent, through disengagement from the agent when the user is no longer in need of it.

G3 **Users should be able to invoke a dismissed agent.**

Once an agent has been dismissed, the user should be able to bring the agent back. This is based on G7 of human-AI guidelines [28].

G4 **Users should be able to provide feedback so the AI can learn user preferences over time.**

This is based on the human-AI guideline G13, that the agent should learn from the user behaviour over time. G15 also encourage granular feedback over time, providing feedback for the AI to learn user preferences over time.

#### Communication with the User

This category refers to the guidelines which impact the different levels of communication between the user and the agent.

G5 **The system should communicate what the agent can do.**

The system should be able to communicate what the agent can do so the user knows the scope of the AI. Noessel [10] brings up that being able to convey the capabilities and limitations of the system is a very important design challenge for agentive technology. G1 of human-AI guidelines also talks about this important aspect.

G6 **The system should communicate what the agent can't do.**

The system should also be able to provide the user with its limitations, further narrowing down the scope of the AI. The guideline is based on the same Noessel [10] principle as mentioned in the above guideline (G5).

G7 **The system should communicate the status of the agent's work.**

When the agent is performing an action, or has performed an action, the user should be updated. This guideline is based partially on multiple human-AI guidelines, for example G16 and G18. It also includes having sufficient error handling, or being clear when an agent is finished, active, or waiting.

G8 **The system should have sufficient error handling.**

Both Noessel [10] and the human-AI guidelines [28] mention error handling. Examples of error handling specifically for insights would be a message "no predictions found" or "data set too simple to analyze".

G9 **The system should notify the user if anything interesting can be found in the data.**

While this is slightly based on the notion that the agent should notify the user of any changes, G18 of human-AI guidelines [28], and showing contextually relevant information, G4 of human-AI guidelines, this is more focused on the fundamental aspect of what system generated insights are. They are supposed to show interesting things in data, and one way of doing so is for them to be system generated and notify the user when something interesting is found.

### User Experience

The guidelines in the User Experience category impact the general user experience of the system, with regard to user workflow and ease of understanding.

G10 **The user's workflow should not be disrupted.**

The user should be able to perform an analysis without being too disrupted by the agent and system-generated insights.

G11 **The agent should show results based on context.**

The agent should be able to provide the user with relevant insights based on the task they are performing and the data they are looking at. This is a guideline formed from one of the recommendation axes of the VisRec system [9].

### Analytical Reasoning

The guidelines presented below reflect on the analytical process.

G12 **The systems should provide sufficient information for why it is doing something.**

To provide transparency for the user, the agent needs to inform the user why it is showing certain insights.

G13 **The user should be able to backtrack an insight.**

In this context, backtracking means that the user should be able to follow the thought process of the AI, and look at the data that the insight is based on.

G14 **The visualization should clearly indicate what is generated from data vs system.**

Being able to separate between the data that is generated by the system and what is based on the data set is important for the understanding of the user.

G15 **The system should show the certainty of insights, if applicable.**

This guideline if mostly aimed at the insight type forecast, where showing the certainty of the prediction can be useful for the user. By showing the certainty of insights, the user can decide whether the insight is relevant for their use case or not.

## 6.2 Iteration 1: Product and user research

The focus for the first iteration was to get a better understanding of the Product and its users. For this, interviews with internal stakeholders at the Company was conducted. This phase was also about exploring more about what system-generated insights are and how they can be presented and utilized. To achieve this, a competitive analysis was performed on different products with some type of functionality for system-generated insights. In the end of this iteration, the set guidelines were updated and refined into Guidelines version 1.

### 6.2.1 Competitive Analysis

To understand how system-generated insights work in current analytics software, a competitive analysis was done, see more about this method in section 4.4.1. The analysis was done on four different software: Power BI Cloud, Power BI Desktop, Qlik, and Tableau (see more in section 2.1 for further descriptions of the different software).

For the competitive analysis, a matrix was used to gather different kinds of data for each software. The main idea was to see how they utilize insights and the pros and cons of each software's application of insights. Three heuristics were also set up to get an even scoring between the software. The following is a list of what was gathered:

1. Product name
2. Platform
3. Purpose of AI function
4. Types of relevant functions
5. Pros
6. Cons
7. Insight WITHIN a visualization?
8. Competitive advantages
9. **Heuristic 1**: Can you accomplish your primary task without disruption for both ad hoc analysis and consumer analysis?
10. **Heuristic 2**: Is it easy to understand why suggestions have been made? What are the breadcrumbs to show the way..?
11. **Heuristic 3**: Are the visuals easy to understand and informative? Is it cramped?
12. General notes

The goal of the exercise was to get a better understanding of the current problems with system-generated insights, and also see what design choices have been made previously.

Since one of the resulting guidelines of the pre-study was that the insights should not disrupt the user's workflow, it was a primary focus within the competitive analysis. Two other main goals of the competitive analysis was to look at the level of automation the current systems use as well as what advantages and pitfalls there could be with system-generated insights.

### Tableau

As explained in Section 2.1.2, Tableau's insight system is called Explain Data, and the software tried out for this analysis was the web version. The purpose of their AI is to get explanations for "the why" behind specific data points. It is used to find relationships, outliers, and other relevant information after clicking on a data point as a reference. Explain Data shows up as a pop-up window, above the current visualization. The user triggers it by clicking on a data point that they deem interesting. The pop-up can be seen in Figure 6.1, and it is possible to see how it blocks much of the underlying content. The interaction with the insights is entirely user-driven; the system does not show beforehand if there are any available insights or not.

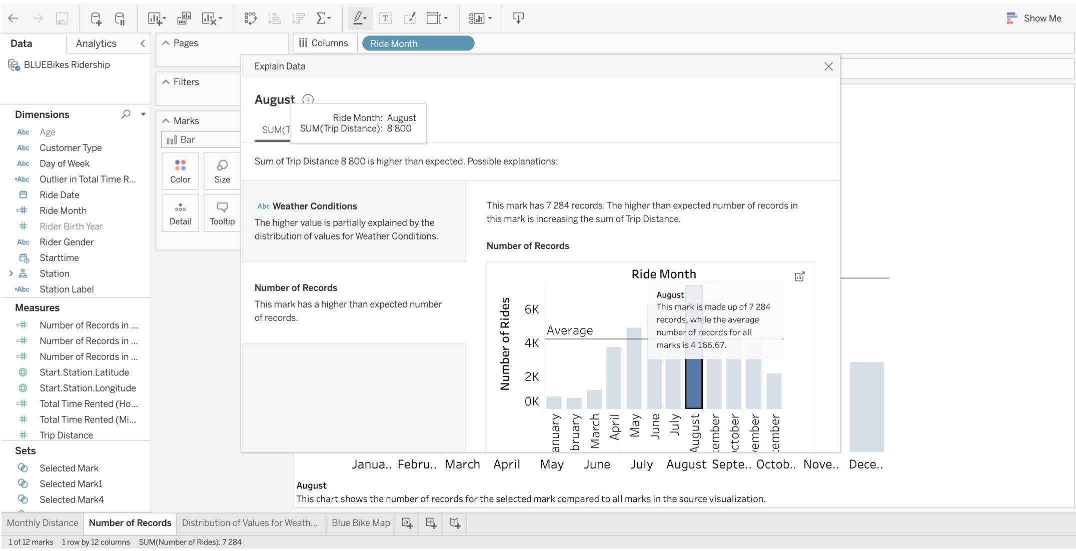



**Figure 6.1:** Explain Data pop-up with suggested insights, which showed up after clicking on a specific data point and asking for an explanation for that data.

Some pros discussed was that it was easy to access the Explain Data window. The window also had explanations as to why the AI suggested a specific insight, for example, "a value is higher than expected compared to the rest of the data set". Another positive aspect of the feature was that it could suggest more than just recommended visualizations, and go as far as to suggests a possible relationship between different data points.

However, there were some cons remarked, as well. The Explain Data pop-up opens up and covers the whole screen when it is engaged. The pop-up makes the insights hard to relate it to the original visualization, and it disturbs the analytic workflow. It was also not possible to add an insight to the current visualization, but rather add a whole new worksheet with the visualization from the insight, removing the user from the original workflow. While the information in the pop-up is very useful, it becomes a bit cluttered, making it difficult to take it all in.

In general, the information provided regarding the insights helped with the understanding said insight; however, it quickly resulted in information overload. The interaction is entirely user-driven, and is specific to one data point. It allows for a deep dive into the insight, but with certain user experience flaws.

Something to note is that the competitive analysis was made on a demo version of Tableau's Explain Data, and that the visualizations and data was already set up by the software. A specific demo guide was followed as a result, with the software going through each task of the program. Thus the software was not analyzed in its natural state, with other data than the data set provided by the demo. It is difficult to tell how well the Explain Data feature works with a different data set.

### Power BI Cloud

Power BI Cloud can be read about in more detail in Section 2.1.1. However, the service is web-based and has two types of insight systems: a visualization based drill-down and a general insight search of the whole data set.

The fact that there are two types of insights that can be generated is quite a positive feature. If the user wants to get an overview of the data set, they can get Power BI to scan the whole database, presenting potentially interesting visualizations and insights within that data. This can be seen in Figure 6.2. It would be a useful feature for the target user Serena for example, allowing her to explore any new data set by giving her interesting starting points to go off of. However, the user can also request insights using a specific visualization as a starting point. This process can be seen in Figure 6.3. Each insight can then be further explored by opening it in a more focused view.

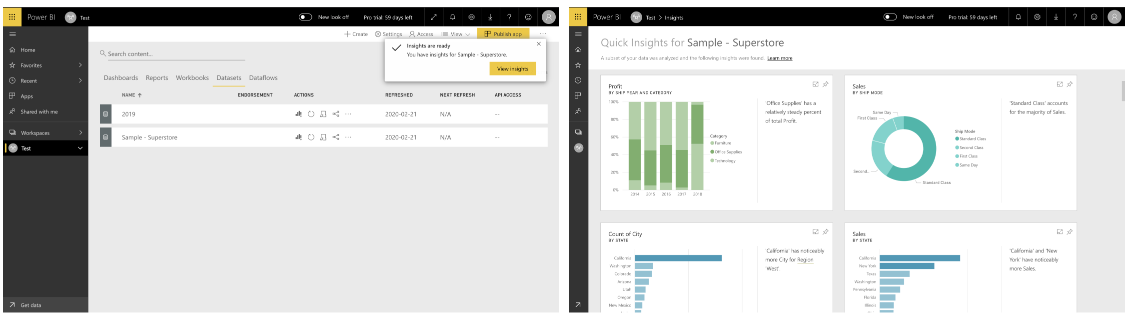



**Figure 6.2:** The process of generating general insight on a data set (left) and the resulting list of insights (right).

While the features that exist seem to be useful in theory, they are quite disruptive to use. To get insights on the whole data set, the user needs to go out of the analysis

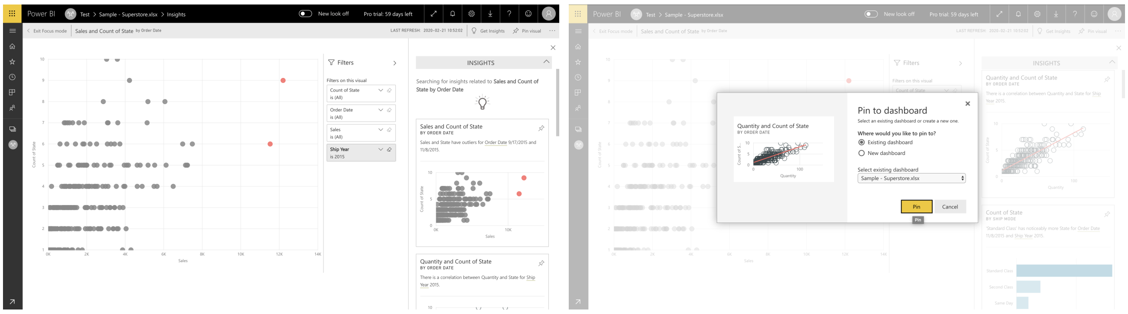


**Figure 6.3:** Visualization specific insights (left) and the process of pinning the visualization to a dashboard (right).

and into the database page and go from there. The whole process felt like it was unnecessarily complex to get "Quick insights", as the insight system in this software was called. While there is an option to get insight from within a visualization, it is not possible to drill down while also still seeing the original visualization. This disrupts the user's workflow, and it is quite frustrating how it is not possible to look at more than one visualization at the same time. It is possible to add insights to a dashboard, but there is no possibility to get any insights about all the visualizations in the dashboard. The user has to enter focused mode, looking at one specific visualization to get insights.

What is interesting for Power BI Cloud is the large number of types of insights [16]. They use their documentation to inform the user about what the system can do and its limitations, which can feel like it is removed from the system itself, however, it allows for a more thorough run-through of how the feature works.

The look and feel of the insights was less cluttered than the Tableau version, with a short sentence explaining what is interesting and why it is shown. However, that also means a lack of information about the insight. The insight panel was also modeless, which helped when wanting to look at the original visualization as well. Unfortunately, similar to Tableau, it was not possible to explore both the original visualization and the insight at the same time, as the insight opened as a new state.

### Power BI Desktop

Power BI Desktop, described in more detail in Section 2.1.1, uses insights in a different way than the Cloud version. The desktop version is still user-driven, but with a focus on insights based on specific data points, rather than the whole visualization or the whole data set. There seems to be two types of potential insights depending on the type of graph used: explain the distribution or explain the decrease/increase. Both types of insights can be seen in Figure 6.4. However, only line charts and bar charts were used for the analysis within the Power BI Desktop software.

There are fewer types of insights that the system can produce, but they seemed to be very good at finding hidden relationships for why something has changed. While the insight panel was a pop-up, it still allowed for some visibility of the original

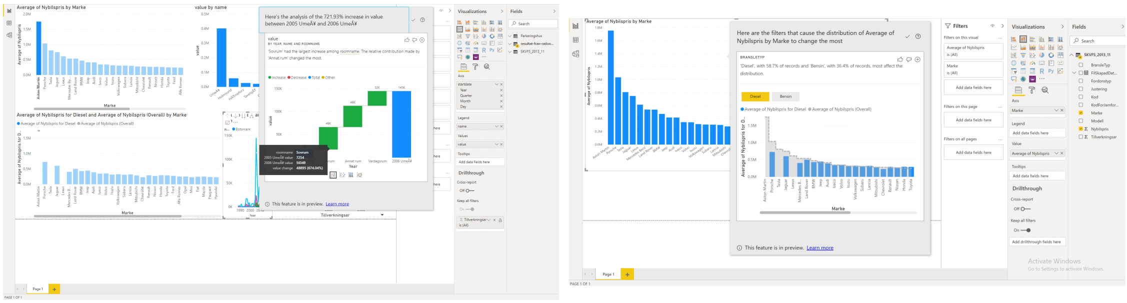


**Figure 6.4:** Printscreens of Power BI Desktop, showing two types of insights

visualization. The insight panel was less cluttered, and showed more information than the Cloud version, leading to a quicker understanding of why the insight is relevant.

Since the insights are based on the user selecting a specific data point they want to get an explanation for, the insights are very context-specific. This leads to a natural understanding of the context of the insight; however, it does limit the system to produce insights specifically for data the user has selected.

A positive feature for this type of insight was that the explain increase/decrease insights provided options for how to display the insight. The user was allowed to view the insight using four different types of visualizations, deciding for themselves which one is the most helpful.

This type of insight seemed to work well for dashboards where the user is specifically looking for an answer to a question, for example, "why is there a decrease in sales in 2015". However, it does not allow for a good general overview of interesting insights into a data set. It is also very user-driven.

### Qlik

Much like the Power BI Cloud software, Qlik allows for the user to get insights based on the whole data set or a specific visualization. Qlik has been described in more detail in Section 2.1.3. However, Qlik's system seems to be column-based. So the user is allowed to select all columns, or specific columns to see what insights can be generated on those columns, as seen in Figure 6.5.

The visualization specific insights, so-called Associative Insights within the program, seems to be more numbers-based, either pointing out specific numbers which seem interesting in the data or comparing two types of metrics, as seen in the right image in Figure 6.5.

While the workflow was still hindered in the program by certain design choices, the insights were readily available throughout the analysis process. It allowed for easier switching between the generated insights and the visualization the user is currently looking at. While it was readily available, it still meant that the user

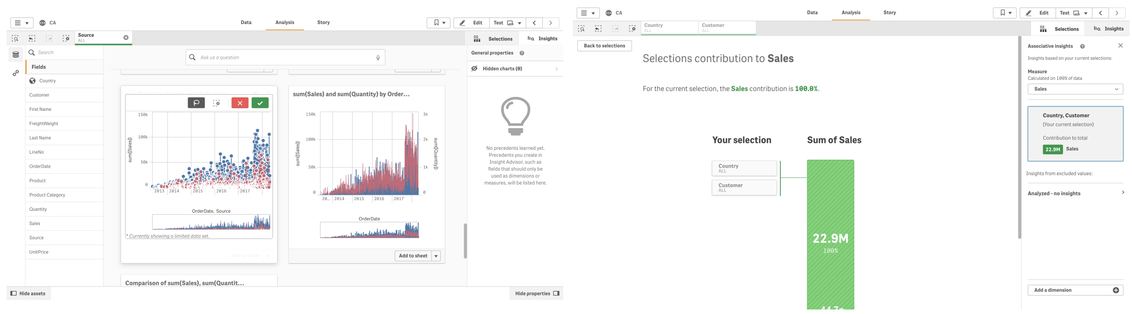



**Figure 6.5:** Generated insight from the whole data set (left) and Associative Insights based on one visualization (right).

had to switch between the two views, breaking the user's analysis workflow. The Associated Insight were available on the same page as the original visualization, however.

Qlik's generated insights seemed more like a recommendation system for visualizations, rather than providing insights. The insights themselves did not have much explanation as to why they were shown, but rather just stated the data it was based on. This made it much harder to understand why it was shown. While it is possible for an experienced analyst to figure out the reason for showing the insight, the other programs mentioned previously did a better job at explaining their generated insights.

### 6.2.2 Interviews with Internal Stakeholders

The functionality in the Product currently closest to the topic of this thesis, AI-generated insights, is the recommendation system. This system can aid the user by suggesting different visualizations for the data being analyzed, based on one or several categories selected by the user. To gain more knowledge about the Product and its users, four interviews with different internal stakeholders were conducted. Two product managers, one developer and one designer at the Company were interviewed. The questions were both about the recommendation system and a potential system for AI-generated insights.

The interviews were semi-structured with a list of predefined questions (see more about interviews in Section 4.4.4). There was no strict order to the questions as they were rather seen as something to get the conversation started, and often the interviewee would cover several questions without the interviewer needed to ask them. Follow-up questions were added during the interviews, as well for clarification or if something deemed extra interesting. During the interviews, audio was recorded, and notes were taken. Afterwards, the interviews were analyzed by summarizing the key takeaways from each interview.

First, the product managers were interviewed since they have a good understanding of the users, market and the Product as a whole. Questions asked to the product managers included:

- What do users currently think about the recommendation system?
- What is the purpose of AI generated insights?
- How do you see AI generated insights developing in the future and what could these look like?
- How do you think a user would interact with such a system?
- We will be looking at the level of automation for such a system. What level of automation do you think will be the most fitting for this type of functionality?

An interview with a developer was also conducted in order to get a further understanding of the Product. Here the questions were more about the implementation of the current recommendation system, and the challenges concerning implementation of this kind of functionality. The list of questions for the developer included:

- How does the current recommendation system work?
- What is your definition of AI generated insights?
- We will be looking at the level of automation for such a system. What level of automation do you think will be the most fitting for this type of functionality?
- What types of opportunities can you see with system-generated insights?
- What do you believe are the challenges with system-generated insights?

A UX designer was also interviewed as a step of getting a better understanding of the design, concepts and challenges considered when implementing the Product's recommendation system. Questions asked to the interaction designer included questions like:

- What was the background for the design of the current functionalities in the Product?
- Did you have any ideas that were left out during the development process?
- Why were they left out?
- How do you see AI-generated insights developing in the future?
- What level of automation do you think will be the most suitable for this kind of system?

### 6.2.3 Reflections

The first iteration was used to understand the current design space and how system-generated insights are currently used within products. It also allowed for interviews with stakeholders in order to get a better understanding of their view of system-generated insights, how a potential AI would work, and the level of automation within these types of systems.

Through the competitive analysis, a better understanding of system-generated insights in the current market was gained. What was found was that there seemed to be three different categories of system-generated insights, with regards to the level-of-detail for the different functions:

1. **Analysis of the whole data set.** This category was used by Qlik and Power BI Cloud, allowing for easy ad hoc analysis and the insights acts as inspiration as to where to begin analysing a data set. It also allows for suggesting potential visualizations or relationships that the user has missed themselves.
2. **Analysis of a subset of the data.** The subset referred to here could be based on choosing a specific subset within the data table, like within Qlik, or by suggesting insights based on the current visualization in focus. The software that have this category of insights are Power BI Cloud and Qlik.
3. **Analysis of a specific data point.** In Tableau and Power BI Desktop, the user can choose specific data points to explore and get insights on. These insights are very context specific.

From the software that was analyzed, an apparent trend could be seen. The system-generated insights are very user-driven. No insights are suggested automatically, and the user has to specifically request the insights when they want them. Many of the systems are also difficult to use, with many different panes and windows the user has to operate. In all of the analyzed programs, the user is not able to simultaneously work with the original visualization or dashboard, and the visualizations the system has produced. No program seemed to work much better than the rest, as many design choices have some negative consequences, and they all have to balance different consequences.

Takeaways from the internal stakeholder interviews included a better understanding of the Product and its users, but also the field of analytics in general and how system-generated insights belonged here. During the interviews, the design choices, implementation, challenges and opportunities regarding the recommendation system in the Product were brought forward, whose functionality is similar to the functionality in the research question, the AI-generated insights. Understanding this background helped in moving forward with the design process of AI-generated insights and helped with the refinement and update of the guidelines. How different levels of automation effected the systems was also discussed during the interviews.

One of the major challenges with more automation in a system (both the existing recommendation system, but also a system for generating insights) is how to provide the user with relevant results when the user is not able to show its intent. It was believed that there had to be some user input in order for the system to generate useful insights. One suggested way of get the context needed would be to use the user's intent when creating the visualizations in the dashboard. Another way could be to learn user preferences over time in order to provide the user with better and more relevant insights. Another, more user-driven approach already used in the Product, is a "More like this" functionality.

Another aspect that was mentioned to give better and more relevant recommendations or insights for the users was to incorporate domain-specific knowledge. Domain-specific insights could, for example, be generated from scripts today, but that requires someone knowledgeable within that specific field to create these.

For the future of the existing recommendation system, there were several comments about developing it to be able to tell why a recommendation had been generated, or explain the ranking of them. When the system provides a recommendation (or an insight in the case of the project), it would also be able to back it up by providing a reason for why it has been generated, for example, by explaining the relation in data or the prioritization in a list. Also, today the recommendations are triggered by specific UI-elements. However, a possible feature for the future and away towards more automation in the system, would be that the system would continuously run in the background, alerting the user of recommendations when there were any recommendations available.

It was also commented on that even though a user did not utilize the existing recommendation system; it did not bother the user. This feature lets the user choose whether or not to be assisted by the system.

The main target user group was originally only the persona Serena, doing ad-hoc analysis for a company. Through the competitive analysis and the stakeholder interview, it became clear that while there is not much research done within the field of system-generated insights, the current application of insights within analytics software is centered around a similar workflow as Serena, focusing on providing insight when creating the analysis. Thus, another target user group was introduced, in the form of a persona Ethan, the consumer. This was done to broaden the scope to see if there was an interesting intermediate between the analyst and the consumer.

### 6.2.4 Guidelines v.1

Based on the outcome of the competitive analysis and the interviews with the stakeholders at the Company, the set of guidelines were updated from the initial version in the pre-study.

