Strategies for reducing the perception of complexity in a GIS interface

CLAUDIA ZELAZOWSKA
Abstract

Complexity in software is often a result of companies adding more and more features in order to make the product attractive in the marketplace (a phenomenon called ‘featuritis’ or ‘bloated software’) and a lack of a user-centered approach in the software development process. This causes a number of problems, such as the feeling of frustration and confusion amongst users and more time and resources spent on learning the software or executing a task.

The main purpose of this research project was to explore strategies that mitigate frustration and confusion caused by complexity for applications with instrumental aims, by engaging with concepts from interaction aesthetics and through qualitative and quantitative evaluation methods. This was done by focusing on redesigning dpPower, a GIS software developed by the company Digpro. Three strategies were identified which help reduce the perception of complexity and at the same time introduce concepts from interaction aesthetics to the system: clear information architecture, which increases fluency; adaptable interface, which concerns pliability; and consistency in interacting with the graphical user interface through visual design guidelines, which introduces rhythm to the interaction.
Strategier för att reducera upplevd komplexitet i GIS-mjukvara

Sammanfattning

Komplex mjukvara kan uppstå på grund av att företag lägger till fler och fler funktioner för att försöka göra sin produkt mer attraktiv på marknaden (detta fenomen kallas *featuritis* eller *bloated software*) men också för att de inte använt ett användarcentrerat tillvägagångssätt i sin mjukvaruutveckling. Detta kan skapa ett antal problem för användaren: känslor av frustration och förvirring samt att det kostar användaren mer resurser i form av tid att lära sig bemästra mjukvaran och att genomföra sina arbetsuppgifter. Målet med detta forskningsprojektet var att undersöka strategier som kan mildra användarnas frustration och förvirring, när de handskas med komplex mjukvara med instrumentala mål, genom att använda koncept från *interaction aesthetics* och genom kvalitativa och kvantitativa utvärderingsmetoder. Detta gjordes genom att fokusera på att omdesigna dpPower, som är ett GIS utvecklat av företaget Digpro. Tre strategier identifierades, som kunde hjälpa till att mildra den upplevda komplexiteten och de introducerade koncept från *interaction aesthetics* in till programvaran: tydlig informationsarkitektur, vilket ökar *fluency*; anpassningsbart gränssnitt, vilket introducerar *pliability*; och följdriktighet i interaktionen med det grafiska gränssnittet genom visuell design riktlinjer, som i sin tur introducerar *rhythm* till interaktionen.
Strategies for reducing the perception of complexity in a GIS interface

Claudia Zelazowska
School of Electrical Engineering and Computer Science
KTH Royal Institute of Technology
Stockholm, Sweden
claudiaz@kth.se

ABSTRACT
Complexity in software is often a result of companies adding more and more features in order to make the product attractive in the marketplace (a phenomenon called 'featuritis' or 'bloated software') and a lack of a user-centered approach in the software development process. This causes a number of problems, such as the feeling of frustration and confusion amongst users and more time and resources spent on learning the software or executing a task. The main purpose of this research project was to explore strategies that mitigate frustration and confusion caused by complexity for applications with instrumental aims, by engaging with concepts from interaction aesthetics and through qualitative and quantitative evaluation methods. This was done by focusing on redesigning dpPower, a GIS software developed by the company Digpro. Three strategies were identified which help reduce the perception of complexity and at the same time introduce concepts from interaction aesthetics to the system: clear information architecture, which increases fluency; adaptable interface, which concerns pliability; and consistency in interacting with the graphical user interface through visual design guidelines, which introduces rhythm to the interaction.

KEYWORDS
GIS; redesign of an interface; complexity; interaction aesthetics; information architecture; adaptable interface.

INTRODUCTION
Complexity in user interface design can be considered as something both good and bad. It can be a positive thing when defined as the level of sophistication of a programme [12]. However, complexity can also be quite problematic for users. In this paper, I refer to complexity in user interfaces as something that leads to users’ confusion, in which the location of interface components is often arbitrary, and the interaction techniques required to use them are not intuitive. This complexity leads to systems being viewed as complicated, which could be treated as a measure of a programme being confusing and behaving unpredictably. Complexity seen in this way can have negative effects both for the users and the companies making such products. Some problems which arise with complexity are a higher learning curve, not many people being 'expert' users and the users spending much time in searching for functions or trying to understand the meaning of a command.

One way in which complexity could be handled is introducing a user-centered approach to the development process but this isn’t the case with all software companies. Instead, more and more features are being added to commercial programmes as a way of improving the product and making it more attractive in the marketplace, a phenomenon called 'bloated software' or 'featuritis' [7]. Such applications have complex graphical interfaces presenting an overwhelming number of functions to users, with menus offering long lists of options and toolbars with a lot of ambiguous icons [11].

A more specific example of software with complex user interfaces are geographic information systems (GIS) which will be the main focus of this paper. A GIS is a system designed to capture, store manipulate, analyze, manage and present all types of geographical data [4]. An example of GIS with a complex interface is dpPower. The application is developed by the company Digpro and
provides a visual tool (in the form of a map) that helps customers plan, design and operate electrical networks. The main goal of this paper is thus to explore different strategies for reducing the perception of complexity amongst users of this particular GIS. I am not focusing on reducing complexity as such but rather on its perception as a subjective experience, in line with the argument that one can’t design a product with a certain quality, but a design might increase the chances that a product is experienced in a certain way [9]. Additionally, with this paper, I am also introducing a user-centered approach to redesigning this type of software as there has not been much research about using user-centered evaluation methods in the development of GIS. Even though GIS is pretty common at work or even at home, there has not been much research dedicated to studying the use of GIS in those environments [19].

RELATED WORK

The history of GIS software and complexity in relation to interaction aesthetics are the main related fields of research that will be presented in the next section. The former describes the history of GIS software with more detail on the dpPower system, while the latter presents approaches arguing against complexity in design.

History of GIS software

GIS software was introduced in the 1960s when computing systems were first handling and processing geographic information. The industry dedicated to the development of GIS software started to emerge in the early 1980s and since then is constantly developing. This was the time in which the graphical user interface of GIS went from command-style query languages understandable only by expert users to hiding those behind icons and menus [5]. To this day, GIS are characterized by overwhelming interfaces presenting a great number of functions which results in a high learning curve for the users. This should be considered as a serious problem preventing a more widespread (but also more enjoyable) use of GIS software [1]. Since GIS have a very complex information structure and a lot of functions, the way of making them more accessible to users lies in their graphical user interface. A successful user interface should create an ‘illusion’, sort of hide the underlying technology and complex architecture understandable to the programme and translate it into something easily understandable for the user [8].

In the case of dpPower (Figure 1), we can identify some aspects which led to an increase in its complexity. First of all, it has been developed over the past years without taking a user-centered approach into consideration. Moreover, many features have been added since the beginning of the application that resulted in an interface which presents a great number of functions in an arbitrary way. All this has led to the users (both new and expert) viewing the interface as something complicated and difficult to use. According to previous customer surveys¹, the interface feels ‘chaotic’ and ‘not modern’. Moreover, the system doesn’t feel immediately accessible to all users, as there aren’t many customers who consider themselves ‘expert’ users.

Figure 1: Interface of dpPower.

Complexity and interaction aesthetics

When redesigning a legacy system, aesthetic qualities, defined as the ‘static appearance of a screen layout’, are often the first thing designers address [10]. The concept of aesthetics in interaction design is however not only confined to what the users see but can be applied to interaction, that is to the feeling one has when using a product, how a product is experienced [10]. Aesthetic statements can be both of positive and negative valence. In this paper, I am concentrating on the latter, namely the perception of complexity of a certain product as a source of frustration and confusion. Moreover, GIS software is a perfect example of a type of product that is considered instrumentally— meant as products used for executing specific tasks—often omitting the experiential

¹ Information accessed through the company’s survey results from 2015 and 2018
dimensions that such systems can have [9]. In our case, by trying to reduce the perception of complexity of dpPower, I will also introduce different types of aesthetic qualities, further explored in the discussion.