#### Interactions with Agent

G1 **The AI should be easy to set up.**

This guideline was kept from the previous iteration.

G2 **The user should be able to dismiss the agent.**

This guideline was kept from the previous iteration. During the internal stakeholder interviews, it was mentioned the users who did not utilize the recommendation system in the Product were not be bothered by it. This seemed like a nice feature, that the user could be able to turn off or dismiss the agent if they do not want its assistant in the analysis.

G3 **The user should be able to invoke a dismissed agent.**

This guideline was kept from the previous iteration.

G4 **The user should be able to provide feedback about the insights, so the AI can learn user preferences over time.**

This guideline was kept from the previous iteration. During the interviews with the internal stakeholders, the subject of having an agent present relevant and useful insights were brought up several times. By letting the user give feedback to the system would allow it to learn and give better and more relevant results.

#### Communication With User

G5 **The system should communicate what the agent can do, as well as the agent's limitations.**

This is an updated guideline where the guidelines *G5 The system should communicate what the agent can do* and *G6 The system should communicate what the agent can't do* from the previous set of guidelines have been merged into one.

G6 **The system should communicate the status of the agent.**

This guideline was kept from the previous iteration, but was reformulated from *G7 The system should communicate what the status of the agent's work*. Status in this guideline refers to progress, success or error that the agent can communicate to the user.

G7 **The system should have sufficient error handling.**

This guideline was kept from the previous iteration.

#### User Experience

G8 **The insights should be based on domain-specific knowledge.**

This guideline has been added in this iteration, based on the internal stakeholder discussion about the more specific an insight can be, the more value can it give the user.

G9 **Interaction with the insights should not disrupt the user's workflow.**

This guideline was kept from the previous iteration, but was reformulated from *G10 The user's workflow should not be disrupted* in order to clarify the guideline.

G10 **The analysis should always be the main focus, and the AI should be in the background, assisting the user.**

A new guideline, based on the competitive analysis. It was concluded that when the analysis was not the main focus, it could cause frustration for the user.

#### Analytical reasoning

G11 **The system should provide an explanation for why the agent generated an insight.**

This guideline was reformulated from *G12 The systems should provide sufficient information for why it is doing something* in order to be more specific towards system-generated insights. Included here is for example when the system provides the user with a list of recommendations. If there is a ranking there, what is it based on? This guideline might be even more important if there is a more automatic system, where for example some insights have been chosen to be displayed compared to others, on order to provide transparency for the user to why those insights are deemed "the best" by the system.

G12 **The user should be able to investigate the origin of an insight.**

This guideline was reformulated from *G13 The user should be able to backtrack an insight* in order to indicate more clearly what the insight is about. Backtrack means here for example look at the data the insight is based on.

G13 **The visualization should clearly indicate what is generated from the data vs the system.**

This guideline was kept from the previous iteration.

G14 **The system should show the certainty of insights, if applicable.**

This guideline was kept from the previous iteration.

Guidelines that were removed from the initial set of guidelines was:

- Previous *G9 The system should notify the user if anything interesting can be found in the data* - This was removed because it seemed to more recommend a specific level of automation for the generated insights, which might more depend on the context of insights.
- Previous *G11 The agent should show results based on context* - This aspect felt like something needed to be explored more in this thesis.

## 6.3 Iteration 2: Concept generation

Based on the outcomes and developed guidelines from the first iteration, the second iteration was about generating and exploring different concepts on the topic of AI-generated insights within visual analytics. The iteration started with a design workshop at the Company, moving on to a concept generation through brainstorming. Finally, the concepts were evaluated in a design critique.

### 6.3.1 Ideation Workshop

An ideation workshop was performed with both designers and developers at the Company. The hopes were to get a broader perspective of the problem at hand, and possible solution, by involving different disciplines.

The focus of the workshop, ways of incorporating AI-generated insights in a visual analytics tool, was introduced to the participants, along with the purpose and target group. Also, the guidelines formulated in the previous iteration was presented, together with a list of four focus question to help the participants with the ideation:

- How do we integrate insights within the context of the visualization without disrupting the user's workflow?
- What does the interaction between the user and the insight look like, and vice versa?
- How do we build trust?
- How do we explain the insight?

The workshop was divided into two parts, in the first part the ideation method Sketchstorming was used (see more about the method in Section 4.4.2). The ideation method Design Charette (see Section 4.4.2) was used in the second part. Figure 6.6 shows two images from the ideation workshop held at the Company.

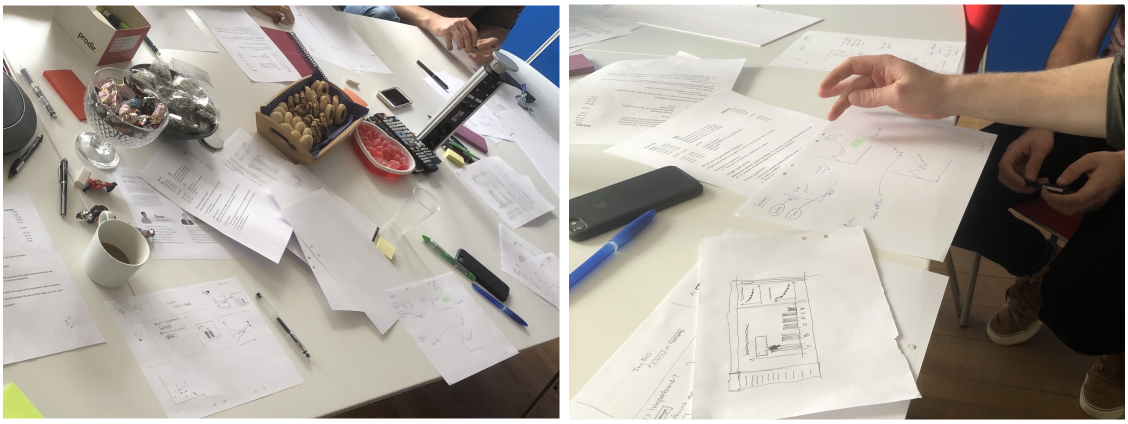


**Figure 6.6:** Images from the ideation workshop held at the Company with designers and developers

The workshop generated some ideas on how AI-generated insights could be incorporated, with different levels of automation. A list with some of the more or less incomplete ideas from the workshop is presented below:

- If there is a very automatic system, use "better to ask forgiveness than permission" and enable the user to easily revert the system's actions that has been automatically applied.
- Previews of the insights could be a way of helping the user see the insight without actually going through with selecting the insight and add a graph to the dashboard.
- Have an AI that the user can toggle, or even better, an AI so that the user can decrease and increase level of automation themselves.
- Insights can be used as hooks into the data, and be more about presenting a very fast, effective and inviting way of creating the first visualizations in a dashboard.

### 6.3.2 Creation of a Design language

The design workshop generated ideas on how to incorporate an insight into the user's flow, how to alert a user of available insight and how an insight could be presented. The input from designers and developers helped in bringing some clarity into what an insight could be and how it could work in its context, but it also leads to more confusion and more options on how a system-generated could work and be interacted with. Many different design challenges emerged, and different aspects of system-generated insights seemed to affect the design choices in the system.

While conceptualizing and exploring potential ideas for this phase, it became apparent that the problem at hand grew in complexity as more and more aspects of the system were considered. As mentioned above, many of the solutions to some of the design challenges, conflicted with solutions of other aspects of the design. While trying to communicate regarding these challenges and the different aspects of the design, it became difficult to get ideas across.

As different ideas were explored, categories of challenges and design aspects emerged. In an effort of trying to aid in communicating these challenges and organize the different possible aspects of a system-generated insight, a design language was formed. There was a need for formalizing defined terms and concepts for communication purposes, as well as organizing the different design ideas in order to be able to move forward in the process. It also helped to make the application of system-generated insights more tangible by defining dimensions which affect the design as a whole. The language started as a few key categories, such as the type of user, level of automation, and transparency of the insights. As time went on, and more ideas were explored, new aspects were added to the list of dimensions.

Because many of the different aspects of an insight seemed to be tangled together, creating ripple effects in the form of contradictions in the design or new design challenges when they were changed or added, this type of organization of thoughts was very helpful. The creation of a design language and the effort of listing and categorizing the different dimensions did not solve all problems, but helped on the way. The different properties of each dimension in the design language also helped with ideation, as it was possible to either describe ideas using the properties, or skew the properties to come up with new concepts.

Below are the different dimensions listed, sorted into the different topics Context, UI, Trust, and Position. For each dimension, some identified properties are presented.

#### Context

The context is the mental focus of the user, sometimes perceived as the digital focus of the program. This can be applied both to the information in the insight, but also to what type of user will be using the system.

**User** can be *Consumer*, *Analyst* or *Both*. An analyst creates visualization and dashboard, and analyzes data, while the Consumer receives an already existing dashboard that someone else has created. Depending on whether the user is a creator or consumer of content in dashboard, it might affect the interaction with a system-generated insight.

**Type of insight** was in this thesis project limited to *Correlations*, *Comparisons*, *Outliers* and *Forecasts*. This list of properties for this dimension can be expanded if needed.

**Level of insight** refers to the context from which the insight is derived. The different properties listed here are *Data point specific*, *Visualization specific*, *Categorical* and *Whole data set*. The context can sometimes be perceived as the digital focus of the program, or the mental focus of the user (depending on the level of automation, system-driven vs user-driven). It could be described as which data the insight is based on. For instance, it can be the *Whole data set* that the system-generated insight is based on, as seen in some software mentioned in 6.2.3. *Categorical* means that the insight is partly based in data that belongs in a existing visualization, for instance a correlation between two variables/columns (and user has already shown interest in one of the columns by using it in a visualization) and could be seen as an insight by the system. A forecast in an existing visualization can be seen as an *Visualization specific* insight, since it is based on the data connected to specific visualization. An insight in the form of a detected outlier could be seen as a *Data point specific* insight.

**Scope of insight** refers to how specific or general an insight is, and in this case there were two properties identified for the thesis, *Specific* and *General*. This dimension about whether the insight is connected to a specific part of dashboard, for example a specific visualization, or not. This compared to a more general insight that is still based on the existing data being visualized in the dashboard, but cannot visually be connected to a specific visualization.

**UI**

This is more user interface specific dimensions, such as UI elements or interactions.

**Automation** is more of a scale and is used as a dimension to assist in looking at the level of automation within a system of system-generated insights. Different part of the system can be more/less automatic, while others are fully automatic for example. The identified properties here were *User-driven*, *Assistive* and *Agentive*. To read more about assistive and agentive technology, see section 3.3.

**Intrusiveness** is a dimension mostly acting as a reminder to think about the workflow for the user and how a system-generated insight can be incorporated there, where the properties were named *Very disruptive*, *Sometime demands action* and *None*.

**Alert** refers to how to inform the user that there are available insights by *Push noti-*

*fication*, *Notification icon*, *Button*, *Context based button* and *None*. This dimension is both about the system alerting the user, but also how the user can understand there are insights, without the system explicitly notifies the user. For example, in more user-driven scenarios, buttons can be used as a place for the user to access insights. In this dimension, several properties can be applied to the design.

**Format** refers to the format of the insight when it is presented to the user, for example *Modal pop-up*, *Preview*, *Modeless pop-up*, *List* or *Single element*. For this dimension several properties that can be applied to the design.

**Manifestation of insight** is about how an insight is shown. Are they applied straight away, so they are only consisting of a visualization, or is the user presented with an explanation and preview before it is applied? Some possible properties for this dimension were *Picture of another visualization*, *Applied straight to visualization*, *Text based*, *Only numbers*, and *Animation*.

**Actions** refer to the possible actions the user (or system depending on level of automation) can do with the insight, for example *Add to visualization*, *Add to dashboard*, *Change visualization*, *Custom actions* and *No actions*.

**Interaction** is what type of interaction will be used to interact with the insights, for example *Hover*, *Click*, *Mark*, *Drag and drop* and *No interaction*. This dimension is relevant if the insight allows for actions to be taken.

**Can be hidden** is whether or not insights can be toggled to show/hide, for example the applied insights or any insight triggers in the dashboard, *Convenient*, *Not convenient* and *Not applicable*.

#### Trust

Different dimensions for the transparency in the system and how to create trust.

**Transparency** is dimension for a scale of perceived transparency within the system. The transparency can be modified in many places in the system and be different in different parts of the system, *Not transparent*, *A little transparency* and *Very transparent*.

**Certainty** is maybe more relevant to insights in the form of forecasts for example, but it can also be applied to each insight and the system's certainty of the correctness of the insight, identified properties are *Shown*, *Not shown* and *Provided, but hidden*.

**Traceability** is the ability of the user getting an explanation for what the insight is based on, allowing them to decide for themselves if they are able to trust the insight or not, for example *Deep dive*, *Short presentation of numbers*, *Access to the underlying data* and *No traceability*.

**Position**

Specific to the manifestation of the insight, and the position of the different parts of an insight's anatomy.

**Position of insight** refers to the position of the insight itself. Because of the nature of insights and the many different forms they can take on, this can either refer to a visualization being changed automatically or possibly the position of an insight card. Depends on the manifestation of the visualization, but some identified possibilities are *Inside the visualization*, *Next to the visualization*, *In margins of visualization*, *Inside dashboard*, *Next to dashboard*, *In-place (automatically added)*, *Separate from dashboard* and *Not applicable*

**Position of results** is a dimension also affected by the manifestation of the insight. The result is referring to what happens after the user has interacted with an insight, for example where it could be added, *Inside the visualization*, *Next to the visualization*, *In margins of visualization*, *Inside dashboard*, *Next to dashboard*, *In-place (automatically added)*, *Separate from dashboard* and *Not applicable*. If there are no actions to be taken, then there will not be any results for example, and then this is not applicable.

**Position of trigger element** is the position of the actual trigger button for making an action from an insight. This depends on the type of action taken and if there are actions at all, the different properties were *Inside the visualization*, *Next to the visualization*, *In margins of visualization*, *Inside dashboard*, *Next to dashboard*, *In-place (automatically added)*, *Separate from dashboard* and *Not applicable*.

This set of different terms regarding the system-generated insight (its possible actions, interactions, position, and triggers) was created as a help to understand the different possible aspects of a system-generated insight. It is not a formal design language as it was mainly used for communication and ideation of different concepts within the context of this thesis. What was realized, based on this, was that system-generated insights could be quite complex, especially without fixating some of the properties. Some properties would even contradict each other, making it hard to find a design pattern that would work for all types of insights. For example, a specific insight could fit within the context of a visualization, and be automatically added in-place, since that type of insight would have a connection to visualization, compared to a general insight that would not be able to be applied in the same way.

### 6.3.3 Concept generation

The different ideas from the workshop, together with tweaking of properties mentioned in the previous section, would inspire the upcoming concept generation. This lead to the identification of six different concepts for system-generated insights, listed in more detail below.

#### 6.3.3.1 The "List View" concept

In the first concept, the idea was that the generated insights are presented in an "insight panel" on the side of the dashboard, see Figure 6.7. Here several insights can be displayed at the same time, and easily accessed by the user. The hope was that this layout would not disrupt the workflow of the user, with the panel not covering the dashboard, letting the user explore the insights but still see the content of the dashboard.


**Figure 6.7:** The "List View" concept

#### The "Popup" concept

In this concept, insights, together with their explanations and actions, were presented in a pop-up after the user implicitly had told the system to generate insights by clicking "Get insights," see Figure 6.8. The pop-up would provide space for an explanation of the generated insights and a designated place for the user to explore the available insights. One negative aspect was that the pop-up would cover the dashboard and the existing visualizations. The pop-up was believed to be able to disrupt the user's flow, and the question was if this were to be balanced by the large space for managing the insights. This layout for presenting insights was a common pattern for the existing insight functionality in different software.

#### The "Preview in Visualization" concept

In the third concept, the generated insights were indicated by the small light bulb icons in the visualization, see Figure 6.9. From here, the user could choose to take a closer look at the generated insights by clicking on the light bulbs and open a small explanation of the insight. If there were several insights based on the same data point(s) in the plot, these could be accessed from the same place in the visualization.

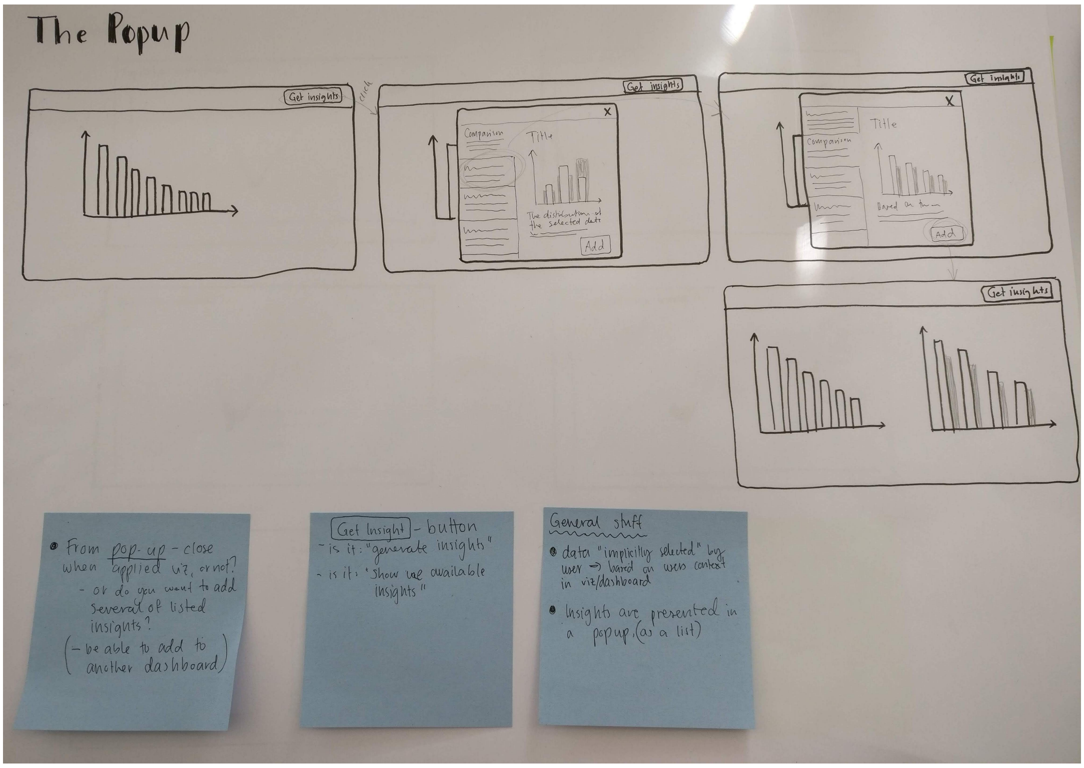



**Figure 6.8:** The "Popup" concept

This very contextual highlight in the form of a light bulb beside the plots would give the user an effective way of understanding where the generated insight was connected to the visualization. Different possible actions for the insights could be accessed directly in-place in the visualization, for example, adding a new graph, or remove an outlier.

### The "Spellcheck" concept

Another concept, which was inspired by spellcheck in Google Documents, can be seen in Figure 6.10. A highlight would alert the user that something interesting had been found in the visualization. If this was deemed interesting, the user could access the insight by clicking on the highlight. Here a couple of options would appear, for example, "Explain the decrease" in a dip in a line chart or "Apply" for insight in the form of a forecast at the end of the line chart.

### The "Spot the Bot" concept

The fifth concept came from the idea that the insights could be generated from a chatbot, see Figure 6.11. The bot would alert the user if any insights had been found and provide explanations for them along with possible actions for the user to take, but stay hidden the rest of the time. Here an idea about enabling the user to provide input for the system emerged.

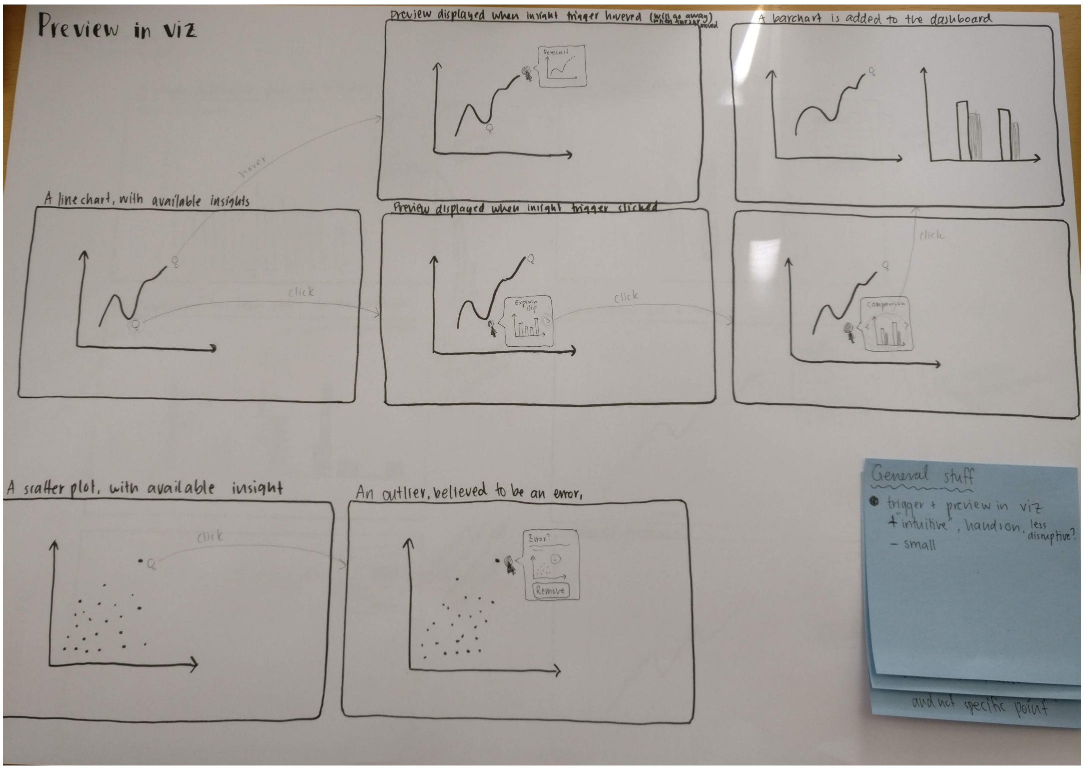



**Figure 6.9:** The "Preview in Visualization" concept

**The "Summary" concept**

This concept was about summarizing essential insights, explaining in text what is interesting in the visualizations, see Figure 6.12. In that text, parts could behave like hyperlinks, enabling the user to show the results of the generated insights in the dashboard.

### 6.3.4 Evaluation through a Design Critique

In order to evaluate the different concepts and be able to move forward in the design process, a design critique was performed (see Section 4.4.4). One product manager and two UX designers from the Company attended and gave their feedback and expert opinions on the six different concepts.

At the beginning of the design critique session, the background and purpose of the session was presented to the participants. Then each concept was presented along with a description, and then discussed for 5 - 10 minutes. The outcome consisted of a list of pros and cons for each concept, along with some comments and possible issues with the concept.