**Approaches against complexity**

One way designers think about reducing complexity, is in terms of reducing the visual noise and by encouraging simplicity in design. This entails making products following a minimalistic approach that aims to create ‘systems so simple they are no longer recognizable as systems, but fade into the background, quietly enhancing our abilities’ [14]. Simplicity is also a significant factor when it comes to the users determining how easy it is to use a programme [20]. Understandability is another term used when speaking about complexity. IEEE Standard Glossary (std. 610.12) says complexity is ‘the degree to which a system or component has a design or implementation that is difficult to understand and verify’ [3]. Finally, accessibility is another concept related to the discussion around complexity, as it is about making products that can be easily used by all different kinds of users. Research around accessibility and universal usability started with focusing on disabled users, elderly and young users and expanded to those who have slow internet and limited technology [17].

When considering complex GIS interfaces, all those approaches are undeniably relevant. Simplicity, understandability, and accessibility (intended in this case as creating products that can be used by both novice and more experienced users) are important aspects and examples of strategies to reduce complexity in interfaces. The main purpose of this research project is to explore other, more concrete strategies to mitigate frustration and confusion caused by complexity for applications with instrumental aims. This will be done by engaging with concepts from interaction aesthetics and through qualitative and quantitative user-centered evaluation methods.

**METHOD**

The method used in this research project is research through design (Rtd). Rtd is an approach used in Interaction Design and in Human Computer Interaction (HCI) that defines research where design practice generates new knowledge [22]. I found this method to be the most in line with how I planned to address the problem of this research project, as reducing the perception of complexity through design work was the main focus. The process was divided into two main phases: an exploration phase which opened a design space and the design work [6].

**Opening a design space**

During this phase I empathized with different stakeholders through quantitative and qualitative evaluation methods. The main goal of this phase was to collect different views and perspectives on the interface, coming from both customers and the people developing the application, in order to identify the most important problems that contribute to the perception of complexity. The evaluation methods included: an online survey sent to the customers using the system; three individual interviews with ex-customers now working at Digpro; field studies in two companies using the application; and a heuristic evaluation with people working with the development of GIS software, namely a UX designer and two developers. I also took a beginner’s course on the system and interviewed the teacher of the course afterwards. Everything was conducted in English, however since the mother tongue of the participants is Swedish some received answers were in Swedish and were consequently translated to English. All those activities helped me identify three main categories of problems within the interface, which opened a design space further discussed in the results section.

**Design Work**

After having identified the three main types of problems, I brainstormed different solutions to them, through sketches on paper [16] and an interface overview of the different parts of dpPower. During the design work I also considered three different ways in which one could deal with complexity, namely information architecture (IA) principles [18], multi-layered design [11], and visual design guidelines [15]. Finally, I made a prototype using Adobe XD, in order to represent the idea in more detail and allow the users and myself to engage more with the resulting design.

**RESULTS**

This section will first present the results of the phase that opened a design space, followed by a description of the design work and the most important design decisions made to reduce the perception of complexity.
Opening a design space

During this exploration phase, three main types of problems that contributed to the perception of complexity were identified, namely a confusing IA; the not customizable, all-in-one type of interface; and an inconsistent interaction with the graphical user interface (GUI). Each problem will be discussed in more detail in the sections below.

Confusing Information Architecture (IA)

One of the issues raised in the interviews was that many functions are repeated and can be accessed through different menus. For example, the attribute form, which displays all the information about an object, can be accessed through the object, the right-click menu, the toolbar, the menu bar, and through the command tab in the sidebar (Figure A in Appendix). This repetition of many commands of the system in different places results in overwhelmingly long menus. For example, the right-click menu had 29 commands (Figure B in Appendix) while experts suggest a length of 10-12 commands [13].

An additional problem is the labeling of information in the different menus. For example, the survey showed that a significant number of users think it’s difficult to find things in the menu bar (37% of respondents). During the interviews, the users confirmed this by stating that unless they remember where a function is, it is hard to find it: ‘The menu bar is not divided in a natural way, for example the command ‘show sidebar’ is under settings. Why would that be obvious?’ (ex-customer now working at Digpro).

All-in one type of interface

Another issue raised in this phase was that the users are presented with an overwhelming amount of information. This is especially problematic at the beginning as they don’t get any sort of introduction after the first log-in. Moreover, there are many different types of users who are all being presented with the same interface without taking into account their tasks, skills and roles. Some examples from the individual interviews and field studies illustrate this problem further: ‘There are many buttons, a lot of which I never used for my job’ (worker in operations center); ‘I see a lot of functions I don’t even have access to’ (construction leader).

Moreover, some users pointed out that some sort of customization is missing from the interface: ‘Often when sitting and working with a task, the same 10-20 tools are used all the time. If one could temporarily make them more accessible that would make things much easier’ (from online survey). The users also pointed out that from the place object tab, they usually use 2-3 types of objects and they would like to have a way to hide the rest (Figure C in Appendix). The same applies to the map tab, which displays a long list, making it harder to find things (Figure C in Appendix). Finally, even if there are functionalities which allow for customization of the interface, such as customizing the toolbar, it is difficult to access them. The results from the survey showed that 47% of respondents didn’t even know there was a possibility of personalizing the toolbar.

Additionally, some parts of the interface seemed a bit redundant, especially in the sidebar. For example, not all the tabs in the sidebar are being used, in particular the command tab (Figure D in Appendix) and the tree tab. The command tab is just a repetition of the menu bar, while as for the tree tab, no one really knew its function.

Inconsistent interaction with the graphical user interface (GUI)

Interaction with the GUI is another area that needed consideration. The majority of users mentioned at some point that in certain cases they lack feedback from the system. I’ve seen this myself during an interview with a customer in which she had to complete some tasks I had prepared beforehand. She tried to connect objects with different kiloWatts, however she couldn’t complete the action and there was no feedback as to why it wasn’t successful. The user finally remembered on her own that it’s not possible to connect objects with different kW, however getting no feedback in such instances can definitely add up to the perception of complexity of the system. In certain instances (for example when placing an object) the system is suggesting the next steps to complete the command, but this guidance is not immediately visible (the font is quite small, located at the bottom of the interface).

The language of the system was another point that came up during the interviews. Some technical concepts should be explained to new users, for example the concept of ‘changeset’ or the difference between components, objects and attributes. Some examples include: ‘The first time I used the system I didn’t understand why it says show/change attribute form when all I needed is to see information about an object’, ‘The most difficult thing for the user is to learn the
database model’ (from survey). This makes understanding the system even more difficult.

Additionally, the way in which a command is executed in the system is not intuitive nor transferable from the interaction with other systems. For example, when splitting a view, the user has to double right-click to finish the command. Moreover, in order to confirm the placement of an object on the map, the user has to right-click on the map. Finally, moving an object is not executed in a natural way for the user, as they have to select an object and choose the ‘move object’ function from the toolbar, right-click menu or the menu bar while in other systems the user would simply drag and drop the object.

Interaction with the GUI also concerns its visual appeal. A problem that arose during this research project are the icons. The GIS uses an overwhelming amount of icons that often represent very abstract concepts, and as a result are ambiguous. The look of some icons is also not directly related to the functions they want to represent. As a consequence, the users feel like it’s hard to remember the meaning of every icon. The map tab for instance uses a pens icon to symbolize the functions related to the map in the sidebar. The same icons are also repeated for different commands: the search tab has the same icon as the query tool in the toolbar. Moreover, the tab structure present in some parts of the interface creates visual noise. Finally, some users said that the interface feels ‘old’, with some parts overcrowded with functions and a lot of ‘dead space’ unused.