### 6.3.5 Reflections

Even though the design workshop did not generate concrete concepts, it served as an inspiration to get going and to figure out different aspects of system-generated

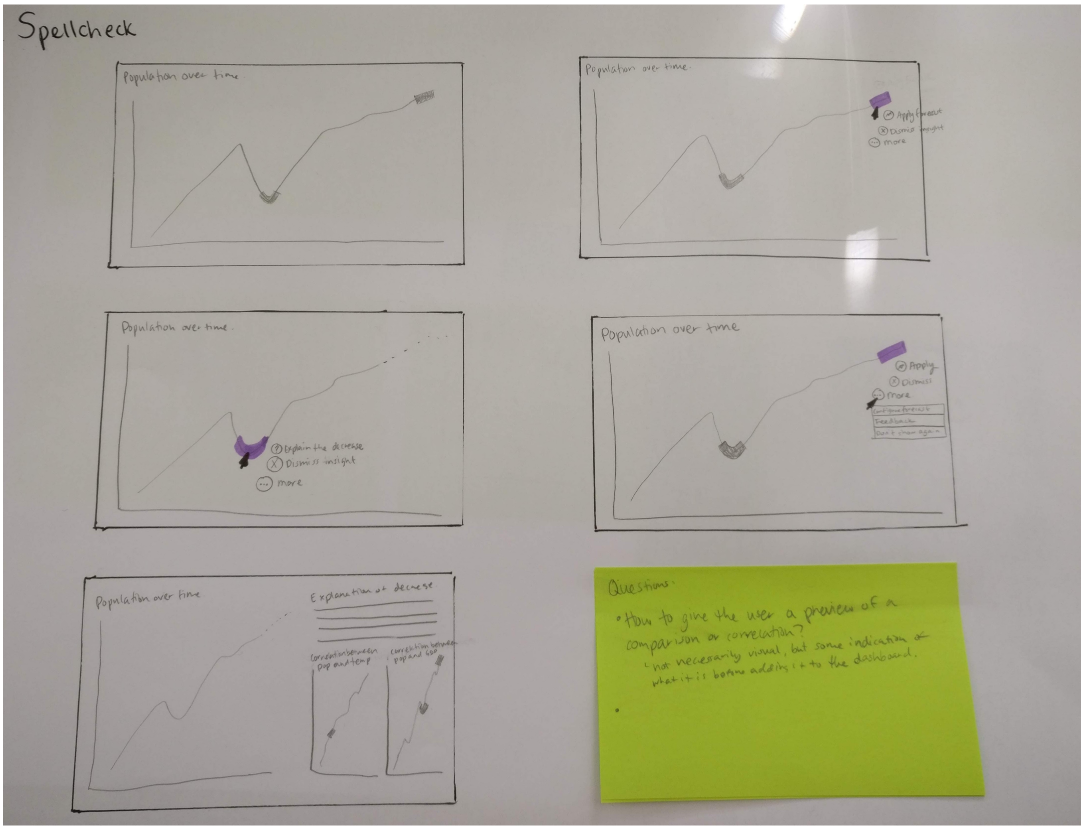



**Figure 6.10:** The "Spellcheck" concept

insights to focus on. Before this stage, what a system-generated insight was and what it could do was a bit unclear. However, the design workshop with designers and developers at the Company helped in order to get a better understanding of the different issues and possibilities connected to applying system-generated insights. The rough ideas from the workshop lead to the creation of a design language that later assisted the design of the concepts. This was done by helping to extract different properties of the system-generated insights that could be skewed in order to explore different possible concepts.

Even though some issues regarding the concepts were known beforehand, it was interesting to see the reactions and feedback. After the design critique, all different concepts were listed along with summarized notes about their possibilities and issues that came up at the design critique:

**The "List View" concept**

- A list provides a designated place for available insights and it would not cover the dashboard.
- It is possible to show many insight at the same time, making it easier for the user to get an overlook of available insights.
- The list could be a place where to manage insights, for example filter and sort

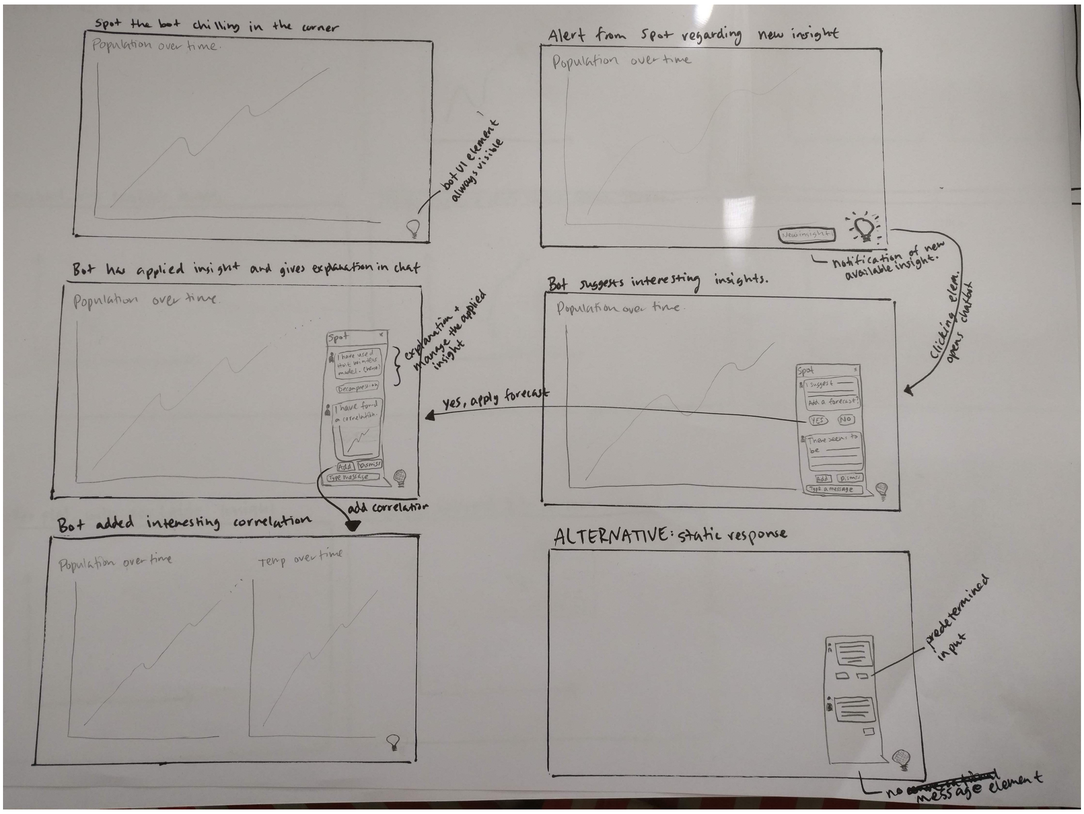



**Figure 6.11:** The "Spot the Bot" concept

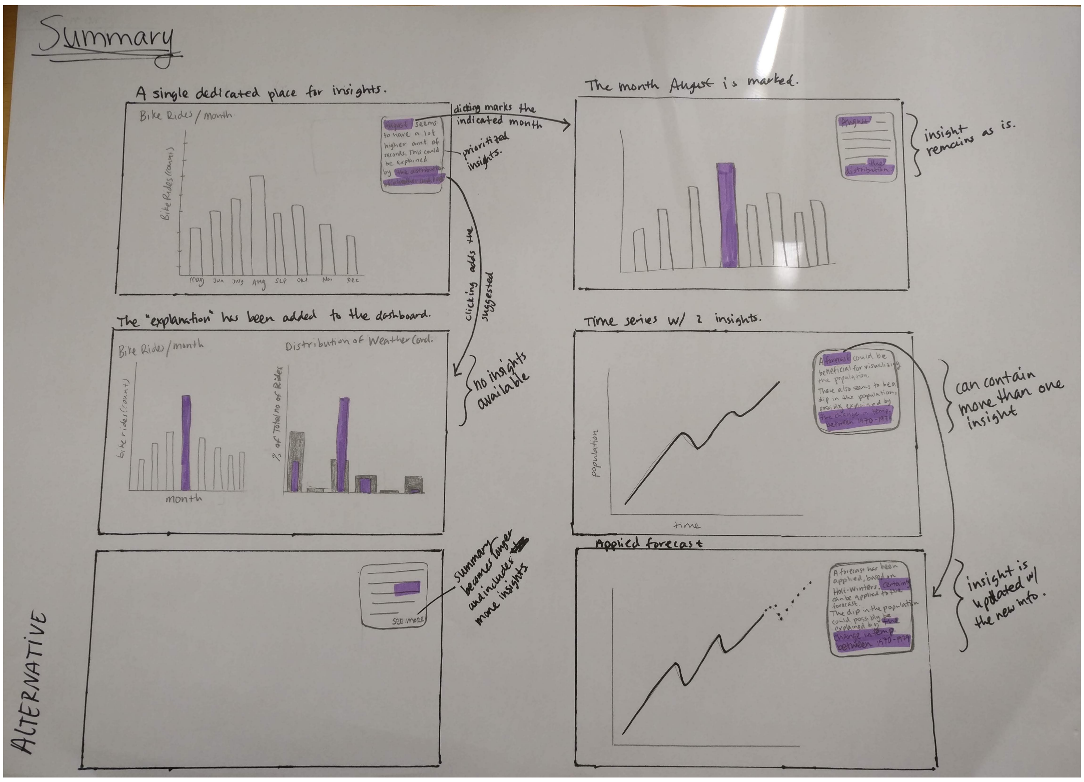



**Figure 6.12:** The "Summary" concept

them.

- It is difficult to provide the user with a context for an insight when the insights are in the list and have no connection to the visualizations in the dashboard.
- How can the user be notified from the visualizations that there are available insights?
- The list could benefit from domain-specific insights, providing relevant insight for the user, otherwise the risk is a lot of scrolling.

**The "Popup" concept**

- A positive aspect is that available insights could be accessed easily in the popup.
- A lot of space for an explanation of an insight.
- A con for the popup is that it covers the dashboard.
- A popup is quite disruptive and might disturb the workflow of the user, and draw attention from the analysis, but this could also be an advantage if attention on insights is preferred.

**The "Preview in Visualization" concept**

- One advantage is that the previews are very hands on and intuitive, and provide the context of an insight to the user.
- They are less disruptive than a popup for example.
- A preview is a powerful way of quickly showing the user what the insight is about.
- One negative aspect is the size, it would be hard to present information about the insight in a clear way, since it it often requires some text to explain an insight.
- The small space would also require a strong ranking of the generated insights, in order to just present one insight at a time.
- What type of actions would a user take from these previews?

**The "Spellcheck" concept**

- It is quite minimalistic and does not take much space.
- It clearly indicates where the insight can be found, providing the context of an insight for the user.
- The small size is one of the negative aspects since it very hard to understand what the insights is about.

**The "Spot the Bot" concept**

- The advantage is that it can be flexible and the conversation only opened when the user wants to.

- A lot of features could be combined in the same ui element, such as explanations of insights or the user can be able to provide feedback on the relevance on the insights.
- A known pattern in the form of a chatbot could be a way of getting a beginner started.
- A negative aspect is that a chatbot have to be really good in order to be perceived useful, people seems to be spontaneously annoyed by them.
- The user might think the bot is smarter than it is actually is, ending up disappointed.
- It would be really hard for the user to access and sort between insights if there are a lot of insights generated.

**The "Summary" concept**

- It is a way of providing the user with the top ranked insights, but that is also one of the negative aspects, that a relevant ranking of insights is hard to accomplish.
- The hyperlinks can give a good context for the insights, taking the user to the insight from the text explanation.
- One negative aspect is the limited number of insights that can be displayed.

### 6.3.6 Guidelines v.2

After the second iteration and its design activities, the guidelines were refined and updated again. In this set of guidelines, the Analytical reasoning category has been divided into two, resulting in a new category called Transparency.

**Interactions with Agent**

G1 **The user should be able to dismiss the agent**
This guideline was kept from previous iteration.

G2 **The user should be able to provide feedback about the insights, so the AI can learn user preferences over time**
This guideline was kept from previous iteration.

**Communication With User**

G3 **The system should communicate what the agent can do, as well as the agent's limitations**
This guideline was kept from previous iteration.

G4 **The system should communicate the status of the agent**
This guideline was kept from previous iteration.

**User Experience**

G5 **The agent should be configurable, and domain-specific knowledge should be put into the agent and it should be told how to determine when domain specific knowledge or general knowledge is the most relevant**

This is reformulated from *G8 The insights should be based on domain-specific knowledge* from previous iteration.

G6 **Interaction with the insights should not disrupt the user's workflow**

This guideline was kept from previous iteration. Also, G10 from previous iteration is incorporated in this one, (*G10 The analysis should always be the main focus, and the AI should be in the background, assisting the user*), since G10 was believed to be a way of achieving G6.

G7 **If an insight will change a visualization or add a new visualization when applying it to the dashboard, a visual preview of that change should be shown**

This guideline was added in this iteration as way of handling insights when applying them would manipulate existing visualization.

G8 **Insights that have been applied should be able to be removed easily**

This guideline was added in this iteration as a way to give more control to the user if an applied insight would accidentally manipulate the dashboard in an unwanted way.

**Analytical reasoning**

G9 **The system should provide an explanation for what is presented in the insight and provide the context of the insight**

This guideline was added to the set of guidelines since a major part of the insight is its explanation of what it is about, and also how it is connected to the data being analyzed.

G10 **Visual insight cues could be shown automatically by the system, without the user having to interact with the insight function**

This guideline was added to the set of guidelines because the outcome of the design critique, where highlights in the visualizations could be a powerful way of indicating available insights within the visualization.

**Transparency**

G11 **The user should be able to see how and why the agent ended up with a specific insight**

This guideline was reformulated from *G11 The system should provide an explanation for why the agent generated an insight* and *G12 The user should be able to investigate the origin of an insight.*

G12 **The visualization should clearly indicate what is generated from the data vs the system**

This guideline was kept from previous iteration.

G13 **The user should be able to access all of the insights gathered by the system, in one way or another**

This guideline was added to this set of guidelines since it was mentioned during the design critique that ranking insights in a good way was difficult, but by giving the user access to them all could be a way of not letting the user miss any of the generated insights.

Guidelines that were removed in this iteration was:

- Previous *G1 The AI should be easy to set up* - This aspect of system-generated insights started to become a bit outside the scope of this project.
- Previous *G3 The user should be able to invoke a dismissed agent* - This guideline was seen as redundant because G1 could cover the this case
- Previous *G7 The system should have sufficient error handling* - This guideline was incorporated in the new G4, including error as part of the agent's status
- Previous *G10 The analysis should always be the main focus, and the AI should be in the background, assisting the user* - This guideline was believed it could go under the the new G6 instead
- Previous *G12 The user should be able to investigate the origin of an insight* - This guideline was believed it could go under the new G8, and that it also was a bit too specific to fit in the set of guidelines
- Previous *G14 The system should show the certainty of insights, if applicable* - This guideline was also deemed to be too specific

## 6.4 Iteration 3: Low-Fidelity Prototyping

In this iteration, three low-fidelity prototypes were created based on the more promising ideas and features in the concepts from the previous iteration. See Section 4.4.3 to read more about different types of prototypes. After this, usability tests were performed in order to evaluate the different concepts, see Section 4.4.4. The outcome of the evaluation was then used to update and refine the guidelines. Due to the global pandemic COVID-19, this phase was affected by the need for social distancing [61][62], forcing parts of the testing to be done remotely.

### 6.4.1 Ideation

Based on the feedback from the design critique, the plan was to merge the most promising concepts and features from the previous iteration into three designs realized as low-fidelity prototypes. After some more brainstorming, three ideas emerged: the "List View" concept, the "Preview in Visualization" concept, and the "Spot the Bot" concept. The first two ideas were the ones believed to have real promise, while the chatbot concept was added as a wild card. The idea with the wild card was to use it as a way to broaden the design space, even though it was believed to not be a good solution for how to apply system-generated insights.

### 6.4.2 Prototyping

For the low-fidelity prototypes, the prototyping tool Balsamiq was used. The argument for this was that Balsamiq would allow the look of prototypes to resemble sketches and not a finished product. The lower fidelity of the prototypes would let the test participants focus on the bigger picture and give feedback on the concepts, not on details or colors. The idea was to use the same scenarios and tasks for the three prototypes in the testing. The prototypes were used as a way to explore concepts of system-generated insights further. However, with a set scenario, the prototypes focused on testing the guidelines produced in the last iteration.

Guidelines *G1 The user should be able to dismiss the agent* and *G2 The user should be able to provide feedback bout the insights so that the AI can learn user preferences over time* were implemented in each prototype to test different solutions. *G4 The system should communicate the status of the agent* is also implemented. *G6 Interaction with the insights should not disrupt the user's workflow* was one of the main focuses for these prototypes, as the general workflow was looked at as a whole. The user experience and analytical reasoning guidelines were tested on a broad scale. For example, *G7 If an insight will change a visualization or add a new visualization when applying it to the dashboard, a visual preview of that change should be shown* was explicitly implemented in each prototype to test out different ways of doing this. Different ways of explaining the visualizations were implemented, as per *G9 The system should provide an explanation for what is presented in the insights and provide the context of the insight.* All prototypes automatically show insight cues, even if there were slight differences, as per *G10 Visual insight cues could be shown automatically by the system, without the user having to interact with the insight function.* Because of feedback from the design critique, three different ways of accessing a longer list of insights were implemented, as per *G13 The user should be able to access all of the insight gathered by the system, in one way or another.*

A description of the different prototypes can be seen below:

**The "List View" concept**

The idea for this concept was to have the generated insights in a panel on the side of the dashboard. The focus of this prototype was the lack of disruption to the workflow (G6), by allowing the user to see both the insight panel and the dashboard at the same time. Another focus was the visibility of insights (G13), as it allowed for easy access to many insights.

Each insight in the panel is presented with an explanation and a visual representation of the insights, as per G9 and G10. From there, the user can select an insight, and the insight will be highlighted in red both in the list and the plot, as seen in Figure 6.13. The sidebar can be hidden and thus letting the visualizations fill the entire dashboard, again trying not to disrupt the user's workflow. From the list, the user can manage the different available insights by scrolling the list or performing the actions provided by the insight. In order to notify the user that there are insights available, highlights in the plot would appear in place where the insights originate, according to G10. Also, a notification in the top right corner tells the user that there are available insights in the list.

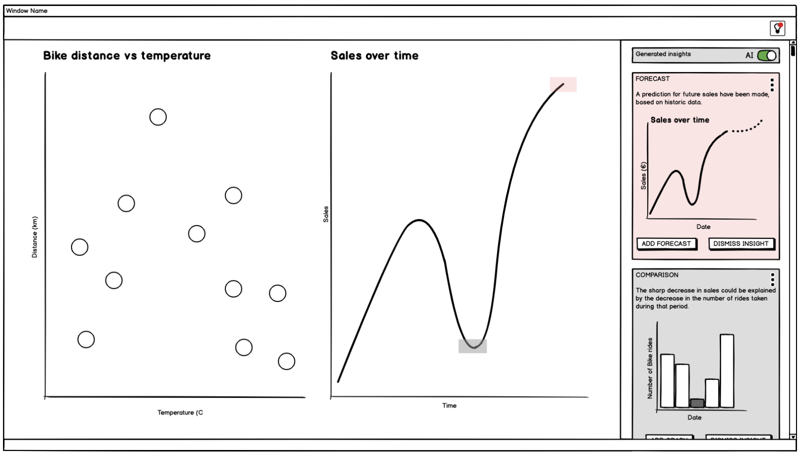



**Figure 6.13:** A wireframe of the "List View" concept illustrating a highlighted insight

In Figure 6.14 has the selected insight, in this case a forecast, been added to the line chart in the dashboard. That insight card is then removed.

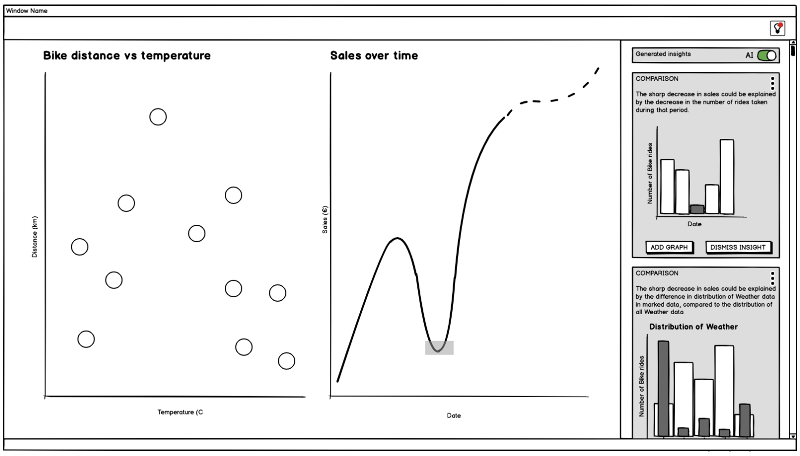



**Figure 6.14:** A wireframe of the "List View" concept illustrating an applied insight in the form of a forecast

In "List View", the user is able to dismiss all of the insight highlights from the insight panel. This removes all of the gray highlights, until the AI is invoked again. This is one example of how to implement G1.

### "Preview in Visualization" concept

In this concept, an insight would be displayed in a small popup right next to the visualization it refers to (see Figure 6.15). The idea was that this would help the user understand what the insight was about and why it was generated and how it is connected to the data presented in the dashboard. The actual insight is very similar to "List View", with an explanation and a visual preview of the insight, as per G7 and G9. The insight cues are also still present, according to G10. However, the user can dismiss the insights per visualization, which is a variation of how one can implement G1. This concept is another example of an implementation of G6,

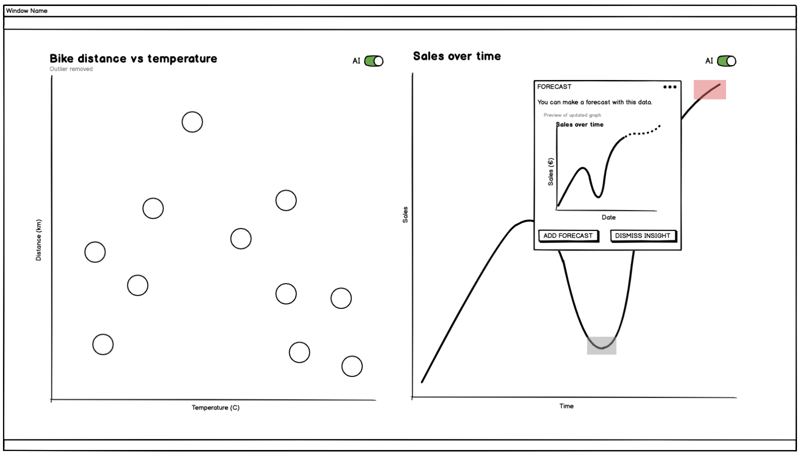



**Figure 6.15:** A wireframe of the "Preview in Visualization" concept

not disrupting the user's workflow. Since the insight is within the context of the visualization, the idea is that it could be adopted as being part of the analytic process, as the user is analysing a visualization.

#### "Spot the Bot" concept

This concept was the wild card, and despite not being very well received at the design critique, it was kept as one of the concepts to see if any positive features could be derived from it. Here insights were accessed through a conversation with a chatbot, see Figure 6.16. The user would be alerted when there were any available insights in the dashboard, and could also give input to the chatbot. The user would still receive visual previews and explanations of insights, as with the previous concepts. However, it would allow for a conversational UI, where the user could ask further questions regarding the explanations. It would also allow for more freedom for the user, where they can dismiss the AI through conversation (G1) or give feedback (G2) by providing it in the chat. It would be more difficult to follow G13, on the other hand.

### 6.4.3 Usability testing

The three prototypes were then evaluated through usability tests conducted with six test participants. These participants were students, the majority of which were in the field of interaction design. Some of the tests were performed remotely due to the pandemic COVID-19. The participants were presented with an interactive prototype, along with some background information about the topic and the prototypes. The same scenarios and tasks were performed for each of the prototypes,

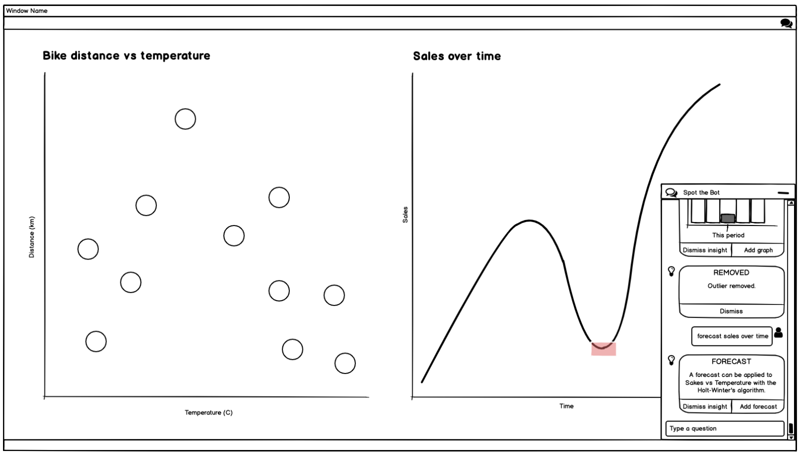



**Figure 6.16:** A wireframe of the "Spot the Bot" concept created in Balsamiq

with a set of questions to answer after completing all tasks for each prototype. They were encouraged to think aloud when they performed the tasks, and the screens and voices were recorded.

The tasks for each prototype are listed below:

- Remove the outlier in the “Bike distance vs temperature” visualization
- Add a forecast to the line chart in “Sales over times” visualization
- Explore if there is a possible explanation for the decrease in “Sales over times” visualization
- Turn off the automatic generation of insights

The closed questions to be answered after each prototype were:

- Do you think that these insights could help you in the process of analyzing data? (Think more of the presentation of the insights, than the content of them in these specifics examples). On a scale from 1: Not helping to 5: Very helpful
- Was it easy to understand if there were any available system-generated insights available? On a scale from 1: Not easy to 5: Very easy
- How disruptive did you think the presentation of the insights was? (both the “highlights” in the plot for indicating available insights and the insights themselves). On a scale from 1: Worse than Clippy to 5: It was very smooth
- How easy was it to understand the context of the insights, for instance, how they were connected to the existing visualizations? On a scale from 1: They were connected? to 5: Easy to understand!
- Do you think the level of automation of the generated insights was appropriate for these tasks? (level of automation from fully automatic to fully user-driven) On a scale from 1: More system-driven to 5: More user-driven

These closed questions were followed by the open questions "Were something you thought was unclear?" and "Do you have any suggestions for improvements?".

For each concept, the results were analyzed and listed as a summary of the concept, together with lists of pros and cons.

#### "List View" concept

**Summary of concept:** This concept was the favorite for most of the participants in the usability test. It seemed like a promising concept with its designated place for the insights allowing a good overview of them. The list also provides the flexibility of adding more functionality; for example, it allows for space for more detailed information about an insight. It was also suggested to have a search and filter function helping the user if there were a lot of insights available. One of the challenges here is connecting the insights in the dashboard to the ones in the list in a clear way.

Advantages of the design:

- Easy to see all the available insights in the list.
- Flexibility with a designated place for the insights, and additional functionality were suggested that could fit in that area as well.
- The list does not cover any of the existing visualizations in the dashboard.
- The least disruptive alternative among the three concepts according to the users participating in the test.
- Using color to connect the insight in the list, to insight highlight in visualization, made it more clear to the user where in the dashboard an insight originated.

Issues or problems with the design:

- It was harder for the participants in the test to connect highlight of insight in a visualization to its insight in the list on the side of the dashboard.
- It was not clear how many available insights there were connected to one data point/interval in the plot.

#### "Preview in Visualization" concept

**Summary of concept:** The users appreciated the concept for providing the context of the visualization. Being able to take some action in conjunction with the actual insights was also mentioned to be a useful feature. The big challenge for this concept was the limitation in space for the actual insight. Another challenge was how to present the relevant information and actions for the insight in-place in the visualization. It was suggested that this concept could be merged with the “List View” concept allowing extra space for further explanation and information about the insight in a panel on the side.

Advantages of the design:

- Provides the context of the insight, making it clear what kind of insight it was and were in the dashboard it originated.
- For the task of deleting an outlier for example, this concept was appreciated for being able to do that in-place.

Issues or problems with the design:

- The small space available for the insight (the popup) provided some limitations on how much information can be given in conjunction to the insight in the visualization.
- If there were many insights connected to one place in the visualization, it would troublesome how to display them all.
- It was also unclear if there were more insights than just the one displayed, since there was nothing on the presentation of the insight that indicated that there were more available.
- It as commented on that the concept was a bit disruptive since the popup for the insight covers a part of the dashboard.

**"Spot the Bot" concept**

**Summary of concept:** This concept was the wild card. Despite not being very well received at the design critique, it was kept as one of the concepts to see if any positive features could be derived from it. The idea of a bot was also the least favorite among the participants of the usability tests. However, there were still some promising features that could be incorporated in the final design in the high-fidelity prototype.

Advantages of the design:

- While they thought the idea of having to ask questions was “annoying” and sometimes confusing, a few users pointed out that it would be a good addition to a more “click-bases” system. For example, there could be a possibility of asking questions within the context of the insights or maybe searching through a list of insights.
- Text based queries are good for obscure and unusual questions.
- The system is good for when you know the question you want an answer to.

Issues or problems with the design:

- The chat window covered part of the dashboard, and when a comparison was added in the prototype, the window would cover part of the visualization and it was easy to miss certain things in the dashboard.
- Dismissing the agent was thought to be done with the minimize button on

the chat window. However, the idea was that it would be possible to either minimize the chat or dismiss the agent, and thus those were treated as different things. The users did not think it was intuitive to dismiss the agent with the same button they used to invoke the agent.

- Part of the interaction was focused around the idea that the user would ask questions. Most users commented that it was a problem that they did not know what to ask, and it was difficult to know what the possibilities and limitations were. Thus, some users wanted more automation and did not like the fact that they had to guess what the appropriate question/command would be.
- The users were unsure what to write. They did not know what would be appropriate, and could end up having to try many different combinations if the system did not understand their query (if the system is not good at processing natural languages).
- Getting to know about the model used for the forecast without any inquiry is perhaps a little too much information straight off the bat.

The results were also analyzed and categorized in a affinity diagram in order to discover overall trends in the test results, seen in Appendix A.

### 6.4.4 Reflections

A summary of some general outcomes from the tests can be seen below:

- The "List View" concept was the most popular, according to the closed questions after each concept.
- Almost all participants in the test mentioned that a combination of the "Preview in Visualization" and the "List View" concept could be a possible design for the system-generated insights.
- The "Preview in Visualization" concept illustrated that providing context by presenting an insight in the plot, right where the insight were connected to the visualization and data insights based on, was very powerful in terms of understanding the context of the insight.
- By having some kind of visual preview to the insight helped in providing a context for the insights, especially when the plot in insight looked like the original visualization. This was more difficult when the suggested visualization did not look like the original, for example if an insight consisted of a bar chart explaining something in a line chart.
- The level of automation could have been more user-driven, according to the closed questions after each concept.
- A very general thing derived from the results was to be clear! Give feedback, possibly categorize by color and make it as clear and transparent as possible.

### 6.4.5 Guidelines v.3

The guidelines were adapted and a longer explanation was added to each guidelines, with more motivation or examples from the generated prototypes.

#### Interactions with Agent

G1 **Consider letting the user be able to dismiss and invoke the agent**

Depending on the user and context, the insight system would probably be used in different ways. A user might sometimes want the assistance of generated insights, and other times see no need for it. Here the user experience could benefit from enabling dismissal (and then invocation) of the agent, resulting in giving the user a sense of control. This could be especially important in a more automatic insight system, allowing control back to the user. The dismissal could be done globally, or maybe even locally, letting the user dismiss the agent with regard for specific visualizations. The guideline is also still based on the previously mentioned theory, such as G8 in human-AI guidelines [28]. In the prototypes for this iteration, dismiss and invoke was interpreted as hide and show the automatic highlights. However, this can be done in many different ways.

G2 **Consider letting the user be able to provide feedback about the insights, so the AI can learn user preferences over time and give more value to the user**

Letting the user be able to give feedback about the generated insights to the system could help the system in providing more correct and relevant insights over time. The guideline is both based on G13 and G15 from the human-AI guidelines, as well as the findings in the usability test. Helping the system learn user preference over time could for example help with the prioritization and ranking of the generated insights when presenting them to the user. If the AI is accurate and relevant, then it will probably increase the user's willingness to be assisted by the AI.

G3 **Consider letting the user to be able to set up the "insight agent" and be able to customize it**

The guideline was added as it would allow for more domain-specific and personalized insights. By being able to set up the agent and adopt it to different domains, industries and user it is believed that the agent and its generated insights could give more value to a user. This also partly refers to the recommendation axes user preferences and competencies in the VisRec system [9]. Allowing the user to customize the agent to their own competency could provide a more useful experience.

#### Communicating with User

G4 **Consider letting the system communicate what the agent can do, as well as the agent's limitations**

Think about how to communicate to the user the possibilities of the system and what the agent can do. This could for example be done by a small tutorial for a first-time user describing what kind of service the system can provide. Another way could be to provide easy access to documentation about the system, or just an icon giving some small explanation to the possibilities of the agent, what insights are, how they can help, and how to access them. This could be seen as one way of provide transparency to the user. Also think about letting the user know what the agent cannot do in order to avoid misunderstanding and frustration from the user. This could also involve telling the user why there might not be any available insights, for example if a data set is too simple for the system to be able to generate any insights from that data. By somehow telling the user what the agent is capable of could help the user set expectations of what to expect from the system, and what not to expect, and thus avoid disappointment or frustration.

G5 **Consider letting the system communicate the status of the agent**

Status of the agent could include progress, success or error, and by letting the agent communicate this to the user, could help the user understand what is going on. Any type of feedback signaling the process of the agent might be helpful and reduce frustration of the user. Showing the status could for example be showing the user when there insights available, either just by a label, or some other kind of notification. Also, differentiate between when there are no available insights because the system could not generate any based on the data, or when the agent is in the process of generating insights, in order to avoid frustration or confusion from the user to why there are no insight available. Error handling is also important, to tell the user why something did not work, and help the user recover from it in some way. It could just be about notifying the user that the data set is too small generating insights on, and that is why no insights have been found, and if there are any actions for the user to take to change this.

### User Experience

G6 **Consider letting the agent be configurable, allowing domain-specific knowledge to be added to the agent and enabling more relevant insights for different domains**

By being able to configure the insight agent and add domain-specific knowledge to it could help in giving users better and more relevant insights. One example is that it could be used to improve relevance in the ranking of the insights when presenting them. This must not be possible from an end user point of view, but some sort of administrator access could be able to change the domain model the agent is using. A standard model could also be used as default for the insight agent, without any domain-specific knowledge. This was also mentioned in the stakeholder interviews in the past iteration, as well as mentioned as a recommendation axes in the VisRec system [9].

G7 **Consider not letting interaction with the insights disrupt the user's workflow**

In visual analytics, the analysis is the main focus. Let the agent be in the background, assisting the user in a non-intrusive way. There are a number of things that could disrupt the user's workflow. It could be popups visually blocking the content of the dashboard, buttons or icons not being clear about what will happen when clicking them. Avoid unexpected surprises for the user.

G8 **If applying an insight would manipulate the existing dashboard in some way, consider a showing a visual preview of that change**

If an applied insight would result in manipulating the existing dashboard, like change or add something to an existing visualization, or add a completely new visualization, consider ways of showing this change for the user, without the user needing to manipulate the dashboard right away. This preview could be done in several ways, for example some kind of mask applied over the visualization, or some effect when hovering over the insight showing what the new visualization would look if insight were to be applied, or just a miniature of the insight together with its explanation. Being able to see a visual preview of the insight, the user could easily determine whether completely apply the insight is a good idea or not. This preview could also be a way of explaining what the generated insight it about. It could also be used for showing the context of the generated insight, if the look and/or placement of the preview would give clues about what the insight refers to.

G9 **Consider letting the user be able to easily remove applied insights**

Consider to make it easy for the user to easily revert a change. In more automatic scenarios, this could really help the user feeling in control by being able to remove an automatically applied insight if the user is not satisfied with it.

**Analytical Reasoning**

G10 **Consider letting the system provide an explanation for that is presented in the insights and provide the context of the insight**

This could be seen as the presentation of the actual insight, what the agent has found that it believes would be interesting for the user. An explanation of the insight could include a small text about why it was regarded to be an insight, for example "this data point seems to be an outlier since it has a 100 % higher value than the rest of the data". It could also be a text accompanied by a preview of the insight. It could also be something to connect the insight to an existing visualization (if there is any connection) and thus provide some kind of context for it. Anything that would give user understanding of the insight basically.

G11 **Consider letting visual cues to be shown automatically by the system, without the user having to interact with the insight function**

The idea is the agent serves the user with the available insights, by automatically telling the user when there are any insights available.

**Transparency**

G12 **Consider letting the user be able to see how and why the agent ended up with a specific insight**

A way of being transparent to the user is to provide an explanation for how an insight has been generated, for example what kind of statistical model that have been used. Furthermore, if there is a ranking between the generated insights (they are presented in a list for instance), then an explanation for that ranking could be presented. In a more automated system, where some insights might have been compared to others, it is also important to show why that selection has been made.

G13 **Consider letting the visualization clearly indicate what is generated from the data compared to what is generated by the system**

It is important to be transparent about what is actual data and what is an insight calculated by the system, since part of visual analytics is about analysing data in order to take informed decisions. For instance, an insight in the form of a forecast is probably based on some statistical model and therefor a calculation and can be misguiding the user if not presented for what it is, a prediction. This is more challenging if an insight is about cleaning data, for example remove an outlier. When something is removed, it is even harder for the user to see that something has happened to the visualization, compared to an insight that is applied to the visualization. A way of indicating a difference between actual data and what is system-generated could be differentiate them by form or/and color. Another way is to notify the user with some kind of feedback when something has happened, for instance if something have been removed. This is extra important in the more automatic scenarios.

G14 **Consider letting the user be able to access all of the insights gathered by the system**

If the highest ranked insight is automatically applied, make sure the user can access the rest of the generated insights (if any available) that the system excluded, in order to give back control to the user. For the system to select "the best insight" is probably impossible, both from a user, domain and technical perspective. The more automation of a system, the more would this ranking of insights take away control from the user. But displaying all insights, if there are many, is also a challenge and a recommendation would then to have for example a filter function enabling the user to go through available insights in an easy way, allowing the user to decide what they are interested in.

## 6.5 Iteration 4: High-Fidelity Prototyping

In the final iteration of the project, a high-fidelity prototype was created. The focus of this prototype was to evaluate the guidelines further, along with exemplifying some of the design concepts from the previous iterations. More ideation was involved before finalizing the concept into the more detailed and partly interactive high-fidelity prototype. Usability tests were performed and analysed before formulating the final set of suggested guidelines. Due to COVID-19, this phase was also affected by the need for social distancing [61][62].

### 6.5.1 Planning

The first step in the fourth iteration was to summarize the outcomes from previous iteration and come to a decision about what the focus would be for the final phase. From the evaluation of the low-fidelity prototypes it was concluded that the final prototype would be a merge of the most promising concepts and features from the "List View" concept and the "Preview in Visualization" concept.

There weer still a lot of questions left to be worked out in order to complete a high-fidelity prototype. It was both issues confirmed and discovered during previous usability tests, along with more detailed design decisions. It was also suggested by the Company to take another look at the level of automation when designing and applying the system generated insights, and possibly to explore more about what would happen if this level would be shifted to a more automatic one.

In the low-fidelity prototypes, the focus had been on the presentation, position and interaction with the system-generated insights, because in these previous iterations, the level of automation had shifted towards a more user-driven approach. Current visual analytics tools lean towards user-driven approaches regarding insights, and it also something that also was expressed was preferred in the evaluation of the low-fidelity prototypes. But since this is a master's thesis it was believed that testing the limits for the level of automation could bring more interesting findings. Thus, the focus for the ideation in this phase come to be an attempt on more automation when exploring how to design system-generated insights.

### 6.5.2 Ideation

Before starting to implement the final prototype, more ideation was done in order come up with possible solutions for the different issues still needed to be addressed. This issues included problems such as what types of insights, how to interact with them or how to visually connect an insight to a visualization (if possible, some insight were more "general", and not connected to a specific visualization in the dashboard but rather connected to the data itself). How to provide the user with transparency when the system automatically derives insights from data is also a challenge. Some limitations were created in order to help with some of the questions, for instance, what types of actions an insight could take were limited down do "Apply" or "Dismiss", leaving it open to what this could more specifically mean for an insight.

Some of the issues was brought forward at a weekly UX meeting, asking for input and feedback from the UX team at the Company. The two concepts which were to be merged in this final iteration were presented, together with some of the problems facing this iteration. The main question was how the interaction and result of a more automatic insight could look like. Some of the question that were presented at the meeting:

- What happens when there are a lot of insights at the same time?
- If insights are applied automatically, what happens if the user thinks the system has applied the wrong insights?
- What explanations are needed for the automatically applied insights?
- How can the user still be in control?

It was also discussed whether to explore "the best solution" meaning going forward with the user-driven approach, already used in the Product, and preferred by the test participants of the low-fidelity prototypes. Or to try out "the extreme solution", with a more system-driven approach, even though it would deliberately be "too much" automation in order to explore some interesting aspects arising in this type of solution. Both solutions were considered to be able to bring value.

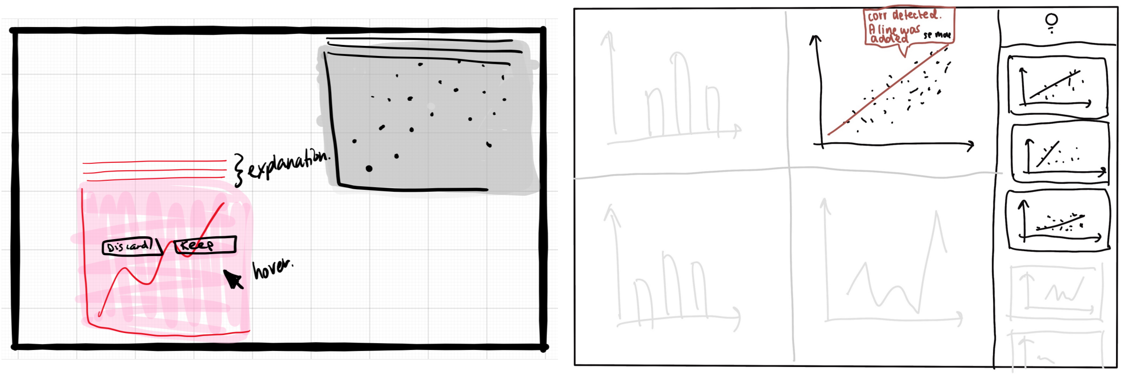
After this meeting, it was decided to try out a more automatic approach, which lead to some more problems to look into. What it is that can be automatic about insights what could help a user in the analytic process? Is it the generation of the insights? Is it to automatically show the generated insights? Is it to automatically apply available insights? And when the system do all these things automatically, how could what the system did then be explained to the user? How can the design give back control to the user?

Ideation in the form of a sketching session was performed, with a variant of the ideation method Crazy 8 (see Section 4.4.2). The focus was on quantity rather than quality to get inspiration for the different possibilities when designing for level of automation, and more specifically on what would happen after an insight has been automatically applied. It resulted in different ideas on how to manage the situation after the system automatically had done things (for example, generated an insight, or applied an insight if possible). Figure 6.17 shows two of the sketches from sketching session, where (a) shows how hovering over an applied insight enables the user to accept or dismiss it and (b) illustrates how to show several relevant insights, but let it be up to the user to change it the automatically applied did not seem to be correct or relevant.

A small workshop for some additional ideation was done with an UX designer and the Manager UX and Technical Communication at the Company. The purpose of this was to discuss the different ideas and get some final input and feedback before starting the implementation of the prototype.

### 6.5.3 Prototyping

A final prototype, called AIm$^2$, was produced using the software Axure RP (see Section 4.5.1). Despite a steep learning curve, Axure RP was deemed suitable as it




**(a)** A sketch of how to accept an automatically applied insight

**(b)** A sketch of several available insights for the same visualization

**Figure 6.17:** Sketches from ideation session

is possible to quickly construct a complex and dynamic prototype.

The evaluation of AIm$^2$ would be done remotely on the online platform usertesting.com, seen in Section 4.5.2, which put some constraints on the prototype and its interactions. The scenarios, tasks and questions would have to be formed in a way enabling the test participants to perform the test without any aid of a test facilitator.

The prototype can be seen as a vertical prototype (see more in Section 4.4.3) since only the functionality connected to the insights was implemented, in this case interacting with the applied and available insights in an already existing dashboard.

Scenarios for different possible use cases were provided from a senior product manager. One of these scenarios was picked in order to enable the prototype and its interactions to resemble an actual possible use case for a customer of the Product. In this scenario a warehouse manager responsible for inventory levels of goods, along with the ordering of items to ensure the right inventory level, is looking at a relevant dashboard. When the manager opens the inventory dashboard to look at inventory levels and plan what to order, the Product recommends the user to take a closer look at the inventory level of saffron, since historical data shows that the delivery time of saffron from suppliers usually more than doubles the upcoming time of year. The manager can access the insight and see the delivery time for saffron each year, along with a predicted delivery time for this year together with an explanation of this prediction. The prototype was based on this scenario in order to illustrate a possible use case for the system-generated insights. As mentioned above, it was decided to try out a quite automatic approach, which here meant scenario was tweaked a bit in order to try out a higher level of automation for the interaction with the system-generated insights.

### 6.5.4 The Design of AIm$^2$

The design of AIm$^2$ was a merge between the "List View" and the "Preview in Visualization" concepts from the previous iteration. From "List View", the insight panel was incorporated into the final design. When opened, it would not cover the dashboard and thus enable the user to see both the visualizations in the dashboard along with the available insights. From "Preview in Visualization", the idea of having contextual indicators within the visualization showing were an available insight originated, was included into the design. In the case of this prototype, these indicators are the insight labels under the title in top visualization for example, informing the user that "1 insight applied and 2 insights found", see Figure 6.18.


**Figure 6.18:** Dashboard just opened with a notification of available insights displayed in AIm$^2$

The initial look of the Inventory dashboard when just opened can be seen in Figure 6.18. There are three visualizations presented here "Inventory level over time", "Product categories, total value" and "Inventory turnover". A notification is displayed in the top right corner notifying the user that one insight has been applied.

In this first figure the guideline *G5 Consider letting the system communicate the status of the agent* have been used in order to communicate that there are insights available. Also, in this specific scenario with a higher level of automation, the system informs the user that an insight has been applied.

By presenting the generated insights list in a panel to the side of the dashboard, the guideline *G7 Consider not letting interaction with the insights disrupt the user's workflow* have been adhered to in the way that the insights does not cover any existing visualizations. This insight panel can be seen in Figure 6.19. The panel can be accessed by either clicking on the insight label indicating available insights in the visualization, or by the insight icon in the top toolbar. In the insight panel the insights can be applied to the dashboard by clicking on them. Clicking again and the insight is removed, based on the guideline *G9 Consider letting the user be able to easily remove applied insights.* All available insights are displayed in that list, following the guideline *G14 Consider letting the user be able to access all of the insights gathered by the system.* At the top of the insight panel there are some filters, enabling the user to filter insights if there are many.