**Design work**

The design work consisted in brainstorming design solutions to the three types of problems mentioned above, through sketches, an information structure overview, and a prototype made in Adobe XD.

**Clear Information Architecture (IA)**

The aim with introducing a clearer information structure is for the users to find it easier to search for information, understand better how to navigate the system and reduce the cognitive overload of the interface. To improve the information architecture in the interface and decrease the number of functions accessible through different menus and their repetitions, I prepared a spreadsheet with an overview of the information structure in the menu bar, the toolbar, the sidebar and the right-click menu (Figure 2). In order to decide which functions should go into which menu, a clear distinction between them had to be made. I decided the toolbar should only include the functions related to manipulating the map, the menu bar should have global functions and the right-click options should only include functions that are directly related to the selected. The sidebar is a good place to have functions that display a long list of items (often in the form of a tree), this is why I decided to keep the place object, search, map, thematic maps and favorites tabs there (and removed them from other locations).

While making the overview of different commands in the menus, I also highlighted the ones likely to be used by beginners, basing myself on the handbook I got after the course provided by the company. I later used this information during the creation of the pre-set views aimed at customers with no prior knowledge of the system, further discussed in the next paragraph. Additionally, I tried to group similar functions under sub-menus, to reduce the amount of options on the lists in the menu bar.

I also deleted some redundant functions, such as the command tab and the tree tab in the sidebar (however, I made sure it is possible to bring it back through the customization options). Since the command tab allowed for searching a specific function of the menu bar, I created a search function on the right side of the menu bar instead (Figure 3).

![Figure 2A: Old menu bar structure. 2B: New menu bar structure](image-url)
The problem of an all-in-one, unmodifiable interface made me realize the system should allow users to make some choices regarding the interface, to create a better compromise between the great number of functions it offers and what users need. The aim with creating a customizable view is that after personalizing it, it will only contain the functions a specific user finds necessary for their work.

However, since not all users know what kind of functions they need for their tasks or are interested in the possibility of customizing the interface, I also propose two pre-set views (which still allow for some sort of customization in the sidebar). One view is a beginner view, tailored to users who log into the system for the first time, and the second one is a standard view with all the functions for the users that don’t feel the need to customize the interface. The users can also create a new view, where they have full control over what to see in the menu bar, the sidebar, the toolbar, the right-click menu, and the attribute form. They are also able to hide the sidebar and the toolbar from the main view.

1. Beginner and standard view

In the beginner and standard view, the user is not able to change much regarding the interface. The only thing that can be customized in those views is the sidebar (more specifically the place object, the map and the favorites tab). This is because different customers use different maps and commands and place different objects on the map. The customization in those tabs happens by either choosing and ‘starring’ and object or command which then gets transferred to the main list or ‘decluttering’, meaning hiding certain items from the main view (Figure 4). When the users first log into the system they can choose which view to log into (Figure 5). When a user changes the view from beginner to standard for the first time, the new functions in the menu bar are highlighted in an effort to make this transition smoother and more understandable (Figure E in Appendix). They disappear automatically after the first session; however, it is possible to bring them back through the ‘edit view’ function under ‘customized view’ in the menu bar.

2. Adaptable view

A user can also choose to create their own customized view of the interface. Toggling between different views is possible through the menu bar. In this view, the user has full control over what to include in the menu bar, the sidebar, the toolbar, the right-click menu and the attribute form. The customization here happens by hiding functions that the user doesn’t want to see in the different menus, a sort of visual decluttering (Figure 6). It is also possible to show/hide the toolbar and sidebar.
and customize the content of different tabs in the sidebar (such as in the beginner and standard view).

![Image](image.png)

**Figure 6A & 6B: Hiding options that are not needed in the attribute form and in the menu bar.**

**Consistent interaction with the graphical user interface (GUI)**

Finally, another problem discussed during the interviews was interacting with the GUI, which concerns both how the system is designed to respond to user actions (how feedback is given and in what instances) and what the GUI looks like (its visual noise).