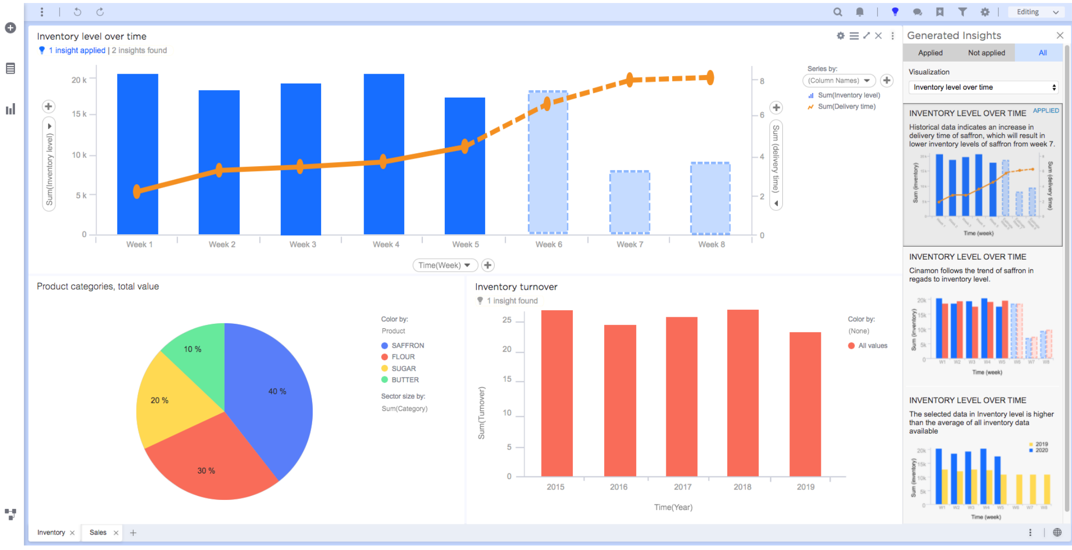



**Figure 6.19:** Insight panel open, with filter applied in the high-fidelity prototype

In Figure 6.20 the user is hovering over the the one available insight for the "Inventory turnover" visualization. When this happens, the visualization connected to this insight lights up in the dashboard. This is designed to follow the guideline *G10 Consider letting the system provide an explanation for that is presented in the insights and provide the context of the insight*, by provide a context to the insight. This guideline is also illustrated in the insight card by a text explaining what the insight is about. By the current design of the insight cards, a text with a preview of what the applied insight would look like, is it following the guideline *G8 If applying an insight would manipulate the existing dashboard in some way, consider a showing a visual preview of that change.*

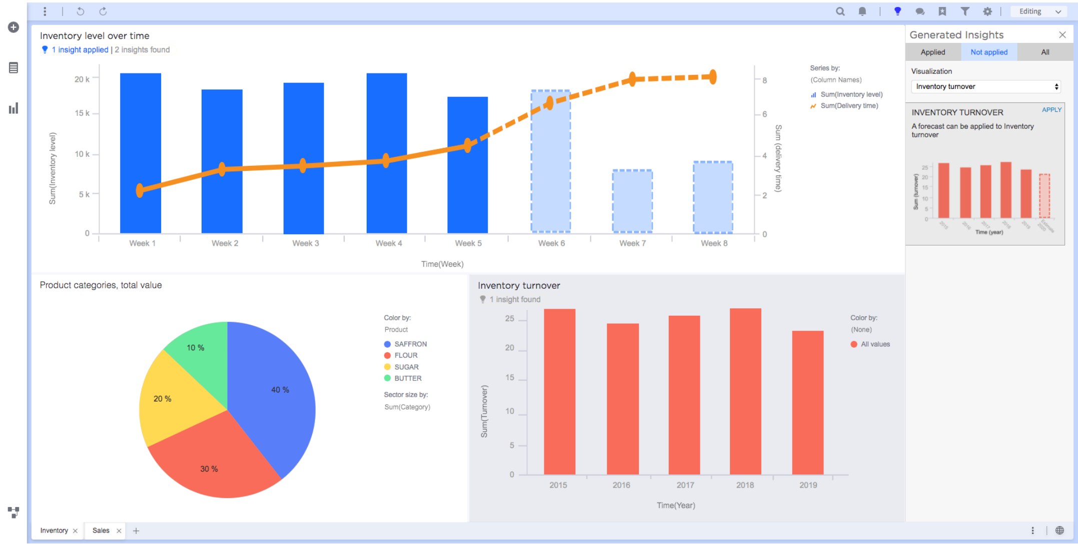



**Figure 6.20:** Insight panel open with user hovering over the insight in the list, with visualization corresponding to that insight highlighted, displays the connection between insight and visualization in the high-fidelity prototype

When an insight is applied, like in the "Inventory level over time" visualization in Figure 6.20, the insight in the form of a forecast looks different from the rest of the data in order to differentiate generated data from actual data imported into the visualization. This design strives to follow the guideline *G13 Consider letting the visualization clearly indicate what is generated from the data compared to what is generated by the system.*

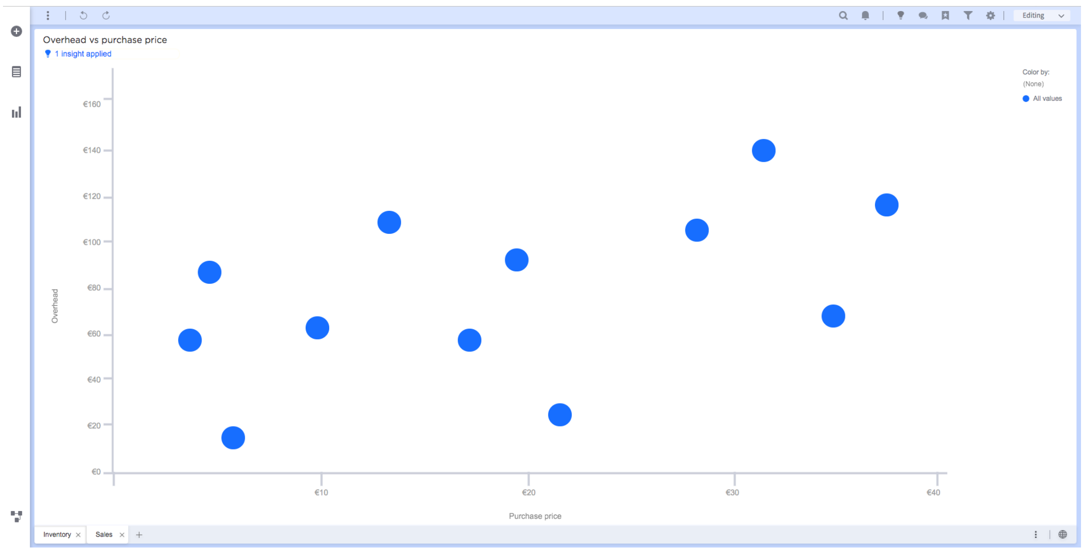



**Figure 6.21:** The Sales tab is opened with one insight automatically applied in the high-fidelity prototype

The user can switch to the Sales tab in the bottom left corner. In this dashboard there is one visualization, a scatter plot, see Figure 6.21. Also here an insight has been applied, and by clicking on either the insight label under the title or the short cut in the toolbar, that insight can be accessed in order to get information of what has been applied. In this case it was an outlier that had been removed automatically, see Figure 6.22. This illustrates the more tricky possibility of an applied insight actually being something that has been removed from a visualization, and not applied.

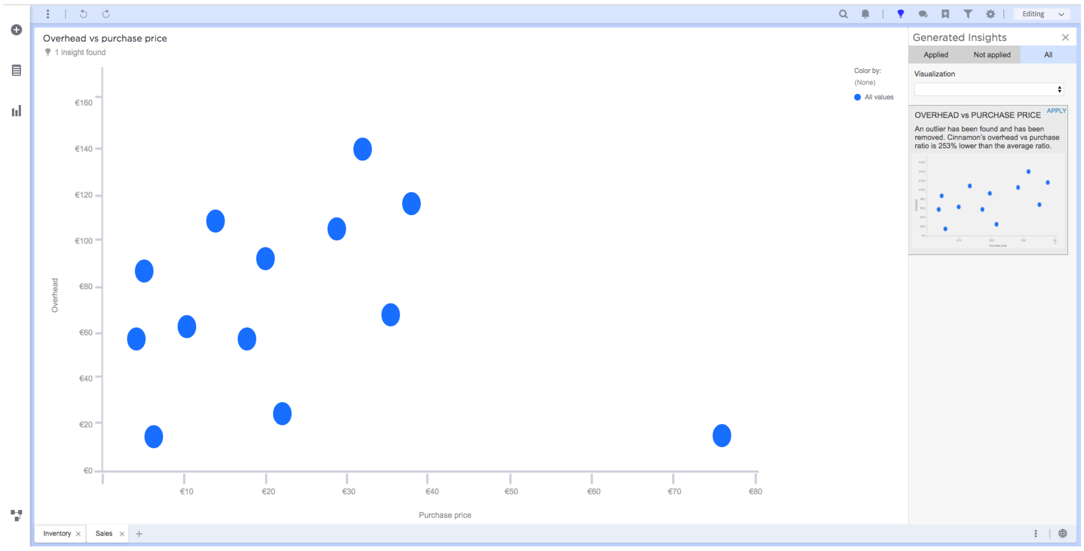



**Figure 6.22:** In the Sales tab, the insight panel in open with the automatically applied insight now removed in the high-fidelity prototype

### 6.5.5 Usability testing

A formative evaluation was done to get user input on the current design and prototype. The purpose of the test was to see what the users would think about the interaction between the visualisations and the system-generated insights and how much they understand how the insights work (especially now that the level of automation has shifted to more automatic). In general, the idea was to get user's thoughts on the level of automation and evaluate the guidelines.

The prototype, and thus the test, was developed for our second persona Ethan, a consumer of the product rather than an analyst, and the prototype only includes specific insights and only adds one insight per visualisation at any time.

#### Setup of test

For this user test, the tool Usertesting.com, mentioned in Section 4.5.2, was used. This allowed for test to be performed remote, over a specific platform designed for usability tests.

The test was formed as a formative evaluation, asking for the user's input and feedback rather than testing how well the prototype worked. The target user group set in Usability.com was:

- Age: 18-65+
- Employment status: Employed full-time or self-employed
- Web expertise: Advanced

Advanced web expertise was a requirement mostly to make the remote test go smoothly, as the test is performed without any support from the facilitators. However, the prototype is aimed towards expert users of analytical software, thus the computer experience of the user is presumed to be high.

A screening question was created, and all applicants had to answer yes to the question to be allowed to participate in the test. The screening question was "Are you analysing data in visualisations using Excel or an Analytics tool (like TIBCO Spotfire, Power BI, Tableau etc) as part of your job?" Once again, since the prototype and concept is aimed towards a professional consumer of analytics software, and thus this final usability test was aimed towards those users in particular.

A pilot test was first performed on one user. It showed that the tasks and questions needed further clarification. The final test was performed with five users and consisted of three part: an introduction with a scenario, tasks to be performed with the prototype together with questions throughout, as well as follow-up questions. The tasks and questions can be seen in Appendix B.

**Results from tests**

Here is summary of the results from evaluation of the high-fidelity prototype:

- All participants said the interaction with the system was easy and straightforward.
- All test participants saw that something had been applied to the dashboard when they opened it, indicating that the insight labels or notification text succeeded in notifying the user about applied/available insight
- Using a lighter color and dashed line seemed to be indicate for the majority test participants that they understood there was a difference between what in had been generated from data vs the system in the visualization.
- All participants believed it was really easy to access the generated insights
- Applying and removing an insight did not seem to be a problem for the participants
- All participants believed they still were in control despite automatically applied insights since the insights could easily be removed
- There was some confusion because the name of two of the visualizations were quite similar, creating difficulties on some of the tasks.

### 6.5.6 Design critique

The final prototype was presented to a data scientist at the Company in a small design critique in order to a data scientist's feedback on the high-fidelity prototype specifically, but also view on system-generated insights in general. Regarding the level of automation, a high level of automation is challenging, especially if data were to automatically be removed. One of the big takeaways were that transparency are very important, that the system is able to explain what has happened and how, to explain the decisions in the data. Another takeaway was simplicity, try to keep it simple, if users lost then it is too complicated anyway.

### 6.5.7 Reflections

Since the prototype was created to fit together with an existing product, it naturally affected the final design. If this project would not have been done at the Company, the resulting design would probably have looked conceptually differently. Using usertesting.com for the evaluation of AIm$^2$ also affected the prototype, since it would have to work without any help of a test facilitator, narrowing down the amount of possible scenarios. Already with these quite simple tasks the test participants could easily get out of sync, missing tasks on the way or misunderstand them.

Once the design was realized, and the whole scenario implemented, a realization was made regarding the last task. The task is for the user to see a visualization on a dashboard with an automatically applied insight. The insight has removed an outlier which the system deemed statistically probable to be irrelevant. Unlike the other tasks and scenarios used in AIm$^2$, this scenario removed data from the visualization. The task highlighted a question which had not been answered previously: should there be a difference between insight which remove data and insights which add data? While transparency is very important in all cases regarding insights, it is harder to enlighten removed data than added data, as there is nothing to point to. The question has not been answered in this thesis, but it highlights that there are many design challenges yet to explore within the field.

### 6.5.8 Guidelines v.4

Based on the reflections above, and a general look a the whole process, the decision was made to reconstruct the list of guidelines. The guidelines below is the final list of guidelines, and the previous version of guidelines has been generalized into larger design challenges under the topic insight specific guidelines. A process topic and a boundaries topic were also added. The process topic deals with considerations when it comes to the process of designing applications of system-generated insights. The guidelines under the boundaries topic are larger considerations for when designing these types of systems, and that affect most of the guidelines under the insight specific guidelines topic. They are overarching, design challenges that will affect many of the smaller design decisions that need to be made. The guidelines are presented below for clarification of changes between the Guidelines v. 3 and v. 4, however they are further clarified in Section 7.3.

### Process

- G1 **Use Other Guidelines**: Look at other relevant guidelines within human-AI interaction and visualization recommendation systems
- G2 **Follow a common design language**: Think about following a common design language for conceptualizing system-generated insights

### Set Boundaries

- G3 **Defining Insights**: Try to define what type of insights the system will generate
- G4 **Level of Automation**: Consider what level of automation is appropriate for the system-generated insights
- G5 **Understand the Agent**: Take time to understand the technology behind the generation of insights and think about what boundaries it will set on the design

### Insight Specific Guidelines

- G6 **Transparency**: Think about providing ways of showing transparency within both the system and the insights
- G7 **User in Control**: Consider how to make the user feel like they are in control of the system
- G8 **Workflow**: Consider what the appropriate workflow would be for the user when interacting with the system-generated insights
- G9 **Understanding the Insight**: Think about how to make the user easily understand the system generated insights
- G10 **Input to Agent**: Consider what input the user will be able to give to the system to help with the refinement of the insights algorithm

# 7

# Results

In trying to answer the research question "What should be considered when applying system-generated insights within visualizations in a visual analytics tool?", an iterative design process was used. Three results were produced throughout the thesis: a high-fidelity prototype, a design language for conceptualizing system-generated insights, and a final set of guidelines.

The guidelines can be used by future projects trying to design system-generated insights, both by designers and other possible stakeholders. It also gives a deeper understanding of what decisions are needed throughout the design process. The prototype, together with the description of the features, can be seen as an annotated artefact, as described in Section 4.1. The design language was primarily created to be used as an aid in the project. Thus, it is not fully developed, but can be used as a helping guide for others trying to conceptualize system-generated insights.

## 7.1 AIm$^2$: the Final Prototype

AIm$^2$ was implemented in the software Axure RP to illustrate how system-generated insights could be incorporated into a data visualization tool. It is a result of the iterative design process during the project. This system-generated insights could aid the user's analytical process with discovering overlooked trends in the data, or speed up the workflow by doing tasks automatically. Figure 7.1 shows a dashboard with three visualizations, two of them with available insights. The look and feel of the prototype has been designed in an effort to seamlessly fit with the existing Product. The prototype was created with a higher level of automation for the system-generated insights than maybe ordinary preferred or used in a visual analytics software. This is order to further explore how a more automatic approach for these insights would effect the user experience of the tool.

AIm$^2$ follows a specific path as it is just used as a tool to explore certain aspects of applying system-generated insights. However, it is interactive and more advanced than simple wireframes. It reacts to different inputs and actions from the user, and can handle multiple types of interactions, such as hover, click, filtering lists, and notifications. The idea with the high-fidelity of AIm$^2$ is to provide a familiar environment to other analytics programs, thus observations can be made regarding how the insights fit into the user's analytical workflow.

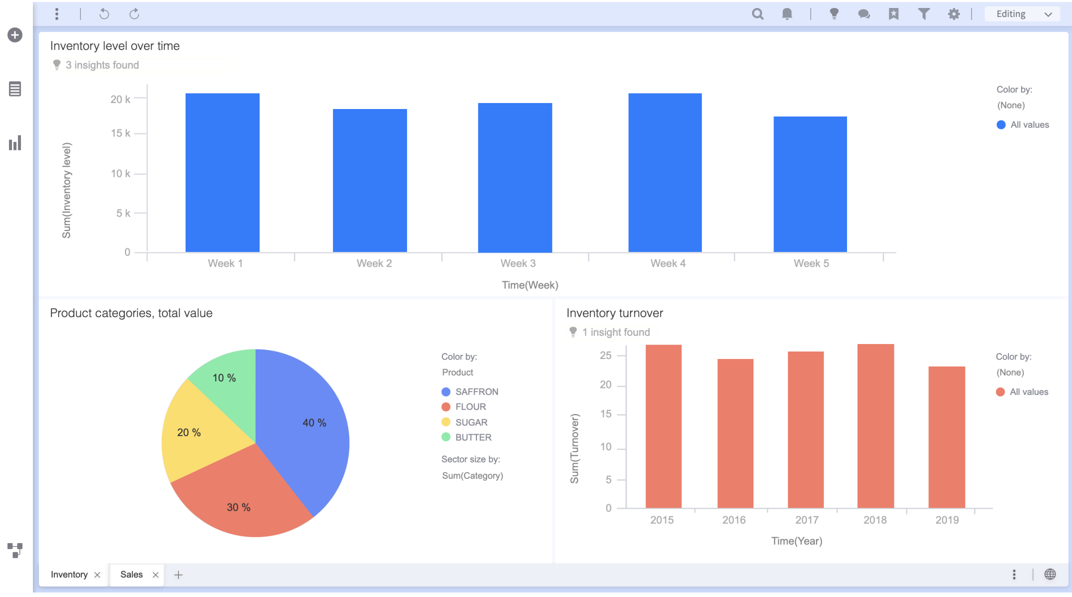



**Figure 7.1:** A dashboard with three visualizations, with insights available for two of the them

### 7.1.1 Contextual highlights

When there are insights applied or available it is indicated in the visualization, see Figure 7.2. This "highlight", in the form of a label under the title for the visualization, informs the user that there are system-generated insights available, or even applied, for that specific visualization. This prototype only handles insights referred to as "specific insights", in this project meaning insights that have a connection to data in existing visualizations in a dashboard.

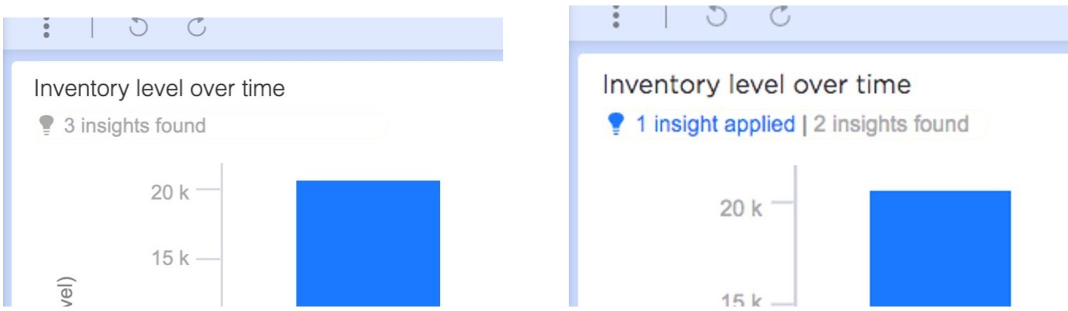



**Figure 7.2:** Contextual highlights in the form a label informing the user of available and applied insights for that specific visualization

### 7.1.2 The Insight panel

The insight panel can be found to the right side of the dashboard, see Figure 7.3. Here all available insights for the current worksheet can be found (in this case the current worksheet is the Inventory dashboard). The panel can be opened from the insight icon in the toolbar or directly from the visualization by clicking on the insight label explained above in Figure 7.2.

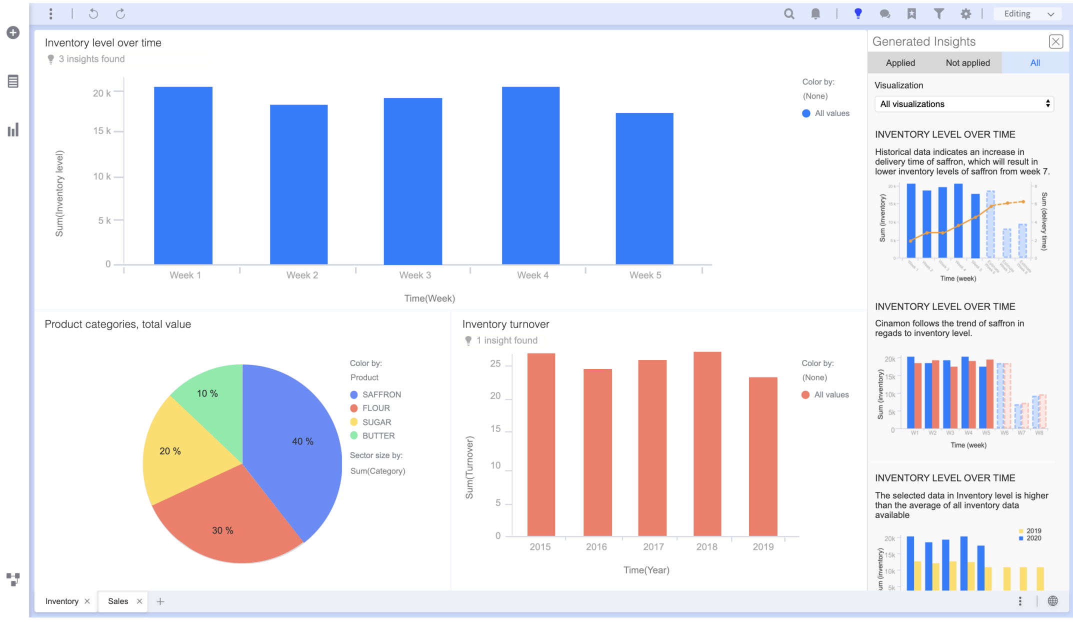



**Figure 7.3:** The insight panel with all available insights displayed

#### 7.1.2.1 The insights

All generated insights can be accessed in the insight panel. An insight can be applied by selecting it in the list of insights in the panel, see Figure 7.4. In this figure, the first insight in the list have been applied to the top visualization. This insight was a forecast of the inventory, based on the data in the "Inventory level of time" plot. Also a line chart of the underlying data the forecast was based on was applied here, delivery time over time, illustrating the prediction of a decrease the inventory level.

The actions provided for an insight in this prototype is just apply and then to remove it again from the visualization. What type of functionality an insight could consist of is flexible, depending on system, type of insight, domain, etc. Figure 7.5 shows different states of the insights in the list. The insights' different states are indicated with visual feedback. Figure 7.5a shows an available insight that is not applied to the dashboard. There is a mouse over effect, seen in Figure 7.5b when a user hovers over the insight, showing a label with APPLY indicating that it can be applied to the dashboard. Similar for when an insight is applied, there is a mouse over effect indicating that selecting the insight again will remove it, see Figure 7.5c.

#### 7.1.2.2 Filtering insights

There is a filtering functionality in the top of the list, enabling the user to filter insights, see Figure 7.6. A filter can be applied to the insights in the list by selecting either the filtering button, filtering for applied or not applied insights, see Figure 7.6a. Or the user can filter insights on preferred visualization in dashboard instead,

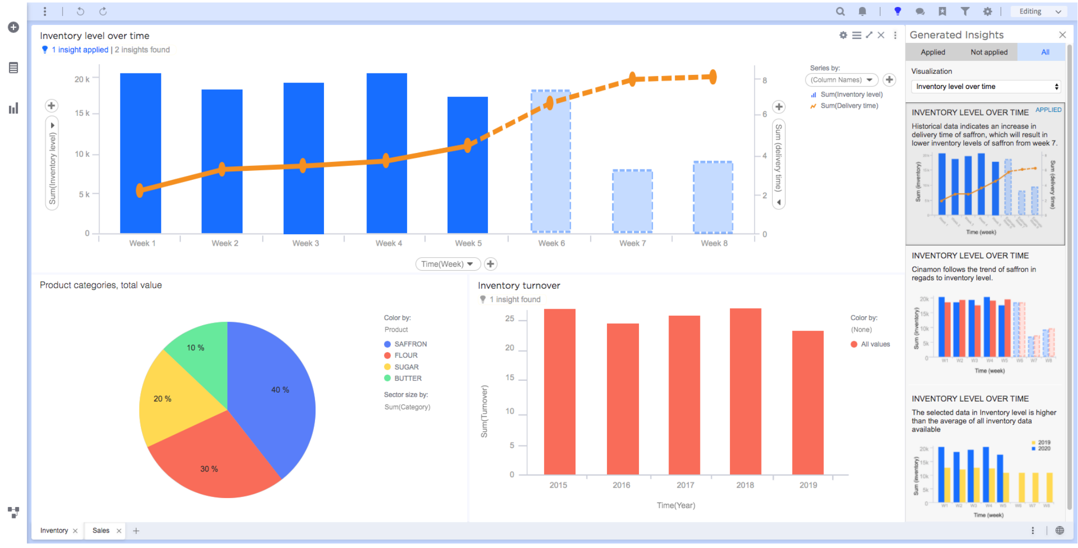



**Figure 7.4:** The insight panel open with one of the insights applied to a visualization in the dashboard

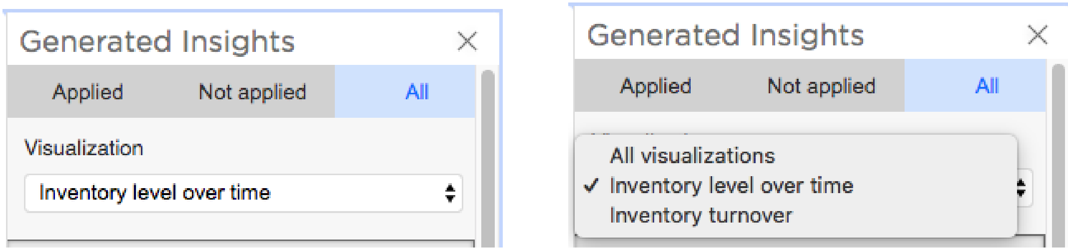

**(a)** An available but not applied insight

**(b)** An available but not applied insight with mouse over effect

**(c)** An already applied insight with a mouse over effect

**Figure 7.5:** The different states of insights presented

if any insights available, see Figure 7.6b. A filter is automatically applied if insights are accessed by the user through the insight label in the visualization, then only insights for that specific visualization is presented in the list in the insight panel.




**(a)** Show only applied, not applied or all insights with filtering button

**(b)** Filter based on the visualization the insight originated from

**Figure 7.6:** Filtering insights is possible in the top of the insight panel

#### 7.1.2.3 Connection between insight and visualization

When a user hovers over an insight in the insight list, it changes color along with its corresponding visualization in the dashboard in order to help the user identify which visualization the insight belongs to, see Figure 7.7. The aim is to provide the user with a context for the insight, hopefully speeding up the process of understanding what the insight is about. This works for both applied and not applied insights.

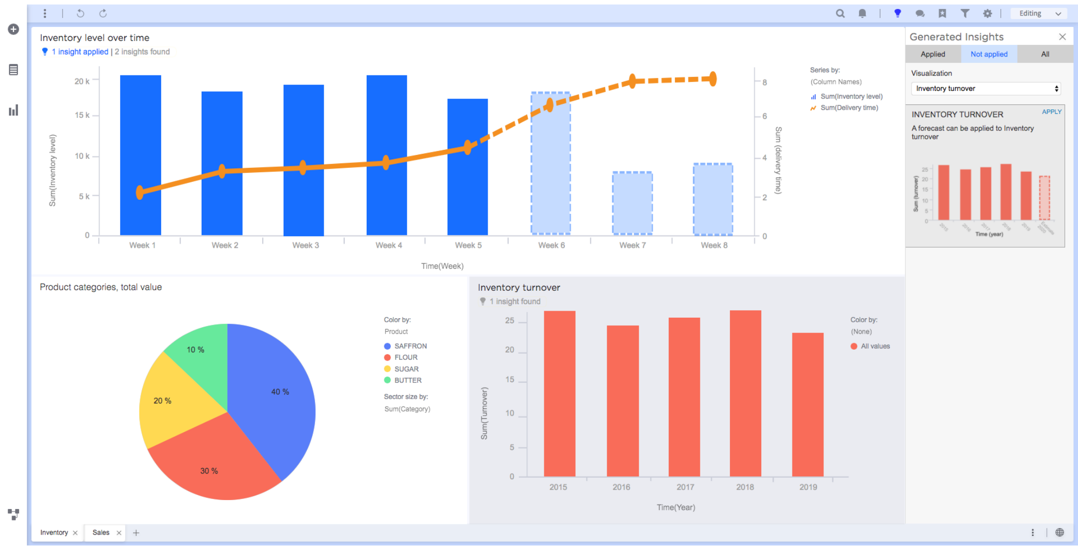



**Figure 7.7:** When hovering over an insight in the list, the corresponding visualization in the dashboard will change color, indicating where in the dashboard the insight is connected

### 7.1.3 Real data compared to data generated from the system

In order to differentiate between actual data imported by the user and data generated by the system, the applied forecast is visually different from the actual data, by the lighter color and dashed line (see Figure 7.8).

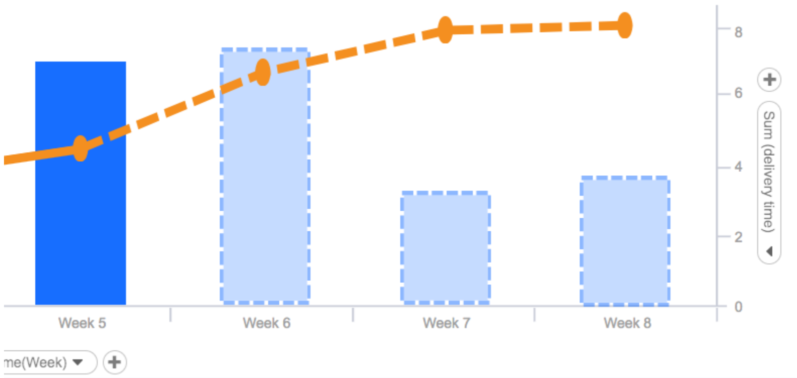



**Figure 7.8:** Part of the visualization "Inventory level over time", with applied data generated from the system, a forecast

## 7.2 Design Language

To aid in the communication between designers and help with skewing the properties of different concepts, a design language was produced. This design language can be used when conceptualizing designs for system-generated insights, and act as a reminder of different design challenges that need to be addressed. The terms from this language was used throughout the project, but it is important to note that the focus of the thesis was the final guidelines and the final prototype, thus the design language mentioned in this section has not been as thoroughly iterated upon.

The design language can be used as a way of exploring different concepts by using the different dimensions and properties to tweak the designs. For example, what happens to the intrusiveness of agentive insights if there are no actions to be taken from an insight. What does intrusiveness mean in that case and how does it affect the format of the insights? These things can be discussed between designers in a more straightforward way when there are defined dimensions and properties to use as discussion points. While the following language is formed from the experience of this specific thesis, it can be used as a starting point when discussing the design of system-generated insights.

The language is divided into different topics:

1. **Context**: The context is the mental focus of the user, sometimes perceived as the digital focus of the program. This can be applied both to the information in the insight, but also to what type of user will be using the system.
2. **UI**: More user interface specific dimensions, such as UI elements or interactions.
3. **Trust**: Different dimensions for the transparency in the system and how to create trust.
4. **Position**: Specific to the manifestation of the insight, and the position of the different parts of an insight's anatomy.

## Context

| Dimension | Property |
|---|---|
| **User** | Consumer \| Analyst \| Both |

*Examples of things an analyst does: create plot/dashboard. Examples of things a consumer does: receive a dashboard from someone else.*

| | |
|---|---|
| **Type of insight** | Correlations \| Comparisons \| Outliers \| Forecasts |

*These types are specific to this thesis project, as these are the insights within the scope of the project. However, it is possible to expand this list as needed.*

| | |
|---|---|
| **Level of insight** | Data point specific \| Visualization specific \| Categorical \| Whole data set |

*The context is the mental focus of the user, sometimes perceived as the digital focus of the program. However, in this category we apply this so the agent can know what level the insight should be on.*

| | |
|---|---|
| **Scope of insight** | Specific \| General |

*How generalized an insight is. Can it be contained to one part of the dashboard or visualization, or is it a more general insight for the whole dashboard. This is similar to the level of the insight, however this can be applied to the different levels of the insights.*

## UI

| Dimension | Property |
|---|---|
| **Automation** | User-driven \| Assistive \| Agentive |
| *This is more of a scale and is used as a category to assist in looking at the level of automation within a system of system-generated insights. Different parts of the system can be more/less automatic, while others are fully-automatic for example.* | |
| **Intrusiveness** | Very disruptive \| Sometimes demands action \| None |
| *This dimension mostly acts as a reminder to think about the workflow and how disturbing an insight is.* | |
| **Alert** | Push notification \| Notification icon \| Button \| Context based button \| None |
| *How do you show the user that there is an insight available? How does the system alert the user to the fact that there is an insight?* | |
| **Format** | Modal pop-up \| Preview \| Modeless pop-up \| List \| Single element |
| *What format are the insights presented in?* | |
| **Manifestation** | Picture of another visualization \| Applied straight to visualization \| Text based \| Only numbers \| Animation |
| *How are the insights shown? Are they applied straight away, so they are only consisting of a visualization, or is the user presented with an explanation and preview before it is applied?* | |
| **Actions** | Add to visualization \| Add to dashboard \| Change visualization \| Custom actions \| No actions |
| *What is the user supposed to do with the insight?* | |
| **Interactions** | Hover \| Click \| Mark \| Drag and drop \| No interaction |
| *What type of interaction will be used to interact with the insights? More relevant if the insight allows for actions to be taken.* | |
| **Can Be Hidden** | Convenient \| Not convenient \| Not applicable |
| *Whether or not insights can be toggled to show/hide for example the applied insights or any insight triggers in the dashboard.* | |

## Trust

| Dimension | Property |
|---|---|
| **Transparency** | Not transparent \| A little transparency \| Very transparent |
| *A scale of perceived transparency within the system. The transparency can be modified in many places in the system and can be different in different parts of the system.* | |
| **Certainty** | Shown \| Not shown \| Provided, but hidden |
| *These types are specific to this thesis project, as these are the insights within the scope of the project. However, it is possible to expand this list as needed.* | |
| **Traceability** | Deep dive \| Short presentation of numbers \| Access to the underlying data\| No traceability |
| *The ability of the user getting an explanation for what the insight is based on, allowing them to decide for themselves if they are able to trust the insight or not.* | |

## Position

| Dimension | Property |
|---|---|
| **Position of Insight** | Inside the visualization \| Next to the visualization\| In margins of visualization \| Inside dashboard \| Next to dashboard \| In-place (automatically added) \| Separate from dashboard \| Not applicable |
| *The position of the insight itself. Because of the nature of insights and the many different forms they can take on, this can either refer to a visualization being changed automatically or possibly the position of an insight card. Depends on the manifestation of the visualization.* | |
| **Position of Results** | Inside the visualization \| Next to the visualization\| In margins of visualization \| Inside dashboard \| Next to dashboard \| In-place (automatically added) \| Separate from dashboard \| Not applicable |
| *This dimension is also affected by the manifestation of the insight. The result is referring to what happens after the user has interacted with an insight. If there are no actions to be taken, then there will not be any results for example, and then this is not applicable.* | |
| **Position of Trigger Element** | Inside the visualization \| Next to the visualization\| In margins of visualization \| Inside dashboard \| Next to dashboard \| In-place (automatically added) \| Separate from dashboard \| Not applicable |
| *The position of the actual trigger button for making an action from an insight. Depends on the type of action taken and if there are actions at all.* | |

## 7.3 Final Guidelines

The main goal of the thesis project was to produce guidelines which aids in the design of system-generated insights. Ten final guidelines were produced and are presented below. The guidelines covers the design process, general boundaries that need to be considered, as well as more insight specific guidelines. They are comprised of recommendations for the design process as well as more specific design challenges which came up throughout the thesis project.

The guidelines are based on this thesis, and were iterated upon throughout the project. However, it is important to note that the following are only recommendations of things to think about during the design process, and were created as an aid to future designers. They were designed specifically for system-generated insights within visualization tools.

### 7.3.1 Process

#### Guideline 1: Use Other Guidelines

*Look at other relevant guidelines within human-AI interaction and visualization recommendation systems*

While there seems to be a lack of research specifically for designing system-generated insights, both Noessel [10] and Microsoft [28] have produced relevant work within the field of AI. These guidelines are mostly aimed towards designs with some element of automation, as they become more relevant then. However, the human-AI guidelines can still be used for more user-driven designs.

The guidelines mentioned above were the basis of the first versions of the final guidelines presented in this chapter. However, the final guidelines evolved into being more specifically aimed towards the challenges faced when designing system-generated insights. Thus, more general human-AI guidelines, such as the ones mentioned above, should also be looked into when appropriate. They can still be used throughout the design, while also following the guidelines found in this thesis, as seen in Figure 7.9.

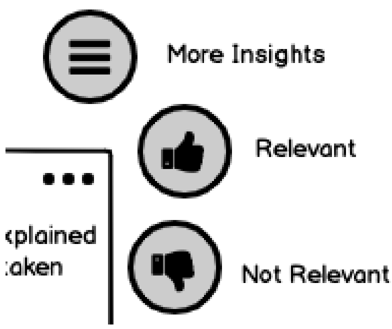



**Figure 7.9:** A low-fidelity prototype encouraging granular feedback from the user, according to G15 in Human-AI guidelines [28].

Even more relevant to the design of system-generated insights is the VisRec system proposed by Vartak [9]. While it does not explicitly talk about system-generated insights, but rather recommendation systems in general, the recommendation axes and recommendation criteria are highly relevant to the deign process of insights. More specifically, the recommendation criteria can for example be used when designing the insights themselves, and what they should contain. The recommendation axes describe the interaction with insights in part, however, the guidelines below has been designed to be used as a further aid when designing.

### Guideline 2: Follow a common design language

*Think about following a common design language for conceptualizing system-generated insights*

As an aid in concept generation in the second iteration, a small design language with regard to different insights and their dimensions and properties were created. Because of the nature of system-generated insights, there are many dimensions which can be tweaked such as the type the insight, the level of automation of the system, where transparency is needed, the interactions with the user, and the look and feel of the insight. Many of these choices affect the other, both in positive and negative ways. The common design language helps in the design process as it is possible for the designer to identify, discuss and try many different combinations, to see which combination yields the best experience for the user. For example, in the low-fidelity phase, different prototypes were produced that were similar but with some parameters changed. This can be seen in Figure 7.10.

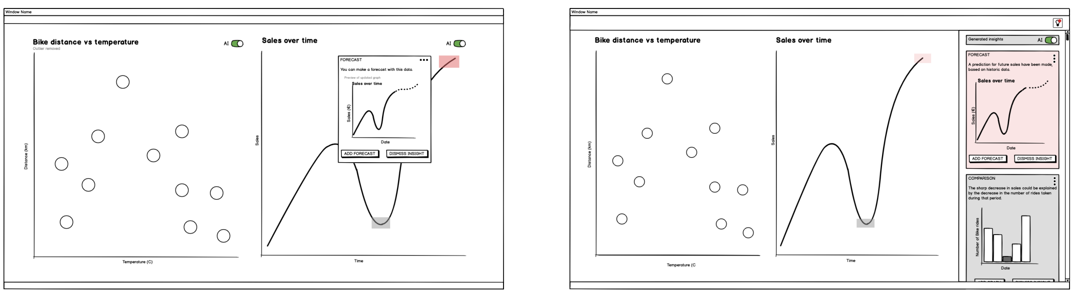



**Figure 7.10:** The dimension format was explored through different prototypes, for example one with a focus of within visualization and another with a focus of a list view.

Having a shared design language also provided a foundation for communication between designers throughout the project, making it easier to refer to the same thing.

The design language used in this thesis can be seen in Section 7.2. While the design language can be used as it is, it is also possible to form a new design language based on the specific task at hand. However, the recommendation is to have a design language to guide the concept generation, communication and discussion, and allow for skewing different dimensions of the concept.

### 7.3.2 Set Boundaries

#### Guideline 3: Defining Insights

*Try to define what type of insights the system will generate*

Using different methods, it is possible to define what insights the system will generate. If there is a clearly defined and simple use case, then the definition might be something which can be decided upon quickly. However, most likely, this will need to be iterated upon, to find out what the optimal insight will be for each specific design problem. An important note is that the whole system that is being designed will be very dependent upon the definition of the insights. Without this clear definition, it is possible that parts of the design will not satisfy all problems that will arise because of the complex nature of insights. Therefore, this is considered as a guideline set boundaries, affecting other parts of the design.

One way to define insights is to use guideline 2, and follow or create design language to follow. This was one way to set boundaries for the final prototype. An example of an insight can be seen in Figure 7.11. The following are also some examples of defining properties of the insights in AIm$^2$:

1. **User**: Consumer, but the concepts originated with a focus on both
2. **Type of insight**: the final prototype has examples of comparisons, outliers, and forecasts.
3. **Scope of insight**: Specific
4. **Automation**: Agentive, as there was an agent taking actions for the user and applying insights for that user
5. **Format**: List
6. **Interactions**: Hover, click

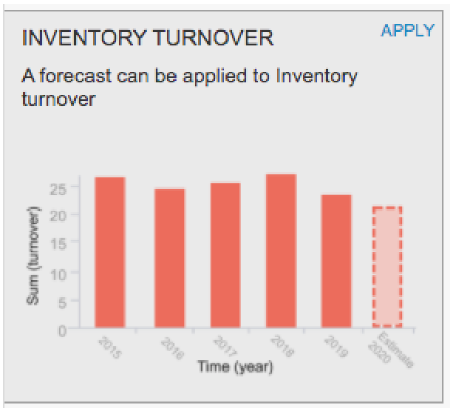



**Figure 7.11:** An example of a forecast insight, which includes a text explanation as well as a picture showing what the applied insight would look like.

Another method of defining the insight would be to use guidelines 6-10 stated below.

They can help with asking the right questions to further define the insight.

One of the main reasons to define what an insight is because of the complex nature of insights. An insight in and of itself is quite an abstract idea, and making the idea more tangible makes it easier to work towards a working design. Having a broad scope, for example trying to accommodate for both specific and general insights, will make the design much more complex.

Context specific UI elements work very well with specific insights, as specific insight is based on a specific part of a visualization. Pairing the insight to a specific part of a visualization seemed help with the understanding of the insight, according to the lo-fi user study in iteration 3. General insights, on the other hand, are difficult to visually place within a visualization. While it is possible to use context specific UI elements there as well, it becomes more difficult to find a general design pattern where both specific insights and general insights use context to their maximal effectiveness. These types of trade offs are generally part of the design process for designing system-generated insights, and knowing the scope that needs to be addressed is important to know where the trade offs need to happen. If a system only needs to be able to show specific insights, then it is possible to design it differently than if it needs to accommodate both general and specific insights.

#### Guideline 4: Level of Automation

*Consider what level of automation is appropriate for the system-generated insights*

One of the main things explored in the thesis was the level of automation of a system for presenting insights. The basis of this is that the use of a rule based system or AI to generate the insights, also allow for the use of such technology throughout the system. The level of automation is seen as a scale in this thesis, ranging from user-driven to agentive.

The level of automation can be different in different parts of the system. Some of the areas to looked at in this thesis are listed below:

1. Generating insights
2. Showing insights
3. Applying insights

Many of the current insight systems seem to have a bigger focus on user-driven design. In those systems the automation is mostly focused on number 1, generating insights. In $AIm^2$ the insights are generated, shown, and applied automatically, in some cases, as seen in Figure 7.12. The system does not apply all insights automatically, but chooses depending on domain-specific knowledge and user preferences.

While automation allows for the user to specify the context of the insights, making sure the generated insights are specifically relevant to the user's task, it also brings

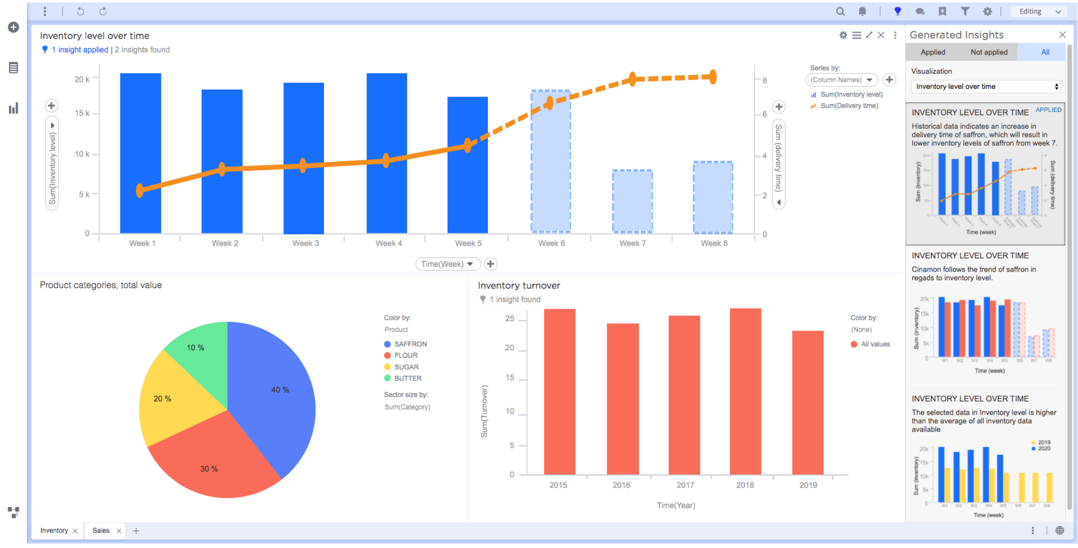



**Figure 7.12:** One insight is applied automatically to the visualization Inventory Level Over Time, but no insight is automatically applied to Inventory Turnover, despite the system finding one.

some negative aspects with it. For example, if the user has to generate the insights themselves, it is difficult to know beforehand if there are any relevant insights. The system also then depends on the user deciding when they want insights, so relevant insights can be missed because of it.

Generating the insights automatically would provide good feedback to the user whether or not there are any relevant insights. However, if the insights are to be only data point specific, and the design is built around choosing a data point to get insights about, then generating the insights would be difficult. The designer will be faced with the same problem for this guideline as with guideline 3: boundaries need to be set because the rest of the design will depend on the level of automation.

A few different levels of automation was explored, either as a prototype or just as a concept. An example of a more automatic application of insights is the final prototype. If there are insights that are generated and have a high enough probability of being relevant, it will be applied when opening a new dashboard.

### Guideline 5: Understand the Agent

*Take time to understand the technology behind the generation of insights and think about what boundaries it will set on the design*

When it comes to system-generated insights, a list of insights is generated, no matter if they are visible or not. These will then be ranked in some predetermined order. The capability of the algorithm that produces that list will affect the design of the system-generated insights. What information will the algorithm be able to supply based on the set rules of the system, and what types of calculations can it make.

The VisRec system [9] is very relevant to this topic. Vartak writes that the semantics or domain knowledge will dictate the usefulness of a recommendation. If the algorithm is not able to actually use domain-knowledge, certain dimensions of the design need to adapt.

For example, the automatic nature for applying insights in the final prototype is built around the notion that the algorithm is able to accurately enough predict what the most relevant insight for a specific visualization would be. A hypothetical algorithm was used, which generated insights based on domain knowledge and ranked the insights based on the probability of relevance and certainty. This can be seen in Figure 7.12, where an insight has been automatically applied to Inventory Level Over Time, but not to the visualization Inventory Turnover. The certainty of the most relevant insight for Inventory Turnover was considered to be too low to actually apply it to the visualization. If the system is built around an algorithm which is not accurate enough, the usefulness of a higher level of automation for applying insights is reduced. It can even become a pain to always have to correct the system. Therefore, it is important to think about designing around the algorithm which powers the insights.