Subtle improvements to the interaction with the system were made. I increased the font size in the status bar, added some text explaining the new features, and introduced pop-ups with guidance (Figure E in Appendix). Another idea aimed at improving the interaction was that the system should suggest the next step to the user by highlighting the objects that become ‘clickable’ after a certain function is enabled.

For the new design of the interface, visual design guidelines were considered. They provide a standardized look of the system which makes it feel more familiar to the users. In an effort to make the interface as easy to use as possible, I implemented some of the Windows design guidelines\(^2\) (with the exception of the search inputs, which come from the Material Design guidelines\(^3\)) to the interface of dpPower, as all the customers are using the GIS on a Windows operating system. I made sure the elements of the interface are aligned by using a grid in an effort to provide a more harmonious look of the interface. The tabs present in different parts of the interface were replaced with a more modern solution: a vertical side panel where the elements are aligned vertically in order to reduce the visual complexity (Figure 7). To make it easier to customize both the sidebar and the toolbar, three dots symbolizing a settings feature were added to both. Finally, some icons were also replaced with new ones, for instance the map tab, which had a pen-shaped icon, was replaced with one that matched better the concept of a map. Moreover, the query tool and the search icon in one of the tabs in the sidebar were the same, therefore the search icon was replaced with a new one to avoid confusion. The icons that were used in the tree lists in the sidebar, in the form of yellow folders, were also replaced with more minimalistic arrows to decrease the visual noise. As for the colors, the dominating grey color was replaced with light orange and dark blue, which are the colors from the logo of Digpro.

![Image](image.png)

**Figure 7A: Old design of attribute form & interface. 7B: New design of attribute form & interface.**

**DISCUSSION**

During the research project three main types of problems emerged, contributing to the perception of complexity: confusing IA; all in one, not-customizable type of interface; and inconsistent interaction with the GUI. As we saw in the results of the exploration phase, they all contributed to complexity in different ways. A confusing IA makes it harder for the users to understand where to find the information needed. It causes the less experienced users to ask more experienced ones for help. The all-in-one type of interface, on the other hand, displays the same type of information to all types of users, which results in customers seeing functions

\(^2\) [https://docs.microsoft.com/pl-pl/windows/uwp/design/](https://docs.microsoft.com/pl-pl/windows/uwp/design/)

\(^3\) [https://material.io/](https://material.io/)
irrelevant to them. The interface seems complex as only a subset of the functions is being explored and used. Finally, the inconsistent interaction with the GUI was not immediately understandable for the users. The programme prescribes behaviors that are inconsistent with respect to other commonly used programmes and communicates with the users in a language that is not immediately clear. The interface also presents a lot of visual noise, which makes it harder to understand the navigation and the placement of different commands.

**Strategies for reducing complexity**

Three main types of strategies were introduced with the aim to solve those three problems and introduce aesthetic qualities to the interaction. The three main strategies were: a clear information architecture, an adaptable interface through multi-layered design and consistency with visual design guidelines. I chose to focus on these three strategies as they were strongly evidenced by existing literature, they seemed the most relevant to the types of problems found and were prioritized by the company. Additionally, apart from being potential ways of reducing the perception of complexity, they also introduce aesthetic qualities to the interaction, further discussed below.

Information architecture addresses the problem of structural complexity. It’s related to organizing content, describing it clearly and providing ways for people to get to it [18]. Its importance lies in the fact that how we label, group and structure elements of our interface directly affects how clear it is for the user to find or use a certain function. When considering the aesthetic quality that this brings to the interaction with the system, a well-organized information structure reduces the demands for our conscious attention and information processing, which relates to the concept of fluency. Fluency is defined by Löwgren [10] as ‘the graceful dance among media streams’ and is one of the ways in which a designer can think of dealing with complexity. A clear IA makes it easier for the user to navigate and process information in a structured, but also more fluent way. When engaging with the prototype the users said they felt relief and ease while navigating the interface, as it wasn’t as ‘cluttered’ as before. Achieving fluency is however, not an easy task as a designer has to understand how to structure the information in a way that makes sense to the users. What I noticed during the interviews was that many users had doubts about the labeling of the information in the different menus, but they were not unanimous as to where a particular command should be placed, which should be further explored in future studies.