### 7.3.3 Insight Specific Guidelines

#### Guideline 6: Transparency

*Think about providing ways of showing transparency within both the system and the insights*

Transparency within a system-generated insights concept can provide a good understanding of the underlying automatic processes as well as build trust between the user and the agent. There are different parts of the system that can be subject to this design problem, as showing what has been applied automatically in an automatic system is important, but also allowing the user to see the thought process behind an insight, and how what the agent is basing its calculations on.

Ever since the pre-study, the idea of transparency has come up multiple times throughout the project. G11, G16, and G18 of the human-AI guidelines [28] all relate to transparency, and talk about making it clear why the system did what it did, letting the user know about consequences of their actions as well as about changes made to the system. It was also brought up for example both during our iteration 1 interviews with stakeholders as well as during the design critique of the concepts in iteration 2.

An example of how transparency can be provided in a system would be to be clear about what the system can do, as well as provide the user with knowledge of its limitations. Consider also having a way for the user to explore and deep dive into the calculations of the agent and provide an answer as to why it has suggested an insight. Transparency can also be needed within a visualization or dashboard, notifying the user of what insights have been applied or if anything has changed.

The problem with not providing any transparency within the system is partly that the user can loose trust in the system. However, specifically in analytics, it can also be very important to be able to back up your claims. Allowing Serena to be able to deep dive, or drill down, into the insight and the underlying data will allow her to judge if it is statistics that she can actually trust. In certain fields, it is not allowed to use certain methods, for example, so the user would need to be able to backtrack the system's thought-process.

A good example of a so called deep dive is Tableau's Explain Data, as seen in Figure 7.13. While the UI is a little bit cluttered with information, it provides the user with a good understanding of both the reason why the insight is significant to the user's chosen task, and the underlying calculations for why the system is recommending that insight. The transparency give the user the ability to trust the system and also allows for the user to judge whether the insight actually is relevant or not for their specific case.

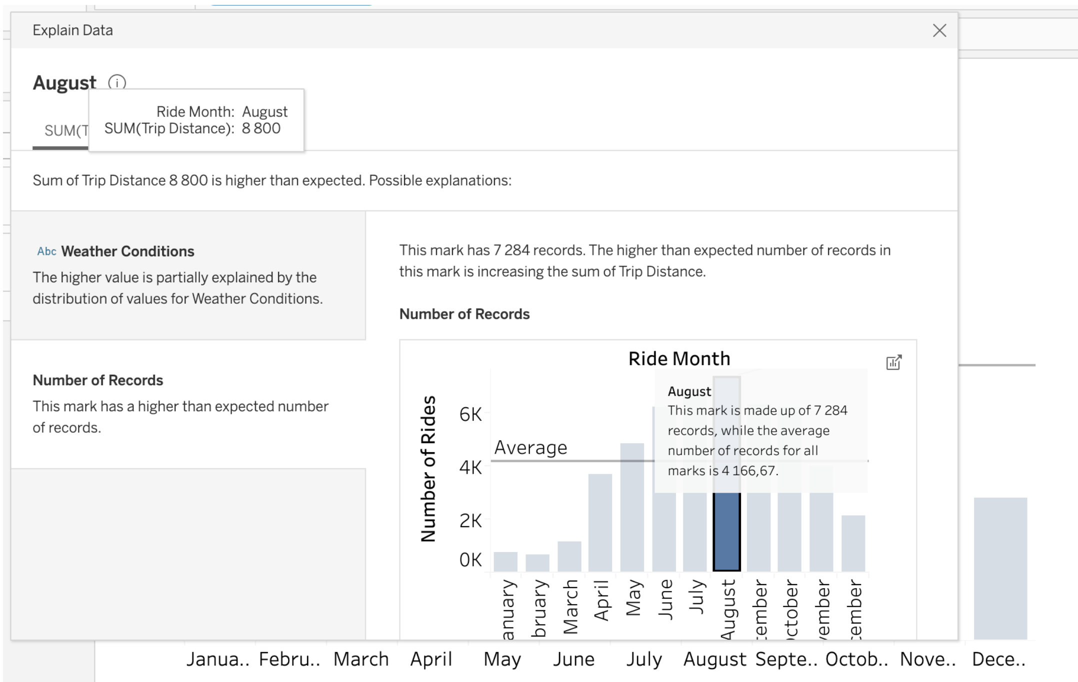



**Figure 7.13:** The Explore Data view within Tableau, with a short summary of why an insight is interesting, as well as a longer text with visualizations to let the user understand more in-depth the statistical significance of the insight

The Tableau example above is an example of transparency in a user-driven system. While a similar deep dive can be useful in more automatic systems, there is often a need for even more transparency within such systems. In the final prototype of this thesis, certain insight have been applied automatically. New problems arise then, where the system needs to be transparent with what has happened without the user having made any decisions. The insights notification as well as the link under the title of each visualization are examples ways of conveying changes to the user. They can be seen in Figure 7.14.


**Figure 7.14:** Examples of ways of notifying the user that the system has changed something within the dashboard automatically.

Another important note is that there are many different ways of providing transparency within a system, and that one must think about the particular system and the type of system-generated insights that are used. Adapt the transparency methods depending on the system.

### Guideline 7: User in Control

*Consider how to make the user feel like they are in control of the system*

Both Noessel [10] and Microsoft's human-AI guidelines [28] state that even if the system takes liberties and takes control of the experience at times, the user should always be in control. This is also the case when designing system-generated insights. The user should always feel like they are in control.

Some examples of providing control to the user is allowing them to easily remove and add insights themselves or allowing them to manipulate the insights in the same way as with the original visualizations. Letting the user know about the status of the agent is also a good way of instilling control in the user.

The user's control becomes an important design challenge especially within a system with higher levels of automation. In iteration 2, during the ideation phase, it was clear that it was difficult to balance the user's need for control while also applying and changing things within a dashboard automatically. A balance needs to be found, and in more automatic systems other control measures needs to be taken. For example, something which came up within the thesis was the ability to show and hide all applied insights, without completely removing them. This example can be seen in Figure 7.15.

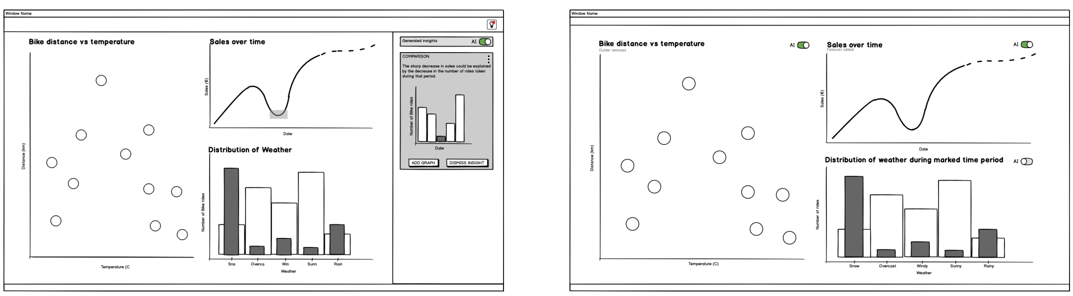



**Figure 7.15:** Examples of concepts providing a toggle function to turn off applied insights.

In the final prototype, work was also done to make sure the user felt in control when applying and removing insights, especially when they had been automatically removed. As seen in Figure 7.16, the status of the insights can be seen, and the user can easily add or remove insights. The title link, seen in Figure 7.14, also provides a quick way of opening the insight panel, and filtering it to match the specified visualization.

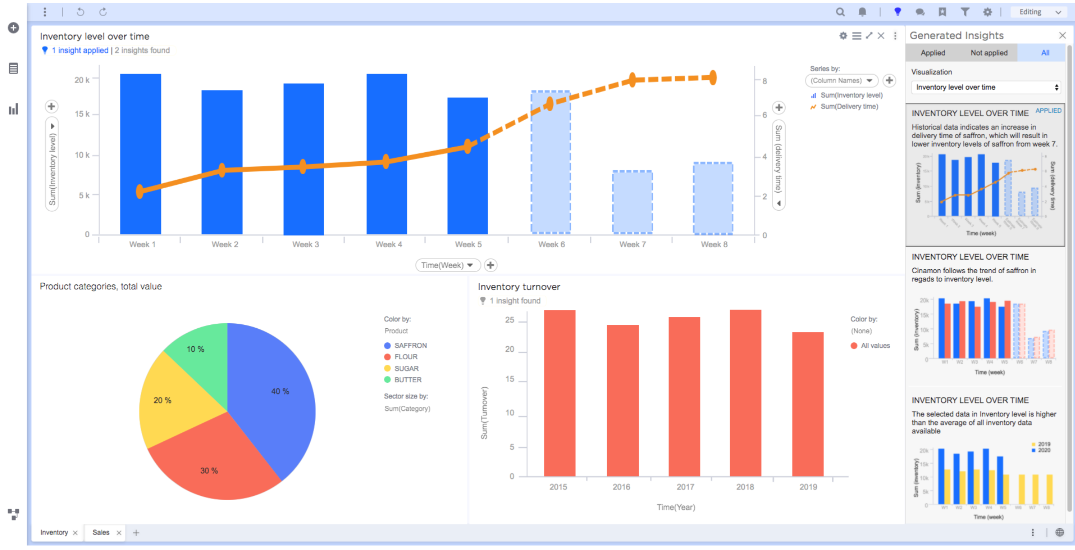



**Figure 7.16:** The insights panel is opened, showing the generated insights. Multiple features for user control can also be seen, including showing which insights have been applied and how many insights there for each visualization.

Guidelines 5, transparency, also plays a part in the user feeling in control. Being able to know what has been done in the system can also provide the user with a feeling of control.

### Guidelines 8: Workflow

*Consider what the appropriate workflow would be for the user when interacting with the system-generated insights*

This guideline is a more general user experience guideline, which might seem obvious when talking about interaction design. However, during the competitive analysis (see Section 6.2.1 for further information), it was obvious that many existing system-generated insights solutions seem to have chosen very modal approaches to the interactions with the insights. This lead to a very frustrating experience and the insights being disassociated from the analytics workflow.

Throughout the thesis project, one goal has been to try to not let the insights disrupt the user's flow. Different strategies were used to see what worked the best for this particular use case. For example, being able to interact with the visualization and the insights at the same time, like in the low-fidelity prototype "Preview in Visualization", garnered some positive comments because of the context it provided for the insights. However, the downside was that the pop-up would cover part of the screen, which was considered disruptive.

Another example of how to design around the workflow of the user can be seen in the final prototype, which took a different approach than the above-mentioned "Preview in Visualization" prototype. The idea was that the user should be able to work within the dashboard as normal, while also seeing the insights. An insights panel, seen in Figure 7.16, was used as a way to modelessly show the insights at the same time as the user works within the dashboard. The user is also able to filter the insight directly from the visualizations, using the link under the visualization title, and work has been done trying to connect the insights with the corresponding visualization.

Finding the perfect balance between functionality and a smooth workflow is difficult, and needs to be adapted to each design's specifications.

#### Guideline 9: Understanding the Insight

*Think about how to make the user easily understand the system generated insights*

The idea with system-generated insights is to provide the user with interesting observations and a deeper understanding of the data. To do this, the system needs to convey that message somehow. There are two main challenges with this guideline: how to make the user understand what the insight is trying to convey and how to show the context of that insight, with regards to the visualization or dashboard.

The VisRec system [9] mentions two important recommendation axes: **visual ease of understanding** and **user preferences and competencies**. Therefore, it is important to think about how to ease the understanding of the insights. It is also important to design for different types of visual literacy and statistical ability.


**Figure 7.17:** An insight, with a heading, an explanation, and a preview.

In Figure 7.17, an insight in the final prototype can be seen. The insight consists of three main parts: a heading related to the corresponding visualization, a short explanation for what the insight is regarding, and a preview of what the applied insight would look like.

The insight mentioned above is only an example of what can be done to try to help the user understand the system-generated insights. In some systems, an insight might only consist of a text, or possibly only a visual to convey the message. However, to be able to deal with different levels of visual literacy, the above approach was chosen.

Another way to provide an ease of understanding is to provide helpful hints about the context of the insight. For example, allowing the user to quickly see which visualization it is referring to. The context is conveyed in the final prototype, by highlighting the corresponding visualization when hovering over the insight, as seen in Figure 7.18.

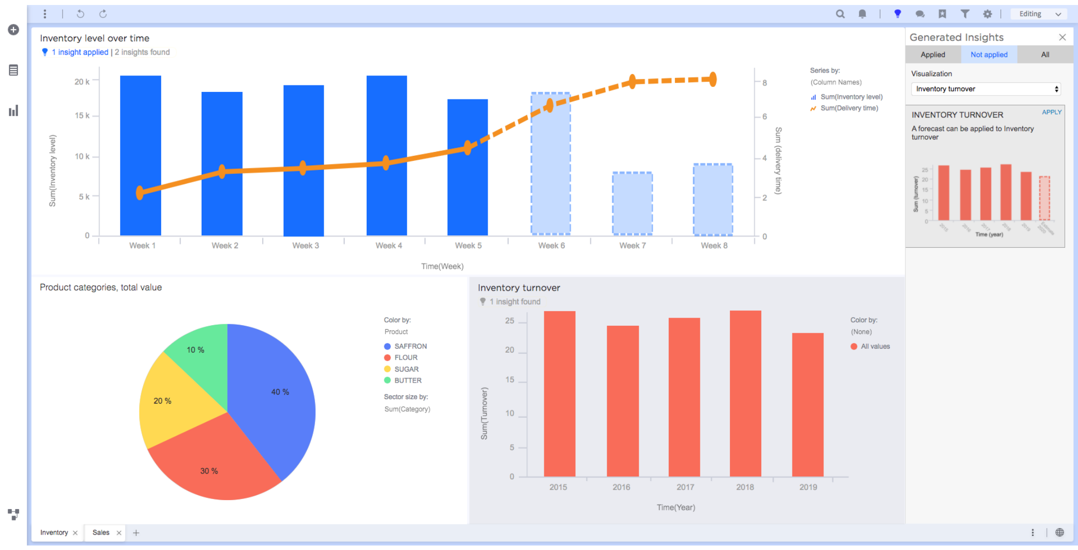



**Figure 7.18:** When hovering over an insight in the list, the visualization in the dashboard will be highlighted, indicating what visualization the insight is referring to.

### Guideline 10: Input to Agent

*Consider what input the user will be able to give to the system to help with the refinement of the insights algorithm*

Whether a more user-driven or agentive approach has been chosen, the fact that the list of insights is built around an algorithm, creates the need for the user to be allowed to input their preferences to the system. This can be done in multiple ways, such as setting up personal preferences or providing feedback to the agent.

The reason this guideline is part of the final guidelines, despite being very related to G9, G15, and G17 the human-AI guidelines [28] and user preferences mentioned as a recommendation axes in the VisRec system, is because of the importance of user input. No system of this sort will be flawless, and being able to learn what insights the general user thinks is relevant. This can be seen as part of the "Preview in Visualization" concept in Figure 7.19

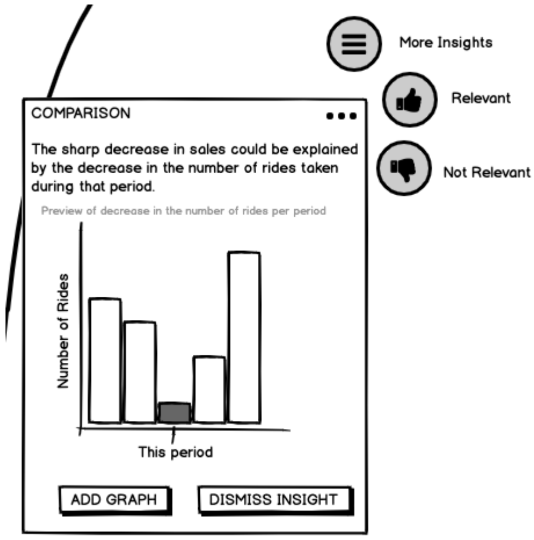



**Figure 7.19:** The "Preview in Visualization" prototype, with a feedback system for users to give input to the system whether an insight was relevant or not.

Once again, the level of automation will change the importance of this guideline. While user-driven applications of system-generated insights still need feedback from the user to produce relevant lists, more agentive systems will be reliant on feedback and user preferences to make the experience of the system-generated insight better. Accuracy is important when it comes to doing things for the user. The user might accept a mistake once or twice, but if the system continues to make the same mistake the user could lose trust in the system.

# 8

# Discussion

A discussion of the different parts of the thesis can be read below. Both the process and the results are discussed, as well as the generality of the project and relevant ethical aspects.

## 8.1 Process

As the whole field of system-generated insights seems very new within interaction design, much of the process was spent on trying to figure out the right path to take to explore the application of insights. Much time was spent figuring out the task at hand, and the literary study could have been spent more efficiently. However, at the time, the pre-study was used to map out the field and see if there were relevant research within the field. Unfortunately, the lack of previous research made it hard to know where to start our research and narrow down the study. Because of this, the study was exploratory, producing many different concepts to test different theories.

The scope of the study was broadened after the pre-study, as the course forward was not clear. As can be seen in the design language produced, the design space of system-generated insights is very complicated. There are many challenges, and many have not been explored yet within the academic field, as far as the pre-study was able to find. However, the broadened scope lead to more general guidelines, talking about the design challenges faced when applying system-generated insights, rather than an evaluation of a specific design.

Through the iterative design process and the broader scope, it was possible to explore many different types of concepts while working on narrowing down both guidelines and a prototype. Being able to explore many different concepts in both iteration 2 and 3 was very useful. For example, it allowed for exploration into more niche concepts, such as Spot the Bot, even if it was theorized from the beginning that it would not be the ideal design. While it took more time to make this approach, it allowed for exploration into a subject that lacks much research.

While producing many concepts was the idea from the beginning, it was not expected that the study would be as broad as it ended up being. The final prototype is still quite shallow, and many design challenges still need further exploration and research. One reason for this could be the addition of the second persona. Serena was initially the only target user group, but Ethan was added as the consumer perspective seemed like an interesting addition to the target user group. However, for a project of this size, a narrower focus would have provided needed limitations for many of the design challenges.

Because AIm$^2$ was not a finished product, the usability testing during the last iteration was formative, rather than summative. The test was formed as a formative test, trying to get different ideas and input into the design. However, the nature of the remote usertesting.com service did not allow for much discussion around the subject. The test were done remotely without the ability to do any further interviews after the test. While some input gathered from the test was useful, the state of the final prototype would have lent itself better to a face-to-face usability test with an unstructured interview afterward to be able to discuss further some of the things which came up during the test.

Much of the project has also taken place remotely. After the prototyping of the low-fidelity concepts, all of the work was done remotely, apart from half of the usability tests of the lo-fi prototypes. While it has gone surprisingly smoothly, working on the project remotely, ideation, and communicating sketches and ideas has been cumbersome. Sharing ideas on a whiteboard or on paper would have been helpful in many situations.

## 8.2 Results

The produced results of the thesis project are good reflections of the process and knowledge gained throughout the project. The broad scope of the project allowed for an exploration into many different parts of the design space of system-generated insights. The results presented in the thesis try to provide a good foundation for future research, trying to highlight potential design challenges to look into further in the future. The results are based on both usability testing as well as practical applications of them throughout the process.

The main result of the study is the final guidelines. The ten guidelines produced answers the question "what should be considered when applying system-generated insights within a visualization in a visual analytics tool". The exploratory nature of the project has led to many different design alternatives being produced, allowing for useful comparisons between different design choices.

While there is a final prototype, it was used to explore design challenges, much like the earlier concepts. The higher fidelity of the final prototype allowed for exploration in more detail, so it was still a valuable part of the process. However, the exploratory nature of the project makes it, so all prototypes and concepts work together to form an annotated artifact exploring different design challenges within this design space. The final prototype was the final concept, with a more detailed exploration, but certain design challenges are better represented in the earlier concepts.

The design language produced was a byproduct of trying to communicate about the abstract topic of insights. However, it is a very relevant result of the thesis as it turns out the design space of system-generated insights is very complex. There are many moving parts and changing one aspect of the design has rippling side-effects on other parts. Thus the use of a design language helps in communication and organization of ideas when looking at what to consider when applying system-generated insights. The result highlights the need for clear communication when looking at designing system-generated insights. While the design language produced in this thesis has not been fully evaluated apart from being used throughout the project, it could work as a good base for creating a more common design language regarding insights in visual analytics.

An important note is that the whole thesis was done in collaboration with the Company, and is built around the Product presented in Section 2.1.4. The authors have received support from people at the Company who have a lot of experience within the field, but also have heard of buzz-words and have a preconceived notion of what the application of system-generated insights should look like. This has affected the final prototype and final guidelines, mostly through the design decision that have been made throughout the process. For example, the level of automation of applying system-generated insights is said to be of interest to the subject. While the authors believe this to be true after having explored the design space, the fact that the Company has had that in mind could have boosted the importance without the author's knowledge.

The final prototype is also designed around the existing Product. This has both imposed a useful limitation so as not to make the task too time-consuming. But these limitations have also affected the final prototype, and previous concepts, and the design had developed from the existing design.

There are certain recommendations and insight systems in the Product today, but the design does not take them into account. This was done deliberately, so as to look at applications of system-generated insights on a more general scale, and not get too focused on the original product.

While the results could be evaluated further, they are formed in order to provide a foundation within the field of system-generated insights for future exploration. They are the result of a lack of previous research within the field, and can hopefully be used as a stepping stone into more detailed research of the design challenges presented. The results successfully mapped out the design space and provides a number of design challenges to tackle in the future.

## 8.3 Generalisation and validity of results

The focus of the results of this thesis has been specifically aimed towards answering the research question stated at the beginning of this thesis, which asks what should be considered when applying system-generated insights within visual analytics tools. However, the final guidelines are general enough to possibly be relevant or be adapted to be relevant to other applications that applied system-generated insights. Many of the guidelines, such as G3-G5, G7, and G9-10, are general enough to present design challenges that could be faced in applications other than visual analytics. While G6, transparency, and G8, workflow, could still be relevant considerations, they are more likely to have to be adapted depending on the situation. It is important to note that this is just speculation and not something that has been evaluated in the project. If the final guidelines were to be used in other applications, evaluating their relevance would be of great importance.

The creation of a design language can also be seen as a more generalized contribution. While the design language was formulated specifically around system-generated insights in visual analytics, the process of using the design language shows the usefulness of such design languages when exploring large and complex design challenges. The creation of the language, and the way it helped in the actual design process, could provide some useful guidance for other complex design problems.

As described in Section 4.1, the thesis has focused on producing design theory to answer the research question posed. The validity of the result is difficult to argue for, as they have been produced to act as rationales for design choices taken. The choices made are based on both user and stakeholder input, as well as the exploration of many concepts. However, the thesis has not tried to pursue an ideal version of science, so it is difficult to definitively argue for the validity of the results as they are specific to this project and the circumstances surrounding it. For example, as mentioned in the section above, the results have been influenced by the fact that the thesis is done in collaboration with a company. This has influenced both the prototype, the guidelines, and the process, in both positive and negative ways. However, the results have aimed to be based on the insights gathered in the process, from usability tests and self-reflections to exploring multiple alternatives and design challenges.

## 8.4 Discussion of ethical aspects

Ethical aspects of the thesis were explored during the pre-study of the project, and can be seen in Section 1.3. The three big issues mentioned were biased agents, misleading the user, and transparency of data. These issues are as relevant now as before. However, the first two issues were explicitly designed for.

During the design process, the last ethical issue mentioned, transparency, was the issue which was most thought of for the design—being transparent with what data has been generated by the agent and what visualizations are taken straight from the data set is essential. However, another part of transparency not mentioned is the transparency of what has happened to a visualization. G6, in the final guidelines, even addresses this issue. While it is something that was looked at for the design, the transparency of the system is something that will be discussed further in the next section, as it is not there are issues with it in the final prototype.