Fluency can also be introduced by having the system anticipate where information and guidance should be provided. In the case of this research project this was done by introducing infoboxes explaining new features and the system highlighting the next step or all the clickable objects. In fact, during the follow-up interviews the users suggested new ways in which the system could provide guidance, introducing a higher level of fluency. Overall, it is an important and valued quality for systems used instrumentally, where saving time and being able to process information in a faster way is crucial for the organization.

The second strategy was having multiple layers which correspond to different types of users and expertise. The idea of multi-layered design has already been suggested before [11, 17]. In order to satisfy the ideas of both first time and more expert users Schneiderman [17] proposed having two layers of the interface corresponding to different levels of expertise, with novice users starting from layer 1 with the possibility to switch to a more advanced layer whenever they feel ready. Dividing the user flow into different layers is a way of showing only the functions a beginner user will need, with the goal of reducing complexity in an interface and facilitating the learning of a piece of software. Another idea related to multi-layered interfaces is providing the user with an adaptable layer, which includes desired features only [11].

The concept of having multiple layers in an interface is an important idea to consider in order to make the interface more tailored to a user’s needs and competencies. Moreover, having an adaptable layer, introduces an experiential quality to the system defined as pliability [9]. A pliable interaction is defined as a ‘sense of shaping the digital information with fingertips, even though the actual artifact might employ standard, non-tactile interaction techniques such as mouse keyboard, and a display monitor’ [9]. In this research project, I took a slightly different take on pliability and focused on letting users adapt the interface in a manner that might lead to a sense of pliability. The users are able to hide the irrelevant information and declutter the interface, which increased its ‘sense of malleability’. The feedback from the follow-up interviews on the prototype confirmed that being able to manipulate the interface, reduces the feeling of frustration and helplessness that
one inevitably has when being presented with an overwhelming amount of information. Instead, the users felt more in control. This, of course, concerns the more experienced users, as first-time users would have to learn the interface first (and go through the beginner and standard layer before using the customizable one). Moreover, there is further potential for increasing pliability in the system which could be done by reducing the divide between the hand operating the mouse and the screen, making the interaction with the GUI seem more responsive and as a result more enticing. Introducing other forms of interactions (for example, dragging objects onto the map instead of right-clicking on the map to confirm the placement) could make the interface feel more dynamic and the interaction more involving by reducing the current divide between action from the user and response from the interface.

Finally, visual design guidelines were the third strategy used to address the problem of visual complexity. They are a way of making the interface look more ‘familiar’ by providing a consistent and standardized look and feel. This standardization allows users to reuse the skills they acquired in a previous interface [15], making the new interface feel more predictable and understandable. ‘Rhythmic patterns and temporal predictability’ [10] create rhythm, which is another experiential quality used to describe a product experienced in a rhythmical way. Rhythm focuses on the interaction with the system and introduces some sort of standardization and ‘repetitive patterns’ [10] to it. A design that follows design guidelines and consequently introduces a consistent and predictable look, makes it easier for the user to apply the knowledge from other systems to a new one. Rhythm is, in other words, about introducing designs made out of components that follow a certain pre-established pattern. An example of this could be making buttons that all follow a certain style, or H1 headers in a specific size.

However, predictability and consistency are not limited to the look of the system and can also be applied to the interaction. The interaction with dpPower is not consistent with respect to other systems. Introducing ways of interacting with the interface that are already known from other systems would also be a way of expanding this experiential quality and could open interesting future work.

Reflections on introducing a user-centered approach

Up until now, we focused on how people experience technology and what can be done to shape the user’s experience [21]. However, we can’t ignore the potential implications on the larger organizational context. Qualities such as fluency (good IA), pliability (customization) and rhythm (consistency) could potentially impact the companies buying and selling digital products. If the learning curve is reduced, perhaps there will be no need for ‘expert’ users at the customer’s offices having to onboard the new ones. This also means less time is ‘wasted’ in learning about the different functionalities of the system before it is used for actual work. Some other benefits could include faster and more efficient task execution and less frustration coming from employees using the system (which could impact their performance overall). We can thus come to a conclusion that in this case the value of the product isn’t only in the interaction itself, but goes beyond it and lies in the ‘lasting value of enduring outcomes’ [2]. From the perspective of the company Digpro, selling a product that addresses customers’ needs (such as having the possibility of customizing the interface according to the tasks of each user) could make the product more desirable in the market. As mentioned by Löwgren [9]: ‘When people are surrounded by digital products and services they can choose freely between (or choose not to use), aesthetic qualities are becoming every bit as important as the instrumental ideals of human computer interaction’. If we look at interaction aesthetics from a company’s perspective, we could argue that design work creates commercial value for the product. This means that consumers will see more value in well-designed products and will choose them over competitor products [2]. This is an important thing to consider for companies, which are nowadays focused mainly on adding more features to programmes without taking into consideration the benefits of introducing a user-centered approach in the software development process.

Future work

Having reflected over the results of this research project, there is still more that can be done to measure the effects of the strategies and explore new possibilities for tackling complexity.

Due to time limits, the evaluation of the prototype was made through follow-up interviews, but I could have asked the users to engage with the prototype over a longer period of time and deliver their feedback in the form of diary entries where they would note the feelings
and experiences connected to using the GIS. This could give me more insights and evidence as to what extent do the strategies proposed reduce the perception of complexity and how everyday work tasks are affected.

As for the GUI, more consideration towards accessibility should be made in the future (for example checking if the colors have sufficient contrast). Moreover, if we look at the idea of multi-layered design, more research on how to make the transition from beginner to standard view smoother is needed perhaps by studying how new users learn the system and how their interaction with it develops over time. It would also be interesting to explore whether there is a significant difference between adaptive interfaces (which automatically respond to how often a function is used through artificial intelligence) and adaptable ones (where the user has to manually change the interface), in terms of user satisfaction and effectiveness.

CONCLUSION

Complexity in instrumental software such as GIS often has negative consequences, such as time and resources which are ‘wasted’ on learning the software and executing daily tasks while the users feel overwhelmed and frustrated. Complexity is often a consequence of an overwhelming number of features being added and a lack of a user-centered approach in the development process. The main purpose of this research project was to explore strategies that mitigate frustration and confusion caused by complexity for applications with instrumental aims by engaging with concepts from interaction aesthetics and through qualitative and quantitative evaluation methods. The three strategies considered in response to the problems found during an initial exploration phase were: information architecture, multi-layered design and visual design guidelines. They all tackle complexity in different ways introducing at the same time different experiential qualities to the interaction namely fluency, pliability, and rhythm. An organized information structure enables the users to process information in a more efficient and satisfying way. Letting the user be in control through customization features and different layers reduced frustration as the users were able to get rid of irrelevant information. Finally, visual design guidelines increased the system’s predictability and helped decrease the visual noise. However, introducing a user-centered approach and concepts from interaction aesthetics benefits not only the users, but also the companies developing instrumental types of products, which is something firms should take into consideration.

ACKNOWLEDGEMENTS

I would like to thank my supervisors at KTH, Karey Helms and Marianela Ciolfi Felice, for all the valuable feedback and helpful discussions. I would also like to thank my supervisor at Digpro, Martin Andersen, for all the support and help throughout this project.

REFERENCES


APPENDIX

Figure A. Different ways of accessing the same show/change attributes command.

Figure B. Right-click menu.

Figure C. Place object tab and Map tab in the Sidebar.
Figure E. Improvements to the interaction with the system: infoboxes & highlighting new functions in the standard view.

Figure D. Command tab in the Sidebar.