In the second part of the prototype, an insight has been automatically applied. That insight deals with cleaning data, and so it has removed an outlier the system deems to be an anomaly. An issue that arises is that if the outlier has been removed automatically, the user will not know what the original visualization would look like. It is important to remember to design for transparency for all changes to a visualization or data, even when data has been removed.

## 8.5 Future Work

The lack of previous research into the topic of applying system-generated insights means that there are many opportunities for evaluation and exploration within the problem area. What has been done in this thesis is to explore the design space, and annotated artifacts have been produced. However, there is a good opportunity to continue the work presented in this thesis, as many of the artifacts produced should be subject to further evaluation.

One main possibility of future work would be to evaluate the guidelines. While the guidelines produced are based on insights gathered during the design process and from usability tests, actually subjecting them to evaluation by using them in designing an application of system-generated insights in a visual analytics tool. The generality of the guidelines could also be evaluated by using them for designing applications of system-generated insight within other fields outside of visual analytics.

Another exciting possibility of the final guidelines is that they allow future work to build upon them and start to design different types of system-generated insights and evaluate them.

Through the design of different types of system-generated insights, the design language could also be evaluated. It was created explicitly for this thesis and not for general use. However, it would hopefully serve as a basis for a common design language when it comes to system-generated insights.

The final prototype is, as mentioned, mostly a tool to help with exploring different design challenges. It is still only part of a system, and some features which were not explored in this thesis could be looked into further. For example, the final prototype does not explore G10, user input to the agent. Different ways of giving feedback would be interesting to look into. How can the system fail gracefully and take help from the user to correct its actions?

New research possibilities open up using the artifacts presented in the thesis. Some interesting aspects to look into would be:

1. How to present the information in an insight in a useful way? How much information is really needed for the user to trust an insight?
2. How can you make use of natural language processing, together with system-generated insights, to make them even more useful?
3. Design patterns for applying system-generated insights could be created. Is there a general design pattern that would work for all, or at least many, types of insights?
4. Can all insight be automatically applied, or are there limiting factors?

# 9

# Conclusion

In this thesis, the application of system-generated insights has been explored in collaboration with a company. More specifically, the following question has been answered.

*What should be considered when applying system-generated insights within visualizations in a visual analytics tool?*

An iterative design process has been used, and five iterations have been carried out, starting with a pre-study. The iterative design process was chosen to be able to iteratively explore different concepts and allowed for multiple evaluations of guidelines produced from each iteration. Each iteration ended with producing and updating a set of guidelines, based on prototypes and evaluations done in the iteration. Formative evaluations have been carried out to evaluate the prototypes produced.

The research question has been answered by producing design theory, and the result is an annotated artifact, $AIm^2$, as well as two methodological contributions: a set of guidelines and a design language.

What should be considered when applying system-generated insights within visualization in a visual analytics tool can be seen in the following guidelines:

**Process**

- G1 Look at other relevant guidelines within human-AI interaction and visualization recommendation systems
- G2 Think about following a common design language for conceptualizing system-generated insights

**Set Boundaries**

- G3 Try to define what type of insights the system will generate
- G4 Consider what level of automation is appropriate for the system-generated insights

G5 Take time to understand the technology behind the generation of insights and think about what boundaries it will set on the design

**Insight Specific Guidelines**

G6 Think about providing ways of showing transparency within both the system and the insights

G7 Consider how to make the user feel like they are in control of the system

G8 Consider what the appropriate workflow would be for the user when interacting with the system-generated insights

G9 Think about how to make the user easily understand the system generated insights

G10 Consider what input the user will be able to give to the system to help with the refinement of the insights algorithm

AIm$^2$, a high-fidelity prototype, was also produced. It served as a tool for exploration, and as an example of the application of the above guidelines. During the process, a design language was also formed to be able to discuss different challenges and to skew different dimensions of concepts. Both of these results could be used in future research when looking at how to design the application of system-generated insights.

This thesis has explored part of the design space for system-generated insights. While the scope for this project was insights within visual analytics, the guidelines are general enough that they could be modified to fit other contexts as well. The final result will also hopefully act as a stepping stone for other research, having mapped out the design space and stated certain challenges within the field of system-generated insights. There is still a lot to explore, but the results from this thesis provide a good starting point for further development.

# Bibliography


[1] W. Cui, “Visual analytics: A comprehensive overview”, *IEEE Access*, vol. 7, pp. 81 555–81 573, 2019.

[2] J. J. Thomas, *Illuminating the path:[the research and development agenda for visual analytics]*. IEEE Computer Society, 2005.

[3] R. Chang, C. Ziemkiewicz, T. M. Green, and W. Ribarsky, “Defining insight for visual analytics”, *IEEE Computer Graphics and Applications*, vol. 29, no. 2, pp. 14–17, 2009.

[4] Microsoft, *PowerBI*, [Online]. Available: `https://powerbi.microsoft.com/en-us/`. [Accessed: Feb. 10, 2020], 2020.

[5] Tableau, *Explain Data - AI-driven explanations for your Data*, [Online]. Available: `https://www.tableau.com/products/new-features/explain-data`. [Accessed: Feb. 10, 2020].

[6] M. Tarallo, *The Qlik Insight Advisor - More than JUST ASKing questions!*, [Online]. Available: `https://community.qlik.com/t5/Qlik-Design-Blog/The-Qlik-Insight-Advisor-More-than-JUST-ASKing-questions/ba-p/1465810`. [Accessed: Feb. 10, 2020], Oct. 2018.

[7] Ç. Demiralp, P. J. Haas, S. Parthasarathy, and T. Pedapati, “Foresight: Recommending visual insights”, *arXiv preprint arXiv:1707.03877*, 2017.

[8] K. Wongsuphasawat, D. Moritz, A. Anand, J. Mackinlay, B. Howe, and J. Heer, “Voyager: Exploratory analysis via faceted browsing of visualization recommendations”, *IEEE transactions on visualization and computer graphics*, vol. 22, no. 1, pp. 649–658, 2015.

[9] M. Vartak, S. Huang, T. Siddiqui, S. Madden, and A. Parameswaran, “Towards visualization recommendation systems”, *ACM SIGMOD Record*, vol. 45, no. 4, pp. 34–39, 2017.

[10] C. Noessel, *Designing agentive technology*. Rosenfeld Media, LLC, 2017.

[11] N. Andrienko and G. Andrienko, *Exploratory Analysis of Spatial and Temporal Sata: A Systematic Approach*. Springer Science & Business Media, 2006.

[12] L. Kalbag, *Disruptive Design: Harmful Patterns and Bad Practice*, [Online]. Available: `https://www.youtube.com/watch?v=cVhySHv9bb8`. [Accessed: Feb. 28, 2020].

[13] J. Silberg and J. Manyika, "Notes from the ai frontier: Tackling bias in ai (and in humans)", *McKinsey Global Institute (June 2019)*, 2019.

[14] M. Blythe et al., *Apply insights in Power BI Desktop to explain fluctuations in visuals (preview)*, [Online]. Available: `https://docs.microsoft.com/en-us/power-bi/desktop-insights`. [Accessed: Feb. 10, 2020], Feb. 2019.

[15] ——, *Generate data insights automatically with Power BI*, [Online]. Available: `https://docs.microsoft.com/en-us/power-bi/service-insights`. [Accessed: Feb. 10, 2020], Aug. 2019.

[16] P. Baumgartner, *Find more insights in your Power BI dashboards with Quick Insights*, [Online]. Available: `https://powerbi.microsoft.com/en-us/blog/find-more-insights-in-your-dashboards-with-quick-insights/`. [Accessed: Feb. 10, 2020], Apr. 2016.

[17] TABLEAU SOFTWARE, LLC, *Tableau*, [Online]. Available: `https://www.tableau.com/`. [Accessed: May 22, 2020], 2020.

[18] Qlik, *Qlik Sense*, [Online]. Available: `https://www.qlik.com/us/products/qlik-sense`. [Accessed: May 22, 2020], 2020.

[19] M. Tarallo, *Qlik Associative Insights - A Simple Yet Practical Example*, [Online]. Available: `https://community.qlik.com/t5/Qlik-Design-Blog/Qlik-Associative-Insights-A-Simple-Yet-Practical-Example/ba-p/1573869`. [Accessed: Feb. 10, 2020], Apr. 2019.

[20] W. W. Eckerson, *Secrets of Analytical Leaders*. Technics Publications, 2012.

[21] S. Negash and P. Gray, "Business intelligence", in *Handbook on Decision Support Systems 2*, Springer, 2008, pp. 175–193.

[22] A. Bologa, R. Bologa, *et al.*, "Business intelligence using software agents", *Database Systems Journal*, vol. 2, no. 4, pp. 31–42, 2011.

[23] D. Loshin, *Business intelligence: the savvy manager's guide*. Newnes, 2012.

[24] J. Ranjan, "Business justification with business intelligence", *VINE*, 2008.

[25] S. Chaudhuri, U. Dayal, and V. Narasayya, "An overview of business intelligence technology", *Communications of the ACM*, vol. 54, no. 8, pp. 88–98, 2011.

[26] M. Card, *Readings in information visualization: using vision to think*. Morgan Kaufmann, 1999.

[27] N. Bikakis, "Big data visualization tools", *arXiv preprint arXiv:1801.08336*, 2018.

[28] S. Amershi *et al.*, "Guidelines for human-ai interaction", in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 2019, pp. 1–13.

[29] C. Frayling, "Research in art and design", *Royal College of Art Research Papers*, vol. 1, no. 1, pp. 1–5, Apr. 1993.

[30] B. Martin and B. M. Hanington, *Universal methods of design*. Rockport Publishers, 2012.

[31] W. Gaver, “What should we expect from research through design?”, *Conference on Human Factors in Computing Systems - Proceedings*, May 2012. DOI: 10.1145/2207676.2208538.

[32] J. C. Jones, *Design Methods*. John Wiley & Sons, Inc, 1992, ISBN: 0471284963.

[33] A. Williams, “User-centered design, activity-centered design, and goal-directed design: A review of three methods for designing web applications”, in *Proceedings of the 27th ACM international conference on Design of communication*, 2009, pp. 1–8.

[34] International Organization for Standardization, *ISO 13407:1999 - Human-centred design processes for interactive systems*, [Online]. Available: https://www.iso.org/standard/21197.html. [Accessed: Feb. 4, 2020], Jun. 1999.

[35] ——, *ISO 9241-210:2019 - Ergonomics of human-system interaction — Part 210: Human-centred design for interactive systems*, [Online]. Available: https://www.iso.org/standard/77520.html. [Accessed: Feb. 4, 2020], Jul. 2019.

[36] J. Preece, Y. Rogers, and H. Sharp, *Interaction Design - Beyond Human-Computer Interaction*, 2nd ed. West Sussex, United Kingdom: John Wiley & Sons Ltd., 2015, [E-book], ISBN: 9781119066019.

[37] B. R. Gifford and N. D. Enyedy, “Activity centered design: Towards a theoretical framework for cscl”, in *Proceedings of the 1999 conference on Computer support for collaborative learning*, 1999, 22–es.

[38] R. Buchanan, “Wicked problems in design thinking”, *Design Issues*, vol. 8, no. 2, pp. 5–21, 1992, ISSN: 07479360, 15314790. [Online]. Available: http://www.jstor.org/stable/1511637.

[39] H. W. J. Rittel and M. M. Webber, “Dilemmas in a general theory of planning”, *Policy Sciences*, vol. 4, no. 2, pp. 155–169, 1973, ISSN: 00322687, 15730891.

[40] R. Hartson and P. S. Pyla, *The UX Book - Process and Guidelines for Ensuring a Quality User Experience*. Elsevier, 2012.

[41] A. Gibbons, L. Botturi, and J. Nelson, “47 design languages”, *Handbook of Research for Educational Communications and Technology*, Jan. 2008.

[42] C. Rohrer, “When to use which user-experience research methods”, *Nielsen Norman Group*, 2014, [Online]. Available: https://www.nngroup.com/articles/which-ux-research-methods/. [Accessed: Feb. 11, 2020].

[43] J. Levy, *UX Strategy: How to devise innovative digital products that people want*. " O’Reilly Media, Inc.", 2015.

[44] Neil Turner, *A guide to sketch storming – a design game for ideation*, [Online]. Available: http://www.uxforthemasses.com/sketch-storming/. [Accessed: May. 18, 2020], Jun. 2017.

[45] C. Butler, *Honing Crazy Eights with randomness*, [Online]. Available: https://uxdesign.cc/honing-crazy-eights-with-randomness-1c8100b94824 [Accessed: May 24, 2020], 2017.

[46] Google, *Crazy 8’s*, [Online]. Available: https://designsprintkit.withgoogle.com/methodology/phase3-sketch/crazy-8s4 [Accessed: May 24, 2020].

[47] H. Sharp, J. Preece, and Y. Rogers, *Interaction Design: Beyond Human-Computer Interaction*, 5th ed. John Wiley & Sons, Incorporated, 2019.

[48] S. Gibbons, “Design critiques: Encourage a positive culture to improve products”, *Nielsen Norman Group*, 2016, [Online]. Available: `https://www.nngroup.com/articles/design-critiques/`. [Accessed: May. 18, 2020].

[49] Balsamiq Studios, LLC, *Balsamiq wireframes*, [Online]. Available: `https://balsamiq.com/`. [Accessed: Feb. 19, 2020].

[50] Marvel Prototyping LTD, *Marvel*, [Online]. Available: `https://marvelapp.com/`. [Accessed: Feb. 28, 2020].

[51] Figma, *Figma*, [Online]. Available: `https://www.figma.com/downloads/`. [Accessed: Feb. 28, 2020].

[52] Adobe, *Adobe XD*, [Online]. Available: `https://www.adobe.com/products/xd.html`. [Accessed: Feb. 28, 2020].

[53] Axure Software Solutions Inc., *Axure RP*, [Online]. Available: `https://www.axure.com`. [Accessed: Feb. 28, 2020].

[54] Bohemian Coding, *Sketch*, [Online]. Available: `https://www.sketch.com/`. [Accessed: Feb. 28, 2020].

[55] InVisionApp, *InVision*, [Online]. Available: `https://www.invisionapp.com/`. [Accessed: Feb. 28, 2020].

[56] UserTesting, *Usertesting*, [Online]. Available: `https://www.usertesting.com/faq` [Accessed: May 24, 2020].

[57] Google, *Google Scholar*, [Online]. Available: `scholar.google.com` [Accessed: May 19, 2020].

[58] Elsevier B.V., *Science Direct*, [Online]. Available: `https://www.sciencedirect.com/` [Accessed: May 19, 2020].

[59] IEEE, *IEEE Xplore*, [Online]. Available: `https://ieeexplore.ieee.org/Xplore/home.jsp` [Accessed: May 19, 2020].

[60] ACM, Inc., *ACM Digital Library*, [Online]. Available: `https://dl.acm.org/` [Accessed: May 19, 2020].

[61] C. for Disease Control and Prevention, *Social Distancing*, [Online]. Available: `https://www.cdc.gov/coronavirus/2019-ncov/prevent-getting-sick/social-distancing.html` [Accessed: May 24, 2020].

[62] Folkhälsomyndigheten, *Vad gäller för friska, sjuka och isolerade?*, [Online]. Available: `https://www.folkhalsomyndigheten.se/smittskydd-beredskap/utbrott/aktuella-utbrott/covid-19/skydda-dig-och-andra/vad-galler-for-friska-sjuka-och-isolerade/` [Accessed: May 24, 2020].

# A

# Affinity Diagram from Low-Fidelity Usability Test

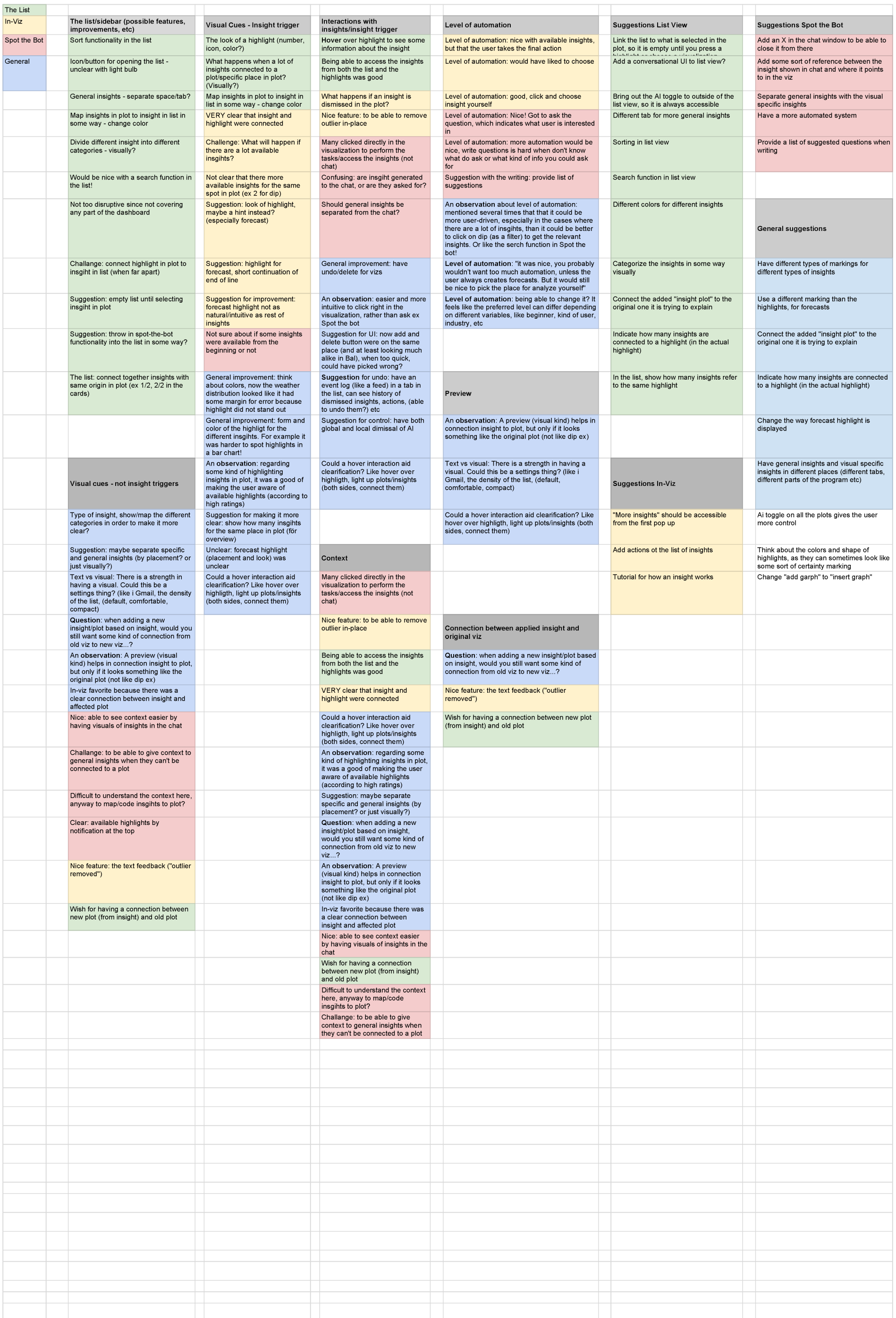



**Figure A.1:** Affinity diagram created in Google Sheets.

# B

# Tasks For the High-Fidelity Usability Test

The user was provided with the following scenario: *This is a prototype for a master's thesis looking at system-generated insights for visual analytics. The indented target is a consumer of dashboards created by someone else. Some examples of different types of insights that can be generated by the system based on the imported data are forecasts, correlations, comparisons, outliers, or a combination of the aforementioned. Imagine that you are interested in inventory management of spices, and use this dashboard to keep track. Use the password ***** to access the prototype. Please note that the prototype can be a bit slow, bear with us!*

While the user is taking the test, they are able to see the prototype and are also able to follow steps that are part fo the test. Each step is done with no knowledge of the next steps.

The main component of the test consisted of a series of tasks: some tasks meant interaction with the prototype; some tasks were verbal response questions answering questions about the prototype, but required no action within the prototype; and one tasks asked the user to fill in a rating scale based on the difficulty of a task.

1. Open the prototype from the link. Use the password ***** to access the prototype.
2. Open the dashboard "Stock analysis" (click on the image under Recent Analyses).
3. Take a look at the dashboard. Do you think any automatic insights have been applied to it by the system? In thet case, how is that indicated? Please explain. [Verbal response]
4. When looking at the "Inventory Over Time" visualisation, is it clear what is an insight (generated from the system) and what is actual data? [Verbal response]
5. It seems the inventory level will decrease in week 7. Can you find an explanation of why? (You will be guided in the next step, but please try first.)
6. If you have not already, please click on the link below the title Inventory Over

Time to open the insights panel. There you can see an applied insight which says that the inventory level over time will be affected from week 7 because of the increased delivery time.

7. How difficult (1) or easy (5) was it to access the insights panel with he explanation? [5-point Rating Scale: Very difficult to Very easy]
8. Open the insight panel by clicking the link under the title for the Inventory Turnover visualisation.
9. Apply the insight to the visualisation and then close the insight panel.
10. Open the insight panel from the toolbar (in the top right part of the screen) instead.
11. (If you have not already, click on the lightbulb icon)
12. Select to view insight for all visualisations.
13. In the insight panel to the right, select to view insight that are not applied to the dashboard.
14. At the bottom left corner of the page, select the Sales tab.
15. On this page, an insight has been applied - an outlier has been removed. You are interested in the data that has been removed. How do you go about to remove the insight?
16. The sales worksheet has an added hover effect on the insight int he insight panel, compared to the inventory worksheet. When hovering over th insight, the visualisation in the insight changed to who what would happen if the insight was clicked. Did you notice a difference between the worksheets? Was the sales worksheet version of any extra help? [Verbal response]

At the end of the test, usertesting.com prompts the user to verbally answer a standardised question: "Any final thoughts. Please provide any final comments, questions, or suggestions, you have before you stop recording and continue to the follow-up questions."

After the test was completed, the users were prompted with follow-up questions, to be answered in written form. The questions, along with the answers, are listed below:

1. How do you think the interaction with the system generated insights worked? With interactions we mean for example accessing the insight panel, removing/applying insights etc

    - It was very simple and easy to use. I found it to be very self explanatory and the steps were easy to follow

    - it worked well. very friendly. two different way to find them. i liked the pllied and not applied part of the page

- Very well, not nearly nearly as laggy as I would have expected from the screener - enjoyed the hover effect on the outliers page

- Accessing the insight panel was pretty straight-forward. Hovering function was pretty much crucial

- I think it worked well. Everything was smooth and easy to navigate. No issues once I got used to it.

2. Were you able to easily make a connection between the insight and the visualisation it was referring to?

   - Yes, the visualizations all seemed to make sense to me intuitively, and if not, I could read the explanations in the pop out pane.

   - yea very very clear due to the hover over effect

   - yes - very quickly - instantaneous changing between standard data and filtered/insight applied data.

   - Once the hovering function was in place, yes

   - I was once I got my bearings

3. Do you think these kinds of insights, and how they were applied, could be helpful in an analytical process?

   - Yes I currently work in financial data analytics for an aerospace company, and I feel that they definitely would help

   - yea yea yea! very helpful

   - absolutely - projections of any kind are very, very useful to a business. sometimes the insights will notice what the user does not and can yield surprising benefits for the business

   - Absolutely

   - Absolutely. Any/all extra insights are helpful in data analyses

4. How did it feel when insights were automatically applied, did you still feel in control?

   - I still felt in control because they are so easy to toggle on/off.

   - yea i do. i can change it if i want to

   - yes, of course - at the end of the day I am using your service for that feature and if it has non useful insights I could just turn off. Overall, nice browsing experience with an intuitive interface and useful insights.

- With the removing function in place, it was not an issue
- It felt fine. I never felt like I lost control over the data, no